Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

De la parole aux transcriptions :
optimiser la transcription de l'arménien occidental avec des ressources limitées

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Agathe WALLET

Directeur de mémoire :

Damien Nouvel

Année universitaire 2023/2024

Avant-Propos

Remerciements

Avant toute chose, je souhaiterais remercier toute l'équipe pédagogique du master PluriTAL pour les connaissances et compétences précieuses qu'elle m'a transmises.

Je tiens à remercier tout particulièrement l'ensemble de l'équipe ERTIM pour son accueil chaleureux il y a maintenant six mois. Nos discussions autour de la table ronde, toujours très enrichissantes, m'accompagneront longtemps.

Un grand merci à Johanna et Samuel, sans qui ce mémoire n'aurait pas été possible, ou du moins bien différent.

Merci également à Damien pour son accompagnement tout au long du stage et de la rédaction de ce mémoire, pour ses idées et conseils précieux.

Je voudrais remercier tous mes camarades pour leur solidarité, leur bonne humeur et les rires que nous avons partagés pendant ces deux années.

Enfin, un merci tout particulier à Laura et Clément, qui m'ont accompagné chaque jour ces six derniers mois, dans les bons comme dans les mauvais moments.

Résumé

La transcription automatique de langues peu dotées en ressources est un défi, autant pour l'exploitation des données disponibles que pour le choix des algorithmes et l'évaluation des résultats. Dans ce mémoire, nous présentons nos travaux pour développer un modèle de transcription de l'arménien occidental vers une écriture phonologique (API). Ils ont été réalisés dans le cadre du projet de recherche DALiH (Digitizing Armenian Linguistic Heritage) qui apporte un volume de données transcrites limitées pour l'arménien occidental. Plusieurs stratégies d'optimisation de l'apprentissage ont été explorées, qui comparent plusieurs modes de sélection des données d'entraînement. Ces recherches ont montré qu'un choix judicieux de modèle et de données améliore significativement l'apprentissage automatique, tout en montrant la sensibilité des résultats aux choix des jeux de données et des paramètres des modèles de transcription automatique.

Mots-clés : alphabet phonétique international, arménien occidental standard, langues peu dotées, optimisation d'apprentissage de modèles, reconnaissance automatique de la parole

TABLE DES MATIÈRES

Liste des figures

Liste des tableaux

Introduction

Présentation générale

Le traitement de la parole, et plus particulièrement la reconnaissance automatique de la parole, est un domaine de recherche en pleine expansion. Cependant, les langues moins dotées en ressources linguistiques restent souvent mal prises en charge par les systèmes actuels. Contrairement au traitement de l'écrit, le traitement de la parole a vocation à traiter toutes les langues du monde, dont une grande majorité est uniquement orale, sans écriture associée. Les outils développés dans ce domaine pourraient donc servir à documenter, étudier et préserver ces langues, dont beaucoup sont en voie de disparition. Toutefois, développer des modèles de langue nécessite généralement de grandes quantités de données, ce qui pose problème pour ces langues pour lesquelles les volumes de ressources sont limités.

Les variantes de l'arménien sont des langues généralement peu dotées. Hormis l'arménien standard oriental, langue officielle de l'Arménie, les autres variantes ont un statut plus précaire. Plusieurs projets ont récemment vu le jour autour des langues arméniennes, de leur étude et de leur préservation. Ce mémoire s'inscrit dans le cadre de l'un d'eux : le projet DALiH (Digitizing Armenian Linguistic Heritage), qui vise à créer une plateforme linguistique numérique en accès libre, couvrant une grande partie des variantes de l'arménien. Ce projet inclut l'arménien standard oriental, l'arménien standard occidental, l'arménien classique, l'arménien moyen, ainsi que divers dialectes. L'un des objectifs majeurs de ce projet est de développer un modèle de transcription automatique multi-variétal de l'arménien, permettant ainsi de mieux comprendre et documenter les variations phonologiques entre les différentes variantes de la langue.

Parmi les variantes, l'arménien occidental présente un intérêt particulier en raison de sa situation linguistique unique. Parlée principalement par les membres de la diaspora arménienne, cette variante est de moins en moins transmise entre les générations, ce qui rend compte de l'urgence de sa documentation et de sa préservation. Le projet Rerooted, créé en 2017 par Anoush Baghdassarian et Ani Schug, toutes deux issues de la diaspora arménienne, illustre bien cette volonté de préserver l'arménien occidental. Ce projet a pour double objectif de réunir des témoignages de nombreuses communautés arméniennes et de créer du matériel pédagogique à partir de ces enregistrements. Aujourd'hui, plus de 25 communautés arméniennes ont participé à ce projet, contribuant ainsi à la création d'un corpus riche et diversifié de témoignages, sur lequel nous nous sommes appuyé pour nos expériences.

Problématique et objectif

L'arménien occidental étant une langue peu dotée, notamment à l'oral, il est important de réfléchir aux stratégies possibles pour optimiser l'apprentissage de mo-

dèles de reconnaissance de la parole avec peu de données.

Tout d'abord, comment choisir ses données ? Dans le contexte d'une langue disposant de peu de données transcrites, il est essentiel de déterminer quelles données permettent un meilleur apprentissage des modèles et, par conséquent, quels types de données transcrire en priorité pour développer un système de transcription de la parole performant.

On peut également se poser la question de la modalité de transcription. Les systèmes de Reconnaissance Automatique de la Parole (RAP) traditionnels s'appuient souvent sur des dictionnaires associant une représentation sonore à un mot. Lorsqu'on a peu de données et un échantillon pas toujours représentatif de la diversité linguistique de la langue, cette option n'est donc pas forcément la plus pertinente. La transcription vers l'Alphabet Phonétique International (API) apporte une réponse à ce problème. Le vocabulaire étant plus restreint et ce type de transcription plus généralisable, transcrire vers l'API demande en théorie moins de temps de calcul pour l'apprentissage du modèle, qui serait aussi performant sur tout type de fichier audio en arménien occidental.

De plus, la mesure des distances phonologiques entre les différentes variantes de l'arménien est une dimension importante du projet DALiH. En retranscrivant les variantes de l'arménien dans une écriture phonologique, il devient possible de comparer les systèmes phonologiques de manière précise et systématique. Cette approche permettrait donc non seulement de documenter les variations phonologiques existantes, mais également de comprendre les dynamiques linguistiques qui sous-tendent ces variations.

Plan de lecture

Ce mémoire s'articule en plusieurs parties :

- Traitement naturel et automatique de la parole (chapitre 1) : Cette section explore d'abord la perception du langage par le cerveau humain, puis examine comment ce système organique a inspiré les systèmes de reconnaissance automatique de la parole.
- L'arménien, langue·s plurielle·s (chapitre 2) : Cette partie présente brièvement les différentes variantes de l'arménien et les principales différences phonologiques entre les deux standards actuels.
- Corpus (chapitre 3) : Ici, nous détaillons le corpus utilisé et les pré-traitements effectués pour l'intégrer dans nos expériences.
- Méthodes (chapitre 4) : Cette section présente les différents modèles que nous avons utilisé dans le cadre de nos expériences.
- Résultats (chapitre 5) : Nous y présentons les hyper-paramètres utilisés et les résultats des expériences réalisées.
- Discussion (chapitre 6) : Cette partie discute des résultats obtenus, des limites de nos expériences et des perspectives futures.

partie I

État de l’art et contexte

CHAPITRE 1

TRAITEMENT NATUREL ET ARTIFICIEL DE LA PAROLE

Sommaire

Introduction

Dans ce chapitre, nous abordons le traitement naturel et artificiel de la parole. Nous commencerons par explorer comment le langage parlé est perçu et analysé par le cerveau humain. Ensuite, nous présenterons comment les systèmes biologiques ont inspiré les systèmes artificiels dans le domaine du traitement automatique de la parole.

1.1 Perception humaine du langage

Dans son Cours de Linguistique Générale, [Saussure, 1916] divise le langage en deux composantes : la **langue** et la **parole**. La langue est la partie sociale du langage. C'est un code commun à tous les membres d'une communauté linguistique. La parole, en revanche, est la composante individuelle du langage. Elle dépend d'une personne à un moment donné, dans un lieu donné, dans un contexte donné. La parole est innée et ne suit pas de normes ni de code. C'est dans la parole que des changements linguistiques s'opèrent chez chaque individu.

Le mot «parole» sert également à désigner le mode de communication le plus naturel pour l'être humain. C'est d'ailleurs la seule modalité langagière commune à toutes les langues du monde (on estime à seulement 200 le nombres de langues possédant une écriture sur les plus de 7000 langues répertoriées par l'UNESCO[1] et Ethnologue[2]).

1. https://en.wal.unesco.org/world-atlas-languages
2. https://www.ethnologue.com/

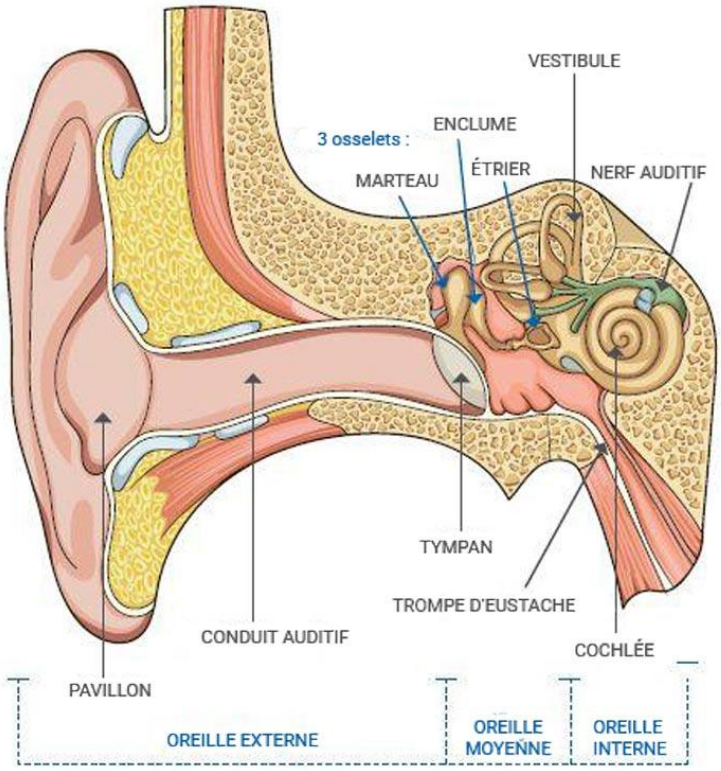

FIG. 1.1 : Schéma anatomique de l'oreille humaine

Comment le langage est-il interprété par le cerveau humain ? Nous verrons comment le signal sonore qu'est la parole est perçu par l'oreille humaine et comment du sens s'en dégage.

1.1.1 Perception de la parole

D'un point de vue physiologique, la parole est un son perçu par l'oreille. Celle-ci est composée de trois parties, permettant un premier traitement du signal sonore (voir figure 1.1) :

- L'oreille externe, qui détecte la provenance des sons grâce au pavillon et concentre les ondes sonores via le conduit auditif vers le tympan ;
- L'oreille moyenne, qui, avec ses trois osselets (marteau, enclume, étrier), transmet les vibrations du tympan aux liquides de l'oreille interne ;
- L'oreille interne, principalement composée de la cochlée, dont les cellules ciliées transforment les vibrations ayant une fréquence comprise entre 20 et 20 000 Hz en signaux nerveux. Ces signaux nerveux sont transmis via le nerf auditif jusqu'au cortex auditif primaire.

Bien que l'oreille soit capable de détecter les sons dont la fréquence est comprise entre 20 et 20 000 Hz, elle ne le fait pas avec la même précision sur tout le spectre. En effet, l'oreille humaine est particulièrement sensible aux sons dont la fréquence est comprise entre 500 et 5 000 Hz, ce qui correspond aux fréquences de la voix humaine (voir la figure 1.2).

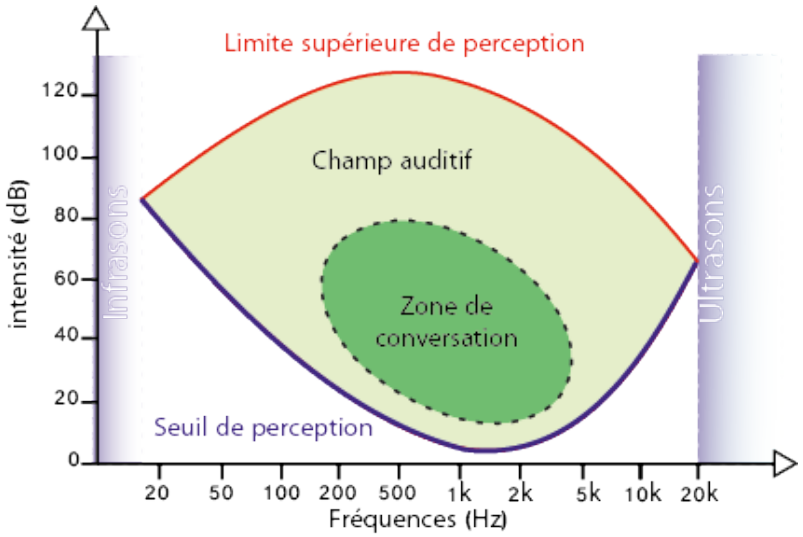

FIG. 1.2 : Courbe de Wegel présentant le champ auditif de l'oreille humaine

1.1.2 Perception de la langue

Une fois le signal sonore de la parole perçu et transmis au cortex auditif primaire, celui-ci décode l'information et la transmet à différentes aires cérébrales pour un traitement ultérieur. Parmi celles-ci, on trouve notamment l'aire de Wernicke, le centre de la compréhension du langage, qu'il soit parlé ou écrit. Une lésion dans cette zone cérébrale peut entraîner une aphasie réceptive, un trouble de la compréhension du langage : la personne atteinte ne sera plus capable de comprendre le langage, mais pourra encore parler, bien que de manière totalement incompréhensible. C'est donc au niveau de l'aire de Wernicke que parole et langue se rencontrent. Cette zone fait appel à la mémoire sémantique pour permettre la compréhension des mots et leur catégorisation.

La mémoire sémantique est la mémoire partagée par tout le monde dans une certaine limite culturelle. Elle stocke des connaissances permanentes, faciles d'accès et ne nécessitant pas d'effort pour être restituées. C'est à ce niveau que sont enregistrées toutes les informations liées aux langues premières (ou natales) : lexique phonologique, morphologique, constructions syntaxiques, etc. [Tulving, 1970] associe d'ailleurs la mémoire sémantique à un «thésaurus mental, un savoir organisé qu'une personne possède sur les mots et autres symboles verbaux, leur signification et leurs référents, sur les relations entre eux, et sur les règles, formules et algorithmes pour la manipulation de ces symboles, concepts et relations»[3].

