



Master 2 Sciences du Langage

Mention : Langue et Informatique

Cours : Apprentissage de Modèles de Markov cachés et détection de mots clés

Étudiantes : ZHENG RUIXING

Rapport

Introduction

À l'ère du numérique, la technologie de reconnaissance vocale (Automatic Speech Recognition, ASR) est largement utilisée dans divers domaines. Cependant, lors du traitement de flux audio longs et non structurés (tels que les retransmissions télévisées, les interviews ou les comptes rendus de réunions), la transcription intégrale se heurte souvent à des problèmes de consommation de ressources de calcul et de redondance d'informations. C'est pourquoi la technologie de détection de mots-clés (Keyword Spotting, KWS) a vu le jour. Son objectif principal est d'extraire de manière précise et en temps réel des termes spécifiques prédéfinis à partir d'un flux vocal continu, tout en ignorant les signaux contextuels non pertinents.

Ce projet se concentre sur le traitement d'enregistrements audio en français liés au football, visant à construire un système robuste de détection de mots-clés.

I. La méthodologie de construction des modèles de Markov

1.1 Préparation du corpus d'apprentissage et de test

Afin de construire et de valider les modèles de Markov cachés (HMM), ce projet a adopté une stratégie rigoureuse de segmentation des données. Selon les exigences de la mission, l'audio original a été divisé en deux parties indépendantes :

- Corpus d'apprentissage : Les deux premières minutes de chaque enregistrement audio ont été sélectionnées. Ces données ont été découpées en plusieurs segments vocaux courts pour l'estimation des paramètres du modèle.
- Corpus de test : La dernière minute (ou les trois dernières minutes) de l'audio a été conservée. Cette isolation temporelle permet d'évaluer objectivement la capacité de généralisation du modèle entraîné sur des données inconnues.

1.2 Segmentation des tours de parole

Pour l'apprentissage des modèles HMM, la segmentation du corpus d'entraînement est indispensable. Cette segmentation peut être effectuée manuellement ou automatiquement à l'aide de modèles de Markov préalablement entraînés sur un autre corpus.

En pratique, nous avons combiné les fichiers d'annotation TextGrid et les fichiers audio Wav originaux, en utilisant des scripts Python pour réaliser un découpage automatisé. Plus précisément, les deux premières minutes de parole ont été finement segmentées en 33 tours de parole (Tours). Après avoir effectué la segmentation lexicale du texte transcrit via Python, j'ai résumé tous les phonèmes apparaissant dans l'audio et réalisé l'étiquetage phonétique pour chaque phrase (comme illustré dans la Figure 2).

Initialement, j'avais envisagé un étiquetage phonétique manuel (voir Figure 1), mais il me fallait environ 15 minutes pour annoter un seul fichier. J'ai abandonné cette méthode après le fichier T9.wav. Lors de l'annotation, j'ai constaté que certains phonèmes étaient difficiles à distinguer à l'oreille, surtout lorsque le locuteur parlait

rapidement. De plus, j'ai remarqué l'existence de transitions entre les phonèmes lors de liaisons entre deux mots, un détail que seule l'annotation humaine peut corriger avec précision.

