UCLouvain
Faculté de philosophie, arts et lettres – FIAL

Faculté de philosophie, arts et lettres

La transcription automatique de la parole spontanée non standard

Étude des modèles Whisper pour la reconnaissance vocale chez les néerlandophones apprenant le français

Autrice : Adélaïde Couplet
Promotrice : Dr Anaïs Tack
Co-promotrice : Pr Anne-Catherine Simon
Année académique 2023-2024
Master en linguistique, à finalité spécialisée : traitement automatique du langage

Remerciements

“Attention is all you need” (Vaswani et al., 2017)

C’est peut-être vrai pour les technologies de TAL actuelles, mais lors de la rédaction d’un mémoire, il vous faudra bien plus que de l’attention. Sachant cela, il me semble important de remercier les personnes sans qui ce mémoire n’aurait pas eu de conclusion, et peut-être même pas d’introduction.

Tout d’abord, un immense merci à Anaïs Tack, pour tout le temps qu’elle m’a accordé, pour les nombreux rendez-vous Teams, pour ses conseils avisés, pour ses connaissances qu’elle a volontiers partagées avec moi et pour son soutien dans mes recherches depuis plus d’un an déjà.

Merci également à Anne-Catherine Simon d’avoir accepté d’être la co-promotrice de ce mémoire.

Merci à Ann-Sophie Noreillie d’avoir partagé le fruit de son travail avec moi par le prêt du corpus réalisé durant sa thèse.

Merci à tous les membres de l’ITEC du campus de Kortrijk de la KU Leuven de m’avoir accueillie durant mon stage de l’été 2023.

Merci à l’équipe du CENTAL de faire en sorte que ce master soit accessible aux étudiant·e·s de la faculté FIAL. Un merci particulier à Hubert Naets qui transmet toujours ses connaissances avec un enthousiasme communicatif.

Merci à ma sœur, Clémentine, et à mon père, Ignace, pour leurs relectures attentives de ce travail.

Merci à mes ami·e·s et à ma merveilleuse famille pour leur soutien de toujours et particulièrement durant cette période de reprise d’études.

Enfin, un merci particulier à Charles, qui a supporté la période de rédaction, pas seulement d’un, mais de deux mémoires depuis que nous nous sommes rencontrés.

Table des matières

Première partie

Introduction

1.1 Thématique et enjeux

La reconnaissance (ou transcription) automatique de la parole (RAP dans la suite de ce travail) est un domaine de recherches lié au traitement automatique du langage (TAL), qui intéresse les chercheurs et chercheuses depuis la fin des années 1940. Le but de la reconnaissance automatique de la parole est de créer un outil qui transforme une production orale en mots ou en phrases écrites.

Notre cerveau humain est habitué à décrypter une production langagière sans même y réfléchir, nous sommes capables de suivre une conversation, de comprendre ce que nous entendons même si plusieurs personnes parlent ou s'il y a du bruit autour de nous. La RAP cherche à créer un modèle qui permettrait à une machine d'être aussi efficace dans ces tâches qu'un cerveau humain.

Le modèle le plus récent et le plus avancé dans le domaine est le modèle Whisper, développé par la société OpenAI. Il est basé sur l'architecture *Transformer*, une technologie de pointe qui a obtenu les meilleurs résultats jamais observés depuis le début des recherches, non seulement pour la RAP, mais aussi pour toutes les questions de TAL.

Malgré les avancées récentes impressionnantes, la transcription automatique de la parole fait souvent face à quelques difficultés. Un premier obstacle concerne la transcription de la parole spontanée. La parole spontanée est la parole que nous produisons quotidiennement. Lors de nos interactions, nous

parlons parfois plus vite, en articulant moins, en produisant des disfluences, des hésitations ou répétitions, cela n'est pas intuitif pour un modèle de RAP.

Une autre difficulté de la transcription automatique est ce qu'on appelle le bruit. Le bruit représente tous les parasites qui rendent plus compliquée la transcription du spectre sonore. Ce sont par exemple des distorsions sonores dues à une mauvaise qualité de l'enregistrement, les conversations alentour, les bruits de la rue, etc.

La parole non spontanée, en revanche, est plus facile à transcrire. Tout d'abord, elle est souvent enregistrée dans de bonnes conditions et surtout, les locuteurices ont tendance à faire plus attention, à préparer leur production orale, à mieux articuler, etc. Un exemple de parole non spontanée peut être la lecture d'un audiolivre.

Une autre difficulté pour la transcription automatique de la parole concerne la transcription de la parole non standard[1]. Nous reprenons dans le terme parole non standard les productions des locuteurices ayant des troubles de la parole, les différents accents du français (créole, québécois, etc.), et également, et c'est sur ce sujet précisément que nous allons nous pencher, les productions des locuteurices non-natif·ve·s.

Dans les cas précités, la parole dévie du type de parole que le modèle a vu lors de l'entrainement et, dès lors, elle est plus difficile à transcrire.

Les modèles récents sont devenus plus performants pour transcrire la

1. En sociolinguistique variationniste, le français non standard ne désigne pas les productions de locuteurices non-natif·ve·s ou de locuteurices atteint·e·s de troubles du langage. En effet, ce terme fait référence aux parlers qui sont éloignés de la norme (souvent, le français de Paris) que ce soit des parlers régionaux, des parlers jeunes, etc. (pour plus d'informations sur le parler non standard tel qu'il est étudié en linguistique, voir, par exemple, la thèse de (Audrit, 2009, 34-40)). Cependant, pour ce mémoire, nous avons décidé de garder le terme de « non standard » car il nous semble adapté pour décrire la parole de locuteurices non-natif·ve·s ou de locuteurices ayant des troubles de la parole. C'est donc cette description étendue que nous sous-entendons lorsque nous utilisons le terme « non standard ».

parole des locuteurices non-natif·ve·s, mais les modèles comme Whisper ont tendance à corriger les erreurs (grammaticales par exemple). Les transcriptions obtenues grâce à Whisper ne sont donc pas vraiment « fiables » si la tâche est de transcrire exactement ce qui a été dit, ce qui rend le modèle difficile à utiliser dans un contexte éducatif où l'on voudrait identifier les erreurs.

1.2 Questions de recherches et organisation

L'objectif de ce mémoire est d'étudier les possibilités actuelles pour la transcription automatique de la parole spontanée non standard. Plus précisément, notre étude vise à évaluer la performance des différentes versions du modèle Whisper pour la transcription de dialogues spontanés d'apprenant·e·s du français.

Pour ce faire, nous avons travaillé avec un corpus créé par Ann-Sophie Noreillie dans le cadre de sa thèse à la KU Leuven (Noreillie, 2019). Ce corpus comprend des interviews spontanées d'étudiant·e·s néerlandophones parlant en français. Deux thèmes sont abordés lors de ces interviews : un rendez-vous chez le médecin et un entretien d'embauche.

En outre, le corpus comprend également des interviews d'étudiant·e·s francophones faisant les mêmes types de tâches dialogiques. Grâce à cela, le corpus nous permet d'évaluer la transcription automatique de la parole spontanée non standard et de la comparer à la transcription automatique de la parole spontanée standard pour un même sujet.

Plus particulièrement, notre étude vise à répondre aux questions de recherche suivantes :

1. Quelle version du modèle Whisper offre les meilleures performances (c'est-à-dire, le plus petit taux d'erreurs sur les mots transcrits) pour la

transcription automatique des dialogues spontanés étudiés par Noreillie (2019) ?

2. Les modèles Whisper sont-ils moins performants pour transcrire automatiquement la parole non standard (par des personnes non-natives) que la parole standard (par des personnes natives) ?
3. Quel modèle Whisper offre les meilleures performances (c'est-à-dire, le plus petit taux d'erreurs sur les mots transcrits) pour la transcription automatique de la parole non standard (par des personnes non-natives) ?

La suite de notre mémoire est structurée en trois parties. La première présentera un état de l'art sur la transcription automatique de la parole (voir partie 2). Dans cet état de l'art, nous reprendrons les grandes lignes de l'évolution du domaine, pour aboutir à la révolution de l'apparition des modèles *Transformers*, comme le modèle Whisper. Nous parlerons ensuite du *Word Error Rate*, qui est une métrique d'évaluation que nous allons employer dans nos analyses. Enfin, nous présenterons également d'autres études sur la transcription automatique de la parole spontanée non standard.

La deuxième partie présentera la méthodologie de notre étude (voir partie 3). Nous décrirons le corpus de Noreillie (2019) plus en détail ainsi que les démarches suivies pour transcrire automatiquement ce corpus avec le modèle Whisper d'OpenAI. Nous ferons également un point sur les métriques d'évaluation que nous avons choisi d'utiliser dans le cadre du mémoire.

Enfin, la troisième partie, dans laquelle nous répondrons aux questions de recherche définies ci-dessus, abordera les résultats de notre étude (voir partie 4).

De possibles ouvertures pour des recherches ultérieures seront présentées dans la conclusion générale de ce mémoire (voir partie 5).

Deuxième partie

État de l'art

Cette partie du mémoire présente un état de l'art (non exhaustif) des avancées en matière de reconnaissance de la parole. Dans un premier temps, nous allons présenter une évolution historique du domaine (section 2.1). Nous nous attarderons sur les modèles de Markov cachés avant de parler des réseaux de neurones pour finalement terminer par la technologie des *Transformers* et le modèle Whisper que nous avons étudié dans ce mémoire.

Dans un deuxième temps, nous parlerons de la métrique d'évaluation *Word Error Rate* et de son utilisation dans le domaine de la transcription automatique (section 2.2).

Enfin, nous allons passer en revue des études se focalisant sur la reconnaissance et transcription automatique de la parole non standard, comme le cas des personnes non-natives (section 2.3).

2.1 Panorama de l'évolution des modèles

2.1.1 Des recherches qui débutent dès le milieu du XX^{e} siècle

Les recherches dans le domaine de la RAP commencent à se développer dès la fin des années 1940. Au départ, le laboratoire Bell crée un système capable de reconnaitre les nombres de 1 à 9, ce modèle est basé sur des enregistrements d'un·e seul·e locuteurice (Davis et al., 1952). La reconnaissance des chiffres de 1 à 9 va encore être utilisée dans le développement de systèmes plus poussés par la suite.

Il y a encore de nombreuses étapes intermédiaires entre le système de Bell et les modèles de notre décennie. Nous allons tout d'abord nous concentrer sur l'avènement de l'utilisation des modèles de Markov cachés (*Hidden Markov Models*, HMM dans la suite du travail) qui ont tout d'abord été utilisés au début des années 1970 et qui vont devenir la norme dans les années 1990.

Pourquoi parler des HMM dans ce mémoire alors que ce modèle n'est plus utilisé aujourd'hui ? Avoir une meilleure compréhension de l'utilisation des HMM dans la RAP nous a permis de mieux comprendre l'évolution actuelle des systèmes de reconnaissance automatique de la parole, c'est pourquoi il nous semblait important de développer plus profondément les explications sur les HMM.

De plus, bien que largement remplacés par les nouvelles techniques de *deep learning* et les réseaux de neurones, les HMM sont encore parfois employés dans les approches hybrides (Deléglise and Lailler, 2020).

2.1.2 Les modèles de Markov cachés et leur fonctionnement en RAP

2.1.2.1 Les origines

Où et quand a commencé l'utilisation des HMM ? Les HMM ont été développés vers 1972 par le professeur Baum et ses collègues de Princeton pour être utilisés en statistiques[2], c'est suite à ces travaux que James K. Baker va utiliser ce modèle pour la reconnaissance vocale.

Baker mettra au point un système nommé DRAGON basé sur les HMM. Dans sa thèse *Stochastic modeling as a means of automatic speech recognition* (Baker, 1975b) il commence par citer et expliquer en quelques phrases les

2. Pour plus d'informations, voir Baum and Eagon (1967) et Baum and Petrie (1966)

systèmes de reconnaissance vocale déjà existants (ARCS, CASPER, IBM-Watson,...). Déjà en 1975 ces systèmes sont nombreux et cherchent tous à faire correspondre un spectre audio avec un mot ou une phrase.

Baker explique qu'à l'époque, traduire un signal sonore en mots est compliqué et que pour ce faire, il faut ajouter d'autres connaissances au signal (connaissances lexicales, syntaxiques et sémantiques). Il explique encore que ces connaissances doivent être mises en forme pour pouvoir être utilisées dans la reconnaissance vocale et que les deux opérations les plus importantes de la discipline à l'époque sont le « *searching and matching* » qu'on peut traduire par « recherche et mise en correspondance » :

> Deux opérations sont essentielles dans tout système de reconnaissance vocale : la recherche et la mise en correspondance. Supposons qu'une source de connaissances, telle que la syntaxe, émette une hypothèse sur un mot ou une séquence de mots. Cette hypothèse ne peut être vérifiée qu'en faisant correspondre les mots avec les événements observés par les autres sources de connaissance, tels que le signal acoustique réel. Une procédure de mise en correspondance est nécessaire pour évaluer une hypothèse particulière. Une procédure de recherche est nécessaire pour explorer l'espace des hypothèses possibles.
> (Baker, 1975b, page 1) [notre traduction]

Baker continue en expliquant qu'au vu du nombre de modèles existants, il y a plusieurs manières de réaliser ce *searching and matching*. Alors pourquoi créer le DRAGON ? Son idée est de représenter dans un modèle unique toutes les sources de connaissances que nous avons citées précédemment afin d'obtenir un outil plus performant et il va pour ce faire utiliser un modèle probabiliste de Markov en l'adaptant à la tâche de reconnaissance vocale.

Le modèle DRAGON est basé sur plusieurs caractéristiques. Tout d'abord, il est basé sur des règles génératives, c'est-à-dire par exemple que si l'on connait un mot donné et sa durée de prononciation, il est possible de déterminer quels

phonèmes ont été prononcés pour dire ce mot. Baker explique que le théorème de Bayes permet, à partir de ces règles, de réaliser des hypothèses *a posteriori* : le modèle note la probabilité d'une certaine suite de phonèmes à partir d'une séquence de mots, et en appliquant le théorème à cette probabilité, on peut obtenir la séquence de mots à partir de la suite de phonèmes.

Ce modèle est aussi un système organisé de manière hiérarchique, une phrase est découpée en plusieurs mots, un mot en plusieurs phonèmes, etc. (Baker, 1975a).

Il existe également une représentation intégrée qui reprend la construction des modèles de Markov en incluant la matrice de transition et les probabilités conditionnelles. Les connaissances nécessaires au modèle sont représentées directement dans cette structure, ce qui facilite sa lecture et son utilisation.

Un autre avantage de DRAGON selon Baker est ce qu'il appelle le « general theoretical framework », un cadre théorique général donc. Le fait de pouvoir représenter les modèles de Markov cachés par une simple équation rend le modèle plus facile à comprendre et à implémenter.

Enfin, ce modèle permet de supporter un large vocabulaire sans que l'espace de stockage pour les calculs soit trop grand et impossible à gérer. « Le nombre total de calculs est linéaire en fonction de la longueur de l'énoncé » (Baker, 1975a, 25) [notre traduction].

À la même période, mais de manière tout à fait indépendante, des chercheurs de l'IBM Thomas J. Watson Research Center[3], dont Frederick Jelinek, proposent également l'utilisation des HMM pour la transcription automatique.

Jelinek et al. (1975) théorisent un décodeur linguistique statistique (LSD). Le but est de réussir à décoder de la parole continue et pas juste de la parole dans un cadre très précis et normé comme c'était le cas avant. Dans l'article

3. Pour plus d'informations sur IBM, https://research.ibm.com/labs/yorktown-heights

Design of a linguistic statistical decoder for the recognition of continuous speech (Jelinek et al., 1975), les auteurs expliquent que, jusqu'à présent, la tâche à décoder est souvent limitée à un domaine. Ils prennent l'exemple d'un jeu d'échecs. Si le cadre est limité par un sujet, le nombre de phrases possibles l'est aussi.

Malgré la volonté de Jelinek et son équipe d'agrandir la recherche, dans ce premier article de 1975, on peut déjà noter quelques limites. En effet, il est dit dans l'article que les phrases sont énoncées clairement et dans un « environnement relativement sans bruit » (Jelinek et al., 1975, 250) [notre traduction]. De plus, le vocabulaire du corpus fait plus ou moins 10 000 mots, ce qui est déjà beaucoup pour l'époque, mais qui aujourd'hui évidemment semble peu.

