Chapitre 10

Développement de l'outil PLSPP

Une chaîne de traitements automatisés a été développée dans le cadre de ce travail de recherche afin d'identifier la position syntaxique des pauses et de mesurer le degré de proéminence acoustique des syllabes à partir des enregistrements du CLES. Cet outil, auquel nous ferons référence par l'acronyme PLSPP (Pauses and Lexical Stress Processing Pipeline) se compose d'une suite de modules recourant tantôt à des outils existants, tantôt à des scripts d'analyses originaux. PLSPP est entièrement open-source est disponible sur ce dépôt GitLab[1].

Ce chapitre présente chaque étape de traitement et chaque module qui composent les versions 1 et 2 de PLSPP. Les versions suivantes seront brièvement présentées en fin de chapitre.

La figure 10.1 présente l'architecture générale de PLSPP. Les modules de traitement sont les suivants :

- Identification du locuteur et segmentation de la parole ;
- Reconnaissance automatique de la parole et alignement au mot ;
- Détection des noyaux syllabiques (acoustique ou phonologique) ;
- Étiquetage morphosyntaxique et analyse par constituants ;
- Annotation des pauses ;
- Extraction des paramètres acoustiques, annotation des proéminence syllabiques et comparaison avec le dictionnaire phonologique de référence.

[1] https://gricad-gitlab.univ-grenoble-alpes.fr/lidilem/plspp

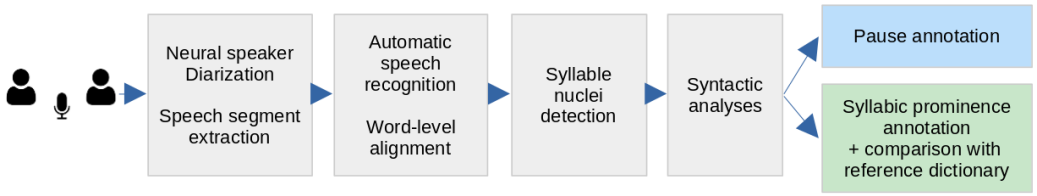

FIG. 10.1 : Architecture générale de PLSPP

10.1 Identification automatique du locuteur

Analyser automatiquement les enregistrements du corpus CLES est un challenge pour plusieurs raisons. Parmi elles, le fait qu'il s'agisse de conversations entre plusieurs locuteurs nécessite de pouvoir identifier qui parle quand. Par ailleurs, il arrive que la parole de plusieurs locuteurs se chevauche par moment, ou que l'un d'eux réagisse brièvement sans pour autant interrompre le tour de parole en cours. Enfin, certains tours peuvent être très courts ou non terminés. La difficulté ici est donc, outre l'identification du locuteur, de segmenter la parole en tours de parole.

L'outil choisi pour effectuer ce travail d'identification du locuteur et de découpage en segments de parole est pyannote.audio (Bredin, 2023 ; Plaquet & Bredin, 2023), combiné à un script de découpage et d'extraction audio paramétrable conçu pour les besoins de la présente pipeline de traitements.

Identification du locuteur Elle est gérée par Pyannote, qui est appelé par le script `diarisationPyannote.py`, qui prend en entrée les enregistrements audio bruts et renvoie un fichier texte par enregistrement listant chaque segment de parole détecté et son locuteur.

Fusion des segments Le script `pyannote2TextGrid.py` convertit ensuite les fichiers Pyannote en format TextGrid, en fusionnant les segments consécutifs du même locuteur. Un seuil paramétrable permet de jouer sur la sensibilité du découpage : plus il est élevé, plus les segments consécutifs ont tendance à être fusionnés, au risque toutefois de contenir des réactions de l'interlocuteur. Plus le seuil est bas, plus on limite la présence de l'interlocuteur dans le segment de parole, mais les segments ont alors tendance à être plus courts. La difficulté ici est de déterminer à partir de quelle durée on considère une prise de parole de l'interlocuteur comme un changement de tour, et jusqu'à quelle durée on la considère comme un simple backchannel qui peut être conservé dans le tour du premier locuteur. Nous avons paramétré ce seuil à 1 s par défaut, ce qui signifie qu'un segment est coupé à partir d'1 s de silence pour le locuteur

courant.

[[ILLUSTRATION AVANT et APRÈS FUSION]]

Extraction Le script `intervalles2wav.praat` extrait enfin chaque segment de parole en fichiers audio indépendants. Les paramètres disponibles sont la durée minimum des segments à extraire (par défaut 8 s) et la marge de découpage avant et après le segment (par défaut 10 ms).