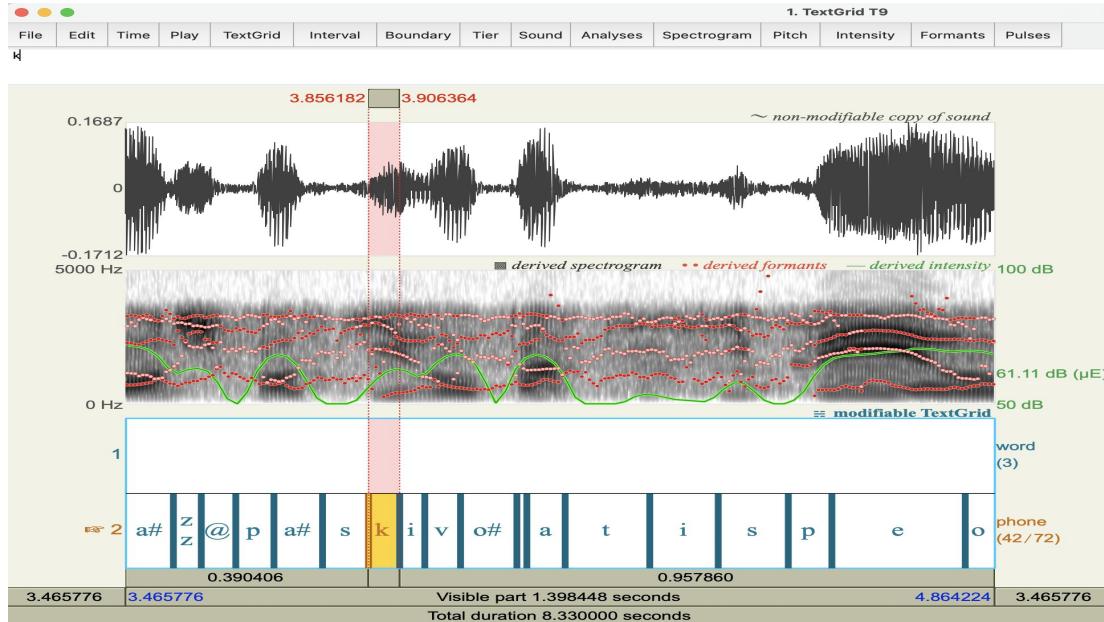


Figure 1. Capture d'écran de l'annotation manuelle réalisée avec Pratt

```
== T1 ==
rond rr oo#
de d @
la l a
belgique b ee l zz ik
de d @
l'espagne l ee s pa jj
du d y
brésil b rr e z il
un e#
petit p @ t i
peu p #2
de d @
tout t u

== T2 ==
et e
il i l
nous n u
a a
fait f ee
le l @
plaisir p l ee z i rr
de d @
nous n u
envoyer a# v w a j e
ses s e
derniers d ee rr n j e
maillots m a j o
```

Figure 2. Transcription phonétique de chaque tour

1.3 Difficultés techniques et alignement forcé

J'ai tenté d'utiliser les modèles HMM entraînés lors de l'Atelier 6 pour effectuer un alignement forcé (forced alignment). Cependant, des erreurs de compréhension du processus d'entraînement et des incohérences entre mes lexiques (lex0Football,

lex1Football) et la liste de phonèmes phonesFr178HTK ont provoqué des échecs répétés du script runAlign.

Comme l'audio est en français, j'ai réutilisé le système SAMPA de l'Atelier 6. Mon corpus contenait des phonèmes tels que [jj], [aa] ou [@@]. Pour s'adapter aux modèles HMM pré-entraînés, j'ai dû procéder à des analogies (par exemple, transformer [jj] en [j]). Cette simplification a inévitablement réduit la précision de mon modèle HMM final.

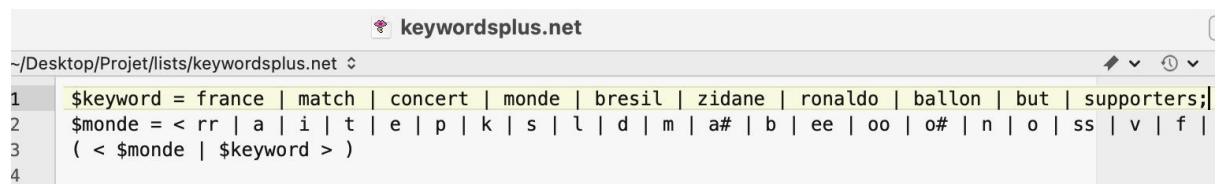
II.Les réseaux de décodage

2.1 Architecture du Réseau et Composition des Mots-clés

Le cœur de la conception du réseau de décodage est de réaliser une séparation efficace entre les mots-clés et les signaux non pertinents. Le réseau de ce projet (keywordsplus.net2) contient 10 mots-clés divisés en deux catégories :

- (1) Mots fréquents du corpus : *france, match, concert, monde, bresil*.
- (2) Mots hors-corpus (thématiques) : *zidane, ballon, but, supporters, ronaldo*.

