Rapport final

Apprentissage de Modèles de Markov cachées et détection de mots clés

Yuyan QIAN

Table des matières

Introduction

« *La détection de mots-clés dans la parole continue est un processus qui implique de reconnaître et de localiser toutes les occurrences des mots d'une liste de mots-clés dans un flux de parole continu. L'objectif de ce projet individuel est de mettre en œuvre une telle détection en utilisant des modèles de Markov cachés phonétiques.* »

Dans le cadre des ateliers de ce module **Reconnaissance de la parole**, nous avons exploré diverses méthodologies pour résoudre une gamme de tâches. Parmi ces approches, nous avons utilisé la règle des K plus proches voisins pour la reconnaissance des formes ; l'Analyse en Composantes Principales (ACP) et l'Analyse Factorielle Discriminante (AED) pour les voyelles ; la comparaison dynamique ; ainsi que les Modèles de Markov Cachés, appliqués aux mots isolés et à la phonétique. Nous avons constaté que plus le champ d'application s'élargit et plus la reconnaissance devient flexible dans des situations réelles, plus le coût de construction du système augmente. Des facteurs tels que le nombre de locuteurs, le genre, les données d'apprentissage, et les unités à reconnaître peuvent influencer la détection vocale. Dans ce contexte, l'utilisation de HTK (Hidden Markov Model Toolkit) a été essentielle pour notre apprentissage de Modèles de Markov Cachés et la détection de mots clés. Ce toolkit, conçu spécifiquement pour la modélisation et le traitement de séquences basées sur des modèles de Markov cachés, nous a permis de développer et d'affiner nos modèles acoustiques de manière efficace. Cette approche a facilité une compréhension approfondie des mécanismes sous-jacents à la reconnaissance de la parole et a significativement amélioré la performance de notre système dans des scénarios d'utilisation réels.

Avant d'entrer dans le détail, un aperçu global est nécessaire pour comprendre l'enjeu de ce projet complexe. Pour illustrer le processus et présenter les étapes enchaînés, je vais me référer au schéma de la figure 1.

Ce rapport se concentrera principalement sur ce dernier aspect. Les étapes clés du projet, qui définissent également la structure de ce rapport, incluent : la segmentation des tours de parole en phonèmes, la construction des Modèles de Markov Cachés (HMM) pour chaque phonème, la construction des réseaux de décodage et l'ajustement des valeurs de récompense et de pénalité pour les mots-clés ciblés, et enfin, la reconnaissance de la parole suivie de l'analyse des résultats. Chaque étape prépare le terrain pour la suivante, nécessitant une révision continue pour améliorer les résultats de reconnaissance de la parole.

La première version de mes HMM est basée sur deux minuites de l'audio *synthRadio04*, qui ne contient qu'une seule fois du phonème ŋ (prononcé dans le mot *ligne*). Cependant, une seule apparition ne suffit pas d'entraîner un HMM à cause d'une manque d'observation. Pour cela, j'ai ajouté deux autres sources d'apprentissage *FR162*[1] et *synthRadio01*[2] pour enrichir les ressources des modèles. Les modèles de Markov cachées dans ce projet sont construites àl'aide des trois sources.

1. L'audio *FR162* annoté, segmenté et phonétisé par Anne FAURY. Elle a partagé les fichiers *.lab*, *.mfc* et *.wav* dont le nom commençant par *FR162_* avec moi.
2. L'audio *synthRadio01* annoté, segmenté et phonétisé par Marceau HERNANDEZ. Il a partagé les fichiers *.lab*, *.mfc* et *.wav* dont le nom commençant par *MH_* avec moi.