En sortie de ce module, chaque enregistrement se retrouve découpé en autant de fichiers audio qu'il contient de segments de parole de la durée minimum paramétrée. Chaque fichier contient en théorie la parole d'un seul locuteur et peut être analysé indépendamment par les modules suivants.

10.2 Reconnaissance automatique de parole et alignement

Pour l'étape de reconnaissance automatique de la parole, WhisperX (Bain et al., 2023) est apparu comme l'outil le plus adéquat car il combine la haute précision de reconnaissance de Whisper avec une étape supplémentaire d'alignement au mot. Les récentes version de WhisperX intègrent maintenant une étape d'identification du locuteur avec Pyannote.audio, mais pour garantir une meilleure flexibilité notamment dans le découpage des segments, nous avons laissé les deux étapes séparées.

Transcription alignement au mot Le script `myWhisperxTG.py` exécute la reconnaissance de la parole et l'alignement au mot avec WhisperX. Il prend en entrée les fichiers mono-locuteur précédemment créés et renvoie la transcription alignée au mot en format TextGrid pour chaque fichier audio. Le script accepte plusieurs arguments comme le modèle utilisé (par défaut base.en), le type de processeur (par défaut CUDA, CPU et GPU en parallèle) et plusieurs paramètres techniques ajustables en fonction du serveur à disposition. Le modèle d'alignement est Wav2Vec2.0, mais celui-ci n'est pas paramétrable pour le moment.

10.3 Détection des noyaux syllabiques

Dans la première version de PLSPP, les mesures acoustiques pour l'accentuation lexicale sont réalisées au niveau des noyaux syllabiques estimés à partir des pics d'intensité par un script Praat de De Jong et al., 2021. En combinant ces noyaux syllabiques avec l'alignement au mot de Wav2Vec2.0, il est possible d'identifier chaque

syllabe des mots et de comparer leur mesures prosodiques. À partir de la version 2 de PLSPP, les mesures sont effectuées au niveau des intervalles vocaliques, localisées grâce à une couche supplémentaire d'alignement des phonèmes.

Détection acoustique des noyaux syllabiques `SyllableNucleiv3.praat` prend en entrée les fichiers audio et génère un fichier TextGrid avec chaque noyau syllabique détecté aligné au signal. Il prend en paramètre les mêmes options que le script original, notamment un band-pass de 300 Hz à 3300 Hz activé par défaut.

Détection phonologique des noyaux syllabiques Ce module ajouté dans la version 2 de PLSPP recourt au Montreal Forced Aligner (MFA, McAuliffe et al., 2017) pour aligner le texte brut transcrit par WhisperX. L'avantage qu'il présente est qu'il effectue un alignement au mot plus juste et ajoute une couche d'alignement phonémique, permettant d'identifier le noyaux vocalique des syllabes. En contrepartie, MFA est plus sensibles aux disfluences et aux écarts entre la transcription et le signal audio, et a tendance à produire des alignement incohérents avec des enregistrements de parole disfluente. Ce module semble donc moins adapté à la parole spontanée.

10.4 Analyses syntaxiques

Deux types d'analyses syntaxiques sont effectuées : un étiquetage morphosyntaxique pour déterminer la catégorie grammaticale de chaque mot, et une analyse par constituants pour obtenir un arbre syntaxique et regrouper les mots en syntagmes et en propositions.

Étiquetage morphosyntaxique Il est effectué par Spacy (Honnibal et al., 2020). Le script correspondant est `spacyTextgrid_v2.py`. Il prend en entrée le fichier TextGrid contenant la transcription alignée et renvoie le même fichier avec une tier supplémentaire indiquant la catégorie de chaque mot. Les paramètres sont le nom du modèle (par défaut `en_core_web_md`) et le nom de la tier contenant l'alignement des mots.

Analyse par constituants Elle est effectuée par Berkeley Neural Parser (Kitaev et al., 2019) via le script `text2benepar.py`. Celui-ci prend en entrée le texte brut de la transcription et génère un fichier texte contenant le résultat de l'analyse par constituants. Il prend en arguments le modèle d'analyse, par défaut `benepar_en3`[2].

[2] Les arbres syntaxiques peuvent être visualisés directement avec un outil tel que `RSyntaxTree` de Yōichirō Hasebe : https://yohasebe.com/rsyntaxtree.