Dans l'article de Jelinek et al., on retrouve également l'emploi de règles phonétiques comme énoncées chez Baker et, évidemment, l'utilisation des HMM.

2.1.2.2 L'utilisation des HMM

Comme mentionné dans la section précédente, avant d'être utilisées pour la reconnaissance vocale, les chaines de Markov sont avant tout un modèle statistique. Le professeur Rabiner décrit ce modèle comme ceci :

> [C'est] un système qui peut être décrit à chaque instant comme étant dans un des états d'une liste composée de N états distincts [...]. À intervalle régulier, le système subit un changement d'état (qui peut potentiellement être un retour à son même état) selon une série de probabilités associées à cet état.
> (Rabiner, 1989, page 258) [notre traduction]

Les chaines de Markov permettent donc de décrire des états changeants. On comprend pour cette raison l'intérêt de les utiliser sur la parole puisque

la parole continue est une suite d'états changeants.

Cependant, il faut aller au-delà de ce modèle. En effet, Rabiner explique que la chaine de Markov, dans sa plus simple représentation, où tous les états sont visibles, est trop restreinte pour expliquer des problèmes plus abstraits. C'est là qu'apparaissent les modèles de Markov cachés. La couche cachée va permettre de rendre le modèle plus adapté à certaines questions, notamment pour la reconnaissance vocale.

Rabiner prend l'exemple d'une pièce lancée et de la probabilité de faire pile ou face. La face cachée est ici le fait que, lors de l'expérience supposée, la pièce est lancée derrière un rideau et que la personne qui reçoit les résultats n'a aucune manière de savoir si une seule ou plusieurs pièce sont lancées, c'est ce genre d'informations qui sont représentées dans la couche cachée (1989, 259).

Les HMM sont définis par plusieurs éléments. Tout d'abord, les différents états. Dans la représentation statistique de base, appelée modèle ergodique, chaque état peut être relié à n'importe quel autre état (1989, 260). Dans la représentation utilisée pour la reconnaissance vocale, on va préférer le système « gauche-droite » qui est une représentation linéaire. On passe de l'état initial, à des états intermédiaires et on termine par l'état final. Ce système est utile pour la parole, car il permet de comprendre un signal qui change avec le temps, comme la parole (1989, 266). Dans la figure 1, on peut observer une représentation du modèle ergodique (a) et une représentation du système « gauche-droite » (b).

Le deuxième élément est le nombre d'états que l'on peut observer. Dans l'exemple de la pièce, il y en a deux, pile ou face. S'il y avait trois pièces, le nombre d'états serait de 9, etc.

Le troisième élément est la probabilité des états de transition. Pour trouver

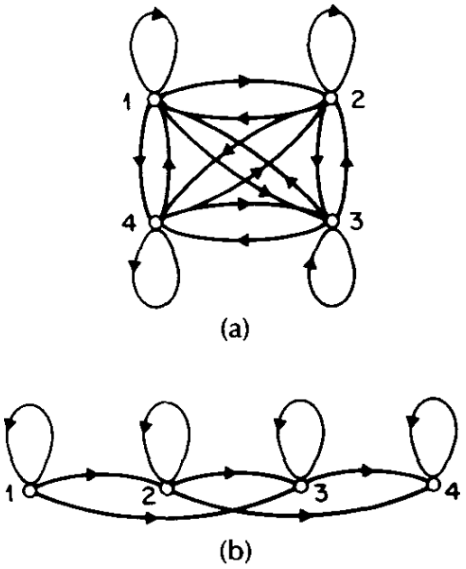

FIGURE 1 – Représentations du modèle ergodique (a) et du modèle « gauche-droite » (b) (pris de Rabiner, 1989, Figure 7, p. 266)

la séquence de transition la plus probable pour passer d'un état à l'autre, on va utiliser l'algorithme de Viterbi qui va permettre de ressortir la meilleure séquence parmi toutes celles qui sont possibles (1989, 264).

Comment adapter le modèle à la reconnaissance vocale ? Rabiner explique que l'utilisation des HMM, transposée à un système de reconnaissance de la parole, se découpe en 5 étapes (1989, 275) :

1. Tout d'abord, il y a l'analyse des caractéristiques : on commence par faire une analyse du signal sonore pour obtenir des vecteurs d'observation. Ces vecteurs vont être utilisés pour entrainer les HMM.
2. Ensuite, il y a la phase appelée « *unit matching system* ».

   Avant d'expliquer cette phase, il faut comprendre ce qu'est l'unité sonore dans ce modèle. Une unité sonore peut être un phonème ou

une syllabe, mais cela peut aussi être un mot complet. Avant de faire le *unit matching system*, il faut choisir l'unité la mieux adaptée à la tâche.

Rabiner explique que pour la reconnaissance de grand corpus de vocabulaire (dans les années 1980, cela veut dire 1000 mots ou plus, ce qui peut sembler peu aujourd'hui mais était conséquent pour l'époque), c'est presque obligatoire d'utiliser des « *sub-word* », c'est-à-dire des parties de mots et non des mots entiers, car « il serait assez difficile d'enregistrer un set d'entrainement adéquat pour créer un modèle de Markov caché pour des unités de la taille d'un mot ou plus large (1989, 275) [notre traduction]. » . Il explique que par contre, lorsque la tâche est plus petite ou que le vocabulaire est plus petit, alors il est possible de choisir le mot comme unité.

Après avoir choisi l'unité sonore, on va l'associer avec une unité prédéfinie faisant partie d'un inventaire qu'on aura obtenu grâce à un entrainement, c'est l'étape du *matching*. Dans cet inventaire, les unités sont représentées sous la forme de HMM. Chaque unité a un début, un milieu et une fin et ces unités peuvent être reliées entre elles. Le système va calculer pour chaque séquence sonore la probabilité qu'elle soit associée aux différentes unités prédéfinies et va choisir celle avec la plus grande probabilité.

3. La troisième étape est le « *lexical decoding* » ou décodage lexical. Une contrainte est ajoutée au modèle, on vérifie que les possibilités explorées correspondent bien à des mots présents dans le vocabulaire. Rabiner explique que lorsque les unités à reconnaitre sont déjà des mots, cette étape est supprimée.

4. La quatrième étape est l'analyse syntaxique. Cette étape ajoute éga-

lement des contraintes. En effet, on s'assure que le mot s'inscrit bien dans une séquence qui est syntaxiquement correcte et qui existe. Tous les mots ne peuvent pas se suivre dans une langue, certaines natures de mots ne se retrouvent jamais les unes derrière les autres. Évidemment, comme pour le point précédent, cette tâche n'est pas nécessaire si la tâche principale est de reconnaitre des numéros, puisqu'ils peuvent tous se suivre. Mais elle est cruciale lorsqu'on veut décoder la parole continue.

5. Enfin, la dernière étape est l'analyse sémantique. Cette tâche est un peu un bonus, elle ajoute des contraintes aux deux tâches précédentes. Cela permet de supprimer certaines phrases syntaxiquement correctes, mais qui n'existent pas en réalité.

Ces étapes sont représentées dans le schéma ci-dessous :

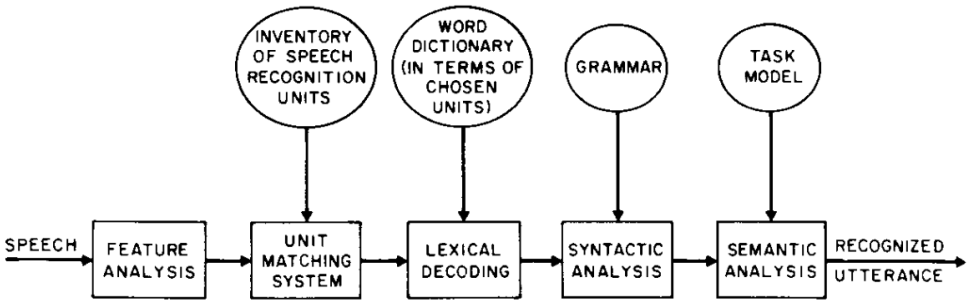

FIGURE 2 – Les différentes étapes d'un système de reconnaissance vocale utilisant les HMM (pris de Rabiner, 1989, Figure 11, p. 275)

Bien que le fonctionnement et l'utilisation des HMM soient connus depuis les années 1970, c'est seulement dans les années 1990 que le modèle a pu être réellement utilisé pour la reconnaissance vocale de manière indépendante du/de la locuteurice et sur un large vocabulaire (Young, 1996, 48). En effet,

avant les modèles montraient surtout de bons résultats sur la reconnaissance de mots isolés, car les recherches sur la parole continue étaient trop chères et demandaient trop de puissance de calcul pour être vraiment investiguées (Rabiner, 1989, 283).

Le modèle va évoluer et les études vont montrer que, pour transcrire la parole continue, on ne peut pas se baser que sur des phonèmes.

Pour rappel, dans la reconnaissance vocale, les phonèmes sont représentés par des HMM et ces HMM sont interconnectés entre eux pour former des mots puis des phrases. Les HMM des phonèmes possèdent normalement trois états : une entrée, un milieu et une sortie. Ces états sont connectés par des arcs et les HMM sont connectés entre eux via les entrées et sorties (Young, 1996, 48).

Pour entrainer les modèles, un seul HMM par phonème n'est pas suffisant. En effet, la prononciation d'un phonème dépend de son contexte, du phonème qui le précède et de celui qui lui succède, c'est pourquoi on a pensé à la technique des triphonèmes « chaque phonème a un HMM distinct pour chaque paire unique de voisins gauche et droite (1996, 48) [notre traduction]. »

Le problème de l'utilisation des triphonèmes est que ça multiplie le nombre de phonèmes d'une langue de manière cubique et que donc cela fait trop d'unités à entrainer (1996, 49). Il faut donc ajouter une technique de *smoothing*.

Une des techniques de *smoothing* dans les années 1990 était le *tying*, qu'on peut traduire par « attacher ». C'était le fait d'associer des paramètres ensemble, par exemple les différents états similaires au sein des HMM. Au départ, chaque triphonème a son propre HMM avec trois états, après le tying, ceux qui sont compatibles phonétiquement vont parfois partager la même entrée, ou la même sortie, etc. Cela permet d'avoir plus de données pour entrainer chaque état des HMM. « L'idée est de mettre ensemble des états

qui sont indiscernables acoustiquement (1996, 49) [notre traduction]. »

Les états sont mis ensemble grâce à un arbre phonétique de décision binaire. Chaque phonème passe par l'arbre, et en fonction des réponses données, finit dans une des feuilles de l'arbre. Les phonèmes appartenant à la même feuille sont mis en commun (1996, 50).

Comme dit précédemment, il y a des règles à respecter dans une langue, ces règles vont permettre de savoir quel mot a le plus de chances de suivre celui qui vient d'être transcrit. On va utiliser les N-grams pour calculer ces probabilités. Les N-grams ont l'avantage « d'encoder simultanément la syntaxe, la sémantique et la pragmatique » (1996, 50) [notre traduction] et d'utiliser le contexte direct : le mot précédent. De plus, il n'est pas nécessaire d'avoir des règles préalables pour utiliser les N-Grams, les probabilités sont calculées directement à partir du texte (1996, 50).

Nous n'allons pas nous étendre plus longtemps sur l'utilisation des HMM, bien qu'il y ait encore beaucoup à dire, puisqu'ils ont été remplacés. Cependant, il nous semblait pertinent de mieux comprendre leur fonctionnement et l'évolution de la discipline pour appréhender les nouvelles technologies. Ce qui est sûr, c'est que les HMM ont marqué l'évolution de la reconnaissance vocale et ont dominé la recherche pendant près de 40 ans avant d'être remplacés par les réseaux de neurones et le *deep learning*.

Maintenant que nous avons une meilleure vision de ce que sont les HMM et de leur utilisation, nous allons introduire les réseaux de neurones récurrents en parlant d'une période intermédiaire : les modèles hybrides.

2.1.3 L'approche hybride et l'apparition des réseaux de neurones

Dans la section précédente, nous avons dit que les HMM ont dominé le champ des recherches durant de longues années, et ce fut le cas. Mais l'utilisation des réseaux de neurones dans la reconnaissance vocale a débuté dès les années 1980 (Morgan and Bourlard, 1995, 742), il a juste fallu attendre quelques années, et un passage par des modèles hybrides (qui mélangent les réseaux de neurones et les HMM) pour obtenir des résultats satisfaisants.

Au départ, les réseaux de neurones ne sont pas utilisés directement comme modèle pour la RAP, mais comme le disent Morgan et Bourlard plutôt comme « un composant clé dans le système » (1995, 742) [notre traduction] de reconnaissance de la parole continue. Dans les années 1990, les modèles entièrement basés sur les réseaux de neurones fonctionnaient pour les petites tâches de reconnaissance, comme la reconnaissance de mots isolés ou de numéros, mais les auteurs expliquent que, pour la reconnaissance de la parole continue sur de larges données, les réseaux de neurones seuls ne sont pas encore suffisamment efficaces (1995, 754).

C'est pour cela qu'on parle d'approche hybride, on va utiliser à la fois les HMM et les réseaux de neurones. Au départ, les RNA (réseaux de neurones artificiels) étaient surtout employés pour modéliser les séquences de caractéristiques et prédire les probabilités des états acoustiques. Ces probabilités permettaient de décider avec plus de certitude quels mots ou phonèmes correspondaient à un spectre sonore. Les RNA remplacent le modèle de mélange gaussien, utilisé dans les versions antérieures des systèmes de RAP, pour émettre ces probabilités (1995, 758-760).

Les systèmes hybrides sont présentés comme la nouvelle alternative aux

HMM. Ils semblent plus efficaces, notamment au niveau du temps d'exécution (1995, 761). Cependant, ce modèle n'est pas encore idéal, en effet, à l'époque, les données annotées à la main sont encore nécessaires pour l'entrainement des réseaux de neurones, ce qui a un coût et est difficile à obtenir (1995, 760). De plus, le coût, en termes de puissance de calcul, de l'entrainement des RNA était beaucoup plus élevé que celui des HMM (1995, 767).

2.1.4 Les réseaux de neurones récurrents et le modèle encodeur-décodeur

2.1.4.1 Les réseaux de neurones

Bien que les réseaux de neurones aient été employés en reconnaissance vocale depuis de nombreuses années, c'est seulement dans les années 2010 que leur utilisation se généralise et que des modèles utilisant exclusivement les RNA, et non plus l'approche hybride, font leur apparition.

L'utilisation de manière *end-to-end*, c'est-à-dire en les utilisant du début à la fin, des RNA va permettre d'améliorer les modèles de reconnaissance vocale et notamment de résoudre un problème récurrent de la discipline, déjà expliqué par Morgan and Bourlard :

> [C]omment une séquence d'entrées (par exemple, une séquence de vecteurs) peut-elle être expliquée par une séquence de sorties (par exemple, une séquence de phonèmes ou de mots) alors que les deux séquences ne sont pas synchronisées (puisqu'il y a en fait de nombreux vecteurs acoustiques associés pour chaque mot ou phonème prononcé) ? (Morgan and Bourlard, 1995, page 753) [notre traduction]

Ce que les auteurs veulent dire, c'est que la transcription d'un spectre sonore en mots écrits est rendue plus difficile, car l'entrée et la sortie ne sont

pas de même taille. Or, ce qui rend difficile l'utilisation des HMM, c'est le fait qu'il faut entrainer le modèle à aligner des données de tailles différentes. Pour ce faire, les modèles doivent être entrainés avec des données annotées manuellement, ce qui crée potentiellement plus d'erreurs.

Les RNN (*Recurrent Neural Network*) sont particulièrement adaptés pour répondre à cette différence de taille entre l'entrée et la sortie du modèle. En effet, comme l'expliquent Graves et al. (2013), l'avantage d'un entrainement de bout en bout (*end-to-end*) c'est que les RNN sont entrainés à passer directement de la séquence acoustique aux phonèmes de sortie. L'alignement des données orales et écrites, le fait d'utiliser des données annotées manuellement par des humains pour la phase d'entrainement, n'est plus nécessaire avec cette approche. On peut associer des fichiers d'entrée et de sortie de tailles différentes sans jamais tenir compte de leurs tailles respectives (2013, 6646).