Seulement, qui dit mémoire dit apprentissage. En effet, bien que la parole soit innée, la langue ne l'est pas. L'apprentissage de la ou des langue·s première·s se fait dans les premières années de vie d'un enfant. Lorsqu'il naît, le nourrisson n'a pas de prédisposition à distinguer certains phonèmes plutôt que d'autres. Il peut discerner et discriminer tous les sons du langage humain. Il va peu à peu perdre cette capacité en n'étant exposé qu'à une langue (parfois plusieurs dans les familles multilingues). Durant les 7 à 8 premières années de sa vie, l'enfant va se former un lexique phoné-

3. "It is a mental thesaurus, organized knowledge a person possesses about words and other verbal symbols, their meaning and referents, about relations among them, and about rules, formulas, and algorithms for the manipulation of these symbols, concepts, and relations."

mique. Au-delà de cette période, qu'on appelle «période critique», l'apprentissage de nouvelles langues mobilisera d'autres zones cérébrales et la distinction phonémique demandera plus d'effort pour les phonèmes qui ne sont pas déjà dans son lexique.

Ainsi, une personne n'ayant été exposée qu'au japonais durant son enfance aura beaucoup de mal à distinguer les phones [ʁ] et [l] car ils ne font pas partie du lexique phonémique du japonais. Ainsi, les productions [ʁe] et [le] seront perçues comme deux productions du même phonèmes. En revanche, une personne exposée au français dans son enfance saura distinguer les deux : les productions [ʁe] et [le] (qu'on pourrait par exemple orthographier «ré» et «les» en français), désignent bien deux entités distinctes.

Phone et phonème : son et identification

Nous avons parlé de phones et de phonèmes mais à quoi correspondent ces termes ?

Selon le dictionnaire de linguistique ([Dubois et al., 2001]), les **phones** sont définis comme «chacune des réalisations concrètes d'un phonème, variables suivant le contexte phonique, le locuteur, et les conditions générales d'émission». En simplifiant, un phone peut être défini comme tout son articulatoire ou tout son de la parole. Les phones n'ont pas tous la même importance au sein d'une langue : la distinction entre certains est plus cruciale que celle entre d'autres, car elle permet de différencier deux mots (comme les phones [ʁe] et [le] en français par rapport au japonais). On appelle **phonèmes** ces phones discriminants au sein d'une langue donnée. Le phonème est défini par le même dictionnaire comme «l'élément minimal, non segmentable, de la représentation phonologique d'un énoncé, dont la nature est déterminée par un ensemble de traits distinctifs».

On appelle **allophones** les réalisations phonétiques possibles d'un même phonème, «de telle sorte qu'aucune d'entre elles n'apparaît jamais dans le même environnement qu'une autre» ([Dubois et al., 2001]). En français, le phonème /r/ a plusieurs réalisations phonétiques possibles (donc plusieurs allophones) :

- le r voisé [ʁ] comme dans [gʁa] (gras),
- le r sourd [χ] comme dans [taχt] (tarte),
- le r roulé [ʀ], utilisé par des chanteurs et chanteuses francophones comme Édith Piaf ou Jacques Brel.

Les phones et phonèmes sont transcrits en Alphabet Phonétique International (API). Le principe de cet alphabet est «un seul signe pour chaque son, un seul son pour chaque signe». Cet alphabet a pour vocation de transcrire tous les phones, c'est-à-dire tous les sons présents dans les langues humaines. La première version de l'API a été publiée en 1886 par l'Association phonétique internationale, à l'époque constituée principalement de phonéticiens français et britanniques. L'alphabet se compose principalement de lettres grecques et latines, auxquelles d'autres signes et diacritiques ont été ajoutés. Sa dernière version, publiée en 2005, comprend 107 lettres, 52 signes diacritiques et 4 caractères de prosodie[4].

4. https://fr.wikipedia.org/wiki/Alphabet_phonétique_international et https://www.internationalphoneticalphabet.org/

Pour distinguer les phones des phonèmes, on inscrit les premiers entre crochets et les seconds entre barres obliques.

Phonème et morphème : double articulation

[Frauenfelder, 1991] s'intéresse à la détection du mot parlé. La parole représente pour lui deux défis principaux :

- la parole est variable : la production orale d'un mot n'est jamais exactement la même et il est pourtant reconnu comme une seule entité (relation plusieurs-à-un ou *many-to-one*) ;
- la parole est ambiguë : une représentation phonologique peut correspondre à plusieurs entrées lexicales (relation une-à-plusieurs ou *one-to-many*). Ex : /œ̃gʁɑ̃tami/ pourrait correspondre à «un grand ami» ou «un grand tamis».

La parole est en effet perçue par notre cerveau comme une suite continue de phones. Lorsque nous entendons parler quelqu'un dans une langue étrangère, nous sommes incapables d'isoler des unités dans ce flux. Pourtant, dans une langue connue, nous pouvons associer à ces phones des phonèmes et les assembler pour former des unités de sens. Les unités minimales de sens sont appelées monèmes ou morphèmes. C'est cette capacité de combiner des unités distinctives (phonèmes) pour obtenir des unités significatives (morphèmes) que [Martinet, 1960] appelle la **double articulation**.

Comme mentionné précédemment, [Saussure, 1916] décrit la langue comme un ensemble de signes codifiés permettant aux locuteurs de se comprendre. Il définit le signe comme une entité psychique, abstraite, composée de deux éléments indissociables : le **signifié**, qui désigne le concept du signe, et le **signifiant**, qui correspond à l'image acoustique ou graphique servant à exprimer le signifié. Ainsi, notre lexique, stocké dans notre mémoire sémantique, associe à un concept une suite de graphèmes ou de phonèmes, nous permettant de donner un sens au signal physique qu'est le son de la parole que nous percevons.

Bien que la perception du langage par le cerveau humain ne soit pas encore complètement comprise et qu'il existe de nombreuses théories à ce sujet, nous avons présenté tous les éléments nécessaires à la compréhension du langage. Nous avons exploré les mécanismes physiologiques de la perception de la parole, les processus cognitifs impliqués dans la compréhension du langage, et les concepts fondamentaux de la phonologie et de la sémantique. Ces éléments constituent une base solide pour comprendre comment le langage est interprété par le cerveau humain et comment ces connaissances peuvent être appliquées à la transcription automatique de la parole.

1.2 Traitement automatique de la parole

Les technologies de traitement de la parole ont connu un essor considérable ces dernières années, notamment grâce à l'avènement des réseaux de neurones, qui tentent de reproduire le fonctionnement du cerveau humain dans le traitement de l'information. En reconnaissance automatique de la parole, par exemple, nous sommes passés en quelques décennies d'un dispositif électronique câblé capable de reconnaître les 10 chiffres (AUDREY, développé par K. H. Davis, Rulon Biddulph

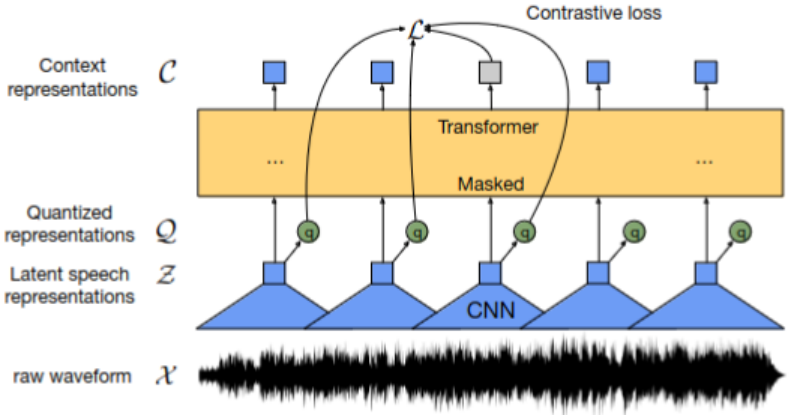

FIG. 1.3 : Architecture de Wav2vec2 apprenant conjointement des représentations de parole contextualisées et un inventaire d'unités de parole discrétisées ([Baevski et al., 2020])

et Stephen Balashek, trois chercheurs des Bell Labs en 1952, comme décrit par [Pieraccini and Rabiner, 2012]) à une multitude d'assistants vocaux, dont Siri (Apple) a été le pionnier en 2011.

Tout comme l'apprentissage d'une langue pour un être humain nécessite des années d'exposition à celle-ci, l'entraînement d'un modèle de reconnaissance de la parole nécessite également son exposition à une grande quantité de données. Cela pose donc un problème pour les langues peu dotées, pour lesquelles peu de ressources sont accessibles, notamment les ressources annotées.

1.2.1 L'architecture Wav2Vec2

Les modèles Wav2Vec2, proposés par [Baevski et al., 2020], tentent de répondre à ce problème. Leur architecture, présentée sur la figure 1.3, permet d'entraîner un modèle performant en utilisant majoritairement des données non annotées, plus facilement disponibles. L'idée est de pré-entraîner le modèle en auto-supervision. On lui fournit des données audio non annotées, et le modèle «apprend» la ou les langues en masquant certaines zones de l'audio qu'il doit compléter en se basant sur le contexte. Cette méthode d'apprentissage se rapproche de la manière dont un enfant apprend une langue, simplement en y étant exposé et en apprenant petit à petit quels mots vont ensemble dans tel contexte.

Une fois ce pré-entraînement effectué sur des données non annotées, les modèles peuvent être affinés en leur fournissant cette fois-ci des données annotées, en bien moindre quantité : on obtient de bons résultats à partir de seulement 10 minutes de données annotées. Cet affinage permet également de spécialiser les modèles sur certaines tâches, telles que la classification d'audio (détection de langue, de prosodie, etc.) ou la reconnaissance automatique de la parole (transcription de l'audio vers l'écrit ou *Speech-To-Text*), la lecture de textes écrits (*Text-To-Speech*), etc. De nombreux modèles pré-entraînés sont disponibles sur HuggingFace[5], certains pré-entraînés sur plus de données et plus de langues que d'autres.

5. https://huggingface.co/

1.2.2 La reconnaissance automatique de la parole

La reconnaissance automatique de la parole (RAP ou ASR en anglais pour *Automatic Speech Recognition*) est le domaine du traitement automatique de la parole permettant de retranscrire la parole sous forme de texte exploitable par la machine et par l'humain.

Aujourd'hui, la RAP est devenue largement accessible au grand public, notamment grâce à diverses applications sur téléphones et ordinateurs. Tous les smartphones sont désormais équipés de leurs propres assistants vocaux, qui deviennent de plus en plus performants, avec l'intégration récente de l'«IA» sur les différents appareils (ici, le terme «IA» se réfère plutôt aux outils de génération de textes ou d'images, comme ChatGPT ou Midjourney).

L'architecture Wav2Vec2 a révolutionné ce sous-domaine du traitement automatique de la parole. Beaucoup de modèles état de l'art s'appuient désormais sur ce type de modèle, souvent couplé avec une autre approche : la Classification Temporelle Connexionniste (CTC pour *Connectionist Temporal Classification*). La CTC, présentée par [Graves et al., 2006], est un algorithme permettant d'entraîner des réseaux de neurones profonds dans des tâches où l'alignement entre les séquences d'entrée et de sortie n'est pas connu. Dans le cadre de la RAP, la CTC permet d'aligner un phone avec sa représentation dans la transcription associée, obtenant ainsi l'alignement le plus fin possible entre l'audio et la transcription. Cela a permis de franchir un nouveau cap dans la transcription automatique de la parole.

Les modèles de RAP, basés sur des modèles de langues pré-entraînés, existent en versions monolingues et plurilingues. Les deux fonctionnent de manière similaire, mais le modèle plurilingue détecte d'abord la langue parlée avant de transcrire la parole. Ces systèmes utilisent généralement un dictionnaire par langue, associant l'orthographe aux différentes prononciations/représentations sonores possibles. Cependant, cette approche est limitée aux langues possédant une forme écrite. Or, la grande majorité des langues du monde n'ont pas de système d'écriture standardisé. Comment, dès lors, retranscrire une langue qui ne s'écrit pas ?

De plus, les systèmes de transcription, qui sont finalement tous plus ou moins monolingues, peuvent rencontrer des difficultés en cas de code-switching, c'est-à-dire lorsque des mots d'une autre langue sont utilisés au sein d'un énoncé. Pour résoudre ce problème, des modèles spécifiques au code-switching, entre deux paires de langues, ont été proposés par [Dhawan et al., 2023].

Une autre option, capable de répondre aux deux problématiques que sont l'absence d'écriture et le code-switching, pourrait être de transcrire non pas vers l'écriture d'une langue spécifique, mais de représenter les sons de la parole de manière standardisée, indépendamment de l'existence d'un système d'écriture pour la langue en question. [Taguchi et al., 2023] proposent de développer un modèle universel capable de retranscrire n'importe quelle langue en utilisant l'Alphabet Phonétique International (API). Ce modèle, utilisé dans nos expériences, sera plus amplement détaillé dans le chapitre 4.

1.2.3 Mesures d'évaluation

Les mesures d'évaluation traditionnelles en RAP sont le WER (Word Error Rate) et le CER (Character Error Rate), qui mesurent respectivement le taux d'erreur du

modèle au niveau du mot ou du caractère. Pour les transcriptions en API, on a également introduit le PER (Phone Error Rate), qui mesure le taux d'erreur au niveau du phone (un phone pouvant être composé de plusieurs caractères). Ces mesures se basent sur la distance de Levenshtein, proposée par [Levenshtein, 1965]. Elle comptabilise le nombre minimum d'opérations nécessaires pour transformer une chaîne de caractères en une autre. Il existe trois types d'opérations :

- La suppression : un caractère est supprimé entre la première et la seconde chaîne de caractères. Par exemple, entre *maire* et *mare*, le i a été supprimé.
- L'insertion : un caractère est inséré entre la première et la seconde chaîne de caractères. Par exemple, entre *mare* et *mares*, le s a été ajouté à la fin de la chaîne de caractères.
- La substitution : le n$^{\text{ème}}$ caractère des deux chaînes de caractères est différent. Par exemple, entre *mère* et *mare*.

Traditionnellement, chacune de ces opérations a le même poids.

Formellement, on définit la distance de Levenshtein de la manière suivante :

$$lev(a,b) = \begin{cases} max(|a|,|b|) & \text{si } min(|a|,|b|) = 0, \\ lev(a[1:],b[1:]) & \text{si } a[0] = b[0], \\ 1 + min \begin{cases} lev(a[1:],b) \\ lev(a,b[1:]) \\ lev(a[1:],b[1:]) \end{cases} & \text{sinon.} \end{cases} \tag{1.1}$$

où a et b représentent les deux chaînes de caractères que nous souhaitons comparer, $|a|$ le cardinal de a (c'est-à-dire son nombre de caractères), et $a[1:]$ la chaîne a tronquée de son premier caractère $a[0]$.

