Synthèse de la parole spontanée et analyse paralinguistique

Yuyan QIAN

Table des matières

1 Phonétisation et Élaboration des Ressources

1.1 Sélection des tours de parole

Audio	Durée (s)	Locuteur
T1	13, 1	A (Homme)
T3	19, 5	B (Homme)
T4	9, 3	B
T9	11, 9	B
T11	9, 8	C (Homme)

Table 1 – Cinq tours de parole sélectionnés à analyser. A, B, C sont les marqueurs de 3 locuteurs différents.

2 Extraction de la Prosodie

2.1 Extraction avec scripts

Dans cette phase du projet, pour garantir l'excellence de l'extraction prosodique, j'ai procédé à une réévaluation minutieuse de l'exactitude de l'étiquetage des phonèmes, en mettant un accent particulier sur la distinction entre les sons voisés et non voisés. Cette précision est cruciale car, sans elle, des phonèmes non voisés pourraient se voir incorrectement attribuer une fréquence fondamentale. Cette étape de vérification est essentielle pour nuancer la fréquence fondamentale au commencement du phonème, compte tenu de la subtilité de la transition entre deux phonèmes.

Parallèlement, lors de l'analyse de cinq séquences de parole, j'ai observé que certains phonèmes étaient inadéquatement représentés en termes de hauteur (pitch) sur Praat. Effectivement, une portion était mal articulée en ce qui concerne le voisement, tandis que d'autres étaient prononcés trop rapidement lors de la transition entre deux phonèmes voisés. C'est-à-dire que, dans l'annotation le début de chaque son voisé doit être un peu derrière le début de ligne de la courbe pitch.

Le script *TextGridExtractPro1.praat* a été utilisé pour extraire les valeurs de la durée et de la fréquence fondamentale moyenne pour chaque phonème voisé. Pour les phonèmes non voisés, je ne garde que la durée.

Plus avancé, le script *TextGridExtractPro2.praat* a été utilisé pour préciser les valeurs de la fréquence fondamentale au début et au milieu de chaque phonème voisé. Le script a généré des lignes qui respectent le format du fichier *pho* pour MBROLA. Par exemple, **a 85 50 116** signifie que le phonème **a** dure 85 ms, avec une fréquence fondamentale de 116 Hz avec une position de la cible au milieu.

2.2 Audio to Speech avec MBROLA - Analyse des résultats

Dans le processus de génération automatique de fichiers de phonèmes (*pho*) pour MBROLA, utilisant des extraits de voix masculines de mon corpus et la voix *fr1* pour la synthèse vocale, plusieurs éléments clés doivent être pris en compte pour assurer une qualité optimale.

Premièrement, le format d'encodage de ces fichiers *pho* doit être en UTF-8, ce qui est crucial pour la compatibilité et le traitement correct des caractères spéciaux.

En second lieu, une limitation rencontrée lors de l'utilisation de scripts sur Praat empêche la notation directe des silences par le symbole _. Cette contrainte nécessite une correction (regex) pour assurer que les silences soient correctement identifiés et intégrés dans la synthèse vocale, maintenant ainsi le rythme naturel et les pauses de la parole.

Enfin, pour les interjections ou mots marqués par de l'hésitation, qui ne présentent pas de fréquence fondamentale détectable par l'analyse dans Praat, une approche créative est adoptée. Plutôt que de les laisser sans modulation de fréquence, ce qui les ferait ressortir de manière non naturelle, une fréquence fondamentale moyenne, extraite d'autres parties du discours, est attribuée à ces phonèmes. Cette méthode permet une intégration plus harmonieuse de ces éléments dans la synthèse vocale, contribuant à un rendu final plus cohérent et naturel de la parole.

3 Génération des Durées

La génération de la prosodie est une étape cruciale dans la synthèse de la parole expressive. Pour faire parler la machine, l'expressivité de la parole synthétisée est un élément clé pour garantir la qualité. Il existe plusieurs approches dans cette phase. Ce que nous avons discuté dans la section 2 est une approche basée sur le corpus. Allons plus loin, cette extraction à partir de l'audio permets d'entraîner les modèles HMM et de générer la prosodie et le Mel cepstrum qui assure la qualité de haut niveau (La méthodologie est semblable à ce que nous avons fait dans le projet Reconnaissance de la parole avec HTK). Retournons à notre sujet de cette section, à part de l'élément le plus basique, la fréquence fondamentale, la durée est un autre facteur essentiel pour la prosodie.

Nous allons voir deux approches principales : l'approche basée sur des règles (Règle de Klatt) et l'approche statistique. La durée des pauses sera discutée dans la section suivante qui est généralement générée à partir de l'arbre syntaxique ou des structures de performance.

3.1 Règles de Klatt

Les règles Klatt sont des règles qui décrivent la durée des phonèmes en fonction de leur contexte. Ces règles sont basées sur des observations de la parole naturelle. En pratique, le calcul est assez complexe particulièrement pour les polysyllabes.

3.2 Utilisation de logiciels tiers : espeak

Le logiciel *espeak*[1] a été utilisé pour générer les durées pour les tours de parole T3, T4, T9 et T11. *espeak* est un synthétiseur de parole open source qui se base sur les formants (F1, F2, F3). Dans les ateliers passés, la parole synthétisée se montre d'une manière « robotique » et peu naturelle.