10.5 Annotation des pauses

En sortie du module de transcription et d'alignement, nous disposons de la position estimée de chaque mot dans le signal audio. Par la suite, les analyses syntaxiques ont permis d'annoter chaque mot de leur partie du discours et d'obtenir l'arbre syntaxique à partir de la transcription de l'extrait. Dans ce module, nous nous intéressons non plus aux mots mais aux intervalles entre les mots. Dans le cas de l'alignement avec Wav2Vec2.0, tous les mots sont séparés par un intervalle vide étiqueté <p:> (parfois d'une durée de seulement quelques millisecondes). Ces intervalles ne sont pas nécessairement vides au sens de silencieux ; ils peuvent contenir des hésitations, des allongements, voire des faux départs – tout ce que WhisperX ne transcrit pas. Il nous a semblé opportun de considérer ces intervalles comme potentielle interruption du flux de parole.

Dans le cas de l'alignement avec MFA, les intervalles vides sont plus rares mais peuvent également être très courts (ex. 30 ms). Toutefois, l'alignement de Wav2Vec2.0 s'étant révélé plus fiable en parole spontanée que MFA, l'annotation des pauses est faite pour l'instant exclusivement à partir de l'alignement de Wav2Vec2.0.

Annotation des pauses Le script `pausesAnalysis.py` prend en entrée les transcriptions alignées au format TextGrid et les analyses par constituants au format texte, et renvoie un tableau listant tous les intervalles inter-mots, leur durée et leur contexte syntaxique : mots précédant et suivant ainsi que leur catégorie, type du plus grand constituant se terminant et commençant ainsi que le nombre de mots qu'ils contiennent et leur profondeur syntaxique à partir de la racine de l'arbre. À partir de là, l'utilisateur peut définir un seuil à partir duquel considérer un intervalle comme pause, et faire les analyses qu'il souhaite.

10.6 Annotation des proéminences syllabiques

L'objectif de ce module est de mesurer le degré de proéminence acoustique de chacune des syllabes des mots polysyllabiques, d'identifier la syllabe proéminente, et comparer sa position avec celle de l'accent lexical primaire tel qu'il est attesté dans un dictionnaire de référence. Les mesures acoustiques sont réalisées sur trois dimensions : la fréquence fondamentale (F_0), l'intensité et la durée. Selon la version de PLSPP, l'annotation est faite tantôt ponctuellement au niveau des pics d'intensité des syllabes (version 1), tantôt sur toute la durée de la voyelle (versions 2 et suivantes). Afin de filtrer les mots potentiellement mal alignés, seuls les mots dont le nombre de syllabes détectées correspond à une réalisation possible selon le dictionnaire de référence CMU

Pronouncing Dictionary[3] sont analysés. Dans les version 2 et suivantes, ce filtrage est optionnel.

Normalisation par locuteur Elle est effectuée de la même manière pour les trois dimensions acoustiques : chaque valeur absolue est convertie en centile pour le locuteur et la dimension en question. La valeur ainsi obtenue s'étend de 0 à 100, avec 50 indiquant la valeur médiane de la dimension donnée pour le locuteur, et 100 la valeur maximale. Cette méthode de normalisation permet de tenir compte de la distribution des mesures pour chaque locuteur, tout en permettant de comparer les valeurs entre elles (50 représente la valeur médiane pour tous les locuteurs). En contrepartie, il est nécessaire d'avoir suffisamment de mesures pour chaque locuteur, sans quoi plusieurs centiles peuvent renvoyer aux mêmes valeurs absolues. Une méthode de normalisation alternative est à l'étude pour permettre une annotation cohérente lorsque moins de données sont disponibles.

Annotation au niveau syllabique Effectuée dans la première version de PLSPP par le script `stressAnalysis.py`. Celui-ci prend en entrée les fichiers TextGrid contenant la transcription alignée, l'analyse morphosyntaxique et les noyaux syllabiques acoustiques (pics d'intensité) ; les fichiers audio, et le dictionnaire de référence CMU Pronouncing Dictionary. Pour chaque noyau syllabique acoustique (pic d'intensité), la F_0 est mesurée à partir du point le plus proche ("Get value at time...", "Hertz", "Nearest"), ou bien par interpolation linéaire si aucune valeur n'est trouvée. La durée est quant à elle estimée à partir des noyaux voisins ou des frontières de mot. En sortie sont générés les fichiers TextGrid avec trois tiers supplémentaires : pour chaque mot cible, le pattern de référence, le pattern observé global consistant en une moyenne des trois dimensions, et le pattern observé sur chacune des trois dimensions acoustiques (*cf.* figure 10.2). Les symboles pour représenter la syllabe proéminente et les autres syllabes sont personnalisables au début du script (par défaut "O" et "o" respectivement).