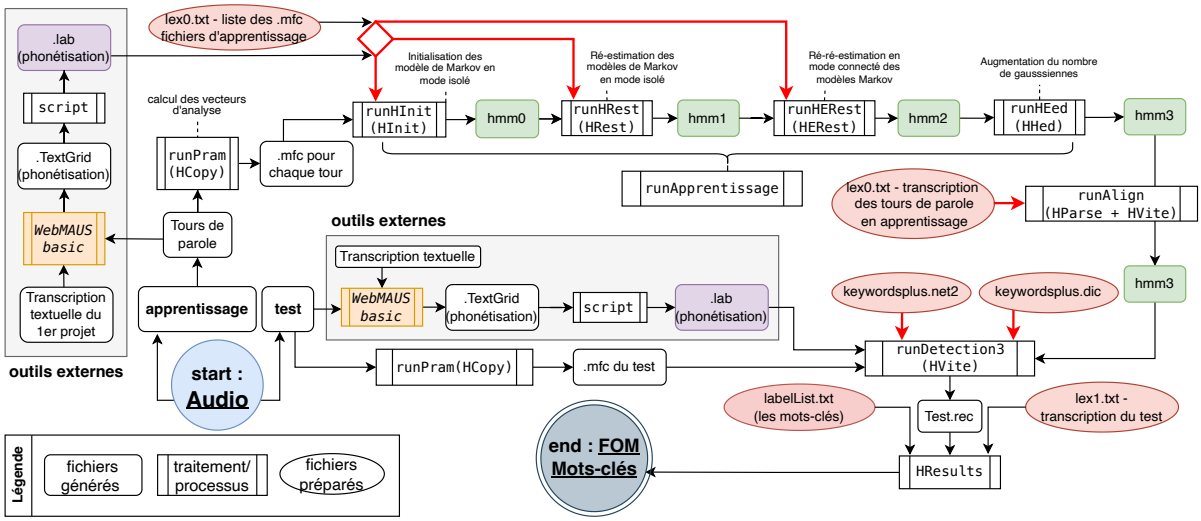

Figure 1 – Flowchart du projet

1 Méthodologie de Construction des Modèles de Markov

1.1 Détail des étapes de construction des modèles

L'étape initiale consiste à la préparation des données, ainsi que leur phonétisation. Pour réduire l'impact des parties en chevauchement dans les tours de parole, j'ai choisi de ne pas les prendre en compte dans la construction des modèles de Markov. En fait, dans les deux fichiers d'apprentissage, la fin de la première minuite (environ 1,2 seconde) sont supprimé et une séquence entre 32,5 s et 34,5 s est supprimée dans le deuxième audio d'apprentissage. Après avoir pré-traiter les duex audios, j'ai découpé les audios en 17 tours de parole.

Grâce à un outil externe *WebMAUS*[3], j'ai réalisé la segmentation phonétisée en SAMPA des tours de parole avec un fichier .TextGrid généré. Pour extraitre les informations dont j'ai besoin, j'ai ecrit un script sur praat *creerLab.praat* qui transforme les annotaitons en format du fichier .lab. La vérification des résultats est autant indispensable pour les fichiers TextGrid que pour les fichiers .lab. D'abord, lors que deux locuteurs changent leur paroles, l'annotation automatique pourrait être perturbée. Ensuite, certains séquences durant millesecondes ne peut pas être identifié avec un marqueur par un point d'interrogation. Dans ce contexte, je les ai relu sur la fenêtre de praat afin de corriger les annotations. En outre, quelques symboles SAMPA dans les fichiers .lab sont remplacées par un codage modifié. Pour la majorité de modification, le remplacement est dû à une adaptation au système : (R E O S Z H J A a o e 9 2)→(rr ee oo ss zz hh jj aa a# o# e##9 #2). Un cas

3. https ://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMAUSBasic

particulier réside dans le changemnt du phonème [œ̃]. J'ai assimilé [œ̃] à [ɛ̃] afin d'adapter aux symboles de HTK.

Cette étape est suivie par la génération des fichiers MFCC de vecteurs d'analyse à l'aide de l'outil HCopy. Les caractéristiques sont résumées dans les analyses vectorielles.

Par la suite, le processus passe par plusieurs étapes de calibration des HMM. La première est l'initialisation des modèles de Markov en mode isolé (HInit), suivie par des ré-estimations successives pour affiner ces modèles. Chaque ré-estimation, effectuée avec des outils comme HRest et HERest, permet d'ajuster les modèles pour qu'ils correspondent mieux aux données de parole. Avec l'idée du réductionnisme, le script de *runApprentissage.pl* est simplifé en version *runApprentissage1.pl* avec le module HInit dans le but de tester et de localiser l'erreur.

Après l'amélioration des modèles par augmentation du nombre de gaussiennes, le système est prêt pour l'apprentissage et la détection des mots-clés. Un renforcement d'alignement est ajouté avant la détection finale. Cette phase finale utilise un réseau de décodage et ajuste les paramètres pour optimiser la reconnaissance des mots-clés spécifiques. Le processus se termine par l'évaluation des résultats (FOM), permettant d'affiner davantage le système au cas où l'on a besoin de modifier la pondérations des relations dans le réseau de décodage.