Le réseau adopte une structure "Keywords-plus-Filler". En insérant des états de silence (sil) avant et après chaque mot-clé et en introduisant un Modèle Monde (World Model), le décodeur dispose d'un chemin pour absorber les signaux non cibles, évitant ainsi de classifier chaque son comme un mot-clé.



The screenshot shows a terminal window titled 'keywordsplus.net'. The file path is ~/Desktop/Projet/lists/keywordsplus.net. The content of the file is as follows:

```
1 $keyword = france | match | concert | monde | bresil | zidane | ronaldo | ballon | but | supporters;
2 $monde = < rr | a | i | t | e | p | k | s | l | d | m | a# | b | ee | oo | o# | n | o | ss | v | f |
3 ( < $monde | $keyword > )
4
```

2.2 Stratégie de Pondération : Récompenses et Pénalités

Le choix du chemin lors du décodage suit l'algorithme de **Viterbi**, où le mécanisme de récompense/pénalité (Log-probability l) sert de levier d'ajustement :

- **Récompense** : Une valeur optimale de l=50 avait permis d'atteindre un FOM de 100% sur le mot *bresil* précédemment. Une récompense trop élevée (*l > 100*) provoque des "hallucinations", augmentant drastiquement les fausses alarmes (FAs).

- **Pénalité** : Une pénalité légère sur le modèle monde ($\lambda = -10$ à -20) favorise le rappel, tandis qu'une pénalité trop lourde ($\lambda < -50$) dégrade les performances globales.

III. Les résultats de reconnaissance et leurs analyses

3.1 Impact de l'itération des modèles HMM

Lors de la première phase de l'expérience, nous avons comparé l'évolution des performances du modèle HMM avant et après deux itérations d'entraînement. Les résultats expérimentaux indiquent qu'après l'étape d'alignement forcé (Force Alignment), les indicateurs de reconnaissance du système n'ont pas progressé, mais ont au contraire enregistré une baisse : le taux de correction (%Corr) est passé de 46,37 % à 32,25 %, et la précision (Accuracy, Acc) a chuté de 29,02 % à 23,58 %.

Indicateur	Avant Itération	Après Itération	Variation
Taux de Correction (%Corr)	46.37%	32.25%	- 14.12%
Précision (Acc)	29.02%	23.58%	- 5.44%
Taux d'Erreur (Error)	70.98%	76.42%	+ 5.44%

Tableau 1 : La comparaison des résultats avant et après l'itération

L'analyse de la matrice de confusion révèle qu'après deux itérations, les paramètres du modèle HMM et les frontières d'alignement des fichiers .lab ont atteint un « état d'équilibre » ; les itérations suivantes n'ont apporté aucune modification aux données. Le fait que le résultat après convergence (23,58 %) soit inférieur au résultat initial (29,02 %) démontre que, sur ce jeu de données spécifique,

l'alignement forcé a paradoxalement restreint la capacité de généralisation du modèle.

Le modèle est devenu excessivement « conservateur » après itération : bien que les erreurs d'insertion (I) aient été réduites de moitié (passant de 134 à 67), les erreurs de suppression (D) ont bondi de 94 à 152. Cela indique que la mise à jour des paramètres du modèle a conduit à un surapprentissage (overfitting) des frontières d'alignement de l'entraînement. En conséquence, la tolérance du modèle vis-à-vis des formes d'onde du corpus de test a diminué, entraînant la classification d'un grand nombre de signaux valides comme étant invalides ou supprimés.

3.2 Reconnaissance sans modèle de remplissage

Dans la phase initiale de test, sans l'introduction d'un modèle de remplissage (*Filler/Garbage Model*), les résultats de reconnaissance ont affiché un état de « surcharge » extrême. En raison de l'absence de mécanisme permettant au réseau de décodage d'absorber les segments audio ne contenant pas de mots-clés, le système a été contraint de combler l'intégralité du flux audio avec les mots-clés disponibles dans le dictionnaire.