Les RNN sont aussi très efficaces car ils ont l'avantage d'être *deep in time*, c'est-à-dire qu'ils gardent en mémoire tous les états cachés précédents (2013, 6645). En effet, dans un RNN, la couche cachée du réseau de neurones possède une connexion récurrente, ce qui permet au modèle d'être connecté non seulement à l'entrée qu'il traite, mais également à toutes les entrées précédentes alors que dans le modèle traditionnel, les entrées sont traitées une à une (Jurafsky and Martin, 2023, 186). Cette couche cachée peut garder en mémoire des informations qui remontent jusqu'au début de la phrase ou même de la séquence d'enregistrement, ce qui en fait un outil précieux pour la prédiction du mot en sortie. Les RNN ne sont pas limités dans leur contexte contrairement aux modèles N-gram qui sont limités aux quelques mots précédents (2023, 189).

Les recherches de Graves et al. ont fortement fait avancer la recherche pour l'utilisation des RNN dans la reconnaissance de la parole. Dans leur article

Speech Recognition with deep recurrent neural networks (2013), ils proposent un modèle entièrement basé sur les RNN pour une tâche de reconnaissance vocale. Les auteurs proposent un modèle inédit basé sur plusieurs avancées technologiques de l'époque.

Tout d'abord, un élément clé dans l'augmentation de l'efficacité des modèles hybrides est l'utilisation de *deep* RNN. Ces modèles sont créés « en empilant plusieurs couches cachées de RNN les uns sur les autres, avec la séquence de fin d'une couche qui est la séquence d'entrée de la couche suivante (Graves et al., 2013, 6646) [notre traduction]. »

Ensuite, il existe plusieurs types de RNN, notamment les RNN bidirectionnels, dont l'architecture est représentée dans la figure 3, qui ont à la fois accès au contexte précédent mais également à ce qui va suivre. Les BRNN possèdent donc deux couches cachées différentes qui permettent de traiter les données à la fois en avant et en arrière (*forward-backward*), ensuite, les informations récupérées sont mises ensemble afin d'anticiper la sortie (2013, 6645).

Les LSTM (*Long Short-Term Memory*), sont aussi une architecture modifiée basée sur les RNN, mais ce modèle « construit des cellules de mémoire pour enregistrer l'information (Graves et al., 2013, 6645) [notre traduction]. » Les LSTM permettent un accès à un contexte plus large lors de l'entrainement.

Les auteurs proposent donc d'implémenter des réseaux de neurones récurrents profonds et bidirectionnels. Cela permet que chaque couche cachée du modèle reçoive un *input* venant à la fois des couches qui le précèdent, mais également de celles qui lui succèdent. Ils proposent aussi d'implémenter cela avec un LSTM au lieu d'un simple RNN. Ils sont les premiers à implémenter un *deep* LSTM, au lieu d'un LSTM à une seule couche, pour une tâche de large reconnaissance vocale (2013, 6646).

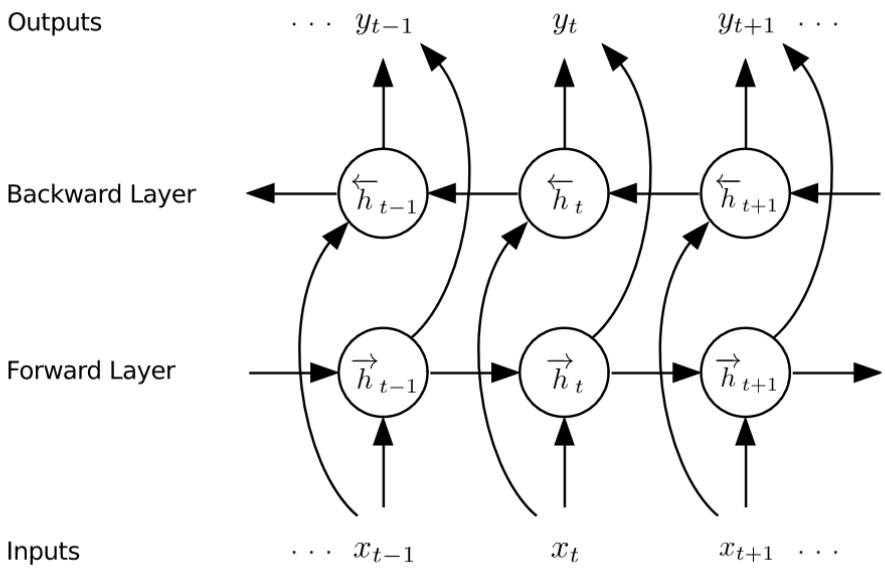

FIGURE 3 – Architecture d'un RNN bidirectionnel (pris de Graves et al., 2013, Figure 2, p. 6646)

2.1.4.2 Le système encodeur-décodeur

Le système encodeur-décodeur est une architecture qu'on peut diviser en deux parties. L'encodeur, tout d'abord, est la partie du mécanisme qui va transformer l'*input* (ici, la parole) en une nouvelle séquence d'*inputs*, aussi appelés des vecteurs de caractéristiques, qui vont rendre les données lisibles pour le décodeur. L'encodeur est souvent un RNN (Chorowski et al., 2015, 2).

Dans les premiers modèles, le décodeur utilise un CTC *(Connectionist Temporal Classification)*[4] ou un transducteur[5], afin de calculer les probabilités *a posteriori*, pour trouver la meilleure séquence de sortie qui va être produite par le décodeur.

Le transducteur est une sorte de version améliorée du CTC, puisque le

4. pour plus d'informations sur le modèle CTC, voir Graves et al. (2006)
5. pour plus d'informations sur le modèle avec un transducteur, voir Graves (2012)

CTC base la distribution de ses probabilités seulement sur l'entrée acoustique du modèle, ce qui en fait un modèle purement acoustique. Alors que les transducteurs se basent aussi sur ces entrées acoustiques, mais ils possèdent également un RNN supplémentaire qui « prédit chaque phonème en fonction des précédents, produisant de ce fait un modèle à la fois acoustique et langagier (Graves et al., 2013, 6647) [notre traduction]. »

Malgré que les transducteurs soient déjà plus efficaces que la méthode CTC, une grande avancée dans l'utilisation du système encodeur-décodeur a été l'apparition du mécanisme d'attention qui va remplacer les deux autres méthodes.

2.1.4.3 L'apparition de l'attention

Les mécanismes d'attention sont plus efficaces, car ils permettent d'aller chercher l'information nécessaire à tous les moments de la séquence d'entrée (qui est la parole) afin de créer les bons vecteurs. Ils permettent donc de gérer des séquences de paroles plus longues et de prendre l'information utile là où elle est, qu'importe sa localisation dans les données orales (Chorowski et al., 2015, 1-2).

Chorowski et al. expliquent que l'avantage d'utiliser l'attention est que ce mécanisme est la dernière étape pour créer un modèle *end-to-end* complet en reconnaissance vocale, car en 2015, les modèles hybrides sont encore les plus utilisés (2015, 2).

Les modèles basés à la fois sur l'attention et sur les réseaux de neurones ont d'abord été développés pour la traduction ou pour la retranscription de l'écriture manuscrite. Mais si l'on considère que « apprendre à reconnaitre la parole peut être vu comme apprendre à générer une séquence (la transcription) étant donné une autre séquence (la parole) » (Chorowski et al., 2015, 1) [notre

traduction], alors il n'y a pas de raison que le système ne fonctionne pas pour transcrire un signal sonore en mots écrits.

L'attention ayant montré de bons résultats sur les deux tâches précitées, Chorowski et al. se sont penchés sur son utilisation pour la reconnaissance de la parole. Cependant, ce processus présentait différentes difficultés. En effet, les séquences d'entrée sont très longues « ce qui introduit un challenge pour distinguer des fragments de parole similaires dans un seul énoncé (Chorowski et al., 2015, 1) [notre traduction]. » De plus, les extraits vocaux sont assez bruyants, ce qui les rend également difficiles à transcrire.

Comparé à un système RNN classique qui utilise le CTC ou les transducteurs, le mécanisme d'attention permet de rendre plus rapide la phase de décodage. Cela s'explique, car la procédure de recherche de faisceaux (*beam search procedure*), qui est une méthode d'optimisation utilisée pour les systèmes *end-to-end* en RAP et qui permet de trouver les meilleures séquences de sorties, est améliorée grâce à l'attention. En effet, avec l'attention, le faisceau de recherche peut être plus petit qu'avec les systèmes RNN classiques, et cela permet de diminuer le temps de calcul et de rendre la phase de décodage plus rapide (2015, 5).

Un autre avantage de l'attention est que l'alignement qui en résulte est « non monotone », c'est-à-dire que l'ordre des éléments dans l'alignement peut changer entre l'entrée et la sortie du modèle, ce qui rend le modèle plus flexible et peut « le rendre adapté à une large variété d'autres tâches que la reconnaissance de la parole (2015, 5) [notre traduction]. » Contrairement à la méthode CTC ou aux transducteurs qui sont « monotones » et récupèrent les informations de façon linéaire par rapport au sens de lecture de la phrase (2015, 5).

Un des résultats positifs des recherches de Chorowski et al. est le fait que

leur modèle est capable de reconnaitre des énoncés plus longs que ceux avec lesquels il a été entrainé, ce qui était un des problèmes des modèles précédents, ils n'étaient pas robustes pour ce genre de tâche (2015, 8).

Nous l'avons compris, l'attention a donc été une grande avancée dans le système encodeur-décodeur, que ce soit pour la reconnaissance vocale ou pour les autres tâches de TAL. Mais un mécanisme va pousser l'attention encore plus loin, le *Transformer*.

2.1.5 Au delà des RNN : les *Transformers*

L'architecture des *Transformers* a été présentée pour la première fois en 2017, par Vaswani et al. dans leur article *Attention is all you need*. Leur idée était de créer un modèle uniquement basé sur l'attention et non plus sur les réseaux de neurones récurrents (2017, 1).

Leur modèle n'est pas un modèle de transcription automatique, mais bien de traduction. Mais les deux disciplines ont des points communs, puisqu'en traduction comme en reconnaissance de la parole, les séquences d'entrées et de sorties ne sont pas forcément de la même taille. En effet, il faut parfois plusieurs mots d'une langue donnée pour en traduire un seul dans une autre.

Les auteurices expliquent que l'architecture des modèles récurrents atteint ses limites lorsque les séquences d'entrées deviennent trop longues et que c'est leur « nature séquentielle inhérente » (2017, 2) [notre traduction] qui pose problème.

Le modèle *Transformer* est le premier modèle de traduction basé entièrement sur ce qu'on appelle la « *self-attention* », ou « *intra-attention* », que nous traduirons par « auto-attention » dans le reste de ce mémoire. L'auto-attention est un mécanisme qui calcule la représentation d'une séquence en prenant et en mettant ensemble les informations de différentes positions venant de cette

même séquence (2017, 2).

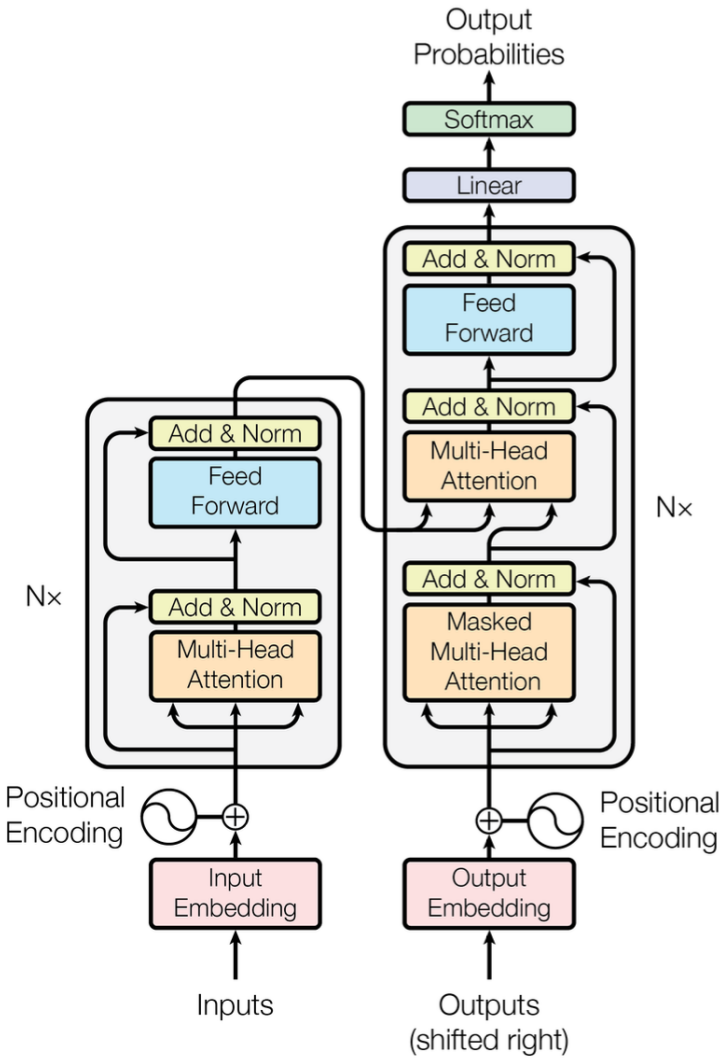

FIGURE 4 – Architecture d'un *Transformer* (pris de Vaswani et al., 2017, Figure 1, p. 3)

Comme on peut le voir dans la figure 4, le modèle *Transformer* est aussi composé d'un encodeur et d'un décodeur. Les deux piles (*stacks*) de l'encodeur et du décodeur sont composées respectivement de 6 couches identiques.

La différence est que, pour chaque couche, l'encodeur a deux sous-couches associées alors que le décodeur en a trois (2017, 3).

La première couche des *stacks* est un mécanisme d'auto-attention multi-têtes. On parle d'attention multi-têtes car ce sont plusieurs couches d'attention qui fonctionnent de manière simultanée comme on peut le voir dans la figure 5. La deuxième couche est un « simple [...] réseau de neurones f*eed-forward* complètement connecté (2017, 3) [notre traduction]. ». La troisième couche, qui est donc présente uniquement dans le décodeur, « effectue [à nouveau] une attention multi-têtes sur la sortie de la pile d'encodeurs » (2017, 3) [notre traduction].

Une des avancées du modèle *Transformer* est donc l'utilisation de l'attention multi-têtes à différents endroits du modèle. En effet, cela permet au modèle d'avoir accès à plus d'informations puisqu'il peut accéder à de l'information venant de positions différentes et de représentations différentes. Les auteurices expliquent que pour ces premières recherches, il utilisent 8 couches (ou têtes) d'attention en parallèle (2017, 5).

La technologie *Transformer*, puisqu'elle n'utilise pas la récurrence comme les RNN, afin de garder l'ordre des séquences, a besoin de « *positional encodings* ». Ils permettent au modèle d'être informé de la position des *tokens* (ici, les mots) dans la séquence. Pour ce faire, ces « *positional encodings* » sont ajoutés en bas de la structure sur l'encodeur et le décodeur juste après la création des vecteurs d'entrée, comme on peut le voir sur la figure 4 (2017, 6).

Vaswani et al. décrivent trois avantages de l'utilisation des couches d'auto-attention par rapport aux modèles récurrents.