Cependant, ces mesures d'évaluation ne prennent pas en compte les similarités phonétiques entre deux phones. Elles accordent, par exemple, le même poids à une erreur entre deux consonnes ayant des lieux d'articulation différents et à une erreur entre deux consonnes dont la phonation est proche. Or, pour entraîner un modèle capable de détecter non pas une langue et des mots, mais des sons, il est crucial de considérer leurs caractéristiques acoustiques dans les mesures d'évaluation. [Mortensen et al., 2016] proposent plusieurs mesures prenant ces paramètres en compte, avec des niveaux de finesse différents. Parmi les différentes mesures proposées, nous avons sélectionné le Feature Error Rate (FER) et ce que nous appellerons ici le Dolgopolsky's Classes Error Rate (DCER).

Pour la première mesure d'évaluation, le FER, les auteurs ont défini un ensemble de 22 caractéristiques phonologiques, principalement articulatoires, présentées dans le tableau A.2. Ils ont également établi un ensemble de règles concernant les diacritiques et les modificateurs[6], ainsi que leurs impacts sur les caractéristiques de base des phones. Ils ont par ailleurs dressé un tableau[7] recensant 6368 phones et les traits phonologiques qui leur sont associés.

6. L'ensemble des règles est disponible ici : https://github.com/dmort27/panphon/blob/master/panphon/data/diacritic_definitions.yml
7. Disponible ici : https://github.com/dmort27/panphon/blob/master/panphon/data/ipa_all.csv

Pour calculer le FER, on détermine d'abord la distance minimale d'édition entre les caractéristiques phonologiques des phones constituant une chaîne de référence et une chaîne hypothétique. Cette distance est ensuite divisée par le nombre total de phones dans la chaîne de référence, ce qui permet d'obtenir un taux d'erreur reflétant la similarité phonologique entre les deux chaînes de phones.

Leur deuxième méthode d'évaluation, le DCER, s'appuie sur les classes d'équivalence décrites par [Dolgopolsky, 1964]. Celui-ci avait créé ces classes dans le cadre d'une étude sur les évolutions phonologiques des langues et constatait que des changements phonologiques avaient plus de chances de se produire au sein d'une même classe qu'entre deux classes différentes (par exemple, un /d/ a plus de chances de devenir un /t/ qu'un /m/). À ces classes, [Mortensen et al., 2016] ont associé les caractéristiques phonologiques précédemment définies. Chaque phone, en fonction de ses caractéristiques phonologiques, est donc remplacé par un label correspondant à une de ces classes, présentées dans le tableau A.1. Une fois la chaîne de caractères transformée en suite de labels, la distance de Levenshtein est appliquée puis divisée par la longueur de la chaîne de caractères la plus longue pour obtenir un score entre 0 et 1.

Il faut tout de même noter que les variations de voyelles ne sont pas prises en compte par cette mesure. En effet, Dolgopolsky estimait que les voyelles étaient trop instables et pouvaient présenter des divergences trop importantes entre différentes langues pour que leur catégorisation puisse être représentative.

La table 1.1 illustre les limites des mesures traditionnelles utilisées en RAP et met en avant l'intérêt des nouvelles mesures proposées.

Tokens en API	CER	PER	FER	DCER
[tabl] vs [fabl]	0,25	0,25	0,046875	0,25
[pavi] vs [papi]	0,25	0,25	0,03125	0,0
[bɔ̃ʒuʁ] vs [boʒuʁ]	0,33	0,2	0,0167	0,0
[bɔ̃ʒuʁ] vs [bɔ̃ðuʁ]	0,167	0,2	0,0167	0,2

TAB. 1.1 : Mesures d'évaluation sur quelques mots d'exemples

On observe d'abord que le taux d'erreur de caractères (CER) et le taux d'erreur de phones (PER) sont identiques pour les paires [tabl] (table) / [fabl] (fable) et [pavi] (comme dans **pavi**llon) / [papi] (comme dans **papi**llon). En revanche, le FER et le DCER varient.

Pour le premier exemple, le lieu et le mode articulatoire de [t] et [f] étant différents, le DCER est de 0,25. Pour le deuxième exemple, le DCER est de 0,0 car les consonnes [p] et [v] sont toutes deux des consonnes obstruantes labiales d'après les classes de Dolgopolsky. De même, on note un FER légèrement supérieur entre [t] et [f] qu'entre [p] et [v]. Cela s'explique par le fait que le premier couple de consonnes présente plus de caractéristiques divergentes que le second, comme le montre la table des caractéristiques de chaque phone sus-mentionnée.

En revanche, on obtient le même FER pour les couples [bɔ̃ʒuʁ] / [boʒuʁ] et [bɔ̃ʒuʁ] / [bɔ̃ðuʁ] car les deux couples de phones qui divergent entre les deux ([ɔ̃]/[o] et [ʒ]/[ð]) ont chacun 2 traits acoustiques différents, auxquels le même poids est attribué. Les divergences de voyelles n'étant pas prises en compte par les classes de Dolgopolsky,

on obtient à nouveau un score de 0,0 pour le premier couple contre un score de 0,2 pour le second, composé de deux consonnes appartenant à deux classes différentes.

Conclusion

Dans ce chapitre, nous avons exploré le traitement naturel et artificiel de la parole. Nous avons commencé par examiner comment le langage parlé est perçu et analysé par le cerveau humain, en distinguant la langue et la parole selon la théorie de Saussure. Nous avons décrit les mécanismes physiologiques de la perception de la parole, depuis la détection des sons par l'oreille jusqu'à leur traitement par le cortex auditif et l'aire de Wernicke. Nous avons également abordé les concepts fondamentaux de la phonologie, en distinguant notamment les phones et les phonèmes.

Ensuite, nous avons présenté un état de l'art sur le traitement automatique de la parole, en soulignant les avancées technologiques récentes, notamment grâce aux réseaux de neurones. Nous avons mis en lumière l'architecture Wav2Vec2, qui permet d'entraîner des modèles performants en utilisant majoritairement des données non annotées. Cette architecture, couplée avec la Classification Temporelle Connexionniste (CTC), a permis de franchir un nouveau cap dans la transcription automatique de la parole. Nous avons également discuté des défis spécifiques liés à la reconnaissance automatique de la parole (RAP), notamment la gestion des langues non écrites et du code-switching et les solutions possibles. Nous avons enfin exploré des mesures d'évaluation traditionnelles comme le WER, le CER et le PER, ainsi que des mesures plus avancées comme le FER et le DCER, qui prennent en compte les caractéristiques phonologiques des phones. Ces mesures permettent d'obtenir une évaluation plus fine et plus précise des performances des modèles de RAP.

CHAPITRE 2

ARMÉNIEN : LANGUE·S PLURIELLE·S

Sommaire

Introduction

L'arménien est une langue indo-européenne formant à elle-seule une branche de l'arbre phylogénétique des langues indo-européennes (voir figure 2.1). En plus de sa position isolée, l'arménien est également une langue pluri-centrique. En linguistique synchronique, on appelle langue pluri-centrique une langue ayant plusieurs variants standard. L'anglais et l'arabe sont parmi les exemples de langues pluri-centriques les plus connus. L'arménien possède deux variantes standardisées : l'arménien oriental standard (*Standard Eastern Armenian* ; SEA) et l'arménien occidental standard (*Standard Western Armenian* ; SWA). Dans un premier temps, nous présenterons les deux standards actuels de l'arménien, puis nous aborderons les principales différences phonologiques entre ces deux standards.

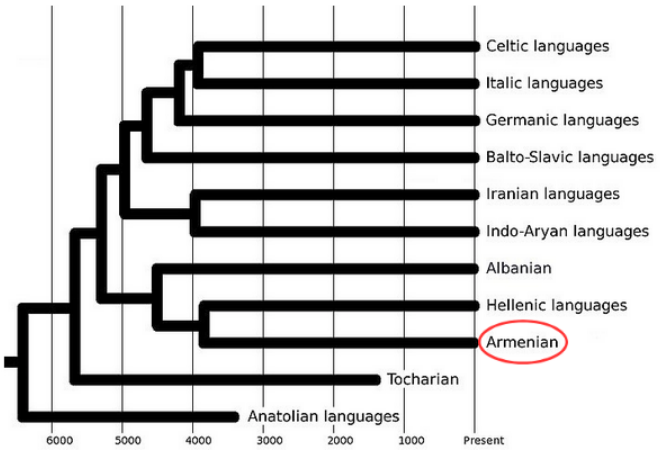

FIG. 2.1 : Relations phylogénétiques des familles majeures des langues indo-européennes, [Chang et al., 2015]

2.1 Les variants standards de l'arménien

La division entre l'Arménie Orientale et l'Arménie Occidentale remonte à 387, lorsque le territoire est partagé entre l'Empire byzantin et l'Empire sassanide (aussi connu sous le nom d'Empire des Iraniens[1]) . L'Arménie Occidentale devient alors la province d'Arménie Mineure sous l'Empire byzantin, tandis que l'Arménie Orientale reste un royaume sous influence perse.

Peu de temps après, en 405, Mesrop Machtots crée l'alphabet arménien, les différents dialectes de l'époque étant exclusivement oraux. Cet alphabet comptait à l'époque 36 lettres, contre 38 aujourd'hui. Il s'agit d'un alphabet très phonémique : on pourrait presque utiliser une fonction bijective pour passer du graphème au phonème en arménien oriental. Avec l'introduction de cette écriture, des normes et des règles pour écrire la langue apparaissent. C'est la première forme standardisée de l'arménien, qu'on appelle aujourd'hui «arménien classique» mais qui porte également le nom de *Grabar*, signifiant «langue écrite». Les seuls dictionnaires et grammaires arméniens de l'époque étaient donc en arménien classique.

Cependant, cette langue était inaccessible à la majorité de la population, qui était illettrée. Celle-ci continuait d'utiliser les différents dialectes arméniens pour communiquer. Les dialectes orientaux et occidentaux ont donc évolué indépendamment, chacun subissant des influences linguistiques de langues différentes. Les zones de localisation des différents dialectes arméniens sont présentées sur la figure 2.2. Les dialectes en orange correspondent approximativement aux dialectes de l'arménien occidental, tandis que ceux en vert correspondent plutôt aux dialectes orientaux.

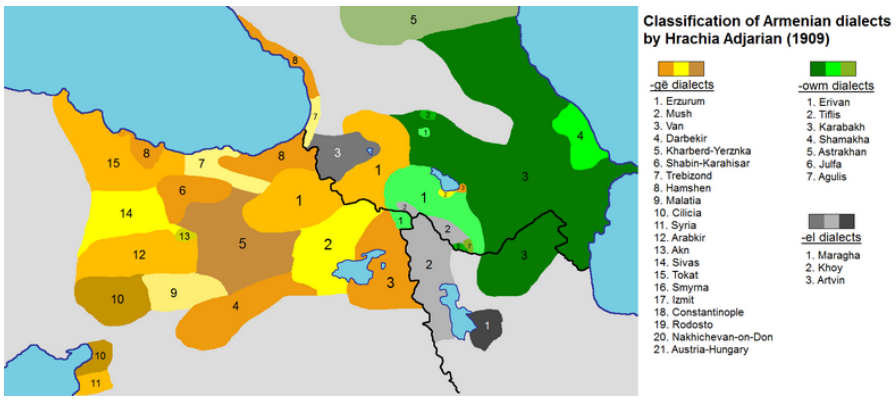

FIG. 2.2 : Carte des dialectes arméniens au début du XXème siècle, [Adjarian, 1909].

Au début du XIX^{e} siècle, avec l'émergence de la linguistique, on observe une dynamique globale de création de langues modernes. Les élites intellectuelles arméniennes de l'époque commencent à s'intéresser davantage à la langue arménienne, avec la volonté de créer une langue unifiée pour pouvoir fonder une nation arménienne. Nous utiliserons les termes «arménien occidental» et «arménien oriental» pour désigner les deux variants standards.

2.1.1 L'arménien occidental standard (SWA)

[Manoukian, 2023] s'est intéressée à la genèse de l'arménien occidental dans l'Empire ottoman. Entre le XVIII^{e} et le XIX^{e} siècle, on voit émerger une troi-

1. https://fr.wikipedia.org/wiki/Sassanides

sième variante de l'arménien, s'ajoutant à l'arménien classique et aux différents dialectes : l'arménien moderne. Il s'agit d'un intermédiaire entre l'arménien classique et l'arménien dialectal de Constantinople, la capitale ottomane. Cette nouvelle variante, au début uniquement écrite, prend peu à peu la place de l'arménien classique, qui n'est plus utilisé que pour les écrits liturgiques à la fin du XIX^{e} siècle. Les dialectes continuent d'être utilisés pour la communication orale. Dans les années 1890, les premières grammaires et les premiers dictionnaires d'arménien moderne sont publiés, permettant désormais son enseignement à l'école. Cet arménien moderne, alors uniquement enseigné dans les écoles arméniennes de l'Empire ottoman, s'est ainsi standardisé pour devenir l'arménien occidental actuel.

Les langues standardisées sont quelque peu artificielles et influencées par les personnes qui les ont prescrites. L'arménien occidental est le résultat de décisions assez arbitraires faites par les élites intellectuelles de l'époque, dans une volonté de créer une langue «pure», sans influence de langues étrangères comme le turc, l'arabe ou le persan. Le vocabulaire de l'arménien occidental actuel est donc un mélange d'arménien classique et de mots du dialecte arménien de Constantinople. Les mots absents de ces deux variantes ont été empruntés à différents dialectes, notamment pour le vocabulaire ayant trait à la vie rurale, à la nature, etc.

L'arménien occidental et les différents dialectes occidentaux ont continué d'être utilisés conjointement jusqu'au génocide de 1915. Suite au départ de la majorité de la communauté arménienne de l'Empire ottoman vers divers pays, des locuteurs de différents dialectes se sont retrouvés à vivre ensemble. Les dialectes occidentaux ont alors peu à peu été abandonnés au profit de l'arménien occidental standard, compréhensible par tous. C'est l'unique variante enseignée dans les écoles arméniennes des différentes communautés diasporiques.

2.1.2 L'arménien oriental standard (SEA)

L'émergence de l'arménien oriental est moins documenté que celle de l'arménien occidental. Il a été formé à partir des dialectes de la plaine d'Ararat, notamment à partir du dialecte de Yerevan, mais également à partir de l'arménien classique. En raison des influences perses puis russes dans cette région, on retrouve également de nombreux emprunts de ces deux langues dans l'arménien oriental actuel.

L'arménien oriental a connu une réforme orthographique sous le régime soviétique en 1922, qui fut partiellement révisée en 1940. Cette réforme orthographique est encore appliquée en Arménie et dans la communauté arménienne en Russie, tandis que la communauté arménienne d'Iran utilise l'écriture traditionnelle. Contrairement aux dialectes occidentaux, les dialectes orientaux sont encore parlés dans les différentes régions d'Arménie.