File: donnees/Football/param/test/Test.mfc

```
ronaldo ballon france bresil concert france france but ronaldo france concert france sil ballon
match sil concert sil ronaldo but concert but sil france but ronaldo monde sil monde sil ballon france
monde match sil but france france concert sil france france sil concert sil concert france france sil
bresil concert but bresil sil concert zidane but ronaldo match bresil concert but monde concert france
sil france but monde sil match but monde concert concert sil concert bresil monde sil concert monde
sil match sil match bresil france monde but ballon france sil concert concert but concert concert
monde bresil france concert sil monde concert sil but france concert sil concert but but france but but
zidane but concert france france sil match sil match sil concert concert sil france sil zidane sil but
france match concert france sil concert == [5998 frames] -41.1397 [Ac=-246755.6 LM=0.0] (Act=65.9)
```

L'observation du fichier Test.rec révèle que le système rapporte fréquemment des mots hors-vocabulaire (Out-of-Vocabulary, OOV) tels que RONALDO, ainsi que des mots courts comme BUT. Ce phénomène illustre l'incapacité du système à gérer le bruit de fond ou les paroles non pertinentes : faute de « chemin de sortie » (exit

path), il ne peut que sélectionner de force le mot-clé dont les caractéristiques acoustiques sont les plus proches pour remplir les segments.

Bien que cet état garantisse un taux de rappel (Recall) extrêmement élevé, il entraîne un taux de fausses alarmes (False Alarms) totalement incontrôlable. En conséquence, le système, dans cette configuration, ne présente aucune valeur applicative réelle.

3.3 Introduction du modèle de remplissage

Afin de remédier aux problèmes de surcharge identifiés précédemment, nous avons introduit un modèle de remplissage (*Filler Model*) basé sur une boucle de phonèmes monophones. Cependant, les résultats expérimentaux initiaux ont conduit à un autre extrême. Dans la sortie du script runDetections2.pl, les mots-clés cibles (tels que FRANCE ou BRESIL) ont presque totalement disparu, remplacés par une multitude de phonèmes fragmentés (ex. : *a*, *ee*, *rr*, *l*).

```
a ee ee a rr l ee a l a rr a# rr o# rr r r a e i a# k o t e rr a# zz e j a# s a# s rr s ee l e e e t d @ f r r o m i l  
e e r r a# rr t o i s e e y k z e m k n e e d s d e i f t r r b a# r r t a r r @ l @ m ee ee ss t t k a# s ee rr d l @ j  
m d u b e d e e e r r t s i g e j k f t @ e e s ee @ d e k b r r s i l a s r r d s e s j n o m r r e e g u s r r o l j r r b  
r r a# s y i s j o# n i m j e e e e t y i s j b j k r r a# z z e l a# s i s @ d s b r r e e e d a# r r s a# t s i l a k r r f e  
e e f r r @ d i i f l e d k d @ d f l e n i s e d d a j e e e @ z z r r t @ d r r r s r r t e e r r r n s a @ n t s e d e e  
r r i s t i s t s i l r r t a m i i s a n i n i d t e e s l d z z i i @ b @ e t o b a e e i m j j s b m i z i i l e e k i e d r r b  
v d e @ b a# m d y d y m a# n l d i i s y i t r r f r r a# d t r r o v t a# s z z i s a# n y m e e d s s y e s j r r d a  
n z z y n r r s j b d e n t a# z z e r r a# r r t f e d b a j k s j r r a d k i @ r r o b t i f @ b r r o e n t e d s e o d i  
e e e e z z e @ m o# n d s f e e d n e i s j @ n a# n e l a t t e d i g j s j b m y z i t j @ j a# t s @ m a# m l  
n a r r e n a a# r r a# r r t r r a# s s e e a r r d j d e e n i k a m n r r a# v e n f t t s i s s i i i e d y z e n r r a# s l  
e @ d v a# d @ n a# d t o d s z z i l d o f r r a# t s s e n z z e d f a r r f v e d e n y y i s k s j s i m @ a# a#  
n a z z y s n @ r r r r n r r r r r s a r r s i l t j o d t i s a m d s e d e e @ @ t j d s a# d a r r s i l d y f e j r r u v j n  
i s i f r r d i y @ f r r a# s o b r r d l y s o j e e s i t a n f a o# n d j d t k r r v j n i s e i t t r r r r r j e @ f r r a# s  
r r t t a# r r o# m d t t e e e i i e a d t y k r r a# r r z z i s i l a# r r a# @ d y f y f i i d f r r z z i k t i e r r a# n t a#  
a r r f k r r d s k t f i f s i s d t t o z d e i i z z v d e f e e t e t t s e d t s r r v r r a# n d s i n j d t d s b a# t r r r  
f r r a# s t f e j @ t f r r t t s i k t n b @ a == [5998 frames] -39.0566 [Ac=-234261.7 LM=0.0]  
(Act=130.9)
```