Tout d'abord, la complexité[6] de calcul de chaque couche. Comme on

6. En informatique, le calcul de la complexité permet de connaitre la puissance de calcul nécessaire pour faire tourner un programme. Elle est étudiée en algorithmique afin de diminuer au maximum les coûts en temps et en mémoire des calculs utilisés en informatique. La plus petite complexité est constante et est représentée par la notation

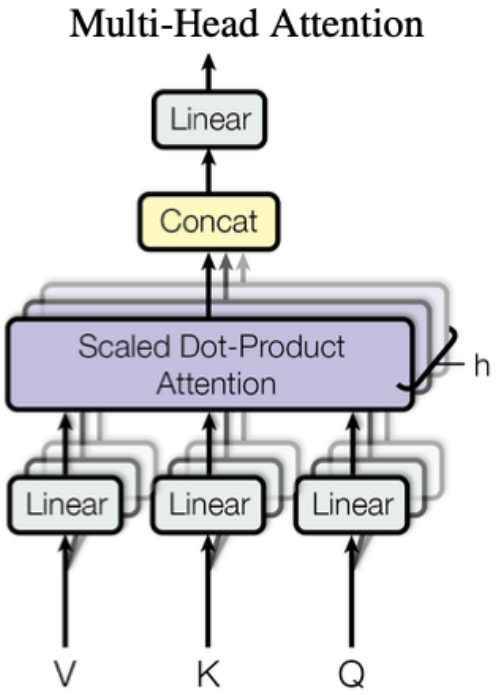

FIGURE 5 – Architecture d'un mécanisme d'attention multi-têtes (pris de Vaswani et al., 2017, Figure 2, p. 4)

peut le voir dans la figure 6 la complexité computationnelle des systèmes d'auto-attention est meilleure que celle des modèles récurrents (*O(1)* VS *O(n)*), car elle produit un nombre constant d'opérations séquentielles pour relier les positions d'entrées et les sorties.

Ensuite, les couches d'attention permettent plus facilement que des opérations s'exécutent en parallèle.

Enfin, les couches d'auto-attention sont utiles, car elles réduisent la lon-

O(1). La complexité la plus élevée est dite factorielle et est représentée par la notation *O(n !)* [source : cours d'*Introduction à l'algorithmique* (LINFO1103), UCLouvain année académique 2022-2023].

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

FIGURE 6 – Résultats comparatifs de la complexité d'un RNN, d'un CNN (Convolutional Neural Network) et d'un modèle utilisant l'attention (pris de Vaswani et al., 2017, Table 1, p. 6)

gueur du chemin que les signaux doivent parcourir entre les dépendances à long terme et facilitent de ce fait les connexions entre des informations en début de séquence qui doivent être connectées à des informations en fin de séquence par exemple. Le contexte complet d'un mot ou d'un bout de phrase peut être exploré plus rapidement si les chemins entre les dépendances à long terme sont plus courts (2017, 6-7).

Sur la tâche de traduction qu'ils s'étaient fixée, les auteurices ont obtenu de meilleurs résultats que ceux de tous les modèles précédents (2017, 8). Ielles ont aussi testé leur modèle sur une tâche de *parsing*[7] pour voir si le *Transformer* allait pouvoir être généralisé pour d'autres tâches TAL et ont obtenu des résultats très satisfaisants (2017, 9).

On le sait aujourd'hui, les *Transformers* sont utilisés dans toutes les tâches de TAL. C'est après cette étude que la généralisation de l'utilisation des *Transformers* s'est répandue dans toutes les disciplines. Aujourd'hui, c'est l'architecture la plus utilisée et la plus performante.

7. Le *parsing* est une tâche de TAL qui consiste à analyser la structure syntaxique des phrases afin de comprendre comment les mots sont reliés entre eux au sein d'une phrase. Le *parsing* permet ainsi de déterminer la structure grammaticale des phrases et, par conséquent, de vérifier si elles sont grammaticalement correctes [source : cours de *Traitement automatique du langage naturel* (LFIAL2620), UCLouvain année académique 2022-2023].

Passons maintenant au cas qui nous intéresse particulièrement pour la RAP, le modèle Whisper d'OpenAI.

2.1.5.1 Le modèle Whisper

Whisper est un modèle basé sur l'architecture encodeur-décodeur du *Transformer*. Il a été développé par Radford et al. (2023) [8] et va pousser les recherches en RAP encore plus loin.

Radford et al. expliquent qu'un des problèmes de l'entrainement des modèles est le manque de données enregistrées. Ils partent du postulat qu'il vaut mieux abandonner le *gold standard* des anciens corpus d'entrainement et qu'il est préférable de miser sur la quantité plutôt que sur la qualité, comme c'était déjà le cas dans la reconnaissance d'images. D'autres auteurices avaient déjà voulu utiliser ces types de données, mais jamais à une échelle aussi grande que pour Whisper. En effet, Whisper a été entrainé sur 680 000 heures de données audio annotées (2023, 2).

Il existe 6 versions différentes du modèle Whisper : tiny, base, small, medium, large et largev2 [9]. Ces versions ont toutes été entrainées avec un nombre de *layers* et de paramètres différents. Le tableau 1 ci-dessous reprend ces données.

Whisper a différents avantages par rapport aux autres modèles de RAP.

Tout d'abord, c'est un modèle qui a la particularité de s'entrainer seul et qui n'a pas besoin d'être affiné (2023, 2).

Whisper s'adapte aussi à de nombreuses langues. En effet, le modèle a été entrainé à la fois pour l'anglais, mais également de manière multilingue. Parmi

8. Pour les informations complètes sur les paramètres de Whisper, voir (Radford et al., 2023, page 3, point 2.2).

9. Il existe en fait actuellement une septième version, la version largev3, mais qui a été affinée après le début de nos recherches, nous ne l'avons donc pas incluse dans ce mémoire.

Modèle	***Layers***	**Paramètres**
Tiny	4	39 M
Base	6	74 M
Small	12	244 M
Medium	24	769 M
Large	32	1550 M
Largev2	32	1550 M

TABLE 1 – Les différents modèles de Whisper et leurs paramètres (Radford et al., 2023, p. 5)

les données, 117 000 heures sont tirées de 96 autres langages que l'anglais (2023, 2).

Un autre avantage est que l'approche de Whisper limite le prétraitement, en effet, la phase de standardisation est supprimée, le modèle est entrainé sur des données brutes, ce qui simplifie sa structure.

Les données d'entrainement proviennent d'Internet, c'est ça qui a permis d'obtenir un nombre aussi élevé de données. Pour sélectionner les données orales, les chercheurs et chercheuses ont pris celles dont il existait déjà une transcription sur Internet. Cela peut poser problème car les transcriptions ne sont pas toujours de bonne qualité, les auteurices ont donc dû trouver des méthodes pour filtrer et améliorer les transcriptions. Ces filtres visaient surtout à supprimer des données les transcriptions déjà produites par d'autres modèles de RAP, car leur ajout aux données risquait d'influencer le modèle de manière négative (2023, 2).

Les fichiers audio sont découpés en segments de 30 secondes. Ses segments sont associés avec la transcription correspondante. Les auteurices expliquent que même les segments sans parole sont utilisés (2023, 2). En effet, ces segments vont servir à entrainer la tâche de détection de la parole. Whisper a été pensé pour être non seulement un modèle de transcription de la parole, mais aussi

un modèle multitâches adapté à toutes les tâches liées à la reconnaissance vocale. En plus d'être entrainé pour la transcription, il est aussi entrainé pour la traduction de n'importe quelle langue vers l'anglais, pour la détection de la voix et aussi pour la détection de la langue. Toutes ces tâches ont été entrainées en même temps pour limiter la complexité du modèle(2023, 4).

Le but de Whisper est de créer un modèle robuste qui aurait de bons résultats sur n'importe quelles données. Pour le tester, les auteurices ont utilisé des *datasets* existants, mais au lieu de diviser ces *datasets* en un set de test et un set d'entrainement, ils ont utilisé la méthode *zero-shot* (2023, 5), c'est-à-dire qu'ils ont directement testé le modèle sur de nouvelles données jamais observées auparavant[10].

Les résultats étaient bons, en effet Radford et al. (2023) constatent ceci :

> Le plus petit modèle *zero-shot* de Whisper, qui a seulement 39 millions de paramètres et un WER 6,7% sur le corpus *test-clean* de LibriSpeech peut presque compétiter avec le meilleur modèle supervisé de LibriSpeech lorsqu'il est évalué sur d'autres *datasets* (Radford et al., 2023, page 6) [notre traduction].

La résistance au bruit a également été testée en ajoutant des bruits sur des données. Le modèle montre une bonne résistance générale(2023, 9).

Enfin, le modèle a également été comparé à la performance humaine en termes de transcription et les résultats obtenus par le modèle sont proches de ceux d'humains transcripteurices professionnel·le·s (2023, 10).

Whisper est donc un modèle très performant qui a montré de bons résultats. Bien que les auteurices disent qu'il peut sûrement encore être affiné pour certaines tâches, il est déjà très efficace et c'est pourquoi nous avons décidé d'étudier ce modèle en particulier dans notre mémoire.

10. Les informations sur la méthode *zero-shot* viennent de HuggingFace : https://huggingface.co/tasks/zero-shot-classification.

Maintenant que nous avons expliqué pourquoi notre choix de modèle s'était porté sur Whisper, nous allons présenter les métriques qui vont nous permettre d'évaluer nos transcriptions.

2.2 Le *Word Error Rate* et les métriques d'évaluation

Le *Word Error Rate* (WER) est un système d'évaluation de la transcription automatique qui existe depuis de nombeuses années et qui est toujours employé aujourd'hui. Le principe est simple, il faut comparer la transcription automatique à la version produite manuellement en comptant, pour la transcription automatique, les mots insérés en trop, les mots substitués à d'autres et les mots qui ont complètement disparu. Après avoir additionné ces observations, il faut diviser le nombre obtenu par le nombre total de mots que compte la phrase et l'on obtient le pourcentage de WER.

Ce système, bien que toujours d'actualité, a cependant quelques défauts qui n'ont pas encore été résolus bien que des pistes existent. Par exemple, la pondération pose problème. En effet, toutes les erreurs ont le même poids, la confusion entre les mots « dans » et « en » aura autant d'importance que la suppression d'un mot plus long comme « dimanche », ce qui ne donne pas toujours des résultats représentatifs lorsqu'on compare deux transcriptions (Jurafsky and Martin, 2023, 343-345).

Une alternative au WER est le CER, le *Character Error Rate*, qui compare les transcriptions, non pas au niveau des mots, mais bien au niveau des caractères qui diffèrent entre les transcriptions. Le CER permet donc de nuancer les erreurs quand deux mots sont très similaires, mais orthographiés différemment.

D'autres pistes ont été proposées pour compléter le WER, il y a notamment celles de Morris et al. (2004) qui propose comme alternatives le MER (*Match Error Rate*) qui est proche du WER, mais pénalise plus fortement les erreurs de substitution que les erreurs d'insertion et de suppression[11].

Morris et al. (2004) a également proposé le WIL, *Word Information Lost*, qui calcule le pourcentage d'informations perdues en se basant sur les erreurs de prédiction de mots[12].

En opposition au WIL, il existe le WIP, *Word Information Preserved* qui, à l'inverse du WIL, calcule le pourcentage de mots bien prédits par le modèle. Un WIP élevé est donc positif, 1 étant un score parfait[13].

D'autres recherches plus récentes comme celles de Bañeras Roux et al. (2022) tendent à améliorer le modèle du WER utilisant les aspects morpho-syntaxiques et sémantiques des mots. Il serait par exemple intéressant que deux mots sémantiquement proches soulèvent une erreur moins importante que deux mots de classes lexicales complètement opposées.

Malgré les nouvelles métriques proposées, le WER est toujours la métrique dominante employée par les chercheurs et chercheuses lors de l'évaluation des modèles de RAP. Nous allons donc principalement nous baser sur le WER dans nos analyses, mais il nous semblait intéressant d'expliquer que d'autres métriques existent et que les recherches pour trouver une métrique plus performante que le WER continuent de nos jours.

Il nous reste un point de théorie à aborder avant de pouvoir commencer à expliquer notre méthodologie, il s'agit du cas particulier des locuteurices non-natif·ve·s en RAP.

11. Voir : https://lightning.ai/docs/torchmetrics/stable/text/match_error_rate.html
12. Voir : https://lightning.ai/docs/torchmetrics/stable/text/word_info_lost.html
13. Voir : https://lightning.ai/docs/torchmetrics/stable/text/word_info_preserved.html

2.3 La reconnaissance de la parole et les accents : le cas des locuteurices non-natif·ve·s

Comme expliqué dans l'introduction, nous avons choisi de nous pencher sur le cas des locuteurices non-natif·ve·s.

Le cas des locuteurices allophones a été étudié à différents moments de l'évolution des recherches en RAP. Les accents des personnes non-natives présentent une difficulté supplémentaire et augmentent de manière générale le nombre d'erreurs dans les transcriptions. Conscient·e·s de cette difficulté, des chercheurs et chercheuses ont cherché des solutions à ce problème en s'aidant des modèles de leur époque et en essayant de les améliorer.

2.3.1 Un sujet de recherche très étendu

Les productions orales des non-natif·ve·s ne sont pas utilisées uniquement pour augmenter la performance des modèles de RAP. En effet, elles ont aussi été étudiées pour repérer les erreurs fréquentes des locuteurices non-natif·ve·s afin de les aider à améliorer leur prononciation, comme dans les recherches de Li et al. (2016).

Les modèles de RAP ont aussi été utilisés pour comparer les compréhensions à l'audition, d'une part d'un·e locuteurice natif·ve écoutant des productions de locuteurices non-natif·ve·s, et d'autre part, d'un système de RAP qui écoute ces mêmes productions. Ces études cherchent à mesurer la différence de compréhension entre un système de RAP et une oreille humaine Inceoglu et al. (2023).

Nous n'allons pas plus développer ces thématiques dans ce point, car elles s'éloignent de nos recherches, mais cela nous semblait pertinent de les

mentionner pour montrer l'étendue de la discipline. Nous allons à présent nous concentrer sur les difficultés qu'ajoute la parole non-native à la transcription automatique en présentant l'évolution de ces difficultés en parallèle avec l'évolution des modèles.

2.3.2 Modifier les modèles pour s'adapter aux données des non-natif·ve·s : le cas des HMM et des RNN

Dans sa thèse de doctorat intitulée *Contributions à la reconnaissance automatique de la parole non-native*, Bouselmi (2008) explique que les difficultés rencontrées sont liées au fait que les modèles avec lesquels il travaille sont stochastiques et statistiques et ont été entrainés pour certains types de propriétés de la parole et que ces propriétés sont différentes de celles des productions des non-natif·ve·s (2008, 121). En effet, en 2008 au moment de sa thèse, ce sont encore les modèles basés sur les HMM qui sont utilisés et ils sont entrainés avec des données annotées et des indications syntaxiques et sémantiques. Il faut donc trouver des techniques pour augmenter ces modèles et les adapter aux productions non-natives.

Les solutions proposées pour adapter les HMM sont les suivantes :

- La première est basée sur le fait que certains phonèmes ne se prononcent pas de la même manière, voire parfois n'existent pas, dans toutes les langues. On va donc modifier les propriétés acoustiques du modèle pour qu'elles correspondent à celles des locuteurices non-natif·ve·s. Mais cette option a le désavantage de diminuer l'efficacité du modèle sur la parole dite « standard » (2008, 121).

- La seconde solution proposée essaie donc de conserver le bon fonctionnement du modèle pour la parole standard en plus de l'augmenter pour la parole non standard. En effet, on va détecter les prononciations non-natives

et modifier le modèle acoustique afin de les y ajouter comme alternative à la prononciation canonique (2008, 121-122).

- Enfin, la troisième est d'ordre grammatical. Les erreurs grammaticales sont détectées et utilisées pour modifier le modèle (2008, 122).

Ces possibilités sont envisagées par Bouselmi pour un modèle basé sur les HMM, mais ces options ont aussi été envisagées pour augmenter des systèmes basés sur les réseaux de neurones. Par exemple, dans leurs recherches, Bada et al. (2020) se basent sur une architecture de réseaux de neurones et leur proposition pour améliorer les transcriptions des personnes non-natives est d'augmenter le lexique en y intégrant les prononciations des locuteurices allophones.

Pour ce faire, ils repèrent les différences de prononciations récurrentes entre natif·ve·s et non-natif·ve·s (leur étude se base sur des personnes francophones parlant en anglais) afin de « générer des règles de création de prononciation » (Bada et al., 2020, 2) [notre traduction].

Mais avec l'évolution des modèles et l'apparition d'outils ultra performants comme Whisper, est-il encore utile de modifier les modèles pour qu'ils s'adaptent en particulier aux cas des locuteurices non-natif·ve·s ?