2.2 Les différences phonologiques entre l'arménien oriental et l'arménien occidental

L'écriture arménienne est grandement phonologique. Seulement, bien que l'arménien oriental et l'arménien occidental aient pour base commune l'arménien classique, qui a donc donné naissance à l'écriture arménienne, leur prononciation respective s'appuie principalement sur des dialectes ayant évolué indépendamment depuis le V^{e} siècle (l'arménien classique étant uniquement une langue écrite). Il n'est

donc pas surprenant que des différences phonologiques significatives existent entre ces deux standards.

D'après [Chakmakjian and Wang, 2022], ces différences résident principalement dans le système consonantique. Dans les deux variantes, celui-ci est constitué de six classes de consonnes : les consonnes nasales, fricatives, spirantes, rhotiques, occlusives et affriquées. Les différences entre les deux standards se trouvent dans les trois dernières catégories de consonnes :

- Les consonnes rhotiques
  - L'arménien oriental distingue deux consonnes rhotiques : /ɾ/ et /ɾ̪/.
  - L'arménien occidental ne fait pas de distinction entre les deux.
- Les consonnes occlusives et affriquées
  - L'arménien oriental distingue trois phonations : voisée, non voisée et non voisée aspirée.
    - /b, p, p^h/
    - /d, t, t^h/
    - /g, k, k^h/
    - /ʣ, ʦ, $ʦ^h$/
    - /ʤ, ʧ, $ʧ^h$/
  - L'arménien occidental ne distingue que deux phonations : voisée et non voisée aspirée.
    - /b, p^h/
    - /d, t^h/
    - /g, k^h/
    - /ʣ, $ʦ^h$/
    - /ʤ, $ʧ^h$/

Nous avons donc 30 consonnes en arménien oriental contre 24 en arménien occidental. Le tableau 2.1, proposé par [Chakmakjian and Wang, 2022], illustre clairement ces différences phonologiques et les ambiguïtés qu'elles peuvent engendrer. En effet, la phonologie occidentale permet la présence d'homophones, ce qui n'est pas le cas en arménien oriental.

Token	SEA	SWA	Traduction
բառ	[**b**ar]	[**p^h**ar]	'mot'
պարի	[**p**ar]	[**b**ar]	'danse'
փառ	[**p^h**ar]	[**p^h**ar]	'placenta'

TAB. 2.1 : Trois mots et leurs prononciations en arménien oriental (SEA) et en arménien occidental (SWA)

Il faut également noter que, l'arménien occidental n'étant parlé que par des communautés diasporiques, cela peut entraîner des variations phonétiques importantes au sein même des locuteurs de cette variante. Contrairement aux locuteurs de

l'arménien oriental, les locuteurs de l'arménien occidental ont (au minimum) une autre langue première. Tout comme cela influence la manière dont les locuteurs structurent la langue, comme l'indique [Manoukian, 2023] qui trouve fréquemment des calques d'expressions idiomatiques françaises, turques ou arabes dans les textes qu'elle traduit, cela influence également la manière dont la langue est prononcée. Bien qu'il n'y ait qu'un arménien occidental standard, sa production évolue différemment dans les diverses communautés arméniennes, sous l'influence de la langue véhiculaire et des éventuelles langues vernaculaires locales. Ainsi, bien que le système vocalique soit le même entre les deux variantes, avec les six voyelles /i, e, a, o, u, ə/, leur production peut varier sensiblement, même au sein des locuteurs de l'arménien occidental.

Conclusion

Dans ce chapitre, nous avons exploré la pluralité de la langue arménienne. Nous avons présenté les deux variantes standardisées de l'arménien : l'arménien oriental standard (SEA) et l'arménien occidental standard (SWA), ainsi que leurs principales différences phonologiques. Nous avons également souligné les influences possibles des autres langues parlées par les communautés diasporiques, tant sur la structure que sur la prononciation de la langue.

partie II

Méthode

CHAPITRE 3

CORPUS

Sommaire

Introduction

Dans ce chapitre, nous commencerons par présenter le corpus utilisé pour entraîner et évaluer nos modèles de transcription automatique de la parole vers l'API. Ensuite, nous détaillerons les prétraitements effectués pour rendre les données exploitables. Enfin, nous présenterons les différents jeux de données constitués pour nos expériences.

3.1 Présentation du corpus

Les données utilisées pour l'entraînement de nos modèles proviennent du projet Rerooted[1], créé en 2017 et porté par deux membres de la diaspora arménienne aux États-Unis : Anoush Baghdassarian et Ani Schug. Leur double objectif, à travers ce projet, est de réunir des témoignages de nombreuses communautés arméniennes et de créer du matériel pédagogique à partir de ces enregistrements pour préserver et perpétuer l'apprentissage de l'arménien occidental. Aujourd'hui, plus de 25 communautés arméniennes ont pris part à ce projet.

Ce projet a débuté au sein de la communauté syro-arménienne, où de nombreux témoignages ont été recueillis, notamment sur les conditions de vie liées à la guerre, les traumatismes associés, mais également sur la communauté arménienne elle-même et la manière dont son identité s'est construite depuis près d'un siècle. Une partie de ces témoignages constitue notre corpus.

Ces témoignages sont sous la forme d'entretiens, filmés puis mis sur YouTube avec les sous-titres correspondants. Les transcriptions que nous avons récupérées sont les sous-titres de ces vidéos. Il est à noter que l'objectif premier de ces transcriptions n'était pas linguistique, mais plutôt didactique. En effet, l'un des objectifs du projet

1. https://www.rerooted.org/

Rerooted est d'utiliser ces vidéos et transcriptions dans un but d'apprentissage de la langue via des exercices de compréhension orale et écrite. Il est donc très probable que les transcriptions aient subi des modifications par rapport à ce qui est dit par les locuteurs, sans parler des éventuels défauts de prononciation qui n'ont pas été retranscrits.

3.2 Pré-traitements des données

Pour pouvoir exploiter ce corpus, nous avons dû effectuer plusieurs pré-traitements. Dans un premier temps, les fichiers audio, initialement au format MP3, ont été convertis en fichiers WAV avec une fréquence d'échantillonnage de 16 000 Hz, un format et une fréquence pris en charge par Wav2Vec2. Les sous-titres des témoignages ont également été convertis en TextGrid, et leur alignement avec les audios a été corrigé manuellement[2]. À partir de ces fichiers TextGrid, nous avons segmenté les fichiers audio en suivant les intervalles plutôt que les ponctuations finales, celles-ci n'étant pas toujours explicitées, ce qui entraînait des segments trop longs pour être traités. Nous avons également généré un fichier texte rassemblant le nom de chaque fichier audio nouvellement créé et la transcription correspondante.

3.2.1 Translittération en API

Les transcriptions étant en arménien occidental, nous avons dû les convertir en API pour pouvoir entraîner nos modèles. Cette conversion a été réalisée à l'aide d'un transducteur morphologique appelé FOMA[3] (pour Free/Open-source Morphological Analyzer). Un transducteur est un automate à états finis avec des sorties. Il effectue des transformations basées sur des règles, permettant de convertir une chaîne de caractères d'un alphabet d'entrée en une chaîne de caractères d'un alphabet de sortie, de manière déterministe ou non.

FOMA, originellement utilisé pour l'analyse morpho-syntaxique, peut également servir à la translittération. Dans notre cas, il s'agit de convertir l'écriture arménienne occidentale en API. Étant donné que l'écriture arménienne est phonologique, une approche par règles pour la translittération vers l'API semblait tout à fait appropriée. Le modèle de translittération de l'arménien occidental vers l'API a été développé par Samuel Chakmakjian, doctorant ERTIM et SeDyL, spécialisé en phonologie. Bien que ce modèle ne soit pas encore totalement achevé, nous l'avons utilisé tel quel, faute d'autres possibilités.

Comme expliqué dans le chapitre 2, l'arménien occidental et l'arménien oriental présentent des différences phonologiques notables. L'une des différences majeures est la présence d'homophones en arménien occidental, c'est-à-dire de mots ayant la même prononciation mais une écriture différente, leur lexique phonologique étant moins riche que celui de l'arménien oriental. Le modèle de translittération pour l'arménien occidental est donc plus complexe. Actuellement, il comprend 75 règles permettant la translittération de l'alphabet vers l'API. Un autre modèle pour l'arménien oriental est également en cours de conception.

Le corpus utilisé contenant des segments présentant du code-switching, c'est-à-dire utilisant des mots ou expressions dans une autre langue, ces parties ont été

2. Données disponibles sur GitHub : https://github.com/jhdeov/ReRooted-ArmenianCorpus
3. Documentation accessible sur : https://fomafst.github.io/

Corpus	«Mots» alphabétiques	«Mots» phonémiques	Caractères alphabétiques	Caractères phonémiques
Rerooted	66 179	66 375	383 950	418 275

TAB. 3.1 : Nombre de mots et de caractères des transcriptions alphabétiques et phonologiques

translittérées à l'aide du site International Phonetic Alphabet[4], qui propose entre autre un outil de translittération pour le français et l'anglais.

Le tableau 3.1 présente le nombre de caractères et de mots dans l'ensemble du corpus Rerooted avant et après translittération. Ces comptes ont été obtenus à l'aide de la commande bash «wc» à partir des fichiers contenant les transcriptions en alphabet arménien et les transcriptions en API, générées grâce au modèle FOMA. Les différences observées, notamment entre le nombre de caractères alphabétiques et le nombre de caractères phonémiques, s'expliquent par le fait que l'écriture arménienne est très phonémique. En effet, une lettre équivaut généralement à un phonème, qui peut, lui, être constitué de plusieurs caractères (comme /dz/ ou /tʃʰ/).

3.2.2 Application d'un détecteur de variants

Après la transcription en API, nous avons appliqué un classificateur de langues à nos données. Nous avons d'abord testé le modèle Massively Multilingual Speech (MMS), développé par [Pratap et al., 2023]. Ce modèle, basé sur Wav2Vec2, est conçu pour diverses tâches, notamment la détection de langues, y compris l'arménien oriental (hye) et l'arménien occidental (hyw). Cependant, lors du pré-entraînement de ce modèle, les différents dialectes d'une même langue ont été combinés, ce qui ne permet pas de distinguer l'arménien oriental de l'arménien occidental ou d'autres variantes non standard. Lorsque nous avons testé ce modèle sur nos données en arménien occidental, il ne détectait effectivement que de l'arménien oriental ou d'autres langues aberrantes comme le japonais ou le breton.

Dans un contexte multi-variant, [Chakmakjian and Wang, 2022] proposaient d'utiliser soit une approche multilingue, soit une approche par détecteur de variantes de l'arménien. La première approche n'ayant pas été concluante avec les modèles actuellement disponibles, nous avons opté pour la seconde.

Par conséquent, un détecteur de variantes de l'arménien a été développé par Johanna Cordova, doctorante ERTIM, dans le cadre du projet DALiH. Ce détecteur repose sur un modèle de mélange gaussien (GMM), un modèle statistique basé sur une densité de mélange, proposé par [Sarmah and Bhattacharjee, 2014]. Ce type de modèle suppose que nos données contiennent plusieurs groupes, ici l'arménien oriental et l'arménien occidental, chacun ayant sa propre distribution gaussienne. Les paramètres sont initialisés de manière aléatoire, puis la probabilité conditionnelle qu'un audio appartienne à un groupe est calculée. Les paramètres sont ensuite mis à jour pour maximiser la vraisemblance des données, et cette opération se répète jusqu'à convergence.

Notre modèle a été entraîné sur 5 heures d'arménien oriental et 4 heures et 40 minutes d'arménien occidental, obtenant un taux d'exactitude d'environ 60% sur des

4. https://www.internationalphoneticalphabet.org/

Variant détecté	Rerooted (occidental)	Corpus oriental
hye	00 : 12 : 59	09 : 31 : 35
hyw	07 : 38 : 33	03 : 33 : 42

TAB. 3.2 : Résultat du détecteur de variants standard sur nos corpus

données totalement différentes, avec une sur-prédiction de l'arménien occidental. Parmi ses données d'entraînement figurent des fichiers extraits de Rerooted, mais nous avons veillé à ce que les données transcrites utilisées dans notre corpus ne soient pas incluses. Le jeu de données a également été enrichi de divers podcasts et émissions pour le rendre plus hétérogène en termes de qualité audio, afin d'éviter que le modèle ne se base uniquement sur des éléments extralinguistiques.

Une fois entraîné, ce détecteur a été appliqué à nos données occidentales et à quelques données orientales issues de corpus rassemblés dans le cadre du projet DALiH, ainsi que de livres audio en arménien disponibles en ligne[5]. Les résultats obtenus sont présentés dans la table 3.2. Nous constatons que le corpus Rerooted est très largement classé occidental, tandis que dans le corpus oriental, 25% des données sont classées comme occidental. Ces résultats confirment une sur-prédiction de l'occidental, sans exclure la possibilité que certains segments du corpus oriental soient occidentaux ou difficiles à déterminer.

Des modèles avec des jeux de données plus conséquents et un meilleur score d'exactitude ont également été développés, mais lorsque nous les testions sur nos corpus, cela avait un impact très négligeable sur le corpus d'arménien occidental et l'arménien occidental semblait être encore plus sur-prédit sur nos données en oriental. De plus, les données ajoutées pour l'arménien oriental provenaient pour beaucoup de source «en arménien», sans spécification du variant. Enfin, nous avons effectué des calculs de spécificités sur les données détectées comme étant de l'oriental et de l'occidental par le premier modèle. La cohérence des résultats a été confirmé par un locuteur de ces langues, avec la présence de termes purement orientaux ou occidentaux dans les données détectées comme telles, et la présence de termes difficilement dissociables d'un point de vue phonologique (mots constitués uniquement de phonèmes communs aux deux standards).

L'ensemble des prétraitements des données est présenté par la figure 3.1.

3.3 Constitution des jeux de données

Le détecteur attribue à chaque fichier audio deux scores logarithmiques de vraisemblance : l'un pour l'arménien oriental et l'autre pour l'arménien occidental. En faisant la différence entre ces deux scores, nous obtenons une mesure qui se rapproche d'un score de confiance du modèle pour estimer si un fichier est en arménien oriental ou occidental. Puisque nous utilisons uniquement les données détectées comme étant en arménien occidental, ce score peut également être vu comme un indicateur de l'«occidentalisation» du fichier audio.