Cette version est actuellement la plus robuste car elle s'appuie sur une combinaison de l'alignement au mot et de la détection acoustique des noyaux syllabiques. Toutefois, les mesures sont effectuées de manière ponctuelle au niveau du maximum d'intensité de la syllabe, et ne prennent donc pas en compte la variation de F0 à travers la voyelle, et les mesures de durée sont plus facilement impactées par la structure syllabique et les allongements de consonnes, notamment les fricatives.

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

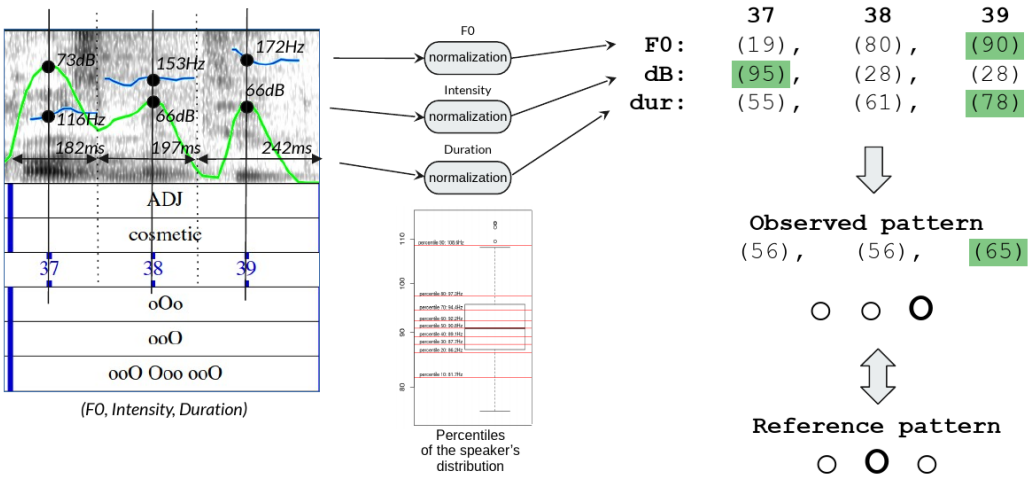

FIG. 10.2 : Extraction des paramètres prosodiques (PLSPP v1). À gauche un aperçu du fichier TextGrid de sortie avec les mesures acoustiques absolues indiquées en surimpression, la courbe bleue indique la F0, la verte l'intensité ; à droite les centiles correspondants au mesures absolue. "Observed pattern" correspond à la position de la syllabe proéminente (moyenne des trois dimensions prosodiques), "Reference pattern" correspond à la position attendue de l'accent primaire pour le mot "cosmetic"

Annotation au niveau vocalique À partir de la deuxième version de PLSPP, les mesures acoustiques sont faites au niveau de l'intervalle vocalique de chaque syllabe. Le script `stressAnalysis_mfa.py` suit la même structure que son équivalent dans la version 1, à la différence qu'il boucle sur la tier des phonèmes plutôt que celle des noyaux syllabiques acoustiques. Pour chaque voyelle, les mesures de F_0 et d'intensité sont faites sur une fenêtre glissante de taille paramétrable (par défaut 10 ms, inspiré de Ferrer et al., 2015) et les valeurs moyenne, minimum et maximum ainsi que l'écart type sont enregistrées. De même que pour la v1, un fichier TextGrid est généré avec les mêmes tiers que listées précédemment (*cf.* figure 10.3).

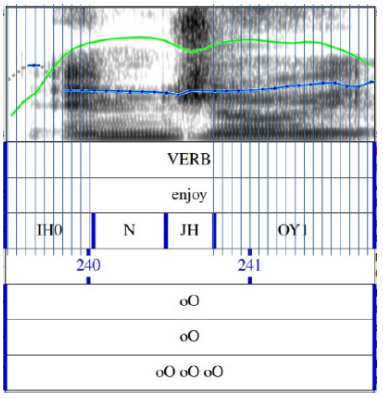

FIG. 10.3 : Extraction des paramètres prosodiques avec PLSPP v2. Les barres bleues ajoutées en surimpression représentent les frames de 10ms pour le calcul de la F0 (courbe bleue) et de l'intensité (courbe verte)

10.7 Évolution de PLSPP

PLSPP se décline aujourd'hui en 4 versions utilisées selon les besoins et le type de parole analysée :