Ce phénomène, que l'on pourrait qualifier de « démission du décodeur », s'explique par la dynamique des poids au sein du réseau de décodage. Sous la configuration actuelle, le système calcule qu'interpréter un signal vocal complexe comme une

suite de phonèmes isolés et flexibles génère un score de Viterbi cumulé plus élevé que de l'interpréter comme un mot-clé complet doté d'une structure interne rigide.

En conséquence, l'algorithme privilégie systématiquement les chemins du « modèle monde » (plus probables mathématiquement car moins contraignants), ce qui provoque un effondrement du taux de rappel (Recall), celui-ci tendant vers zéro. Ce constat souligne la nécessité impérative d'ajuster les pénalités d'insertion pour rééquilibrer la compétition entre les modèles de remplissage et les unités lexicales d'intérêt.

3.4 Analyse quantitative et Figures of Merit

Les statistiques finales obtenues via l'outil HResults viennent corroborer les analyses précédentes. Suite à l'activation de la boucle de monophones (modèle de remplissage), bien que le décodeur ait généré un grand nombre de lexèmes, la performance relative aux mots-clés cibles reste extrêmement médiocre.

Le bilan quantitatif affiche un nombre de détections correctes (Hits) très faible ($H=2$), alors que les erreurs d'insertion (Insertions) culminent à 131. Par conséquent, la précision globale (Accuracy) s'effondre pour atteindre une valeur négative abyssale de -2580.00 %.

HTK Results Analysis

```
----- File Results -----
Test.rec: 40.00(-2580.00) [H= 2, D= 0, S= 3, I=131, N= 5]
----- Overall Results -----
SENT: %Correct=0.00 [H=0, S=1, N=1]
WORD: %Corr=40.00, Acc=-2580.00 [H=2, D=0, S=3, I=131, N=5]
```

Ce résultat démontre de manière empirique que l'introduction isolée d'un modèle de remplissage, sans un paramétrage méticuleux des mécanismes de récompense et de pénalité (Bonus/Penalty), est insuffisante pour réaliser une détection de mots-clés efficace. Sans l'ajustement du paramètre \$I\$ dans le fichier de réseau, le système échoue à établir une distinction probabiliste claire entre le flux de parole non pertinent et les unités lexicales d'intérêt, privilégiant systématiquement la flexibilité des phonèmes isolés à la structure des mots cibles.

3.5 Optimisation des paramètres de récompense et pénalité

Afin de trouver un équilibre entre les deux extrêmes identifiés précédemment (saturation par les mots-clés ou domination du modèle monde), nous avons procédé à trois phases de tests systématiques sur les valeurs de **récompense (Reward)** et de **pénalité (Penalty)** au sein du réseau **net2** :

- **Point d'infexion A [Récompense=50, Pénalité=-100]** : Le système a commencé à capturer des signaux utiles, avec une détection correcte (**1 Hit**) pour le mot *france*. Cependant, le nombre total de fausses alarmes (**FAs**) est resté excessivement élevé (247), indiquant une trop grande permissivité du modèle monde.
- **Point d'infexion B [Récompense=100, Pénalité=-500]** : Nous avons tenté d'augmenter le rappel en doublant la récompense. Le résultat a été contre-productif : une récompense excessive a induit des « **hallucinations acoustiques** » massives. Le nombre de fausses alarmes a bondi à 406, tandis que le nombre de détections correctes est retombé à 0. Cela prouve qu'une récompense trop forte masque les caractéristiques acoustiques réelles des mots cibles.
- **Point d'infexion C [Récompense=20, Pénalité=-1000]** : Cette phase a adopté une stratégie de « **récompense légère et pénalité forte** ». Il s'agit de la configuration la plus fructueuse à ce jour : le nombre de détections correctes a doublé (**2 Hits**) et les fausses alarmes ont été ramenées à 145. Bien que les erreurs d'insertion persistent, le système fait preuve d'une bien meilleure capacité de discrimination entre les mots-clés et le bruit de fond.