Nous avons constitué différents jeux de données à partir de ces scores (voir figure 3.2). Dans un premier temps, nous avons créé un jeu de test unique pour comparer

5. https://www.openslr.org/154/

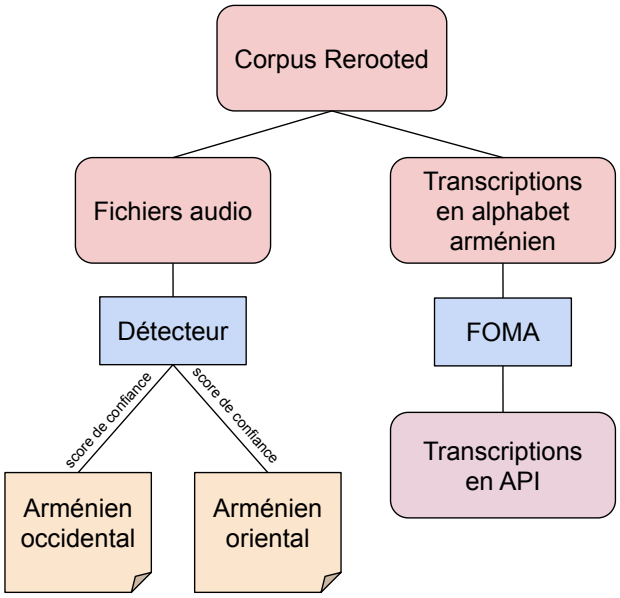

FIG. 3.1 : Pré-traitements des données

de manière objective nos différents modèles. Nous disposons également d'un petit jeu de données de 26 fichiers audio, transcrits manuellement en API par Samuel Chakmakjian à partir des audios et des spectrogrammes, pour plus d'objectivité. Ensuite, nous avons constitué plusieurs sous-ensembles de données à partir des fichiers audio restants, en nous basant sur le score attribué par le détecteur. Nos scores variant de 0,0004 à 8,559 avec un score médian à 4,4, nous avons établi deux seuils :

- $\geqslant 4,4$: on considère que le détecteur est confiant dans sa détection
- $< 2,6$: on considère que le détecteur est peu confiant dans sa détection

Nous avons constitué deux sous-ensembles de jeux de données : l'un de 10 000 secondes (environ 2h47), composé de fichiers audio de plus d'une seconde, et un autre de 11 000 secondes (environ 3h03), composé de fichiers audio de plus d'une demi-seconde.

Ces deux sous-ensemble de jeux de données nous permettront de tester qualitativement diverses configurations pour entraîner nos modèles. Nous pourrons ainsi évaluer si les données détectées comme très «occidentales» améliorent l'apprentissage. Nous examinerons également si un jeu de données homogène facilite l'apprentissage ou si, au contraire, un jeu de données plus hétérogène est plus bénéfique. Enfin, nous testerons si la durée minimale des fichiers audio a un impact sur l'entraînement. Ces expérimentations nous aideront à mieux comprendre les facteurs influençant la performance de nos modèles et à optimiser leur entraînement.

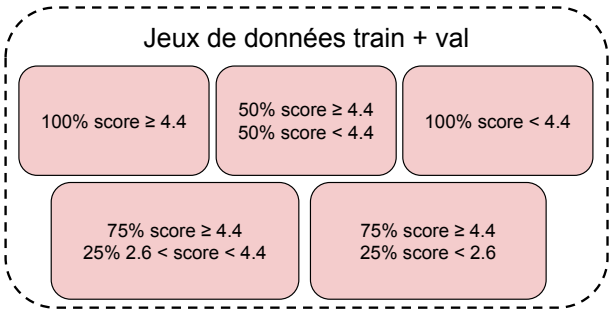

FIG. 3.2 : Composition des deux sous-ensembles de jeux de données

CHAPITRE 4

MODÈLES

Sommaire

Introduction

Comme mentionné précédemment, l'architecture Wav2Vec2 a révolutionné le domaine du traitement automatique de la parole. Les modèles de reconnaissance automatique de la parole (RAP) état de l'art s'appuient pour la plupart sur cette architecture, couplée à un algorithme CTC. De plus, les modèles Wav2Vec2 sont particulièrement pertinents pour les langues peu dotées, nécessitant peu de données annotées pour obtenir des performances satisfaisantes. L'arménien occidental étant une langue peu dotée en ressources, l'utilisation de modèles basés sur Wav2Vec2 nous semblait donc particulièrement appropriée. Nous avons testé deux modèles : le modèle Multipa, proposé par [Taguchi et al., 2023], affiné pour transcrire des fichiers audio vers l'API, et le modèle pré-entraîné Wav2Vec2 XLS-R de Facebook ([Babu et al., 2021]).

4.1 Le modèle Multipa

Le modèle Multipa, développé et présenté par [Taguchi et al., 2023], vise à être un modèle universel (*language-universal*) plutôt qu'un modèle dépendant de la langue (*language-dependant*), comme c'est le cas pour les modèles traditionnels de RAP.

Ce modèle s'appuie sur le modèle pré-entraîné Wav2Vec2-XLSR-53, proposé par [Conneau et al., 2020] et entraîné sur 56 000 heures d'audio provenant des corpus Common Voice, BABEL et Multilingual LibriSpeech, dans 53 langues différentes (aucun variant de l'arménien n'étant inclus). Multipa a affiné ce modèle pour retranscrire des fichiers audio dans n'importe quelle langue vers leur représentation phonétique. Ils ont également intégré l'algorithme CTC pour améliorer l'apprentissage grâce à la fonction de perte CTC.

Multipa a été affiné sur sept langues, phonétiquement éloignées, pour obtenir un jeu de phones le plus diversifié possible : le japonais, le polonais, le maltais, le hongrois, le finnois, le grec et le tamoul. Ces sept langues ont également la particularité d'avoir une écriture très phonétique. Leurs données proviennent de Common Voice et sont retranscrites en API de manière semi-automatique. Plusieurs outils de trans-

littération ont été testés manuellement, et finalement, seuls Epitran[1] et des outils à base de règles développés par les auteurs ont été retenus. Ce choix des langues et des outils de translittération distingue Multipa de Wav2Vec2Phoneme, précédent modèle état de l'art, visant quant à lui à retranscrire les audios vers une forme phonémique plutôt que phonétique.

Dans leur article, les auteurs comparent les résultats de Multipa à ceux d'autres modèles permettant la transcription vers l'API, tels que Wav2Vec2Phoneme et Allosaurus, un modèle de reconnaissance automatique de phones, capable de reconnaître les phones de plus de 2 000 langues ([Li et al., 2020]). Nous reportons ces résultats dans la table 4.1.

Métrique	Modèle	Luganda	Haut sorabe	Hakha Chin	Tatar	Moyenne
PER(%)	Allosaurus	104,1	93,9	79,4	89,6	91,8
	Wav2Vec2Phoneme	64,0	66,1	70,0	63,0	65,8
	Multipa 1000	74,0	68,6	73,0	67,7	70,8
	Multipa 2000	77,0	69,4	72,7	67,5	71,6
	Multipa complet	70,9	52,5	55,3	52,7	**63,2**
PFER(%)	Allosaurus	46,1	36,3	36,3	30,1	34,2
	Wav2Vec2Phoneme	24,2	26,1	19,3	20,0	22,4
	Multipa 1000	20,8	24,0	21,3	18,8	**21,2**
	Multipa 2000	22,7	24,9	21,8	19,4	22,2
	Multipa complet	23,0	23,1	20,3	18,8	21,3

TAB. 4.1 : Performances des différents modèles Multipa et des modèles état de l'art sur des données hors-domaine, telles que présentées dans [Taguchi et al., 2023]

Les résultats de PER (Phone Error Rate) obtenus par Multipa sur des langues ne faisant pas partie de son corpus d'entraînement (*out-of-domain*) sont comparables à ceux de Wav2Vec2Phoneme (63,2 pour Multipa contre 65,8 pour Wav2Vec2Phoneme) et bien meilleurs que ceux d'Allosaurus (91,8). En revanche, lorsqu'on examine le PFER (Phone Feature Error Rate), les résultats de Multipa semblent globalement meilleurs que ceux des deux autres modèles.

Le modèle affiné sur le moins de données (1 000 échantillons pour une durée totale d'environ 9 heures) donne de meilleurs résultats que celui entraîné sur 2000 échantillons (environ 18 heures). Le modèle entraîné sur l'ensemble des données n'étant pas accessible, nous avons donc choisi le modèle entraîné sur 1000 échantillons pour nos expériences.

4.2 Le modèle Wav2Vec2-XLS-R

Le modèle Wav2Vec2 XLS-R, proposé par Facebook, est un modèle Wav2Vec2 pré-entraîné sur près de 436 000 heures d'audio. Ces données proviennent de divers corpus disponibles en libre accès : VoxPopuli, Multilingual LibriSpeech, Common Voice, VoxLingua et BABEL (voir la figure 4.1), parmi lesquelles on compte 55 heures d'arménien. La distinction entre les deux standards ayant commencé à se faire à

1. https://github.com/dmort27/epitran

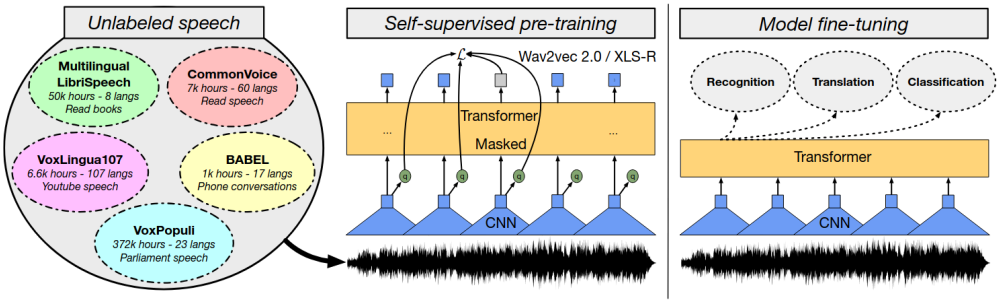

FIG. 4.1 : Apprentissage auto-supervisé de représentations multilingues

	es	fr	it	ky	nl	ru	sv	tr	tt	zh-HK	Avg
Labeled data	1h	1h	1h	1h	1h	1h	1h	1h	1h	1h	
XLSR-53	2,9	5,0	5,7	6,1	5,8	8,1	12,2	7,1	5,1	18,3	7,6
XLS-R (0,3B)	3,1	5,4	4,9	5,1	5,8	6,0	7,2	6,0	4,1	17,0	6,5
XLS-R (1B)	**2,0**	**3,9**	**3,5**	**4,1**	**4,2**	**4,1**	**5,5**	**4,4**	**3,4**	**15,7**	**5,1**

TAB. 4.2 : Comparaison des résultats de PER de W2V2-XLSR-53 et de W2V2-XLS-R sur le corpus Common Voice, extrait de [Babu et al., 2021]

partir de 2018, avec la création du code ISO pour l'arménien occidental, la distribution des standards dans ces données n'est pas précisée, mais il s'agit probablement d'arménien oriental en grande majorité.

Les performances de ce modèle sont comparées à celles du modèle Wav2Vec2-XLSR-53, qui a servi de base à Multipa. Deux tests ont été réalisés : l'un sur la transcription vers le système d'écriture des différentes langues et l'autre sur la reconnaissance de phonèmes. Pour la deuxième expérience, l'unique mesure d'évaluation proposée est le PER (Phone Error Rate). Les résultats sont présentés dans la table 4.2.

Nous constatons que les modèles Wav2Vec2-XLS-R sont généralement meilleurs que le modèle XLSR-53, notamment ceux avec plus de paramètres (nous n'avons pas représenté les performances du modèle à 2 milliards de paramètres, ne l'ayant pas inclus dans nos expériences). Lorsque nous avons testé différents modèles afin de choisir lesquels utiliser pour nos expériences, les résultats obtenus avec le modèle XLS-R à 300 millions de paramètres étaient également meilleurs que ceux obtenus avec XLSR-53. La présence d'arménien dans le jeu de données utilisé pour le pré-entraînement du premier modèle pourrait également expliquer ces différences de performances sur nos données. Aussi, parmi les modèles uniquement pré-entraînés, nous avons sélectionné les modèles XLS-R à 300 millions et 1 milliard de paramètres.

Conclusion

Dans ce chapitre, nous avons donc présenté les deux types de modèles d'analyse de la parole que nous allons utiliser dans nos expériences, tous deux basés sur l'architecture Wav2Vec2 : le modèle Multipa et les modèles Wav2Vec2-XLS-R de Facebook. Multipa, en tant que modèle universel, a montré des performances prometteuses en retranscrivant des fichiers audio en API, même pour des langues non in-

cluses dans son corpus d'entraînement. Le modèle Wav2Vec2-XLS-R, pré-entraîné sur une vaste quantité de données multilingues, a également démontré des performances supérieures aux autres modèles Wav2Vec2 disponibles. La figure 4.2 récapitule les différentes expériences que nous allons mener.

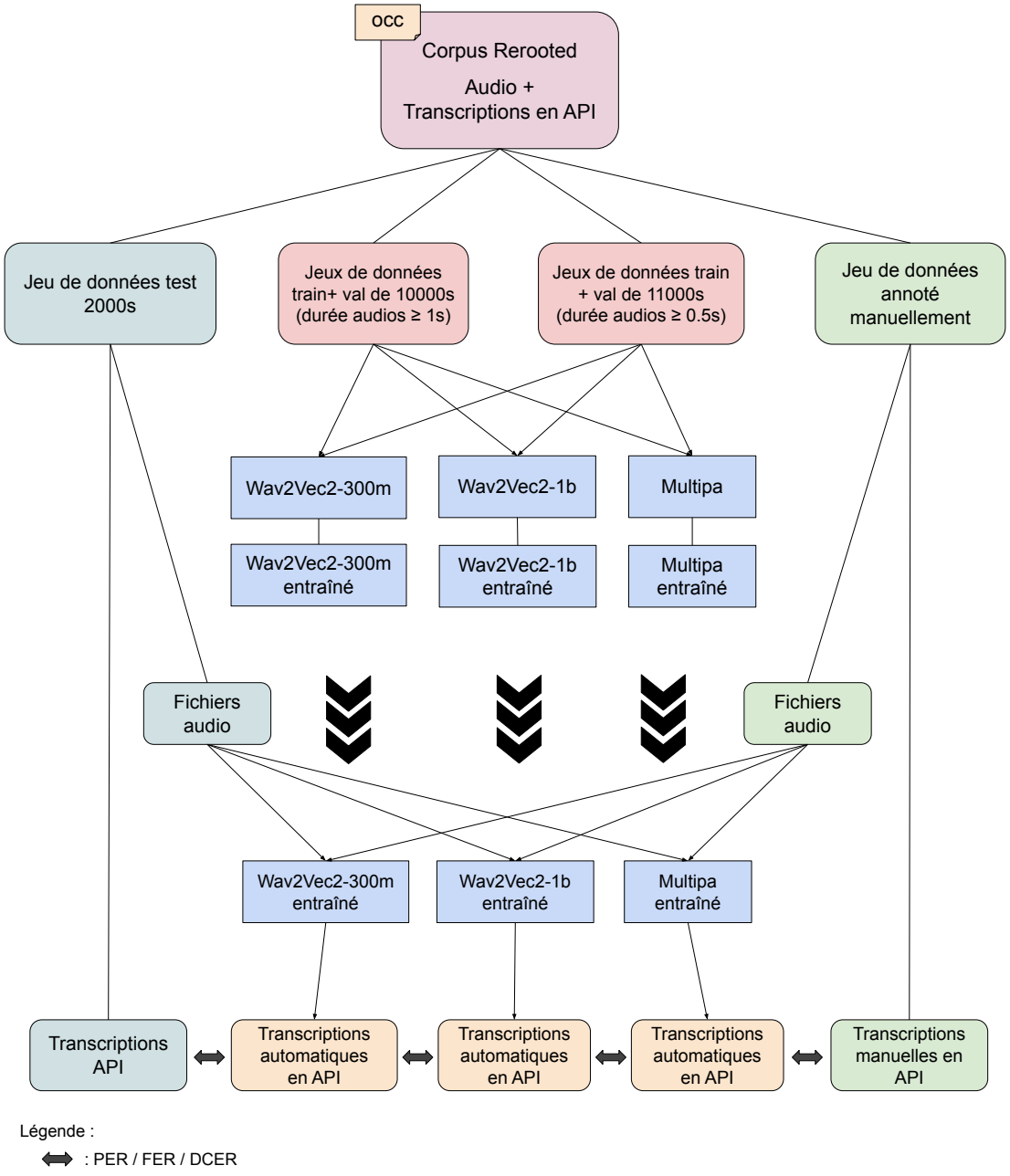

FIG. 4.2 : Présentation des expériences réalisées

partie III

Expériences

CHAPITRE 5

PRÉSENTATION DES RÉSULTATS

Sommaire

Introduction

Nous avons donc testé plusieurs configurations pour l'entraînement de nos modèles afin d'explorer deux pistes principales concernant le choix des données d'entraînement :

- L'impact de la durée des fichiers audio utilisés.
- L'impact du détecteur de variants et de l'homogénéité des données.