- **PLSPP v1** est à ce jour la version la plus adaptée pour analyser la parole spontanée. Elle se base sur une identification acoustique des noyaux syllabiques et a été utilisée pour analyser les corpus du CLES (Coulange, Fries et al., 2024 ; Coulange & Kato, 2023 ; Coulange, Kato, Rossato & Masperi, 2024a, 2024b, 2024c ; Coulange et al., 2023) ;
- **PLSPP v2** se base sur une identification phonologique des noyaux syllabiques, les annotations de l'accent sont plus précises mais moins robustes aux disfluences de la parole et donc moins adaptée à la parole spontanée. Elle a été utilisée pour l'analyse de phrases porteuses ou de textes récités par des locuteurs japonophones, coréanophones et anglophones natifs (Kimura et al., 2024 ; Sugahara et al., 2023, 2024) ;
- **PLSPP v3** est une évolution de la v2 permettant l'analyse des mots monosyllabiques. Elle permet de mesurer le contraste accentuel entre les mots lexicaux et les mots grammaticaux, et a été utilisée sur des textes lus par des locuteurs japonophones et anglophones natifs (Nakanishi & Coulange, 2024) ;
- **PLSPP v4** est une évolution de la v3 qui intègre des mesures de qualité vocalique pour analyser le degré de réduction et de diphtongaison des voyelles, combiné avec des mesures physiologiques d'aperture de la mâchoire réalisées avec un articulographe électromagnétique (EMA, Lezcano et al., 2020). Cette version a été utilisée sur de la parole de locuteurs lusophones (Brésil) et anglophones natifs (Raso et al., 2024)).

Le diagramme 10.4 présente chaque version de PLSPP et leurs différences.

10.8 Interface de visualisation des annotations

Présenter ici l'interface de visualisation interactive des annotations : statistiques globales sur la position et la réalisation de l'accent, avec options de filtrage de la population et des mots cibles ; visualisation segment par segment des mots cibles et de leur accentuation en contexte, et visualisation segment par segment des pauses par catégorie, en contexte, et paramètres associés.

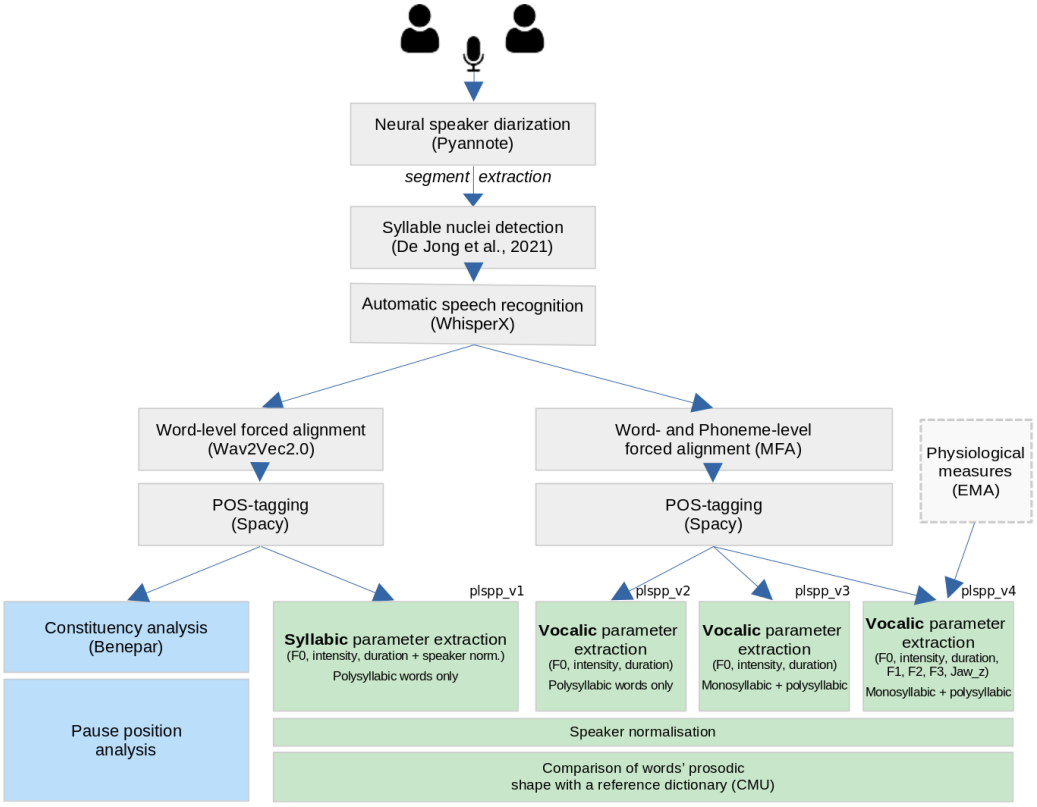

FIG. 10.4 : Architecture des 4 versions actuelles de PLSPP