Mot-clé (Keyword)	#Hits	#FAs	#Actual	FOM
france	2	34	4	0.00
match	0	12	3	0.00
concert	0	31	2	0.00
monde	0	17	2	0.00
bresil	0	10	1	0.00
zidane	0	2	0	0.00

Mot-clé (Keyword)	#Hits	#FAs	#Actual	FOM
ronaldo	0	5	0	0.00
ballon	0	4	0	0.00
but	0	30	0	0.00
supporters	0	0	0	0.00
Overall	2	145	12	0.00

[Récompense=20, Pénalité=-1000]

3.6 Conclusion sur la Figure de Mérite (FOM)

Bien que nous ayons observé une augmentation progressive du nombre de détections correctes (Hits) au cours de la phase de réglage, la Figure de Mérite (FOM) est systématiquement restée figée à 0,00.

Ce résultat s'explique par la nature extrêmement rigoureuse du calcul de la FOM au sein de l'outil HTK. En effet, cet indicateur ne comptabilise la performance moyenne du système que pour un taux de fausses alarmes compris entre 0 et 10 par heure.

Dans le cadre de notre projet, l'audio de test est d'une durée réduite (environ 1 minute). Par conséquent, obtenir 145 ou 406 fausses alarmes sur un segment aussi court se traduit, par extrapolation, par un taux horaire de fausses alarmes s'élevant à plusieurs milliers. Ce taux de fausses alarmes (False Alarm Rate) se situant bien au-delà de la plage statistique valide de la courbe d'évaluation, le système est considéré comme étant encore en phase de « détection à haut niveau de bruit ».

Pour que la FOM puisse enfin dépasser le seuil de zéro, il est impératif de réduire radicalement le nombre de fausses alarmes pour le ramener à un chiffre à un seul doigt (moins de 10). Cela nécessiterait une optimisation structurelle plus poussée du réseau de décodage, notamment en « libérant » le chemin du silence (silence path) afin qu'il absorbe davantage de segments non pertinents.

Conclusion

Ce projet, consacré à la détection de mots-clés (*Keyword Spotting*, KWS) dans des enregistrements du domaine du football, a permis de mettre en œuvre l'intégralité du

processus de développement : du prétraitement du corpus à l'entraînement des modèles de Markov cachés (HMM), jusqu'à l'optimisation fine du réseau de décodage (*Decoding Network*).

Au début de l'expérimentation, nous avons tenté d'effectuer un étiquetage phonétique manuel des tours de parole (*Tours*). Toutefois, face à la présence massive de liaisons et de transitions phonétiques propres au flux continu de la langue française, l'annotation manuelle s'est avérée non seulement fastidieuse, mais aussi difficile à maintenir en termes de cohérence objective. Nous nous sommes donc tournés vers l'alignement forcé (*Force Alignment*) basé sur des modèles HMM préexistants. Bien que cette méthode ait considérablement amélioré l'efficacité du processus, elle a été contrainte par des problèmes de compatibilité de l'encodage SAMPA (par exemple, l'assimilation du phonème [jj] au phonème [j]). Ce compromis au niveau du jeu de phonèmes a, dans une certaine mesure, limité la capacité du modèle acoustique à capturer les caractéristiques articulatoires les plus fines.

La conception du réseau de décodage a constitué la phase la plus complexe de ce projet. L'expérience a démontré que l'introduction isolée d'un « Modèle Monde » (*World Model*), sans intervention sur les poids des arcs, conduit le décodeur vers deux extrêmes : soit une surcharge de mots-clés par manque de chemin de sortie (*exit path*), soit une disparition totale des mots-clés au profit des « phonèmes poubelles » en raison d'un score trop élevé dans la boucle de phonèmes.