Dans ce chapitre, nous présenterons les hyper-paramètres retenus pour l'entraînement de nos modèles, ainsi que les résultats obtenus dans les différentes configurations d'entraînement et les observations qui en découlent.

5.1 Le choix des hyper-paramètres et des mesures d'évaluation

Dans un premier temps, nous avons réalisé des expériences préliminaires afin de déterminer les meilleurs hyper-paramètres à utiliser sur nos données. Nous avons notamment testé divers taux d'apprentissage, plusieurs configurations pour la période de préchauffage du modèle, et fait varier la taille des lots pour optimiser au mieux l'apprentissage de nos modèles.

Après ces expériences préliminaires, nous avons finalement opté pour un taux d'apprentissage unique de 2×10^{-4}. Pour les modèles Wav2Vec2-XLS-R, uniquement pré-entraînés, nous avons choisi de fixer la phase de préchauffage du modèle à 1000 étapes (*warmup steps*), contre 100 pour le modèle Multipa, déjà affiné sur une tâche de transcription de fichiers audio via l'API. Les modèles ont été entraînés sur 30 époques lorsqu'ils utilisaient les jeux de données de 11 000 secondes, car nous avons constaté que l'entraînement s'arrêtait assez rapidement. En revanche, pour les jeux

de données de 10 000 secondes, l'entraînement se poursuivait. Nous avons donc fixé un entraînement de 60 époques, mais les meilleurs résultats ont été obtenus lors des dernières époques. Par conséquent, les performances des modèles entraînés sur ce deuxième sous-ensemble de jeux de données ne sont pas optimales. Enfin, nous avons choisi une taille de lots (*batch size*) de 6, car les expériences préliminaires d'optimisation des hyper-paramètres ont montré qu'un apprentissage plus lent avec des batchs plus petits entraînait de meilleurs résultats. Par exemple, pour le modèle Wav2Vec2-XLS-R à 300 millions de paramètres, le DCER passait de 16,45% à 15,41% simplement en réduisant la taille du batch.

Nos modèles ont été évalués sur plusieurs jeux de données :

- Le corpus de validation, homogène avec le corpus d'entraînement des différents modèles.
- Le corpus de test, composé de fichiers sélectionnés au hasard parmi les fichiers détectés comme étant de l'arménien occidental au sein de notre corpus Rerooted.
- Le sous-corpus de Rerooted détecté comme étant de l'arménien oriental (les 13 minutes indiquées dans la table 3.2).

Nous avons choisi le PER, le FER et le DCER comme mesures d'évaluation principales (décrites dans la partie 1.2.3). Le PER est la mesure classique pour évaluer les modèles de transcription vers l'API. Le FER, quant à lui, est une mesure très précise qui prend en compte les 22 caractéristiques articulatoires de chaque phone (telles que définies dans la table A.2), et accorde une importance moindre aux erreurs entre phones phonologiquement proches par rapport aux erreurs entre phones plus éloignés. Enfin, le DCER est la mesure utilisée pour enregistrer le meilleur modèle. Comme mentionné précédemment, l'arménien occidental est uniquement parlé par des locuteurs ayant une autre langue première, ce qui peut influencer la prononciation. Les voyelles étant moins stables que les consonnes, l'accent se manifeste souvent dans le système vocalique et on risque donc d'avoir plus de divergences sur la transcription des voyelles. Le DCER, en ne prenant en compte que les différences consonantiques, s'avère particulièrement pertinent dans ce contexte.

L'énergie utilisée pour entraîner ces modèles étant également un critère à prendre en compte pour choisir le modèle le plus pertinent, le temps d'entraînement de chaque modèle sur chacun des jeux de données est également indiqué. L'ensemble des modèles a été entraîné sur une machine possédant une carte graphique NVIDIA L40S de 48Go. Le modèle demandant le moins de mémoire GPU est le modèle Wav2Vec2-XLS-R-300M, qui utilise entre 7 et 9 Go de mémoire lors de son entraînement. Il peut donc être entraîné sur un petit GPU. Les deux autres modèles sont plus gourmands : entre 20 et 22 Go pour Multipa et entre 24 et 26 Go pour Wav2Vec2-XLS-R-1B.

Les tableaux complets des résultats sont présentés en annexe (à partir de A.3).

5.2 L'impact de la durée des fichiers

En examinant les résultats obtenus par le modèle Wav2Vec2-XLS-R-300M (tables A.5 et A.6), nous constatons que le modèle entraîné sur 10 000 secondes (et donc sur un plus grand nombre d'époques) produit de meilleurs résultats. Cette amélioration est visible sur le jeu de validation et se confirme sur le jeu de test, ainsi que

sur les données identifiées comme étant de l'arménien oriental par notre détecteur de variants. En effet, lorsqu'on compare le modèle entraîné sur le corpus de 11000 secondes à celui entraîné sur le corpus de 10 000 secondes, les trois mesures de performance montrent des différences de près de 10 points de pourcentage entre les trois sous-corpus de test. De plus, l'écart de performance entre le jeu de validation et les autres jeux de test est réduit lorsque le modèle est entraîné sur des fichiers audio de plus d'une seconde.

Durée totale du jeu entraînement + validation	PER (%)	FER (%)	DCER (%)
10 000	**25,37**	**16,58**	**8,46**
11 000	36,65	23,72	17,19

TAB. 5.1 : Performances du modèle Wav2Vec2-XLS-R-300m sur le jeu de test, entraîné sur un jeu de 10 000 et 11 000 secondes avec les mêmes configurations de score

Cette dynamique se retrouve également pour le modèle Wav2Vec2-XLS-R-1B, avec des écarts de performance encore plus significatifs (voir tables A.7 et A.8). Par exemple, le PER passe de 43,56% pour le modèle entraîné sur 11 000 secondes à 29,87% pour celui entraîné sur 10 000 secondes.

Durée totale du jeu entraînement + validation	PER (%)	FER (%)	DCER (%)
10 000	**29,87**	**18,48**	**17,21**
11 000	51,77	35,32	17,46

TAB. 5.2 : Performances du modèle Wav2Vec2-XLS-R-1B sur le jeu de test, entraîné sur un jeu de 10 000 et 11 000 secondes avec les mêmes configurations de score

Il semble donc que la longueur des fichiers audio utilisés pour l'entraînement ait un impact significatif sur l'apprentissage et la robustesse des modèles Wav2Vec2-XLS-R. Bien que le jeu de 11 000 secondes contienne plus de données, les fichiers de moins d'une seconde semblent perturber l'apprentissage des modèles.

En revanche, le modèle Multipa présente une tendance inverse, obtenant de meilleurs résultats lorsqu'il est entraîné sur le jeu de 11 000 secondes, comme le montrent les tables A.3 et A.4. On observe un écart significatif entre les résultats sur le jeu d'évaluation et les autres jeux de test. Cette différence pourrait s'expliquer par le fait que le modèle est déjà optimisé pour la transcription en API et nécessite donc moins d'entraînement. Bien que les meilleures performances d'évaluation soient obtenues sur les dernières époques d'apprentissage pour le corpus de validation, il est probable que le modèle le plus robuste se trouve à des époques moins avancées. Cela suggère un sur-apprentissage sur les données d'entraînement et d'évaluation.

5.3 L'impact du score et de l'homogénéité des données

En regardant maintenant les résultats en fonction des scores attribués par le détecteur de variants, une autre tendance émerge : pour tous les modèles, les

Durée totale du jeu entraînement + validation	PER (%)	FER (%)	DCER (%)
10 000	64,0	56,08	15,88
11 000	**26,28**	**14,89**	**18,24**

TAB. 5.3 : Performances du modèle Multipa sur le jeu de test, entraîné sur un jeu de 10 000 et 11 000 secondes avec les mêmes configurations de score

performances sont meilleures lorsque le jeu de test est homogène avec le jeu d'entraînement.

Ainsi, les meilleures performances sur les jeux d'évaluation, sont obtenues lorsque les modèles sont entraînés sur des jeux de données dont l'intégralité des fichiers ont un score supérieur ou inférieur à 4,4, c'est-à-dire les modèles entraînés sur des données homogènes (et donc des données homogènes avec le jeu d'évaluation également).

Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)
100% score < 4,4	**23,93**	**14,93**	**12,59**
75% score >= 4,4 25% score < 2,6	27,82	17,89	13,89

TAB. 5.4 : Performances du modèle Wav2Vec2-XLS-R-300m sur le jeu d'évaluation, entraîné sur un jeu de données homogène et non homogène

En revanche, pour le jeu de test composé de fichiers choisis aléatoirement, les meilleurs résultats sont généralement obtenus lorsque le jeu de données d'entraînement est plus diversifié (fichiers choisis parmi l'intégralité des données ou parmi ceux ayant un bon score de confiance, supérieur à 4,4, et un mauvais score de confiance, inférieur à 2,6).

Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)
100% score >= 4,4	27,17	15,49	19,29
75% score >= 4,4 25% score < 2,6	**26,12**	**14,98**	**18,24**

TAB. 5.5 : Performances du modèle Multipa sur le jeu test, entraîné sur un jeu de données homogène et non homogène

Pour les fichiers étiquetés «oriental» par le détecteur, les meilleures performances sont atteintes par les modèles entraînés sur un jeu de données avec un score inférieur à 4,4, c'est-à-dire sur des fichiers audio pour lesquels le détecteur était moins sûr de sa prédiction, des fichiers audio moins «occidentaux». On peut également noter que le modèle Multipa, précédemment affiné sur un jeu de données multilingue, affiche de meilleures performances sur ce corpus test. Il est donc moins dépendant des données spécifiques que nous lui avons fournies pour son sur-affinage.

En fonction de l'objectif du modèle entraîné, il est donc recommandé de choisir un

Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)
100% score >= 4,4	57,59	33,79	28,83
100% score < 4,4	**37,06**	**22,48**	**18,81**

TAB. 5.6 : Performances du modèle Wav2Vec2-XLS-R-1B sur le jeu de données détectées comme «oriental», entraîné sur un jeu de données très «occidental» et un jeu de données peu «occidental»

jeu de données plus ou moins hétérogène. Si l'objectif est d'utiliser les modèles sur un type très précis de données, il est préférable de sélectionner des données proches de celles-ci. À l'inverse, si l'on souhaite un modèle plus robuste face à la variabilité, il est recommandé d'inclure des données très différentes dans le jeu d'entraînement.

5.4 Comparaison avec l'état de l'art

Le modèle Multipa, déjà affiné pour transcrire n'importe quelle langue vers l'API et considéré comme l'état de l'art dans le domaine, offre une base de comparaison intéressante pour évaluer les performances de nos différents modèles. Que ce soit le modèle Multipa que nous avons sur-affiné sur l'arménien occidental ou les modèles Wav2Vec2-XLS-R, la table 5.7 présente les résultats obtenus. Pour chaque modèle que nous avons entraîné, après avoir testé de nombreuses configurations, nous avons sélectionné et présenté dans ce tableau les résultats des configurations offrant les meilleures performances sur le corpus de test.

Modèle	PER (%)	FER(%)	DCER(%)	Mémoire GPU
Multipa non entraîné	81,92	24,73	33,92	X
Meilleur modèle Multipa entraîné	26,12	**14,98**	18,24	20-22Go
Meilleur modèle W2V2-XLS-R-300m	**25,37**	16,58	**8,46**	7-9Go
Meilleur modèle W2V2-XLS-R-1b	29,87	18,48	17,21	24-26Go

TAB. 5.7 : Résultats des différents modèles dans leur meilleure configuration sur le jeu de données test

On note que le modèle Wav2Vec2-XLS-R-300m obtient un DCER particulièrement bas sur ces données. On observe également un écart plus important entre le DCER et les autres mesures pour ce modèle spécifiquement. Comme nous l'avons précédemment mentionné, le DCER ne prend en compte que les divergences de consonnes, contrairement aux autres mesures. Ce score indique donc que ce modèle est particulièrement performant dans la détection des phonèmes consonantiques, mais qu'il est plus sensible aux variations vocaliques que les autres modèles.

Nous constatons donc qu'avec seulement 2 heures et 45 minutes de données annotées pour l'arménien occidental, l'entraînement du plus petit modèle de Wav2Vec2-

XLS-R (Wav2Vec2-XLS-R-300M) produit des résultats nettement supérieurs à ceux du modèle Multipa non sur-affiné. De plus, ce modèle surpasse également les performances des deux autres modèles que nous avons entraînés, à savoir les modèles Wav2Vec2-XLS-R-1B et Multipa sur-affiné, bien que ces derniers soient plus gourmands en ressources GPU.

Conclusion

Dans ce chapitre, nous avons exploré l'impact de la durée des fichiers audio et de l'homogénéité des données sur les performances des modèles Wav2Vec2-XLS-R et Multipa. Nous avons dans un premier temps constaté que, pour les modèles Wav2Vec2-XLS-R, l'entraînement sur des fichiers audio de plus d'une seconde améliore significativement les performances, à l'inverse du modèle Multipa. Nous avons également montré que les modèles sont plus performants sur des données homogènes avec leurs données d'entraînement. Enfin, nous avons constaté que le modèle Wav2Vec2-XLS-R-300M, bien que plus petit et moins gourmand en ressources GPU, produit des résultats nettement supérieurs à ceux du modèle Multipa non sur-affiné, avec seulement 2 heures et 45 minutes de données annotées pour l'arménien occidental.