2.3.3 Whisper et le cas des locuteurices non-natif·ve·s

Nous le verrons dans la partie Analyses (voir partie 4), bien que très performants, les modèles de Whisper ont encore des défauts de transcription. Nous avons choisi de rapporter dans cette section les recherches d'autres chercheurs et chercheuses qui ont également testé les performances de Whisper sur des productions de locuteurices non-natif·ve·s, afin d'avoir une idée des erreurs et difficultés potentielles que nous pourrions rencontrer.

Ballier et al. (2023) ont étudié les productions de locuteurices francophones parlant anglais. Leur approche est assez différente de la nôtre, puisqu'ils ont transcrit des enregistrements de 38 locuteurices de niveau C1 lisant le même passage d'un livre (2023, 3). Leurs données contrairement aux nôtres ne sont donc pas de la parole spontanée. De plus, le texte et le vocabulaire sont toujours les mêmes, ce qui n'est pas le cas dans nos dialogues.

Le fait d'avoir un corpus restreint et un texte identique présente quelques avantages. En effet, cela leur permet de souligner les erreurs récurrentes chez tous les locuteurices et de voir si le modèle Whisper produit les mêmes erreurs d'un·e locuteurice à l'autre et d'une version à l'autre. Ce genre d'étude pourrait être intéressante à tester sur des locuteurices néerlandophones lisant en français également.

Cependant, malgré ces différences, Ballier et al. cherchent à démontrer une hypothèse qui se rapproche de nos questions de recherche. En effet, ils souhaitent voir si les transcriptions varient d'une version de Whisper à l'autre et quel modèle de Whisper reproduit le plus fidèlement les erreurs de langage des locuteurices non-natif·ve·s.

Selon eux, le fait que la version tiny produise un WER plus élevé est un signe que le modèle retranscrit mieux les « bizarreries de prononciation » (Ballier et al., 2023, 3) [notre traduction].

Ils prennent l'exemple du mot « *heat* » régulièrement transcrit par « *it* ». Cette transcription montre la disparition du « h » aspiré, qui est une erreur courante des locuteurices francophones en anglais. Les auteurices soulignent également que selon leurs observations, Whisper accorde plus d'importance à la probabilité phonétique qu'à la probabilité sémantique, ce qui peut parfois donner des résultats étonnants (2023, 7). Nous aurons l'occasion de voir dans nos analyses si cela est vrai pour nous aussi.

Nous souhaitons encore soulever deux points de cet article qui nous seront utiles pour nos propres analyses.

Premièrement, les auteurices expliquent que des erreurs d'hallucination[14] sont produites par les modèles. En effet, ils observent des répétitions : certains mots sont répétés plusieurs fois alors qu'ils ne sont produits oralement qu'une seule fois.

Deuxièmement, les auteurices expliquent que selon eux·elles, la version largev2 est la plus performante sur les autres corpus sur lesquels ielles ont testé les modèles (2023, 8), mais que lorsqu'il s'agit de transcrire de la parole d'apprenant·e·s, le modèle tiny est plus efficace pour pointer les erreurs de prononciation « car il semble plus sensible à la variation phonétique que le modèle plus large (Ballier et al., 2023, 9) [notre traduction]. »

Nous verrons plus tard dans nos analyses si nous arrivons à des constatations similaires. Tout comme Ballier et al., nous souhaitons voir quel modèle est le mieux adapté pour transcrire la parole des non-natif·ve·s, leurs observations nous donnent donc une première idée des phénomènes et résultats que nous pourrions observer lors de nos analyses.

2.4 Conclusion

Dans cet état de l'art, nous avons cherché à nous informer sur l'évolution des techniques utilisées depuis les débuts des recherches en reconnaissance automatique de la parole afin de mieux comprendre ce domaine.

Pour ce faire, nous avons d'abord présenté les systèmes qui fonctionnaient grâce aux modèles de Markov cachés. Ensuite, nous avons présenté l'approche

14. Lorsqu'on parle d'IA générative, le terme « halluciner » est utilisé quand le modèle produit une information inventée. Par exemple, on parle d'hallucination lorsqu'un outil tel que ChatGPT produit une réponse basée sur des sources qui n'existent pas tout en certifiant qu'elles sont correctes et que l'information est bonne.

hybride qui liait à la fois les HMM et les réseaux de neurones. Cela nous a naturellement menée à présenter les modèles basés sur les réseaux de neurones récurrents sur le modèle encodeur-décodeur.

Finalement, après cette évolution historique, nous sommes arrivée à l'apparition du mécanisme d'attention qui a permis de créer l'architecture *Transformer*, qui est le modèle à la base de Whisper, l'outil que nous allons utiliser dans la suite de ce mémoire.

Nous avons ensuite affiné notre sujet de recherche en clôturant notre état de l'art en parlant plus précisément des enjeux et recherches sur la transcription de la parole des locuteurices non-natif·ve·s.

Nous sommes à présent mieux préparée pour la suite de nos recherches et prête à continuer ce mémoire avec des bases plus solides.

Dans la partie suivante, nous allons présenter la méthodologie que nous avons observée pour la partie pratique de ce mémoire.

Troisième partie

Méthodologie

Dans cette partie, nous allons tout d'abord présenter les données sur lesquelles nous avons travaillé et décrire plus en profondeur le corpus de Noreillie (2019). Nous parlerons des données orales, mais aussi des deux formes de transcriptions présentes dans le corpus : le format texte et le format chat.

Nous parlerons également de la diarisation de locuteurice et des limites auxquelles nous nous sommes heurtée lors de nos tests pour réaliser la reconnaissance de locuteurice sur nos données.

Ensuite, nous discuterons de la transcription automatique et du modèle Whisper, nous expliquerons pourquoi notre choix s'est porté sur ce modèle en particulier.

Après cela, nous décrirons les deux outils de transcription et de benchmark que nous avons réalisés pour ce mémoire en détaillant leur utilisation.

Nous terminerons en présentant des recherches effectuées en amont de ce mémoire sur un corpus test.

3.1 Description des données

Les données utilisées dans ce mémoire proviennent du corpus rassemblé par Ann-Sophie Noreillie dans le cadre de sa thèse de doctorat à la KU Leuven (Noreillie, 2019). Il comprend des enregistrements audio (voir la section 3.1.1) et des transcriptions manuelles (voir la section 3.1.2).

Le corpus a été créé dans le but d'étudier le lien entre les connaissances en vocabulaire et le niveau d'apprentissage dans les productions orales de

locuteurices allophones. Cette question étant plus souvent étudiée pour les compétences de l'écrit, l'écoute et la lecture et surtout pour l'anglais, Noreillie (2019) a décidé d'approfondir les recherches pour la production orale en français.

Les enregistrements sont des dialogues entre un·e étudiant·e et un·e examinateurice. Il y a deux locuteurices, d'une part Noreillie, qui est l'intervieweuse, et d'autre part l'étudiant·e interviewé·e.

Le corpus est divisé en deux sous-groupes d'étudiant·e·s. Tout d'abord, des locuteurices néerlandophones de Belgique en dernière année de secondaire qui devaient avoir atteint le niveau B1 du CEFR en français. Ce public a été choisi, car le niveau B1 et le public d'élèves de secondaire sont peu explorés dans la recherche (Noreillie, 2019).

Le second groupe, est composé d'étudiant·e·s francophones de Belgique, également en dernière année de secondaire. Le but de cette deuxième série d'enregistrements était de servir de corpus de contrôle pour les analyses de la thèse.

Deux sujets sont abordés lors des interviews. Le premier dialogue est un entretien d'embauche et le second un rendez-vous chez le médecin. Tou·te·s les participant·e·s, allophones et francophones, ont été enregistré·e·s pour les deux types d'interviews. Ces sujets ont été choisis car ils correspondent au niveau B1 attendu par le CEFR (Noreillie, 2019).

3.1.1 Les données orales brutes

Le corpus de Noreillie (2019) comprend 146 enregistrements. Il y a 51 locuteurices néerlandophones, ce qui fait un total de 102 enregistrements d'étudiant·e·s non-natif·ve·s et 22 étudiant·e·s francophones, ce qui fait un total de 44 enregistrements de locuteurices natif·ve·s.

Le corpus compte 420 minutes d'enregistrements pour tou·te·s les locuteurices confondu·e·s dont 307 minutes pour les locuteurices néerlandophones et 113 minutes pour les locuteurices francophones.

Les différentes informations sur ce corpus sont reprises dans le tableau 2 ci-dessous.

	Loc. néerlandophones		**Loc. francophones**	
	enregistrements	**minutes**	**enregistrements**	**minutes**
Entretien	51	175	22	68
Médecin	51	132	22	45
totaux	102	307	44	113

TABLE 2 – Nombre d'enregistrements par tâche et leurs durées totales en minutes

3.1.2 Les transcriptions

Il y a deux types de transcriptions différentes. Tout d'abord, la transcription au format chat, qui est une transcription qui reprend les propos des deux locuteurices en indiquant quel·le intervenant·e parle. Ce format contient de nombreuses métadonnées utiles et est construit de la manière suivante :

> Trois types d'information sont contenus dans une transcription :
> - des informations à caractère général qui se rapportent à tout l'enregistrement, ce sont principalement les en-têtes (lignes commençant par @) ;
> - des transcriptions réalisées énoncé par énoncé (ou par tours de parole) sur les lignes principales (lignes commençant par *) ;
> - des indications complémentaires se rapportant à un énoncé ou à un tour de parole précis, ce sont les lignes dépendantes (lignes commençant par %)[15].

15. La citation employée pour décrire le format chat vient d'un document en ligne, rédigé par Colette Noyau et Christophe Parisse, et disponible à l'adresse suivante via le site ortolang : https://ct3.ortolang.fr/transferts/pdf/conventions-transferts.pdf - page 2 du document.

Ci-dessous dans la figure 7, un extrait du corpus de Noreillie pour illustrer la construction d'une transcription au format chat.

```
@UTF8
@Begin
@Languages: fra
@Participants:  CEJ Participant, ASN Investigator
@ID:    fra|VALILEX|CEJ||female|MSA||Participant|||
@ID:    fra|VALILEX|ASN|||||Investigator|||
@Media: Cejo_E_CSM_MSA_ASN, audio
@Transcriber:   MDV
@Date:  19-MAR-2018
@Bg:    Entretien
*ASN:   bonjour Célia .
%mor:   co|bonjour n:prop|Célia .
*CEJ:   bonjour euh madame &=laughs .
%mor:   co|bonjour co|euh n|madame&f=madam .
*ASN:   Tu viens pour un job euh d' étudiant ?
%mor:   n:prop|Tu v|venir-PRES&12s prep|pour det:art|un&m&sg n|job&m co|euh
    adv|de n|étudiant&m ?
*CEJ:   oui euh j(e) viens m(e) présenter pour euh le job+d+été dans la
    boulangerie +/.
%mor:   co|oui=yes co|euh pro:subj|je v|venir-PRES&12s pro:obj|me
    v|présenter-INF prep|pour co|euh det:art|le&m&sg
    n|+n|job+prep|d+n|été&m prep|dans det:art|la&f&sg n|boulangerie&f
    +/.
```

FIGURE 7 – Extrait d'un chat venant du corpus de Noreillie (2019)

Cette figure reprend bien toutes les métadonnées susmentionnées et les codes du format chat avec les "*" qui indiquent les tours de parole ; les "@" qui sont des informations métatextuelles, comme la date ou les participant·e·s ; et les "%" qui indiquent des informations complémentaires, ici il s'agit de précisions d'ordre grammatical en dessous de chaque intervention des locuteurices.

L'autre modèle est la transcription au format texte dans laquelle seuls les propos des étudiant·e·s ont été transcrits. Il n'y a donc pas de spécifications pour les tours de parole et les informations grammaticales ne sont pas ajoutées.

Malgré que les deux types de transcriptions soient différentes, dans les deux des métadonnées ont été ajoutées pour indiquer les disfluences et les pauses, hésitations ou erreurs dans la parole des étudiant·e·s.

3.1.3 La reconnaissance des locuteurices

Au début de ce projet, nous pensions utiliser la transcription au format texte, car seules les productions des étudiant·e·s ont été manuellement transcrites dans ce format. Étant donné que ce mémoire porte sur les productions orales de locuteurices apprenant le français, nous trouvions cela plus intéressant de ne pas avoir les interventions de Noreillie dans nos données.

Pour utiliser de manière efficace ces transcriptions, il nous fallait donc créer un outil de reconnaissance de locuteurice afin de pouvoir supprimer les interventions de Noreillie également dans les transcriptions automatiques, et ce, afin que les deux types de transcriptions soient équivalentes et puissent être comparées et que nos analyses ne soient pas biaisées.

Nous avons donc fait des recherches sur la diarisation et cherché une manière de l'appliquer à nos fichiers audio.

Whisper ne propose pas de reconnaissance de locuteurice actuellement. Pour pouvoir effectuer cette reconnaissance et ne garder que les paroles des apprenant·e·s en enlevant celles de Noreillie, nous avons dû utiliser la librairie Pyannote qui est un outil de diarisation.

Pyannote permet de reconnaitre les différent·e·s intervenant·e·s et de segmenter le fichier audio en lui ajoutant des *timestamps* en fonction de l'intervenant·e.

Dans le fichier JSONL que nous obtenons lorsque nous utilisons notre outil de transcription[16], il y a la transcription entière, mais aussi la transcription

16. L'utilisation de cet outil est détaillé plus bas dans ce mémoire, voir 3.2.2.1.

découpée en segments avec des *timestamps* également. Nous avons donc décidé de comparer les *timestamps* de notre transcription automatique avec ceux obtenus grâce à la diarisation. Le but était de parcourir les *timestamps* de notre transcription automatique et de réécrire dans un nouveau fichier les phrases correspondant uniquement aux *timestamps* de l'étudiant·e.

Malheureusement, nos essais n'ont pas abouti à un résultat concluant. En effet, les *timestamps* n'étaient pas exactement alignés, car les *timestamps* obtenus par Whisper et ceux de Pyannote ne découpaient pas l'audio aux mêmes endroits. Nous avons analysé en profondeur les différents *timestamps*, puis réalisé un script qui nous permettait de garder les *timestamps* intéressants en comparant les débuts et fins des *timestamps* de la diarisation à ceux de Whisper. Étant donné que les *timestamps* pouvaient être plus longs, plus courts ou commencer en décalé, nous avions calculé un delta qui devait nous permettre d'inclure toutes les données nécessaires dans nos transcriptions finales.

Après plusieurs essais, des modifications du delta et des techniques utilisées, ainsi qu'une vérification manuelle systématique sur un texte test, nous avons dû abandonner l'idée de la diarisation car elle ne fonctionnait que partiellement, qu'importe nos modifications. Soit nous n'avions que les propos de l'étudiant·e sans ceux de Noreillie, mais dans ce cas-là, certaines interventions de l'étudiant·e (les plus courtes, comme des « oui ») n'étaient pas reprises dans le script, soit, lorsque le delta était trop grand, certaines interventions de Noreillie n'étaient pas bien supprimées du fichier de sortie.

C'est donc à regret que nous avons décidé de ne pas utiliser la diarisation pour ne garder que les propos des étudiant·e·s, car le système n'est pas encore assez optimal, en tout cas pas pour nos données. La diarisation n'étant pas l'objet de ce mémoire, nous ne pouvions pas nous permettre d'y consacrer

plus de temps.

3.1.4 Extraction des données à partir du chat

Les fichiers au format texte n'étant pas utilisables, nous avons travaillé à partir des fichiers chat, qui sont moins faciles à manipuler car ils contiennent de nombreuses informations dont nous n'avons pas besoin pour ce travail, mais les interventions des deux locuteurices sont bien reprises dans leur intégralité. Nous conservons donc les interventions de Noreillie, même si nous sommes consciente que ses propos amènent un biais dans nos analyses.

Pour utiliser les transcriptions au format chat, nous avons dû nettoyer les fichiers. En effet, ces fichiers sont justement utiles, car ils contiennent de nombreuses métadonnées qui peuvent être de précieux outils pour une analyse linguistique, mais nous n'en avions pas besoin pour ce mémoire. Nous avons donc tout d'abord extrait uniquement les lignes qui commençaient par un astérisque en lisant le fichier ligne par ligne et ensuite nous avons appliqué différentes expressions régulières pour remplacer les éléments superflus du texte. Enfin nous avons réécrit les lignes qui nous intéressaient dans des fichiers au format texte afin de les utiliser dans notre outil de benchmark (qui sera détaillé plus loin, voir point 3.2.2.2).