2 Phonétisation « semi-automatique » des Tours de Parole

La sémi-automatisation en segmentation signifie non seulement un traitement rapide et puissant à l'aide de l'outil, mais aussi une vérification et une correction rigoureuse des résultats. Dans le but de faciliter le processus, j'ai testé 3 outils qui m'a permis de automatiser la génération des fichiers .lab. *SPPAS* est un logiciel de recherche scientifique proposé par la laboratoire Parole et Langage en Aix-en-Provence, France. La segmentation phonétique est aussi testée sur un outil *WebMAUS basic* en ligne developpé par l'Institut de phonétique et de traitement de la parole de la Ludwig-Maximilians-Universität, Munich, Allemagne. *eSpeak* est un logiciel de synthèse vocale open-source. Dans ce projet, j'ai testé seulement sa fonctionnalité de phonétisation.

2.1 Segmentation Phonétique avec *SPPAS*

Le segmentation phonétique est d'abord effectué par le logiciel *SPPAS*. SPPAS est un logiciel open source pour l'annotation et le traitement automatique des données de parole et aussi de vidéo. Il est écrit en Python et est disponible pour les systèmes d'exploitation Windows, Mac OS et Linux. J'ai passé pas mal de temps pour apprendre à utiliser ce logiciel. En fait, il est assez difficile à utiliser et les résultats de sortie n'étaient pas satisfaisants. J'ai testé un audio contenant une seule phrase en fournissant une transcription correspondante. La phonétisation n'est pas bonne car il a proposé plusieurs possibilté de transcription qui devrait être un seul phonème et elle n'est pas arrivé à segmenter les

phonèmes avec les marqueurs temporels. Autrement dit, le résultat est que, une phrase correspond à une longe suite de phonèmes sans frontière ou annotation temporelle. Après, lancer l'alignement n'a généré que d'erreur. De ce fait, j'ai décidé de changer de logiciel pour la segmentation phonétique.

2.2 Segmentation Phonétique avec *WebMAUS*

2.2.1 Présentation de *WebMAUS*

WebMAUS[4] est un outil robuste et simple qui m'a épargné à passer énormément du temps à annoter. La segmentation phonétique est aussi testée sur un outil *WebMAUS basic* en ligne. Ce outil fait partie élémentaire de *BAS WebServic* qui est un service de l'Archive bavaroise des signaux de parole hébergé par l'Institut de phonétique et de traitement de la parole de la Ludwig-Maximilians-Universität, Munich, Allemagne.

WebMAUS basic fonctionne en alignant automatiquement le texte transcrit d'un enregistrement vocal avec l'audio correspondant. L'utilisateur fournit un fichier audio et un texte transcrit, et WebMAUS génère les limites temporelles de chaque phonème et mot dans l'enregistrement. Les phonèmes sont directement transcrit en codage de X-SAMPA.

Le principal avantage de *WebMAUS basic* est sa capacité à traiter automatiquement des enregistrements audio en différentes langues, dont le français. Il utilise des modèles acoustiques et des dictionnaires phonétiques pour chaque langue, ce qui lui permet de fournir des alignements assez précis. Cependant, il est important de noter que la qualité de l'alignement dépend en grande partie de la clarté de l'enregistrement audio et de la précision de la transcription.

WebMAUS basic est accessible gratuitement en ligne, ce qui le rend facilement disponible pour tout le monde. Il ne nécessite pas d'installation de logiciel spécifique ou de modification pour l'encodage SAMPA, ce qui facilite largement son utilisation.

Pour notre projet, ce que je devais faire, c'est d'importer les fichiers audio et préparer les transcriptions correspondantes textuelles et de télécharger les fichiers de sortie sous format *TextGrid*. Les fichiers de sortie contiennent les marqueurs temporels de chaque phonème et mot dans l'enregistrement.