CHAPITRE 6

DISCUSSION

Sommaire

Introduction

Dans ce chapitre, nous approfondirons l'analyse des résultats obtenus, discuterons des limites de leur interprétation, identifierons les sources d'erreurs possibles et explorerons les perspectives d'amélioration de nos résultats. Nous aborderons également les limites et les applications possibles de la transcription de fichiers audio vers l'API.

6.1 Des transcriptions de référence imparfaites

Lorsque nous avions présenté les données utilisées pour notre entraînement et les pré-traitements effectués (voir chapitre 3), nous avions déjà mentionné plusieurs sources d'erreurs possibles, qui pourraient empêcher nos différents modèles d'apprendre correctement.

Tout d'abord, les transcriptions à partir desquelles nous avons obtenu nos transcriptions de référence en API proviennent de sous-titres. Par conséquent, les particularités et aspérités langagières liées à l'oral, telles que les schwas ou la prononciation atypique de certains phonèmes, n'ont probablement pas été retranscrites. En effet, ces sous-titres n'ont pas été créés dans un but linguistique, mais plutôt dans une visée didactique et de préservation de la langue.

De plus, comme mentionné lors de la description des pré-traitements effectués sur les données, le modèle Foma utilisé pour translittérer les transcriptions de l'alphabet arménien vers l'API n'est pas totalement achevé. Il est donc possible qu'il y ait des erreurs dans la translittération de certains phonèmes.

Enfin, il convient de noter que les fichiers audio étaient initialement au format MP3 et ont été convertis en WAV. La qualité des fichiers audio n'est donc pas toujours optimale, ce qui pourrait également entraîner une mauvaise perception de certains phonèmes.

Comparaison	PER (%)	FER (%)	DCER (%)
Transcriptions manuelles / FOMA	24,38	11,53	18,11
Transcriptions manuelles / Multipa	30,62	12,97	21,63
Transcriptions manuelles / Wav2Vec2-300M	**26,56**	**12,30**	**19,31**
Transcriptions manuelles / Wav2Vec2-1B	28,26	12,47	20,91

TAB. 6.1 : Comparaison des transcriptions manuelles aux transcriptions produites par FOMA et nos différents modèles

Pour évaluer l'impact de ces différentes sources d'erreurs possibles, nous avons comparé les translittérations de Foma mais également les transcriptions des différents modèles avec le petit jeu de données transcrit manuellement, précédemment mentionné. Nous avons ensuite comparé ses transcriptions à celles produites par Foma ainsi qu'à celles de nos différents modèles (en utilisant les modèles les plus performants sur ce jeu de données). Les résultats sont présentés dans la table 6.1.

Ces résultats mettent en perspective les performances de nos différents modèles. La comparaison des transcriptions phonétiques manuelles avec les transcriptions phonémiques générées par FOMA permet de mesurer l'écart entre le son tel qu'il est perçu et sa représentation phonologique. On note également PER et un DCER légèrement plus élevés entre les transcriptions générées par les modèles et les transcriptions manuelles, par rapport aux transcriptions FOMA (voir les tableaux des résultats). En revanche, le FER est plus faible lorsqu'on compare les transcriptions des modèles aux transcriptions manuelles par rapport aux transcriptions FOMA. Il semblerait donc qu'au niveau des caractéristiques acoustiques des phones, et en particulier des phones vocaliques, les modèles se rapprochent davantage d'une écriture phonétique.

Comme nous l'avons déjà mentionné, l'arménien occidental est principalement parlé par des locuteurs ayant une autre langue principale, dont la prononciation peut influencer celle de l'arménien occidental, notamment au niveau des voyelles. Il est donc possible que les réalisations phonatoires des voyelles présentent plus de variabilité, rendant plus difficile pour les modèles de les associer à un unique phonème.

6.2 Quelles perspectives pour la suite ?

Ce mémoire s'inscrit dans un projet en cours, offrant de nombreuses perspectives pour des travaux futurs. Tout d'abord, il serait intéressant de relancer ces expériences une fois le modèle Foma pour l'arménien occidental achevé, sur plus d'époques pour l'entraînement sur les jeux de données de 10 000 secondes. De plus, lorsque nous disposerons également d'un modèle pour l'arménien oriental, nous pourrons explorer l'impact de l'intégration de données d'arménien oriental, notamment celles détectées comme étant en occidental, dans les jeux de données utilisés pour l'entraînement de nos modèles d'arménien occidental. Nous pourrons également développer un modèle pour la transcription vers l'API de l'arménien oriental.

[Malajyan et al., 2024], dans leur article sur la reconnaissance automatique de la parole bi-dialectale pour l'arménien, soulignent le manque de transférabilité des

modèles de RAP entraînés sur des données en arménien oriental pour transcrire de l'arménien occidental, et vice versa. Ils notent également que l'intégration de jeux de données composés de différents variants de l'arménien améliore mutuellement les performances des modèles pour les deux standards. Il serait donc intéressant de voir si l'entraînement d'un modèle sur les deux variants pourrait également améliorer les performances de reconnaissance des phonèmes sur les fichiers audio en arménien occidental, ainsi que sur tout autre variant de l'arménien.

Une autre piste d'expérimentation consisterait à contrôler l'ordre des données utilisées pour l'entraînement du modèle. On pourrait, dans un premier temps, présenter au modèle des données fortement «occidentales», puis introduire progressivement des données plus ambiguës. Cette approche permettrait d'évaluer l'impact de cette progression sur l'entraînement. En reproduisant ainsi le processus «naturel» de diversification du vocabulaire, similaire à celui d'un enfant apprenant à parler, nous pourrions mieux comprendre comment le modèle adapte son apprentissage en fonction de la nature des données présentées.

Nous avons également évalué les performances de nos modèles en fonction de la durée minimale des fichiers audio utilisés pour l'entraînement, en testant uniquement deux paliers (l'un à 0,5 seconde et l'autre à 1 seconde). Il serait intéressant d'approfondir cette analyse en explorant davantage ce paramètre. De plus, nous n'avons pas encore examiné l'impact de la durée maximale des fichiers audio sur l'entraînement des modèles. Actuellement, les fichiers les plus longs durent environ 8 secondes. Il serait pertinent de tester plusieurs seuils de durée maximale pour évaluer leur impact sur l'entraînement de nos modèles.

Enfin, l'utilisation du détecteur de variants et des scores qu'il attribue pourrait également permettre de transcrire des fichiers audio en dialectes arméniens, même s'ils ne correspondent ni à l'arménien oriental standard ni à l'arménien occidental standard. Ce détecteur permettrait d'attribuer un score d'«occidentalisation» ou d'«orientalisation» au fichier audio, facilitant ainsi la sélection du modèle de transcription le plus adapté. Étant donné que les mesures de distances, et notamment les distances phonologiques, entre les différents variants de l'arménien est un aspect important du projet DALiH (à partir de quelle distance entre deux variants considère-t-on qu'il s'agit de deux langues distinctes ?), ce système de transcription vers l'écriture phonologique pourrait grandement simplifier ces travaux.

6.3 Les limites et applications possibles de la transcription vers l'API

La transcription vers l'API présente certaines limites intrinsèques. En revenant à la manière dont le langage est perçu par l'être humain, et en se référant à la double articulation décrite par [Martinet, 1960], on constate que nos modèles de langue ne prennent en compte que la première articulation. Bien que ces modèles puissent associer des phones à des phonèmes en utilisant une écriture phonémique comme référence, ils ne parviennent pas à passer des unités distinctives (phonèmes) aux unités de sens (morphèmes). Par conséquent, les chaînes de phonèmes produites ne peuvent être directement exploitées par des applications telles qu'un assistant vocal. L'intégration de la prosodie pourrait permettre d'isoler des sous-chaînes de phonèmes correspondant à des syntagmes, offrant ainsi une meilleure exploitation des données

phonémiques.

Cependant, ce type de modèle présente des applications potentielles intéressantes. Ces modèles pourraient par exemple être très utiles pour l'étude des langues uniquement orales, souvent peu étudiées par les linguistes. La description de ces langues demande généralement un travail considérable, car toutes les ressources sont orales et doivent être retranscrites, ce qui nécessitent souvent la création d'un système d'écriture, propre au·x linguiste·s les étudiant. Cela limite donc l'accessibilité des ressources transcrites pour d'autres personnes. L'API a l'avantage d'être universellement compréhensible. De plus, en transcrivant seulement 2 à 3 heures d'audio vers l'API, nous avons démontré qu'il est possible d'obtenir un modèle relativement performant, capable de transcrire automatiquement des heures d'entretien, faisant ainsi gagner un temps précieux.

Enfin, la question du code-switching a également été abordée dans notre chapitre sur l'état de l'art. [Dhawan et al., 2023] proposaient un système de modèles fonctionnant par paires de langues souvent co-occurrentes (comme l'hindi et l'anglais), mais ce type de modèle a ses limites. En revanche, un modèle capable de représenter phonétiquement un mot pour lequel il n'a pas de correspondance proche dans le lexique de la langue détectée pourrait rechercher dans les lexiques des autres langues sur lesquelles il est entraîné. Il pourrait alors potentiellement y trouver un mot dont la représentation phonétique et le sens dans le contexte seraient plus adaptés, améliorant ainsi la gestion du code-switching.

Conclusion générale

Dans ce mémoire, nous avons exploré diverses stratégies pour faciliter l'apprentissage de modèles de reconnaissance automatique de la parole dans le contexte des langues peu dotées, pour lesquelles les données disponibles sont limitées et souvent non exploitables telles quelles par les machines. Nous avons concentré nos travaux sur la transcription vers l'API, qui offre des possibilités uniques par rapport à la transcription automatique de la parole traditionnelle. Cette approche permet notamment de retranscrire des langues uniquement orales et pourrait être plus pertinente pour la gestion du code-switching. De plus, la transcription vers l'API est moins dépendante du vocabulaire utilisé dans les fichiers d'entraînement : on retranscrit un son et non un sens.

Nous avons également formulé plusieurs hypothèses concernant le choix des fichiers pour l'entraînement de nos modèles. La première hypothèse portait sur l'impact de la durée minimale des fichiers audio utilisés pour l'entraînement. Nous avons constaté que, pour un modèle uniquement pré-entraîné, l'utilisation de fichiers audio de plus d'une seconde favorisait l'apprentissage. En revanche, pour les modèles déjà affinés, cette durée minimale semblait avoir moins d'impact, voire un impact négatif. Cette piste sur les modèles affinés reste tout de même à approfondir car la baisse de performance du modèle pourrait également être due à un sur-entraînement plutôt qu'à la durée des fichiers audio utilisés.

La seconde hypothèse concernait l'impact de l'homogénéité des données d'entraînement sur les performances des modèles. Nous avons observé que l'utilisation de fichiers d'entraînement homogènes avec les données que nous souhaitons transcrire était préférable. Cette homogénéité permet de mieux aligner les caractéristiques des données d'entraînement avec celles des données cibles, améliorant ainsi les performances des modèles.

Grâce à ces travaux, nous avons également pu atteindre un nouvel état de l'art pour la transcription de l'arménien occidental vers l'API. Le précédent modèle état de l'art sur cette tâche, Multipa, bien que non entraîné spécifiquement sur l'arménien occidental, se voulait universel. Cependant, même après avoir sur-affiné ce modèle sur des données en arménien occidental, nous avons obtenu de moins bons résultats que le plus petit modèle Wav2Vec2-XLS-R de Facebook après entraînement.

De plus, bien que l'article de [Babu et al., 2021] indique que les modèles Wav2Vec2-XLS-R les plus gros sont généralement meilleurs pour les tâches de transcription phonémique, nous avons obtenu de meilleurs résultats avec le modèle à 300 millions de paramètres par rapport au modèle à un milliard de paramètres, pour une utilisation de GPU trois fois moins importante. Cela montre que la taille du modèle n'est pas toujours le facteur déterminant pour obtenir les meilleures performances.

Enfin, [Taguchi et al., 2023] mentionnent dans leur article que leur modèle obtenant les meilleurs résultats et le plus robuste est celui entraîné sur le moins de

données. Cela souligne l'importance de la qualité et de la pertinence des données d'entraînement, plutôt que de la simple quantité de données ou de la taille du modèle.

Nous avons donc démontré qu'il est possible, avec un volume limité mais bien sélectionné de données et des ressources GPU modestes, d'obtenir un modèle performant pour la transcription de langues vers l'API. Cette découverte rend la tâche de transcription automatique de la parole plus accessible pour l'étude des langues peu dotées.

BIBLIOGRAPHIE

[Adjarian, 1909] Adjarian, H. (1909). *Classification des Dialectes Arméniens*. H. Champion, Paris. – Cité pages 6 et 24.

[Babu et al., 2021] Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. (2021). XLS-R : Self-supervised Cross-lingual Speech Representation Learning at Scale. – Cité pages 6, 37, 39 et 53.

[Baevski et al., 2020] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations. – Cité pages 6 et 18.

[Chakmakjian and Wang, 2022] Chakmakjian, S. and Wang, I. (2022). Towards a Unified ASR System for the Armenian Standards. In *Proceedings of the LREC 2022 workshop on Processing Language Variation : Digital Armenian (DigitAm)*. – Cité pages 26 et 33.

[Chang et al., 2015] Chang, W., Cathcart, C., Hall, D., and Garrett, A. (2015). Ancestry-constrained Phylogenetic Analysis Supports the Indo-European Steppe Hypothesis. *Language, Volume 91, Number 1*. – Cité pages 6 et 23.

[Conneau et al., 2020] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., and Auli, M. (2020). Unsupervised Cross-lingual Representation Learning for Speech Recognition. – Cité page 37.

[Dhawan et al., 2023] Dhawan, K., Rekesh, K., and Ginsburg, B. (2023). Unified Model for Code-Switching Speech Recognition and Language Identification Based on Concatenated Tokenizer. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 74–82. Association for Computational Linguistics. – Cité pages 19 et 52.

[Dolgopolsky, 1964] Dolgopolsky, A. B. (1964). A Probabilistic Hypothesis Concerning The Oldest Relationships Among The Languages Families of Northern Eurasia. In *Typology Relationship and Time - A Collection of Papers on Language Change and Relationship By Soviet Linguists*, pages 27–50. Karoma Publishers. – Cité page 21.

[Dubois et al., 2001] Dubois, J., Giacomo, M., Guespin, L., Marcellesi, C., Marcellesi, J.-B., and Mével, J.-P. (2001). *Dictionnaire de Linguistique*. Larousse. – Cité page 16.

[Frauenfelder, 1991] Frauenfelder, U. H. (1991). Une Introduction aux Modèles de Reconnaissance des Mots Parlés. In *La Reconnaissance des Mots dans les Différentes Modalités Sensorielles : Études de Psycholinguistique Cognitive*, pages 7–36. Presses Universitaires de France. – Cité page 17.

[Graves et al., 2006] Graves, A., Fernandez, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks. – Cité page 19.