Après avoir nettoyé les transcriptions *gold* pour les utiliser dans nos analyses, nous nous sommes penchée sur les transcriptions automatiques et dans la partie suivante nous allons expliquer comment nous les avons obtenues.

3.2 Les transcriptions automatiques

Dans cette section, nous allons tout d'abord expliquer pourquoi notre choix s'est porté sur Whisper, le modèle de transcription d'OpenAI. Ensuite, nous

présenterons les scripts que nous avons utilisés pour réaliser nos transcriptions. Enfin, nous illustrerons leurs utilisations sur un corpus test sur lequel nous avons travaillé avant d'utiliser celui de Noreillie.

3.2.1 Whisper

Avant de décider de nous concentrer sur Whisper, nous souhaitions tester plusieurs modèles de différents acteurs de l'Intelligence Artificielle (Microsoft, OpenAI, Meta, Apple, etc.). Après avoir fait des recherches et en avoir testé certains via les API en ligne, nous nous sommes parfois heurtée à un accès restreint (Google, AssemblyAI) voire à pas d'accès du tout (Amazon, Apple). Nous avons donc commencé à créer notre script autour du modèle Whisper d'OpenAI avant d'adapter le script pour qu'il soit également utilisable avec les modèles de Facebook qui utilisent Wave2Vec[17].

Après plusieurs tests, nous avons décidé de nous concentrer sur le modèle Whisper, car c'est un modèle performant et assez populaire aujourd'hui.

Comme nous l'avons expliqué dans l'état de l'art, il existe 6 versions différentes du modèle Whisper : tiny, base, small, medium, large et largev2. Nous avons testé toutes ces versions au moyen d'un script Python utilisant les pipelines et les *Transformers* et nous permettant d'accéder directement aux modèles via la plateforme HuggingFace.

Les transcriptions des différentes versions de Whisper donnant déjà des résultats différents, nous avons trouvé intéressant de nous pencher en priorité sur ce modèle afin d'en obtenir une connaissance plus approfondie et de pouvoir réaliser une analyse.

17. Pour plus d'informations sur Wave2Vec : https://huggingface.co/docs/transformers/model_doc/wav2vec2

3.2.2 Créations des scripts pour la transcription et l'analyse

3.2.2.1 Création d'un outil de transcription

Après avoir pris connaissance des outils existants et nous être documentée, nous avons commencé la partie pratique du mémoire en créant nos propres outils de transcription et de benchmark.

Pour créer l'outil pour la transcription, nous sommes passée par plusieurs étapes. Notre but était de faire un outil utilisable pour le mémoire, mais également de bien le documenter afin de pouvoir le partager avec d'autres personnes qui pourraient être intéressées.

Tout d'abord, afin d'enregistrer nos différentes transcriptions et de pouvoir facilement faire appel aux modèles que nous avons trouvés sur HuggingFace, nous avons réalisé un module *speech_recognition* contenant une classe *SpeechRecognizer* qui nous permet d'appeler les différents modèles en utilisant les pipelines et nous utilisons ces modèles dans un script que nous avons appelé *transcribe.py*.

Le script *transbribe.py* prend en argument un fichier audio et le modèle que l'on souhaite utiliser et enregistre la transcription dans un fichier JSONL. Le fichier JSONL de sortie a l'avantage de fonctionner comme un dictionnaire Python, il est donc facile d'y enregistrer toutes les informations dont nous avons besoin en les liant à des clés de dictionnaire, comme par exemple le temps que prend la transcription, le texte transcrit, mais aussi le nom du modèle utilisé et le nom de l'enregistrement.

La transcription automatique, surtout pour les modèles Whisper large et largev2, demandant un temps d'exécution important, nous avons eu accès, grâce à Anaïs Tack, à un ordinateur de la KU Leuven muni d'un GPU

Nvidia RTX 3090 pour que l'exécution soit plus rapide que sur notre machine personnelle. Anaïs Tack a réalisé un script bash qui permet de lancer le script de transcription sur tous les fichiers audio et de récupérer toutes les transcriptions dans un fichier JSONL par modèle.

3.2.2.2 Les métriques utilisées et l'outil de benchmark

Dans ce mémoire, nous souhaitions comparer les différentes versions de Whisper et leur efficacité pour la transcription automatique. Pour ce faire, il nous fallait un outil qui compare les différentes transcriptions de manière quantifiable et lisible. Nous avons donc créé un outil de benchmark qui prend en argument un ou plusieurs fichiers JSONL contenant les transcriptions automatiques et un fichier texte avec la transcription *gold*. En *output*, pour chaque audio, nous obtenons un fichier CSV qui reprend les références de l'enregistrement, le temps que prend la transcription pour s'exécuter ainsi que les résultats des différentes métriques que nous allons utiliser pour nos analyses.

Nous avons parlé du WER dans la partie théorique de ce mémoire en soulignant que c'était un outil précieux pour l'analyse de la transcription automatique bien qu'il soit imparfait. Nous allons donc calculer le WER pour nos transcriptions afin de pouvoir mieux les analyser, mais pas uniquement.

Pour obtenir les résultats du benchmark, nous utilisons la librairie JiWER. Cette librairie permet de calculer, en plus du WER, différentes métriques dont nous avons parlé dans notre état de l'art : le CER, le MER, le WIL et le WIP. Nous allons donc illustrer nos recherches également au moyen de ces métriques.

En plus de ces métriques, la librairie JiWER possède aussi une fonctionnalité qui permet de superposer les transcriptions *gold* et automatiques afin

de voir où se situent les erreurs. Cette fonctionnalité nous a été très utile à différents moments de nos recherches, elle nous a permis par exemple de vérifier après l'utilisation de l'outil de benchmark que les bonnes transcriptions *gold* avaient été associées aux transcriptions automatiques correspondantes lors du lancement général du script. Ce qui n'a pas été le cas du premier coup, car deux transcriptions *gold* étaient manquantes. Cet outil nous a donc évité d'avoir des données complètement erronées. Nous l'avons également utilisé à d'autres moments durant nos recherches, nous y reviendrons plus tard dans ce mémoire.

Après l'utilisation de l'outil de benchmark sur nos transcriptions, toutes les métriques sont reprises dans un CSV correspondant chaque fois à un fichier audio. Le tableau contient une ligne par modèle et les scores pour chaque métrique. Un exemple de ces tableaux est repris dans le tableau 3 qui est un tableau réalisé à partir des données du corpus test dont il est question dans la section 3.2.3.

Après avoir obtenu les tableaux pour chaque fichier audio, nous avons fait un tableau récapitulatif par modèle. Ces tableaux contiennent les résultats des métriques de chaque transcription ainsi que la moyenne de chaque métrique. Ils se trouvent en annexe et les résultats calculés seront utilisés dans la partie Analyses de ce mémoire (voir partie 4).

3.2.3 Le corpus de test

Avant d'arrêter notre choix sur le corpus de Noreillie (2019), nous avons travaillé sur un corpus test qui n'est pas celui que nous allons utiliser pour ce mémoire, mais il nous semblait pertinent de le mentionner afin d'expliquer l'évolution de nos recherches.

Étant donné que nous hésitions à travailler sur les troubles du langage

ou sur le français parlé par les personnes non-natives, nous avons cherché un corpus que nous pourrions utiliser pour effectuer nos premières observations.

Pour ce faire, nous avons cherché sur la plateforme Ortolang, qui recense de nombreux corpus, dont certains oraux, un ou plusieurs corpus qui pourraient être utilisés dans le cadre de notre mémoire. Nous avons contacté le professeur responsable du corpus « Paroles Disfluentes » qui est un corpus de personnes atteintes de bégaiement. Malheureusement, à ce jour, nos mails sont restés sans réponse et nous n'avons pas eu accès à ce corpus.

Nous nous sommes donc tournée vers le corpus ALIPE (Damien Chabanal, 2017) qui est en libre accès sur la plateforme. ALIPE est un corpus contenant des enregistrements d'interactions entre des enfants en bas âge et leurs parents. Il y a plusieurs enregistrements de trois enfants différents. Le but de ce corpus est d'étudier l'évolution des interactions entre les enfants et les parents lors de l'apprentissage de la parole.

Ce corpus nous semblait intéressant, bien qu'il ne soit pas directement lié à notre problématique, car la parole des enfants en bas âge est également difficile à transcrire. En effet, certains sons sont mal articulés, les enfants font plus d'hésitations, les parents et les enfants parlent dans les mêmes enregistrements, il y a des bruits parasites, car les enfants sont enregistrés lorsqu'ils effectuent des activités ou lors de repas, etc. L'intérêt de ce corpus est également que les documents audio et les transcriptions sont disponibles en libre accès, ce qui nous a permis de nous mettre directement au travail.

En observant les transcriptions du corpus ALIPE, nous avons constaté qu'il y avait de fortes différences d'efficacité entre les différents niveaux du modèle, ce qui peut s'expliquer par le nombre de paramètres. Le modèle tiny qui a 4 *layers* et 39 M paramètres produit une transcription beaucoup plus rapidement, mais moins correcte, que celle du modèle large qui a 32 *layers* et

1550 M paramètres.

Plusieurs différences étaient flagrantes lors de l'utilisation des modèles. Nous avons tout d'abord constaté que le temps de transcription était très différent d'une version à l'autre. Nous avons donc décidé d'utiliser le package Time dans notre script pour avoir les données exactes de ces différences et les mettre dans notre tableau récapitulatif final.

En plus du temps qui varie fortement, dans les versions tiny, small et base, certains mots sont répétés de manière aléatoire et des pans de phrases complets sont supprimés. Nous avons pu constater cela grâce à la superposition des transcriptions *gold* et automatiques.

Jusqu'à présent, la meilleure transcription est la version large.

Dans le tableau 3 ci-dessous, nous avons compilé les résultats que nous avons obtenus pour ces premières observations sur les extraits du corpus ALIPE. Ce tableau montre les prémices de nos analyses de Whisper.

Model	**Time (sec)**	**WER**	**MER**	**WIL**	**WIP**	**CER**
Tiny	12	1.106	0.916	0.991	0.008	0.789
Base	39	3.245	0.942	0.988	0.011	2.349
Small	54	1.021	0.761	0.922	0.077	0.663
Medium	166	1.289	0.762	0.898	0.101	0.888
Large	260	0.814	0.686	0.879	0.120	0.462
Largev2	326	1.084	0.661	0.804	0.195	1.005

TABLE 3 – Résultats obtenus via l'outil de benchmark pour un des enregistrements du corpus ALIPE

Dans la prochaine partie, nous analyserons les données du corpus de Noreillie et nous verrons si la version large reste la meilleure également pour ces données.

Quatrième partie

Analyses

Dans cette partie, nous allons essayer de répondre à trois questions de recherche en analysant les résultats obtenus grâce à notre outil de benchmark.

La première question est de voir s'il y a une différence significative entre les différentes versions de Whisper. Pour y répondre, nous allons analyser les moyennes des résultats des métriques obtenues pour tou·te·s les locuteurices confondu·e·s (voir section 4.1).

La seconde question est de déterminer si la transcription est significativement meilleure pour les locuteurices natif·ve·s. Pour cette question, nous allons donc comparer les résultats des non-natif·ve·s et ceux des natif·ve·s (voir section 4.2).

Et enfin, l'objectif de la troisième question est de trouver quelle version de Whisper offre les meilleures performances pour la transcription automatique de la parole non standard (voir section 4.3).

4.1 Efficacité des différents modèles sur le corpus entier

4.1.1 Observations générales

Le tableau 4 ci-dessous reprend les moyennes et écarts-types de chaque métrique d'évaluation pour chaque modèle, en tenant compte de l'ensemble des transcriptions du corpus. On remarque qu'au plus le modèle a été entrainé sur un nombre de couches important, au plus les résultats s'améliorent.

L'amélioration est systématique, mais pas constante. En effet, on remarque qu'il y a une plus grande différence d'efficacité entre les modèles tiny et base, on passe de 89% d'erreurs à 71% d'erreurs, qu'entre les modèles medium et large où la différence n'est plus que de 1% d'amélioration.

	WER		CER		MER		WIL		WIP	
	moy	EC	moy	EC	moy	EC	moy	EC	moy	EC
T	0.890	0.41	0.625	0.44	0.724	0.10	0.883	0.06	0.116	0.06
B	0.711	0.26	0.450	0.24	0.635	0.09	0.812	0.07	0.188	0.07
S	0.611	0.13	0.360	0.12	0.572	0.07	0.746	0.07	0.254	0.07
M	0.589	0.10	0.335	0.11	0.560	0.06	0.736	0.06	0.264	0.06
L	0.579	0.10	0.332	0.11	0.551	0.06	0.723	0.06	0.276	0.06
L2	0.572	0.10	0.323	0.11	0.541	0.06	0.714	0.07	0.286	0.07

TABLE 4 – Moyennes et écarts-type des résultats des modèles pour tou·te·s les locuteurices
(T = Tiny, B = Base, S = Small, M = Medium, L = Large, L2 = Largev2)

On observe que cette amélioration est visible pour toutes les métriques, cependant, c'est le WER et le CER qui ont une évolution plus importante. L'évolution du WIL et du WIP est plus faible.

Cette différence peut s'expliquer par le type d'erreurs présentes dans les transcriptions. En effet, pour le WER, toutes les erreurs ont le même poids, elles sont additionnées, et ensuite leur nombre est divisé par le nombre total de mots de la transcription *gold*. Alors que le WIL effectue une pondération, car il donne plus de poids à des erreurs qui selon lui ont un impact plus grand sur l'information véhiculée par le texte. Le WIL élevé nous indique donc que des informations importantes pour la transcription ont été perdues.

En analysant ce tableau, nous pouvons dire qu'au plus le modèle a été entrainé sur un grand nombre de couches, au plus les transcriptions sont

efficaces. L'amélioration diminue cependant au fur et à mesure que le modèle augmente, ce qui laisse un doute quant à l'utilité d'augmenter le modèle large. L'amélioration ne semble en effet pas vraiment suffisamment importante pour que ce modèle soit encore augmenté en une nouvelle version.

4.1.2 Un WER trop élevé ?

Lors de nos observations, nous avons également noté que le WER était plutôt élevé par rapport aux résultats observés dans d'autres recherches traitant du modèle Whisper. Les résultats obtenus par Radford et al. (2023) pour le français sont bien meilleurs que ceux que nous avons obtenus pour notre corpus.

Les figures 8 et 9 sont tirées de cet article et l'on peut constater que, pour le français, les résultats de WER sont compris entre 49,7 % (tiny) et 13,9 % (largev2) sur le corpus *Common Voice 9* (Ardila et al., 2020) et ils sont encore plus bas (32,9 % à 11,4 %) pour le corpus *Vox Populi* (Wang et al., 2021). Dans les deux cas, le WER le plus élevé, et donc le moins bon, reste plus bas que le meilleur WER obtenu sur nos données (57,2 % pour le modèle largev2 sur les données de tou·te·s les locuteurices confondu·e·s).

Après avoir constaté cela, nous avons décidé de voir s'il y avait une explication évidente, si nous avions fait une erreur quelque part qui pourrait expliquer ces résultats trop élevés.