2.2.2 Remarques sur le résultat de *WebMAUS*

Dans les résultats de phonétisation, j'ai remarqué deux points qui sont non-négligeable à mentionner. Premièrement, l'allongement du phonème est identifié particiellement. Par exemple, dans le mot *même [mEm]*, la fin *[m]* est souvent phonétisé comme *[m@]*. Ce phénomène appartient aussi à la disfluence de hésitation. La transcription automatique a réussi à les phonétiser pour certains cas, mais une autre petite partie restait à ajouter manuellemnt. Deuxièmement, le changement des tours de la parole a perturbé un peu l'alignement phonétique car le timbre et les fréquences sont beaucoup différents entre les 5 locuteurs. D'ailleurs, une partie de liaison comme [z] au centre de *six équipes* devait

4. https ://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMAUSBasic

être corrigée dans le processus de vérification. Par contre, la laision [t] dans *huit équipes* est bien identifiée. En outre, l'annotation en codage *SAMPA* a plus de phonèmes que le codage de *HTK*. Donc, avec *WebMAUS*, j'ai rencontre [9∼] qui n'existe pas dans le codage *HTK*, dont ils sont assimilé à [e#] dans les fichiers de sortie. Le phonème [jj] (dans le mot « ligne ») n'existe pas dans la liste de phonèmes appliquée dans les ateliers passés, et il a été rajouté dans la liste.

2.3 Phonétisation avec *Espeak*

eSpeak, dans sa fonction de phonétisation, est un outil essentiel pour la conversion de texte en parole. Cette fonctionnalité permet à *eSpeak* de transformer le texte écrit en une séquence de sons de parole (phonèmes), ce qui est une étape cruciale dans le processus de synthèse vocale. Cette capacité de phonétisation rend *eSpeak* particulièrement utile dans des applications où la précision de la prononciation est importante, comme dans l'apprentissage des langues ou la lecture de textes dans différentes langues. Grâce à cette grande flexibilité, j'ai testé la phonétisation en fournissant la transcription textuelle de mes audios. Un inconvénient remarquable est que cet outil nécessite un temps conséquent. L'opération, bien qu'efficace, peut être relativement longue, surtout pour des textes un peu volumineux ou complexes.

2.4 Réflexion sur la phonétisation

La phonétisation des audios est cruciale pour la performance des modèles de Markov cachés (HMM) dans HTK (Hidden Markov Model Toolkit), surtout dans les applications de reconnaissance de la parole. Cette tâche implique la conversion de la parole en une séquence de phonèmes, éléments de base pour la modélisation acoustique dans les HMM. Après avoir fini la construction des HMM et le test de détection, j'ai revu les fichiers d'entraînement et les résultats de reconnaissance. L'imprécision se trouve peut-être dans les données d'apprentissage, dit autrement, la correspondance entre les données audio et les modèles phonétiques du système. Si la segmentation n'est pas précise, les HMM n'arrive pas à correctement apprendre et reconnaître les patterns sonores spécifiques (mfcc) des phonèmes dans la langue cible. En somme, la qualité de la phonétisation a un impact direct sur l'exactitude et l'efficacité de la reconnaissance de la parole par les HMM.

3 Réseaux de Décodage

3.1 Choix des mots-clés/nœuds

Les réseaux de décodage se composent de deux parties : le vocabulaire et les grammaires. En pratique, le module *Hvite*, basé sur l'algorithme *Viterbi*, fait appel à la grammaire pré-définie afin de comparer un fichier vocal à ce réseau de décodage et produire une transcription (fichier .rec). La grammaire est un fichier texte qui contient les motifs à

rechercher. Mon fichier vocal de test contient un phonème ŋ (prononcé dans le mot *ligne* ou *gagner*) qui n'existe pas dans le vocabulaire appliqué pour les ateliers précédents.

De sorte, j'ai choisi de re-construire un vocabulaire qui contient en total 45 éléments et les transcriptions phonétiques correspondantes, y compris 10 mots-clés, 34 phonèmes [5] et la silence. Ensuite, j'ai ré-créé la grammaire dans le fichier *net* en complétantles phonèmes, les relations et aussi les différents poids.

Tout d'abord, le choix des mots-clés est prédominant dans la reconnaissance de la parole. Différents objectifs changeront largement les résultats attendus. Pour ce projet, j'ai choisi les mots-clés suivants : *France, finale, CoupeDuMonde, Deschamps, MichelPlatini, Chine, Japon, Zidane, Brésil, bordelais* [6]. Ces mots-clés sont choisis en fonction de la thématique du corpus et de leur fréquences sur le corpus du test. D'ailleurs, la séquence « CoupeDuMonde » est collée à l'ensemble comme un seul mot en raison de l'aspect sémantique.