[Levenshtein, 1965] Levenshtein, V. (1965). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Doklady Akademii Nauk SSSR, Vol. 163, No.4*, pages 845–848. – Cité page 20.

[Li et al., 2020] Li, X., Dalmia, S., Li, J., Lee, M., Littell, P., Yao, J., Anastasopoulos, A., Mortensen, D. R., Neubig, G., Black, A. W., and Metze, F. (2020). Universal Phone Recognition with a Multilingual Allophone System. – Cité page 38.

[Malajyan et al., 2024] Malajyan, A., Khurshudyan, V., Avetisyan, K., Dolatian, H., and Nouvel, D. (2024). Bi-dialectal ASR of Armenian from Naturalistic and Read Speech. In Melero, M., Sakti, S., and Soria, C., editors, *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 227–236. ELRA and ICCL. – Cité page 50.

[Manoukian, 2023] Manoukian, J. (2023). How Western Armenian Came to Be : A Story of People, Purism and Global Ideas. Webinar organized by The Promise Armenian Institute at UCLA. – Cité pages 24 et 27.

[Martinet, 1960] Martinet, A. (1960). *Éléments de Linguistique Générale*. Colin. – Cité pages 17 et 51.

[Mortensen et al., 2016] Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., and Levin, L. S. (2016). Panphon : A Resource for Mapping IPA Segments to Articulatory Feature Vectors. In *International Conference on Computational Linguistics*. – Cité pages 7, 20, 21 et 58.

[Pieraccini and Rabiner, 2012] Pieraccini, R. and Rabiner, L. (2012). *The Voice in the Machine : Building Computers That Understand Speech*. The MIT Press. – Cité page 18.

[Pratap et al., 2023] Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., Elkahky, A., Ni, Z., Vyas, A., Fazel-Zarandi, M., Baevski, A., Adi, Y., Zhang, X., Hsu, W.-N., Conneau, A., and Auli, M. (2023). Scaling Speech Technology to 1,000+ Languages. *arXiv*. – Cité page 33.

[Sarmah and Bhattacharjee, 2014] Sarmah, K. and Bhattacharjee, U. (2014). GMM based Language Identification using MFCC and SDC Features. *International Journal of Computer Applications*, 85(5) :36–42. – Cité page 33.

[Saussure, 1916] Saussure, F. d. (1916). *Cours de Linguistique Générale*. Payot. – Cité pages 13 et 17.

[Taguchi et al., 2023] Taguchi, C., Sakai, Y., Haghani, P., and Chiang, D. (2023). Universal Automatic Phonetic Transcription into the International Phonetic Alphabet. – Cité pages 6, 19, 37, 38 et 53.

[Tulving, 1970] Tulving, E. (1970). Episodic and Semantic Memory. In *Organization of Memory*. Academic Press. – Cité page 15.

ANNEXES

A.1 Classes d'équivalences de Dolgopolsky

Classe	Définition de la classe	Label
Obstruantes labiales	[-son +ant -cor + lab]	P
Coronales fricatives	[-son +cor +cont +strid -delrel]	S
Obstruantes vélaires / post-vélaires	[-son -ant -cor +back]	K
Coronales affricatives	[+cor +delrel]	K
Autres coronales obstruantes	[-son +cor]	T
Labiales nasales	[-syl +son +nas +ant -cor +lab]	M
Nasales	[-syl +son -cont +nas]	N
Latérales	[+lat]	R
Spirantes et roulées	[-syl +son -nas]	R
Semi-voyelles palatales	[-syl +son +cont +hi -lo -back]	J
Semi-voyelles labiales-vélaires	[-syl +son]	W
Consonnes laryngiales	[-syl +son +cons +cont -cor -hi -back -lo]	H
Consonnes	[-syl]	C
Voyelles	[+syl]	V

TAB. A.1 : Classes d'équivalence de Dolgopolsky telles qu'utilisées par la bibliothèque Panphon

A.2 Présentation des caractéristiques phonologiques

Caractéristique	Label	Définition
[±syllabic]	syl	Le segment est-il le noyau de la syllabe ?
[±sonorant]	son	Le segment est-il produit avec un conduit vocal relativement dégagé ?
[±consonantal]	cons	Le segment est-il consonantique (pas une voyelle, ni une glissade, ni une consonne laryngée) ?
[±continuant]	cont	Le segment est-il produit avec un flux d'air oral continu ?
[±delayed release]	delrel	Le segment est-il une affriquée ?
[±lateral]	lat	Le segment est-il produit avec une constriction latérale ?
[±nasal]	nas	Le segment est-il produit par le flux d'air nasal ?
[±strident]	strid	Le segment est-il produit avec un frottement sonore ?
[±voice]	voi	Les plis vocaux vibrent-ils pendant la production du segment ?
[±spread glottis]	sg	Les plis vocaux s'écartent-ils pendant la production du segment ?
[±constricted glottis]	cg	Les plis vocaux se rapprochent-ils pendant la production du segment ?
[±anterior]	ant	Une constriction se produit-elle à l'avant du conduit vocal ?
[±coronal]	cor	La pointe ou la lame de la langue est-elle utilisée pour faire une constriction ?
[±distributed]	distr	Une constriction coronale est-elle distribuée latéralement ?
[±labial]	lab	Le segment implique-t-il des constrictions avec ou des lèvres ?
[±high]	hi	Le segment est-il produit avec le corps de la langue relevé ?
[±low]	lo	Le segment est-il produit avec le corps de la langue abaissé ?
[±back]	back	Le segment est-il produit avec le corps de la langue en position postérieure ?
[±round]	round	Le segment est-il produit avec les lèvres arrondies ?
[±tense]	tense	Le segment est-il produit avec la racine de la langue avancée ?
[±long]	long	Le segment est-il allongé ?
[±velaric]	velaric	Le segment est-il produit avec la racine de la langue relevée vers le palais mou ?

TAB. A.2 : Présentation des caractéristiques phonologiques utilisées par Panphon, telles que présentées dans [Mortensen et al., 2016]

A.3 Tableaux des résultats

A.3.1 Multipa

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	**25,28**	**14,86**	17,87	3h38
	100% score < 4,4	26,01	15,34	**17,69**	3h47
	50% score >= 4,4 50% score < 4,4	26,61	15,91	18,35	4h06
	75% score >= 4,4 25% 2,6 < score < 4,4	28,72	16,16	20,23	3h14
	75% score >= 4,4 25% score < 2,6	27,77	16,22	18,83	3h15
Test	100% score >= 4,4	27,17	15,49	19,29	3h38
	100% score < 4,4	27,51	15,26	19,04	3h47
	50% score >= 4,4 50% score < 4,4	26,28	**14,89**	**18,24**	4h06
	75% score >= 4,4 25% 2,6 < score < 4,4	29,65	16,69	20,53	3h14
	75% score >= 4,4 25% score < 2,6	**26,12**	14,98	**18,24**	3h15
Détecté oriental	100% score >= 4,4	40,63	20,19	27,67	3h38
	100% score < 4,4	**30,93**	16,29	**20,63**	3h47
	50% score >= 4,4 50% score < 4,4	31,17	**16,16**	21,11	4h06
	75% score >= 4,4 25% 2,6 < score < 4,4	42,28	20,73	28,47	3h14
	75% score >= 4,4 25% score < 2,6	31,66	16,32	21,28	3h15

TAB. A.3 : Résultats obtenus avec le modèle Multipa sur le dataset de 11 000 secondes (fichiers audios de plus de 0,5 seconde)

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	30,34	19,6	13,95	3h54
	100% score < 4,4	**28,88**	**18,83**	**13,45**	4h31
	50% score >= 4,4 50% score < 4,4	31,99	21,22	14,05	5h07
	75% score >= 4,4 25% 2,6 < score < 4,4	30,95	20,16	**13,45**	3h52
	75% score >= 4,4 25% score < 2,6	33,88	21,79	14,81	3h52
Test	100% score >= 4,4	63,39	55,4	16,76	3h54
	100% score < 4,4	63,47	55,08	16,79	4h31
	50% score >= 4,4 50% score < 4,4	64,0	56,08	**15,88**	5h07
	75% score >= 4,4 25% 2,6 < score < 4,4	63,97	55,78	16,33	3h52
	75% score >= 4,4 25% score < 2,6	**63,31**	**55,07**	16,3	3h52
Détecté oriental	100% score >= 4,4	71,43	60,21	24,67	3h54
	100% score < 4,4	68,36	60,13	17,43	4h31
	50% score >= 4,4 50% score < 4,4	70,42	61,15	**18,1**	5h07
	75% score >= 4,4 25% 2,6 < score < 4,4	70,69	59,8	21,72	3h52
	75% score >= 4,4 25% score < 2,6	**68,08**	**58,98**	18,12	3h52

TAB. A.4 : Résultats obtenus avec le modèle Multipa sur le dataset de 10 000 secondes (fichiers audios de plus de 1 seconde)

A.3.2 Wav2Vec2-XLS-R-300M

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	33,66	21,94	**16,95**	2h58
	100% score < 4,4	**29,95**	**19,66**	17,38	3h00
	50% score >= 4,4 50% score < 4,4	35,94	23,44	17,23	2h52
	75% score >= 4,4 25% 2,6 < score < 4,4	43,82	28,59	21,27	2h31
	75% score >= 4,4 25% score < 2,6	38,05	24,74	19,15	3h09
Test	100% score >= 4,4	35,91	22,92	18,76	2h58
	100% score < 4,4	35,08	**22,38**	18,5	3h00
	50% score >= 4,4 50% score < 4,4	36,65	23,72	**17,19**	2h52
	75% score >= 4,4 25% 2,6 < score < 4,4	43,37	28,18	22,09	2h31
	75% score >= 4,4 25% score < 2,6	**34,98**	22,62	18,59	3h09
Détecté oriental	100% score >= 4,4	62,57	38,1	30,12	2h58
	100% score < 4,4	**43,34**	**28,56**	**20,75**	3h00
	50% score >= 4,4 50% score < 4,4	46,12	30,22	20,8	2h52
	75% score >= 4,4 25% 2,6 < score < 4,4	69,11	44,67	31,74	2h31
	75% score >= 4,4 25% score < 2,6	51,41	33,57	24,58	3h09

TAB. A.5 : Résultats obtenus avec le modèle Wav2Vec2-XLS-R-300m sur le dataset de 11 000 secondes (fichiers audios de plus de 0,5 seconde)

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	26,32	16,6	12,71	6h13
	100% score < 4,4	**23,93**	**14,93**	12,59	4h15
	50% score >= 4,4 50% score < 4,4	27,11	17,68	**8,38**	4h09
	75% score >= 4,4 25% 2,6 < score < 4,4	25,86	16,64	12,54	4h23
	75% score >= 4,4 25% score < 2,6	27,82	17,89	13,89	5h32
Test	100% score >= 4,4	26,94	16,39	15,47	6h13
	100% score < 4,4	27,98	17,48	15,47	4h15
	50% score >= 4,4 50% score < 4,4	**25,37**	16,58	**8,46**	4h09
	75% score >= 4,4 25% 2,6 < score < 4,4	25,78	**15,86**	15,17	4h23
	75% score >= 4,4 25% score < 2,6	26,65	16,9	15,09	5h32
Détecté oriental	100% score >= 4,4	50,59	29,93	24,41	6h13
	100% score < 4,4	**33,43**	**20,95**	17,31	4h15
	50% score >= 4,4 50% score < 4,4	37,08	24,11	**13,35**	4h09
	75% score >= 4,4 25% 2,6 < score < 4,4	44,21	27,44	21,91	4h23
	75% score >= 4,4 25% score < 2,6	36,18	22,14	18,27	5h32

TAB. A.6 : Résultats obtenus avec le modèle Wav2Vec2-XLS-R-300m sur le dataset de 10 000 secondes (fichiers audios de plus de 1 seconde)

A.3.3 Wav2Vec2-XLS-R-1B

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	**40,36**	**25,89**	19,55	3h43
	100% score < 4,4	44,15	28,54	21,5	5h43
	50% score >= 4,4 50% score < 4,4	43,75	29,08	**17,35**	3h50
	75% score >= 4,4 25% 2,6 < score < 4,4	58,76	39,11	24,36	4h59
	75% score >= 4,4 25% score < 2,6	48,95	32,14	24,33	3h26
Test	100% score >= 4,4	**43,56**	**27,83**	21,09	3h43
	100% score < 4,4	48,55	30,25	22,85	5h43
	50% score >= 4,4 50% score < 4,4	51,77	35,32	**17,46**	3h50
	75% score >= 4,4 25% 2,6 < score < 4,4	58,74	38,25	25,58	4h59
	75% score >= 4,4 25% score < 2,6	47,76	31,01	23,25	3h26
Détecté oriental	100% score >= 4,4	69,35	41,93	31,39	3h43
	100% score < 4,4	**59,54**	**38,58**	**26,42**	5h43
	50% score >= 4,4 50% score < 4,4	67,55	44,69	26,46	3h50
	75% score >= 4,4 25% 2,6 < score < 4,4	84,36	52,58	34,9	4h59
	75% score >= 4,4 25% score < 2,6	60,99	39,83	29,77	3h26

TAB. A.7 : Résultats obtenus avec le modèle Wav2Vec2-XLS-R-1B sur le dataset de 11 000 secondes (fichiers audios de plus de 0,5 seconde)

Jeu de données test	Jeu de données d'entraînement	PER (%)	FER (%)	DCER (%)	Temps d'entraînement
Évaluation	100% score >= 4,4	31,5	19,69	**13,9**	5h45
	100% score < 4,4	**28,03**	**17,45**	14,37	8h37
	50% score >= 4,4 50% score < 4,4	31,25	19,93	15,08	6h08
	75% score >= 4,4 25% 2,6 < score < 4,4	30,57	19,49	14,02	5h58
	75% score >= 4,4 25% score < 2,6	34,16	21,96	15,97	8h17
Test	100% score >= 4,4	30,02	**18,44**	17,32	5h45
	100% score < 4,4	31,68	19,36	17,97	8h37
	50% score >= 4,4 50% score < 4,4	**29,87**	18,48	**17,21**	6h08
	75% score >= 4,4 25% 2,6 < score < 4,4	31,4	19,49	17,31	5h58
	75% score >= 4,4 25% score < 2,6	31,41	19,57	17,25	8h17
Détecté oriental	100% score >= 4,4	57,59	33,79	28,83	5h45
	100% score < 4,4	**37,06**	**22,48**	**18,81**	8h37
	50% score >= 4,4 50% score < 4,4	37,79	23,06	20,62	6h08
	75% score >= 4,4 25% 2,6 < score < 4,4	54,21	32,1	26,81	5h58
	75% score >= 4,4 25% score < 2,6	39,6	23,68	20,18	8h17

TAB. A.8 : Résultats obtenus avec le modèle Wav2Vec2-XLS-R-1B sur le dataset de 10 000 secondes (fichiers audios de plus de 1 seconde)