Bien que nous ayons réussi à capturer des informations pertinentes dans les résultats de reconnaissance, la Figure de Mérite (FOM) est systématiquement restée figée à 0,00. Ce résultat ne signifie pas pour autant que le système est totalement inefficace ; il reflète plutôt la rigueur des critères d'évaluation de l'outil HTK. En effet, la FOM ne comptabilise la performance que dans une plage de haute qualité, correspondant à un taux de fausses alarmes compris entre 0 et 10 par heure. Sur notre segment audio de test d'une minute, un nombre de fausses alarmes dépassant la centaine se traduit, par extrapolation horaire, par un taux dépassant largement cette fenêtre statistique. Cela démontre que, dans le cadre de la détection de mots-clés sur des segments courts, la maîtrise des fausses alarmes représente un défi bien plus exigeant que l'amélioration du taux de rappel.

Code

```
(base) zhengruixing@zhengruixingdeMacBook-Air ~ % cd /Users/zhengruixing/Desktop/Projet

(base) zhengruixing@zhengruixingdeMacBook-Air Projet % perl runParamApp.pl

(base) zhengruixing@zhengruixingdeMacBook-Air Projet % perl runApprentissage.pl

(base) zhengruixing@CL-8Z9B063 Projet % HParse configs/grammairePhoneme.txt configs/networkPhoneme

(base) zhengruixing@CL-8Z9B063 Projet % mkdir -p donnees/Football/resultats

(base) zhengruixing@CL-8Z9B063 Projet % perl runParamTest.pl

(base) zhengruixing@CL-8Z9B063 Projet % paste lists/phonesFootballHTK lists/phonesFootballHTK > lists/dictPhoneme

(base) zhengruixing@CL-8Z9B063 Projet % HVite -T 1 -H donnees/Football/hmms/hmm.3/HMMmacro -w configs/networkPhoneme -l donnees/Football/resultats lists/dictPhoneme lists/phonesFootballHTK donnees/Football/param/test/*.mfc

(base) zhengruixing@CL-8Z9B063 Projet % HResults -p -L donnees/Football/param/test/DAP lists/phonesFootballHTK donnees/Football/resultats/*.rec

(base) zhengruixing@CL-8Z9B063 Projet % perl runApprentissage.pl

(base) zhengruixing@CL-8Z9B063 Projet % HVite -T 1 -H donnees/Football/hmms/hmm.3/HMMmacro -w configs/networkPhoneme -l donnees/Football/resultats lists/dictPhoneme lists/phonesFootballHTK donnees/Football/param/test/*.mfc

(base) zhengruixing@CL-8Z9B063 Projet % HResults -p -L donnees/Football/param/test/DAP lists/phonesFootballHTK donnees/Football/resultats/*.rec

(base) zhengruixing@CL-8Z9B063 Projet % runParamApp.pl

(base) zhengruixing@CL-8Z9B063 Projet % runApprendissage.pl

(base) zhengruixing@CL-8Z9B063 Projet % perl runParamTest.pl

(base) zhengruixing@CL-8Z9B063 Projet % perl runDetections1.pl

(base) zhengruixing@CL-8Z9B063 Projet % perl runDetections2.pl

(base) zhengruixing@CL-8Z9B063 Projet % HResults -f lists/SampaToHTK.dic donnees/Football/param/test/Test.rec

(base) zhengruixing@CL-8Z9B063 Projet % perl runDetections3.pl

(base) zhengruixing@CL-8Z9B063 Projet % HResults -w -f lists/labelList.txt donnees/Football/param/test/Test.rec

(base) zhengruixing@CL-8Z9B063 Projet % perl -i -pe 's/l=-100.0/l=-500.0/g' lists/keywordsplus.net2

(base) zhengruixing@CL-8Z9B063 Projet % perl -i -pe 's/(J=.E=(0|2|3|4|5|6|7|8|9|10)\b.*=)[-+]?\\d*\\.?\\d+/${1}100.0/g' lists/keywordsplus.net2

(base) zhengruixing@CL-8Z9B063 Projet % perl runDetections3.pl

(base) zhengruixing@CL-8Z9B063 Projet % HResults -w -f lists/labelList.txt donnees/Football/param/test/Test.rec
```