Deschamps, MichelPlatini, les deux joueurs sont désignés en manière différente en raison de la référence cognitive. Les recherches sur Google ont comfirmé que les gens les appellaient dans les reportage comme une habitude. *Didier Deschamps* est souvent appelé *Deschamps* dans les titres de sorte que je n'ai ajouté que ce prénom dans la liste de lexique. Pour *Michel Platini*, le prénom et le nom sont tout utilisés dans les titres. Si l'on recherche *Platini* ou *Platini football*, on ne trouve presque que les titres contenant *Michel Platini*. Vu ce phénomène, j'ai pris *MichelPlatini* comme un ensemble.

Sur le plan du réseau de décodage, il existe 49 nœuds représentant des mots et 182 arcs représentant les transitions entre les mots.

Bien qu'ils sont nombreux, leur catégories sont limitées et simples. Dans les 49 nœuds, d'un côté, ce sont des 10 mots-clés, 34 phonèmes, la silence comme ce que j'ai mentionné au début de cette section, d'autre côté, 4 nœuds marqués comme *!NULL* : *état initial d'un élément, état final d'un élément*. Si l'on les classifie en fonction des éléments liés (Voir la figure 2), les états de transition sont plus compréhensibles. Un caractèristique de notre système de reconnaissance est que, l'état final d'un mot-clé est toujour le nœud 1, par contre, celui d'un phonème est toujours le nœud 12. Tout les états se termine sur un NULL nœud 46.

En respectant la carctéristique du *network*, j'ai ajouté un nœud et quatre arcs correspondants pour un phonème [ŋ]. Autrement dit, j'ai instancié les arcs et le nœud de la partie en bas de la figure 2.

5. rr, i, a, t, e, p, k, s, l, d, m, a#, b, ee, oo, o#, n, o, ss, v, f, y, z, zz, j, #9, e#, u, #2, hh, @, w, g, jj.
6. Les cinq premiers séquences existent dans le test et les cinq dernier ne sont pas dans l'audio de test.

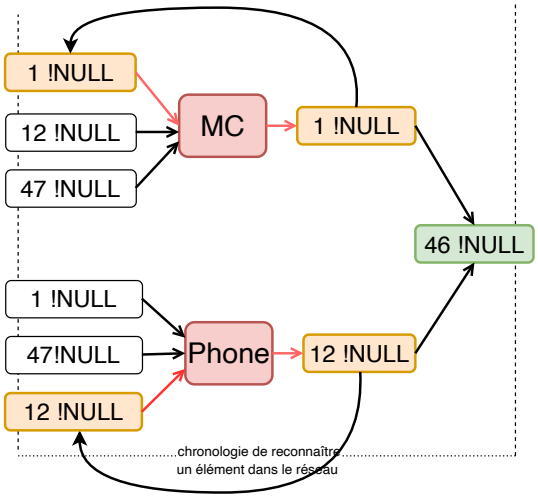

Figure 2 – Simplification du réseau de décodage : les nœuds de transitions sont regroupés selon les nœuds de mot-clé/phonème. *MC* est raccourci des mots-clés et *Phone* représente des phonèmes.

3.2 Pondération des arcs

	Training fichers P1,P2	Training fichers P1,P2,P3	
Nœuds concernants	Poid1	Poid2	Nuances du poid
I=0 W=france	8	10	+2
I=2 W=CoupeDuMonde	40	40	==
I=3 W=MichelPlatini	75	50	-15
I=4 W=finale	32	22	-10
I=5 W=Deschamps	1000	1000	==
Tous les phonèmes	-20	-20	==

Table 1 – Pour préciser les modèles HMM, j'ai ajouté des fhciers d'apprentissage d'un tiers progressivement. P1 désigne Yuyan, P2 représente Anne et P3 est Marceau. Les poids désignent la valeur de récompense et de pénalité aux arcs, conduisant aux 5 mots-clés, aux phonèmes et au silence. Après avoir ajusté les poids, j'ai remarqué que le résultat FOM du test n'a pas été amélioré (toujours à 75%). Cependant, une alarme fausse du mot « france » est réduite.