Le module JiWER possédant une fonction qui permet d'imprimer l'alignement du fichier *gold* et celui de la transcription automatique, nous avons observé plusieurs de ces alignements et nous avons constaté que lorsqu'une ponctuation n'était pas alignée (par exemple, lorsqu'elle est manquante dans le fichier *gold* et ajoutée automatiquement par Whisper), cela comptait comme une erreur comptabilisée dans le WER et les autres métriques. Nos fichiers

Model	Arabic	Bulgarian	Bengali	Catalan	Czech	Welsh	Danish	German	Greek	English	Spanish	Estonian	Persian
Whisper tiny	90.9	79.3	104.1	51.0	79.7	101.8	77.2	34.5	61.9	28.8	30.3	102.1	120.3
Whisper base	84.4	68.1	103.7	39.9	63.1	93.8	57.5	24.5	51.5	21.9	19.6	88.1	99.0
Whisper small	66.4	44.8	118.6	23.8	34.1	65.4	32.1	13.0	31.7	14.5	10.3	67.2	71.9
Whisper medium	60.3	26.7	124.7	16.4	18.8	43.6	19.3	8.5	20.0	11.2	6.9	45.6	49.9
Whisper large	56.0	24.1	106.0	15.3	17.1	40.3	18.3	7.7	18.3	10.1	6.4	41.4	44.8
Whisper large-v2	53.8	19.9	103.4	14.1	13.5	34.2	14.4	6.4	16.0	9.4	5.6	35.1	39.4

Model	Finnish	French	Hindi	Hungarian	Indonesian	Italian	Japanese	Lithuanian	Latvian	Malayalam	Mongolian	Dutch	Polish
Whisper tiny	68.5	49.7	108.3	87.0	49.6	44.5	36.1	103.5	87.8	102.7	123.0	43.6	45.3
Whisper base	52.9	37.3	106.5	71.9	36.1	30.5	24.2	91.3	78.0	122.9	137.0	29.5	32.8
Whisper small	30.5	22.7	43.6	44.4	18.4	16.0	14.0	72.8	54.6	104.8	225.8	14.2	16.9
Whisper medium	18.8	16.0	31.5	26.9	11.6	9.4	10.5	49.4	37.2	137.8	113.4	8.0	10.1
Whisper large	17.0	14.7	25.0	23.5	10.6	8.1	9.4	43.9	34.8	107.1	117.4	7.1	9.0
Whisper large-v2	14.4	13.9	21.9	19.7	8.5	7.1	9.1	35.2	25.5	103.2	128.4	5.8	7.6

Model	Portuguese	Romanian	Russian	Slovak	Slovenian	Serbian	Swedish	Tamil	Thai	Turkish	Urdu	Vietnamese	Chinese
Whisper tiny	35.2	68.2	40.6	104.0	82.0	106.1	58.2	105.7	55.9	53.6	74.7	69.3	52.4
Whisper base	23.7	55.9	28.8	87.2	70.3	103.0	42.4	49.5	32.1	38.6	58.6	51.6	44.9
Whisper small	12.5	33.2	15.0	60.4	45.5	101.3	22.1	28.7	18.1	23.7	39.1	33.3	29.4
Whisper medium	8.1	21.5	9.3	42.0	29.8	85.6	13.7	19.6	10.5	17.7	29.9	24.4	23.2
Whisper large	7.1	19.8	8.2	37.9	25.1	87.4	12.4	17.6	8.8	16.6	28.1	19.9	29.1
Whisper large-v2	6.3	15.8	7.1	31.9	20.6	70.5	10.6	16.1	8.0	14.5	24.2	18.2	26.8

FIGURE 8 – Résultats du WER (%) sur le corpus *Common Voice 9* (pris de Radford et al., 2023, Table 11, p. 23)

Model	Czech	German	English	en_accented	Spanish	Estonian	Finnish	French	Croatian	Hungarian	Italian	Lithuanian	Dutch	Polish	Romanian	Slovak	Slovenian
Whisper tiny	73.5	27.4	11.6	18.8	19.7	99.2	54.1	32.9	72.4	74.5	40.5	93.1	41.9	31.4	65.9	78.7	81.9
Whisper base	54.7	20.6	9.5	17.5	14.4	83.0	39.7	24.9	53.6	52.6	30.8	82.1	29.4	22.1	49.3	63.7	70.5
Whisper small	28.8	14.8	8.2	19.2	11.1	59.2	24.9	15.7	33.7	31.3	22.9	60.1	18.8	13.3	28.6	37.3	50.8
Whisper medium	18.4	12.4	7.6	19.1	9.6	38.2	16.6	12.2	23.9	19.3	19.7	39.3	14.9	10.1	18.4	23.0	36.3
Whisper large	15.9	11.9	7.2	20.8	8.8	33.3	15.5	11.0	19.0	16.8	18.4	35.0	14.0	9.0	17.0	19.1	31.3
Whisper large-v2	12.6	11.2	7.0	18.6	8.2	28.7	12.4	11.4	16.1	13.8	19.0	33.2	12.9	7.8	14.4	15.4	27.9

FIGURE 9 – Résultats du WER (%) sur le corpus *Vox Populi* (pris de Radford et al., 2023, Table 12, p. 23)

gold étant des transcriptions au format chat desquelles nous avons enlevé les indications métatextuelles, la ponctuation est quasi absente de ces transcriptions, car elle n'a pas été ajoutée dans le format chat lors de la transcription

manuelle. Cela est donc probablement une des raisons qui fait que nos résultats sont élevés.

Après avoir regardé plusieurs alignements, nous en sommes arrivée à une première conclusion. Whisper semble produire des transcriptions « trop correctes ». En effet, Whisper n'ajoute pas uniquement la ponctuation, il ajoute également les majuscules (qui n'étaient pas non plus présentes dans nos transcriptions *gold*). Le modèle semble aussi supprimer certaines disfluences, par exemple, les « euh » d'hésitation des étudiant·e·s ne sont pas retranscrits.

De plus, le modèle a tendance à faire des corrections. Par exemple, « de le argent » va devenir « de l'argent » (enregistrement BaCl_E_CSM_FL_ASN, NS), « je préférerais » devient « je préfère » (enregistrement BaCl_E_CSM_FL_ASN, NS), « y en a » devient « il y en a » (enregistrement GuVi_M_HHC_WW_ASN, NNS).

Ces exemples venant à la fois de productions de locuteurices natif·ve·s et non-natif·ve·s, on voit déjà que le modèle surcorrige les productions des néerlandophones mais également celles des francophones. La ponctuation étant absente des transcriptions *gold* pour les deux sous-corpus, son ajout par le modèle influence de manière égale le WER des productions des locuteurices natif·ve·s et non-natif·ve·s.

Aussi, nous avons constaté que les dates ont été retranscrites en lettres dans la transcription *gold*, alors que Whisper les transcrit en chiffres : « vingt-six octobre deux mille » devient « 26 octobre 2000 » (enregistrement BaCl_E_CSM_FL_ASN, NS).

C'est ce genre de différences qui expliquent sans doute que notre WER ne soit pas plus bas.

Cela explique également que le MER soit légèrement meilleur que le WER, bien qu'il soit quand même assez élevé, car cette métrique pénalise

plus lourdement les erreurs de substitutions que les erreurs d'insertion ou de suppression, or les « euh » qui disparaissent sont considérés comme des erreurs de suppression et la ponctuation ajoutée comme une erreur de substitution, et c'est l'erreur la plus fréquente dans nos textes.

4.2 Différence d'efficacité des modèles pour les locuteurices francophones et néerlandophones

4.2.1 Observations générales

Dans le tableau 5, qui reprend les résultats pour les locuteurices francophones, et le tableau 6, qui reprend les résultats pour les locuteurices néerlandophones, nous remarquons la même évolution que pour le tableau général : la moyenne des métriques diminue au fur et à mesure que les modèles s'élargissent.

Nous notons cependant une exception, dans le tableau 5 qui reprend les résultats des locuteurices natif·ve·s. Le modèle largev2 donne de moins bons résultats que le modèle large. C'est le seul cas de figure de ce type parmi nos données. La différence est légère, un peu moins de 1% pour le WER, mais il nous semblait quand même pertinent de la souligner, car cela nous conforte dans l'idée que le modèle large n'a sans doute pas besoin d'être encore plus affiné qu'il ne l'est déjà, en tout cas pour la parole considérée comme standard.

Cette exception est d'autant plus notable qu'elle n'apparait pas dans les résultats des locuteurices non-natif·ve·s. En effet, chez les allophones, la version largev2 obtient un meilleur WER que la version large. Il y a 1,5% de

différence entre les deux résultats, respectivement 58,4% pour la version large et 56,9% pour la version largev2.

NS	WER		CER		MER		WIL		WIP	
	moy	EC	moy	EC	moy	EC	moy	EC	moy	EC
T	0.762	0.15	0.471	0.11	0.697	0.06	0.874	0.04	0.126	0.04
B	0.677	0.13	0.404	0.12	0.631	0.06	0.817	0.06	0.183	0.06
S	0.592	0.07	0.327	0.06	0.566	0.06	0.746	0.06	0.254	0.06
M	0.584	0.04	0.310	0.03	0.562	0.04	0.744	0.05	0.256	0.05
L	0.569	0.05	0.298	0.03	0.545	0.05	0.722	0.05	0.278	0.05
L2	0.578	0.04	0.303	0.03	0.552	0.04	0.732	0.05	0.268	0.05

TABLE 5 – Moyennes et écarts-types des résultats des modèles pour les locuteurices francophones
(T = Tiny, B = Base, S = Small, M = Medium, L = Large, L2 = Largev2)

NNS	WER		CER		MER		WIL		WIP	
	moy	EC	moy	EC	moy	EC	moy	EC	moy	EC
T	0.945	0.47	0.692	0.51	0.735	0.11	0.888	0.07	0.112	0.07
B	0.725	0.30	0.470	0.28	0.637	0.10	0.809	0.07	0.190	0.07
S	0.619	0.15	0.374	0.13	0.575	0.07	0.746	0.07	0.254	0.07
M	0.591	0.11	0.346	0.12	0.559	0.07	0.732	0.06	0.268	0.06
L	0.584	0.11	0.347	0.13	0.554	0.06	0.725	0.06	0.276	0.06
L2	0.569	0.12	0.332	0.12	0.537	0.07	0.707	0.07	0.293	0.07

TABLE 6 – Moyennes et écarts-types des résultats des modèles pour les locuteurices néerlandophones
(T = Tiny, B = Base, S = Small, M = Medium, L = Large, L2 = Largev2)

Il est aussi intéressant de noter que si l'on compare les productions des locuteurices natif·ve·s aux résultats obtenus par Radford et al. (2023) repris

dans les figures 8 et 9, le WER obtenu pour notre corpus est à nouveau très élevé. En effet, notre meilleur résultat est 56,9% pour le modèle large sur les données des locuteurices natif·ve·s alors que pour cette même version, Radford et al. (2023) obtiennent un WER de 13,9% sur le corpus *Common Voice 9* et 11,4% sur le corpus *Vox Populi*.

Le modèle tiny obtient donc de meilleurs résultats (49,7% sur *Common Voice 9* et 32,9% sur *Vox Populi*) que les résultats que nous obtenons avec la version large sur les données de notre corpus produites par les locuteurices natif·ve·s.

Ces résultats sont intéressants car ils nous indiquent que le WER très élevé du tableau 4, qui reprend les résultats de tou·te·s les locuteurices, ne peut pas être imputé uniquement aux résultats des locuteurices non-natif·ve·s. Or, les résultats des non-natif·ve·s auraient pu avoir une influence plus importante sur la moyenne puisqu'il y a plus de données pour les locuteurices non-natif·ve·s, mais nous constatons donc que ce n'est pas le cas ici.

Ces réflexions nous amènent à une autre constatation. En observant ces tableaux, nous avons remarqué que les résultats pour les textes des locuteurices francophones et néerlandophones sont assez similaires pour les modèles Whisper medium, large et largev2, mais que les modèles Whisper small, base, tiny semblent moins efficaces pour les productions des locuteurices néerlandophones que pour celles des francophones. C'est surtout visible pour le modèle tiny qui a un WER de 0,762 chez les natif·ve·s et une WER de 0,945 chez les non-natif·ve·s, ce qui fait une différence de 18,3%, qui est la plus grosse différence observée dans ces tableaux.

Par conséquent, nous observons que les modèles entrainés avec moins de paramètres et un nombre de *layers* plus restreint fonctionnent raisonnablement bien pour les productions des locuteurices natif·ve·s mais sont nettement moins

efficaces pour les productions des locuteurices non-natif·ve·s.

Les modèles entrainés sur plus de couches semblent donc presque aussi efficaces pour des productions de locuteurices natif·ve·s que pour des productions de locuteurices non-natif·ve·s.

4.2.2 Exemples tirés du corpus

Pour illustrer les différences entre les transcriptions, nous avons choisi des phrases extraites de notre corpus dont nous allons superposer les transcriptions *gold* ainsi que celles de Whisper. Nous espérons illustrer au mieux les erreurs soulevées précédemment lors de nos analyses.

Nous avons choisi un extrait venant d'un·e locuteurice natif·ve (enregistrement de ReDe_M_CSM_FLA_ASN) dont les transcriptions sont reprises dans le tableau 7, et une phrase venant d'un·e locuteurice non-natif·ve (enregistrement de JaVb_E_HHC_LT_IFI) dont les transcriptions sont reprises dans le tableau 8.

4.2.2.1 Transcriptions du/de la locuteurice francophone

Versions	Transcriptions
gold	alors ça ce est vous choisissez qui vous voulez en fait mais **euh** je vous conseillerais en fait **euh** moi même parce que **euh beh** je suis je suis un un collègue on va dire assez agréable je pose pas trop de questions oui je ai un petit peu d'humour et à mon avis **ce est ce est** agréable de travailler avec moi
Tiny	Alors ça, vous vous choisissez qui vous voulez en fait, mais je vous pense que **je vais y aller** en fait moi même parce que je suis je suis un collègue on va dire assez agréable, je ne pose pas de **et on a vis ses graibles** de travail avec moi.
Base	Alors ça c'est vous choisissez qui vous voulez en fait mais **je vous pense c'était éré** en fait moi-même parce que je suis un collègue on va dire assez **gréable**, je ne pose pas de question, j'ai un petit peu **et on a vissé s'agréer** à peu de travail avec moi.
Small	Alors ça c'est vous choisissez qui vous voulez en fait mais je vous conseillerai en fait moi-même parce que je suis un collègue on va dire assez agréable, je ne pose pas trop de question, j'ai un petit peu d'humour et **du moure** et en mon avis c'est agréable de travailler avec moi.
Medium	Alors ça c'est**...** vous choisissez qui vous voulez en fait mais je vous conseillerais en fait moi-même parce que je suis un collègue on va dire assez agréable, je pose pas trop de questions, j'ai un petit peu d'humour et **en** mon avis c'est agréable de travailler avec moi. **d'humour et en mon avis c'est agréable de travailler avec moi.**
Large	Alors ça c'est**...** vous choisissez qui vous voulez en fait, mais je vous conseillerais en fait moi-même parce que je suis un collègue on va dire assez agréable, je pose pas trop de questions, j'ai un petit peu d'humour et à mon avis c'est agréable de travailler avec moi. **humour et à mon avis c'est agréable de travailler avec moi.**
Largev2	Alors ça c'est**...** vous choisissez qui vous voulez en fait, mais je vous conseillerais en fait moi-même. Parce que je suis un collègue on va dire assez agréable, je pose pas trop de questions, j'ai un petit peu d'humour, et à mon avis c'est agréable de travailler avec moi. **d'humour et à mon avis c'est agréable de travailler avec moi.**

TABLE 7 – Comparaison des transcriptions obtenues pour le/la locuteurice natif·ve. Extrait venant des transcriptions de l'enregistrement ReDe_M_CSM_FLA_ASN

Nous avons mis en évidence dans le tableau 7 certains éléments qui illustrent des problèmes que nous avons soulevés précédemment.

Tout d'abord, on remarque que même si le·la locuteurice est francophone, les modèles tiny, base et small produisent plus d'erreurs. Certaines sont grammaticalement et sémantiquement correctes, comme « je vais y aller » du modèle tiny : la phrase est correcte mais elle n'est pas la transcription attendue qui est « je vous conseillerais ».

D'autres erreurs par contre sont des mots inexistants dans la langue française : « on a vis ses graibles », « on a vis » n'est pas une forme grammaticale qui existe et « graible » n'est pas un mot de vocabulaire appartenant au français. Les termes « on a vis » ont été inventés par le modèle pour traduire « à mon avis » et « ses graibles » pour « c'est agréable ». On retrouve les mêmes genres de constructions pour le modèle base : « on a vissé s'agréer » en lieu et place de « ce est agréable », ainsi que pour le modèle small « du moure » au lieu de « d'humour ».

Ces observations pourraient aller dans le sens de celles de Ballier et al. (2023) qui expliquaient que la version tiny semblait plus impactée par la variation phonétique. Ceci pourrait expliquer pourquoi les petits modèles produisent des mots dont la sonorité pourrait se rapprocher de ceux de la transcription *gold*, mais qui n'existent pas en français.