Le vocabulaire et la grammaire ne sont pas suffisants pour la reconnaissance de qualité. En effet, les poids des arcs sont aussi importants pour la performance du système. Dans ce projet, j'ai ajusté plusieurs fois des poids aux arcs pour les mots-clés et les phonèmes. Tous les modification des valeurs de poids se conforment à l'idée dichotomique. En général, j'ai

augmenté la valeur de récompense pour les mots-clés ciblés et la valeur de pénalité pour les phonèmes (Plus grande la valeur de pénalité pour les phonèmes, moins de poids dans le réseau de décodage). En fait, le processus d'ajustement n'est pas facile. Il faut faire des essais et l'analyse de cette partie n'est pas utile pour ce projet.

Néanmoins, deux tendences sont intéressantes à discuter et à noter. Premièrement, chaque fois l'ajoute des nouvelles données d'apprentissage, la valeur de récompense pour les mots-clés ciblés devraient être modifiés. Par exemple, avec le même pondération (Poid1 du tableau 1), l'ajouté des données de Marceau a rendu le FOM à 37.50 %. Cela signifie aussi que différentes volumes des données d'apprentissage pour différents phonèmes engend différente précision de reconnaissance. Deuxièmement, la pondération ne fonctionne pas pour les éléments introuvables, comme le mot « Deschamps » dans le test. En effet, la valeur de récompense pour ce mot-clé est déjà très grande (de 0 à 5000), mais il n'a pas été retrouvé. Dans la reconnaissance, le mot « Deschamps » est toujours identifié comme « #2 » de sorte que l'augmentation de la valeur de récompense n'a pas d'effet.

4 Résultats de Reconnaissance et Analyse

A l'aide de la pondération, j'ai réussi à améliorer le FOM de 20% à 75%. Pour chaque mot-clé, la modification de la valeur de récompense n'aurait pas de l'influence positive sur le FOM pour chaque mot-clé par rapport à cette base de résultats. Deux mots-clés n'ont pas un FOM (moins de 60%) : « france » et « Deschamps ». Comme ce que j'ai mentionné dans la section précédente, la valeur de récompense pour le mot-clé « france » est défini en 8 car son augmentation n'introduit que les *Fausse Alarme* sans améliorer la FOM. En outre, bien que j'ai essayé d'augmenter la valeur de récompense pour le mot-clé « Deschamps », il n'a pas été reconnu. Cela est dû à la qualité des HMM. Afin de reconnaître les mots-clés, j'ai essayé d'ajouter plus de données d'apprentissage pour fine-tuning ces modèles utilisés. Bien que le FOM moyenne n'a pas été amélioré, une fausse alarme a été réduite. Ce changement implique que l'ajoute des sources d'apprentissage pourrait influencer les FOM d'une manière positive à condition que la pondération des éléments ciblés est bien ajustée.

De plus, la reconnaissance dépend aussi du choix des mots-clés. « CoupeDuMonde » et « MichelPlatini« ont plus de phonèmes par rapport aux autres mots-clés. Cette caractéristique pourrait finluencer un peu la reconnaissance. Cependant, dans une situation réelle, il est difficile de choisir les mots-clés en fonction de leur nombre de phonèmes ou en fonction de leur fréquence. Donc, cette supposition dans ce projet est un peu artificielle. En partique, les éléments à reconnaître ne sont pas forcément limités au « mot » (qui signifie un token dans le traitement) vu « CoupeDuMonde » choisi dans ce projet. En effet, il est possible de reconnaître une séquence de mots, une phrase ou même une phrase entière dans une situation réelle. Dans ce cas, la reconnaissance de la parole devient plus flexible et plus complexe.

La difficulté de ce projet consiste à la parole continue et la reconnaissance flexible entre phonème et mots-clé. Plus la reconnaissance s'adapte à la vie réelle, plus la difficulté

augmente et plus le bruit s'introduit.

```
====================== HTK Results Analysis =======================
  Date: Mon Jan 15 22:36:33 2024
  Ref :
  Rec : donnees/radio04/param/test/synthRadio04_3.rec
---------------------- Figures of Merit -------------------------
      KeyWord:    #Hits     #FAs  #Actual      FOM
       france:        1        1        2    50.00
       finale:        2        0        2   100.00
 CoupeDuMonde:        2        0        2   100.00
    Deschamps:        0        0        1     0.00
MichelPlatini:        1        0        1   100.00
        chine:        0        0        0     0.00
        japon:        0        0        0     0.00
       zidane:        0        0        0     0.00
       bresil:        0        0        0     0.00
      Overall:        6        1        8    75.00
 649 non keywords found in test files-------------------------------

====================== HTK Results Analysis =======================
  Date: Tue Jan 23 17:48:04 2024
  Ref :
  Rec : donnees/radio04/param/test/synthRadio04_3.rec
---------------------- Figures of Merit -------------------------
      KeyWord:    #Hits     #FAs  #Actual      FOM
       france:        1        0        2    50.00
       finale:        2        0        2   100.00
 CoupeDuMonde:        2        0        2   100.00
    Deschamps:        0        0        1     0.00
MichelPlatini:        1        0        1   100.00
        chine:        0        0        0     0.00
        japon:        0        0        0     0.00
       zidane:        0        0        0     0.00
       bresil:        0        0        0     0.00
      Overall:        6        0        8    75.00
 663 non keywords found in test files-------------------------------
```

Conclusion

Ce projet a permis de mettre en pratique les connaissances acquises lors des ateliers de reconnaissance de la parole, offrant une exploration approfondie des outils de base de la reconnaissance vocale. L'objectif principal était de construire un système de reconnaissance de mots-clés efficace en utilisant HTK dans la parole continue. Bien que les résultats soient encourageants, ils révèlent également la nécessité d'améliorer davantage la performance du système.

La limitation des ressources d'apprentissage, provenant de seulement trois personnes, met en évidence le besoin de diversifier et d'élargir la base de données pour renforcer la qualité et la fiabilité du système. Dans le domaine de la reconnaissance vocale, la variété des données d'entraînement est cruciale pour assurer une bonne généralisation du modèle.

La qualité de la phonétisation, un facteur déterminant dans la précision de la reconnaissance, souligne également l'importance de l'intégration de la recherche avancée en phonétisation automatique. Bien que ce domaine soit en pleine évolution, son incorporation dans le système actuel pourrait conduire à des améliorations significatives.

La familiarisation avec la plateforme HTK, malgré les défis rencontrés et les imprécisions des tutoriels, a été une expérience enrichissante. Elle a non seulement permis une meilleure compréhension des aspects techniques, mais a aussi renforcé les compétences en résolution de problèmes et en autonomie d'apprentissage.

L'observation que l'augmentation de la puissance du système allonge le temps d'entraînement indique un équilibre délicat entre capacités de traitement et efficacité. Cela suggère que de futures recherches pourraient se concentrer sur l'optimisation des algorithmes et l'exploitation de ressources informatiques plus avancées, comme des solutions basées sur le cloud ou le calcul parallèle, pour gérer de manière plus efficace des ensembles de données plus volumineux.

La langue ciblée dans ce projet est le français. Cependant, la reconnaissance de la parole est un domaine qui s'applique à toutes les langues. En effet, l'extension à d'autres langues pourrait être une piste de recherche intéressante mais également un défi, car elle nécessite une compréhension approfondie des outils. Loin de simplement copy-coller les systèmes, adpter les systèmes à une autre langue nécessite une compréhension approfondie des outils et des langues.

En somme, ce projet a non seulement approfondi la compréhension de la reconnaissance vocale, mais a aussi ouvert des pistes pour des améliorations futures et des explorations dans ce domaine dynamique et en constante évolution.

Références

[1] Brigitte Bigi. SPPAS - MULTI-LINGUAL APPROACHES TO THE AUTOMATIC ANNOTATION OF SPEECH. *The Phonetician. Journal of the International Society of Phonetic Sciences*, 111-112(ISSN :0741-6164) :54–69, 2015.

[2] Brigitte Bigi. A phonetization approach for the forced-alignment task in SPPAS. In *Human Language Technology. Challenges for Computer Science and Linguistics, LNAI 9561*, pages 515–526. 2016.

[3] Thomas Kisler, Uwe Reichel, and Florian Schiel. Multilingual processing of speech via web services. *Computer Speech & Language*, 45 :326–347, September 2017.

[4] Steve Young, Gunnar Evermann, M.J.F. Gales, Thomas Hain, Dan Kershaw, Gareth Moore, James Odell, Dave Ollason, Daniel Povey, Valtcho Valtchev, and Philip Woodland. *The HTK Book (from version 3.3)*. 01 2004.