La présence dans la transcription small à la fois du terme « d'humour » et du terme « du mour », qui semble transcrire la même partie de phrase nous permet de mettre en lumière un autre souci de Whisper également souligné par Ballier et al. (2023). Dans certains cas, le modèle opère une répétition d'un mot ou d'un morceau de phrase. En observant nos transcriptions, nous avions remarqué que cela arrivait surtout avec les petits modèles, mais dans l'exemple que nous avons choisi, cette répétition est présente dans toutes

les transcriptions sauf dans la version tiny et dans la version base. Même les modèles les plus performants dédoublent une phrase, et dans le cas des versions medium, large et largev2, un bout de phrase complet est répété. Nous avons constaté ce phénomène dans certaines transcriptions, elles expliquent aussi sans doute pourquoi nos métriques ont des résultats élevés. Actuellement nous n'avons pas d'explications pour comprendre ce phénomène. Ce point serait donc à améliorer dans de futures versions de Whisper.

À part ça, les versions medium (à l'exception de la locution « en mon avis »), large et largev2 ont produit des transcriptions grammaticalement et sémantiquement correctes. Mais ces transcriptions ne sont pas fidèles à la transcription *gold*. En effet, on remarque bien ici que le modèle surcorrige. Les « euh » ont été supprimés, l'allongement du « ce est » du début de l'extrait a été signifié par les trois petits points et la répétition « ce est ce est » a été supprimée. Les modèles ont donc produit une bonne traduction, les propos sont correctement transmis, mais elle n'est pas fidèle à ce que le·la locuteurice a produit comme énoncé.

4.2.2.2 Transcriptions du/de la locuteurice néerlandophone

Versions	Transcriptions
gold	bonjour bonjour je m'appelle euh Jaron et euh je suis au sixième année dans le collège de Waregem et euh oui euh je suis en latin langues et euh c'est mon première fois que je oui euh je veux euh fait un job d'étudiant dans une un supermarché ok donc je n'ai pas pas d'expérience
Tiny	Bonjour, je suis un m'appelle Yaron et je suis au **6e** année dans le collège de **Vargame** et oui, je suis **plataire l'anle** et c'est mon premier fois que je suis**...** C'est une **j'oblite d'union** dans une **supermachie**. Donc, je ne parle pas de expérience,
Base	Bonjour, je suis **m'enpelle** Yaron et je suis au **6e** année dans le collège de **Bargain**. J'ai fait un job **petit-d'udien** dans un supermaché, donc je **ne finis** pas d'expérience.
Small	Bonjour, je m'appelle Yaron et je suis au **6e** année dans le collège de **Vargham** et **je suis latin-langue**. **J'ai fait** un job d'étudiant dans un supermarché. Donc je n'ai pas d'expérience,
Medium	Bonjour, je m'appelle Yaron et je suis au **6ème** année dans le collège de **Barragem**. **euh...** Fait un job d'étudiant dans un supermarché. Donc je n'ai pas d'expérience.
Large	Bonjour, je m'appelle Yaron et je suis en **6e** année dans le collège de **Bargain**. **Je fais un job d'étudiant dans un supermarché donc je n'ai pas d'expérience**
Largev2	Bonjour, je m'appelle Yaron et je suis aux **sixième** année dans le collège de **Barguem**. **euh... Fais un job d'étudiant dans un supermarché.** Oui. Donc je n'ai pas d'expérience,

TABLE 8 – Comparaison des transcriptions obtenues pour le/la locuteurice non-natif·ve. Extrait venant des transcriptions de l'enregistrement JaVb_E_HHC_LT_IFI

Dans le tableau 8 qui reprend les transcriptions de l'extrait du/de la locuteurice néerlandophone, on remarque des événements similaires à ceux observés pour la transcription de l'étudiant·e francophone.

Les hésitations comme les « euh » ont presque tous été supprimées dans

les transcriptions, à part une occurrence dans les transcriptions des modèles medium et largev2.

Les versions tiny et base produisent des mots et des constructions qui n'existent pas en français comme par exemple « plataire l'anle » pour le modèle tiny et « un job petit-dudien » pour le modèle base.

La transcription du modèle small semble être dans ce cas précis plus correcte. C'est d'ailleurs la seule version où la phrase « je suis en latin langues » n'a pas complètement disparu.

Cependant, ce qui est flagrant dans ces transcriptions, surtout pour les modèles plus puissants, c'est que l'extrait a été fortement raccourci. La transcription qui semble la plus correcte au niveau de la grammaire est celle de la version large, mais la mention de l'option latin langues a été complètement supprimée et les paroles de l'étudiant·e ont été modifiées pour être plus grammaticalement correctes. Comme le « au sixième année » qui est devenu « en 6e année ».

Les modèles plus larges ont donc encore une fois tendance à privilégier le fait de produire une transcription qui ait du sens plutôt que de transcrire les propos exacts du/de la locuteurice. Le problème majeur dans l'exemple que nous avons choisi, est que cette transcription s'accompagne d'une suppression importante d'une partie des propos du/de la locuteurice.

Cet extrait nous permet aussi de souligner que le modèle Whisper a été entrainé pour la reconnaissance d'entités nommées (le fait de reconnaitre les noms propres pour bien les repérer dans le texte afin de les transcrire correctement) (Radford et al., 2023) mais en ce qui concerne « Waregem » cette fonctionnalité n'est pas encore au point.

4.3 Quel modèle pour transcrire la parole des locuteurices non-natif·ve·s ?

Notre dernière question de recherche portait sur le choix du meilleur modèle pour transcrire les productions des locuteurices allophones.

Selon nos observations, les versions medium, large et largev2 du modèle Whisper semblent bien adaptées pour transcrire les enregistrements de locuteurices allophones, puisque les résultats obtenus grâce aux métriques ne sont pas très différents des résultats obtenus pour les locuteurices francophones. Il vaut mieux privilégier ces versions que les versions entrainées sur moins de couches, car ces dernières donnent des résultats vraiment moins bons chez les locuteurices allophones que chez les locuteurices francophones.

Cependant, et l'exemple choisi pour l'alignement des transcriptions le montre bien, bien que les résultats des métriques soient corrects, il faut utiliser Whisper en sachant que c'est un bon modèle, mais pas forcément un modèle adapté pour souligner les erreurs de prononciation des locuteurices non-natif·ve·s, ni mêmes retranscrire leurs propos exacts dans leur entièreté.

En conclusion, Whisper est un outil puissant, mais il faut que les locuteurices allophones aient quand même une bonne connaissance du français pour être sûr·e·s de pouvoir vérifier que la transcription obtenue correspond à ce qu'ils ont dit réellement. En effet, comme nous l'avons vu dans notre exemple, même les versions large et largev2 suppriment parfois des bouts de phrases pour favoriser la cohérence grammaticale et sémantique, il est donc important à ce stade de vérifier les transcriptions.

Cinquième partie

Conclusion

5.1 Observations générales

Dans ce mémoire, nous avons tout d'abord présenté une évolution des modèles de la reconnaissance de la parole afin de mieux comprendre les différents enjeux liés à ce domaine de recherches. Nous avons ensuite décidé d'utiliser Whisper comme modèle pour nos transcriptions, car c'est un modèle basé sur l'architecture des *Transformers*, qui est la technologie la plus avancée en RAP, et parce que Whisper est le modèle le plus performant pour la reconnaissance vocale actuellement.

Nous avons analysé les performances des modèles Whisper sur notre corpus de dialogues d'étudiant·e·s non-natif·ve·s. Nous avons conclu que les versions entrainées sur plus de paramètres (medium, large et largev2) obtenaient de meilleurs résultats pour les transcriptions que les modèles plus petits (tiny, small et base). Nous avons également noté que les versions medium, large et largev2 étaient presque aussi efficaces pour transcrire la parole des natif·ve·s que celle des non-natif·ve·s.

5.2 Whisper est-il un outil adapté à notre tâche ?

Whisper, à première vue, en tout cas dans ses versions medium, large et largev2, semble être un bon outil de transcription. Les alignements que nous avons observés sur nos données sont satisfaisants, si nous faisons abstraction

de la ponctuation et des autres problèmes, mentionnés dans la partie 4, qui sont aussi des problèmes liés à notre corpus.

Cependant, dans le cas qui nous intéresse, puisque nous voulons transcrire la parole d'apprenant·e·s dans le cadre d'études pédagogiques, Whisper semble « trop » performant. En effet, enlever les disfluences peut poser problème si par exemple les professeur·e·s souhaitent transcrire automatiquement la parole de leurs apprenant·e·s pour souligner plus facilement leurs erreurs. Il serait intéressant, afin d'utiliser Whisper dans un cadre pédagogique, qu'il ne surcorrige pas les propos des étudiant·e·s dans ses transcriptions.

De plus, un autre risque de cette surcorrection serait que les propos soient déformés et ne correspondent finalement pas à ce que le ou la locuteurice a voulu dire. Nous avons eu ce cas de figure dans nos exemples, avec le modèle qui a produit une phrase grammaticalement et sémantiquement correcte, mais qui n'avait rien à voir avec les propos de l'étudiant·e.

Le modèle semble donc à première vue efficace pour traduire la parole des personnes ayant un accent en français, puisqu'il va jusqu'à corriger les fautes de français et produire une transcription intelligible, mais si le but de la transcription est également de transcrire les disfluences afin de s'approcher au maximum des propos de l'étudiant·e, alors Whisper n'est sans doute pas le modèle le plus approprié.

Cependant, dans un cadre moins pédagogique, Whisper est intéressant car il permet aux locuteurices allophones d'utiliser l'outil de reconnaissance vocale dans d'autres contextes. Par exemple, si le but de l'utilisation est d'écrire un message en français en le dictant directement à Whisper, alors le fait que le modèle corrige les erreurs et ne tienne pas compte des défauts de prononciation peut être très intéressant afin que le message produit soit intelligible et correct.

5.3 Modifications et pistes envisageables

Lors de la préparation de ce mémoire, nous avons envisagé de modifier les transcriptions *gold* pour y ajouter la ponctuation, et nous avons aussi pensé à uniformiser les deux types de transcriptions afin que tous les mots soient en minuscule et que la casse ne soit plus un problème. Cependant, nous nous sommes rendu compte que cela servirait uniquement à diminuer notre WER, mais que ça ne nous aiderait pas réellement à analyser l'efficacité des modèles sur nos données ni à différencier les versions de Whisper entre elles. Nous avons donc renoncé à effectuer ces manipulations, en tout cas dans le cadre du présent mémoire, mais cette piste reste ouverte si nous souhaitons encore travailler sur ces données dans le futur. Refaire une transcription manuelle sans données métatextuelles et en ajoutant de la ponctuation pourrait être bénéfique pour l'utilisation de ce corpus.

Aussi, il pourrait être intéressant de travailler sur un corpus de parole continue, afin de ne pas avoir la difficulté supplémentaire des deux intervenant·e·s. Une autre option pour résoudre ce problème serait de réaliser une diarisation de locuteurice.

Il y aurait encore beaucoup d'autres pistes à explorer, par exemple le fait de se baser sur des productions de plusieurs locuteurices néerlandophones lisant un même texte en français, afin de pointer les erreurs fréquentes de prononciation au moyen de Whisper comme l'ont fait Ballier et al. (2023) pour l'anglais. Mais la reconnaissance vocale est un sujet très vaste et un mémoire ne peut être le réceptacle d'une infinité de sujets.

Puisqu'il est temps de terminer ce mémoire, autant le faire en musique. Nous laissons le mot de la fin aux Beatles : *"Whisper words of wisdom, let it be"*.

Bibliographie

Références

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., and Weber, G. (2020). Common voice : A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Audrit, S. (16 December 2009). *Variation linguistique et signification sociale chez les jeunes Bruxelloises issus de limmigration maghrébine. Analyse socio-phonétique de trois variantes non standard.* PhD thesis, UCL - Université Catholique de Louvain.

Bada, I., Fohr, D., and Illina, I. (2020). Reconnaissance automatique de la parole : génération des prononciations non natives pour l'enrichissement du lexique (In this study we propose a method for lexicon adaptation in order to improve the automatic speech recognition (ASR) of non-native speakers). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 1 : Journées d'Études sur la Parole*, pages 27–35, Nancy, France. ATALA et AFCP.

Baker, J. K. (1975a). The DRAGON systemAn overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29. Conference Name : IEEE Transactions on Acoustics, Speech, and Signal Processing.

Baker, J. K. (1975b). *Stochastic Modeling as a means of automatic speech recognition*. PhD thesis, Mellon Institute of Sciences Carnegie-Mellon University.

Ballier, N., Méli, A., Amand, M., and Yunès, J.-B. (2023). Using Whisper LLM for Automatic Phonetic Diagnosis of L2 Speech : A Case Study with French Learners of English. In Mourad Abbas, A. A. F., editor, *6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, volume 6 of *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, Trento (Italy), Italy. Mourad Abbas, Abed Alhakim Freihat.

Baum, L. E. and Eagon, J. A. (1967). An inequality with application to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(6):360–363.

Baum, L. E. and Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563. Publisher : Institute of Mathematical Statistics.

Bañeras Roux, T., Rouvier, M., Wottawa, J., and Dufour, R. (2022). Mesures linguistiques automatiques pour l'évaluation des systèmes de Reconnaissance Automatique de la Parole (Automated linguistic measures for automatic speech recognition systems' evaluation). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 166–173, Avignon, France. ATALA.

Bouselmi, G. (2008). *Contributions à la reconnaissance automatique de la parole non-native*. Theses, Université Henri Poincaré - Nancy I.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-Based Models for Speech Recognition. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Damien Chabanal, Thierry Chanier, L. L. (2017). Alipe (acquisition de la liaison et interactions parents enfants). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Davis, K. H., Biddulph, R., and Balashek, S. (1952). Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24:637–642.

Deléglise, P. and Lailler, C. (2020). Quel type de systèmes utiliser pour la transcription automatique du français ? Les HMM font de la résistance (What system for the automatic transcription of French in audiovisual broadcasts ?). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 1 : Journées d'Études sur la Parole*, pages 154–162, Nancy, France. ATALA et AFCP.

Graves, A. (2012). Sequence transduction with recurrent neural networks. *ICML Representation Learning Worksop*.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA. Association for Computing Machinery.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.

Inceoglu, S., Chen, W.-H., and Lim, H. (2023). Assessment of l2 intelligibility : Comparing l1 listeners and automatic speech recognition. *ReCALL*, 35(1):89104.

Jelinek, F., Bahl, L., and Mercer, R. (1975). Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21(3):250–256. Conference Name : IEEE Transactions on Information Theory.

Jurafsky, D. and Martin, J. H. (2023). *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition - Third edition (draft).*

Li, W., Siniscalchi, S. M., Chen, N. F., and Lee, C.-H. (2016). Improving non-native mispronunciation detection and enriching diagnostic feedback with dnn-based speech attribute modeling. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6135–6139.

Morgan, N. and Bourlard, H. (1995). Neural networks for statistical recognition of continuous speech. *Proceedings of the IEEE*, 83(5):742–772.

Morris, A. C., Maier, V., and Green, P. D. (2004). From wer and ril to mer and wil : improved evaluation measures for connected speech recognition. In *Interspeech*, pages 2765–2768.

Noreillie, A.-S. (2019). *It's all about words. Three empirical studies into the role of lexical knowledge and use in French listening and speaking*

tasks. (Unpublished doctoral dissertation). PhD thesis, KU Leuven Campus Antwerpen.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., and Dupoux, E. (2021). VoxPopuli : A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Young, S. (1996). A review of large-vocabulary continuous-speech. *IEEE Signal Processing Magazine*, 13(5):45–57.

Crédit des musiques

The Beatles. 1970. *Let it be*, dans l'album éponyme "Let it be". Chanson écrite et composée par Paul McCartney et John Lennon.

Texte cité : *"Whisper words of wisdom, let it be"*. « Murmurant de sages paroles, qu'il en soit ainsi »[notre traduction].

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté de philosophie, arts et lettres
Place Blaise Pascal, 1 bte L3.03.11, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/fial**