Pour les quatre autres tours de parole, j'ai généré les durées dans les fichiers d'extension *pho*. L'option –pho du synthétiseur *espeak* est généralement utilisée avec MBROLA. Bien que les voix MBROLA ne soient pas spécifiées directement dans les commandes (par exemple, *-v fr1*), si l'on envisage d'utiliser MBROLA pour traiter des fichiers .pho, on doit assurer que MBROLA est correctement installés dans un même environnement que *espeak*.

La voix *fr1* de *MBROLA* a été choisie pour générer les durées pour les tours de parole T3, T4, T9 et T11 avec *espeak*. Pour faciliter cette tâche, j'ai ajouté cette voix et j'ai donné les permissions nécessaires pour que *espeak* puisse accéder à cette voix. Voici les commandes que j'ai utilisées :

```
sudo mkdir -p /usr/share/mbrola/voices
sudo mv /home/parallels/Downloads/fr1 /usr/share/mbrola/voices/fr1
sudo chmod +r /usr/share/mbrola/voices/fr1
espeak -v mb-fr1 bonjour
```

Ensuite, j'ai utilisé la commande suivante pour générer les durées pour chaque tour de parole :

```
espeak  -q -v mb-fr1 -f ./T3.txt --pho > ./T3.pho
-q : mode silencieux, ne joue pas le son, mais génère uniquement les données.
-v mb_fr1 : sélectionne la voix de Mbrola avec une préfixe mb.
-f T3.txt : spécifie le fichier d'entrée, espeak lira le texte à partir de ce fichier
--pho : produit les données des phonèmes.
> T3.pho : redirige la sortie vers le fichier T3.pho.
```

3.3 Text to Speech avec MBROLA - Analyse des résultats

Les audios synthétisés avec MBROLA à l'aide de *espeak* ont une qualité de parole peu naturelle. À l'écoute , on peut entendre des ruptures relativement brutales et des pauses inattendues dans la parole. Dans les spectrogrammes, les transitions entre les phonèmes sont trop claires et mal articulées. La parole s'est prononcé un peu trop rapidement à manque de la liaison entre les phonèmes.

1. http ://tvaira.free.fr/bts-sn/activites/preparation-ccf-e52/activite-synthese-vocale.html

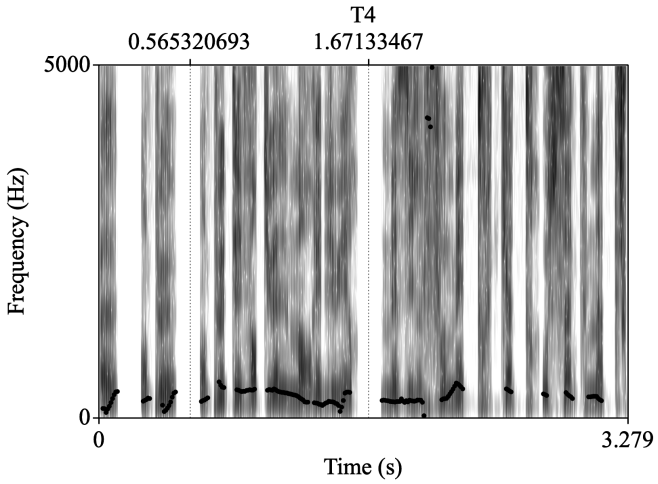

Figure 1 – Spectrogramme et pitch du segment du T4 synthétisée avec espeak et MBROLA. La phrase est « On peut éventuellement réduire. »

Au premier tour avec les règles de Klatt, la parole synthétisée est un peu plus naturelle et agréable. Les transitions entre les phonèmes sont plus douces et les pauses ajoutées durant 200 ms sont plus naturelles. La parole est globalement prononcée à un rythme plus naturel.

La limite de cette approche est qu'elle utilise un nombre de règles limité pour la durée segmentale. Ces règles ne sont pas suffisantes pour couvrir toutes les variabilités et les interactions des facteurs prosodiques sur la durée dans plusieurs contextes, en particulier en cas de forte expressivité. Par exemple, une règle se concentre sur l'allongement pour les contours intonatifs montants. Dans la pratique, l'intonation se prolonge, mais pas de manière uniforme et montante.

D'ailleurs, si l'on mentionne les règles, l'on abandonne en quelque sorte la singularité de chaque locuteur. Les caractéristiques paralinguistiques, comme l'âge, l'émotion, l'accent, etc., ne sont pas prises en compte. Dans cet aspect, espeak a presque le même problème de la parole synthétisée.

4 Génération des Pauses et de la Courbe Mélodique

4.1 Premier tour de parole : arbre syntaxique

D'arpès l'arbre syntaxique de la première phrase du Tour1, « de final, personne aurait regardé footboll, Monsieur Platini. », les groupes accentuels ont été identifiés dans la figure 2. En regroupant les groupes, 5 groupes intonatifs sont formés. (de final - personne - aurait regardé - footboll - Monsieur Platini) Après avoir identifié les groupes intonatifs, les pauses ont été ajoutées pour marquer les limites entre les groupes intonatifs. Entre les prépositions, j'ai ajouté une pause de 400 ms. Entre les syntagmes nominaux, j'ai ajouté une pause de 250 ms.

C	N	N	N	C	N	C	N
de	finale	Personne	aurait	regardé	football	monsieur	Platini
(		) (	) (		)(	) (	)

Table 2 – Formation des groupes prosodiques (C = clitique, N = non clitique)

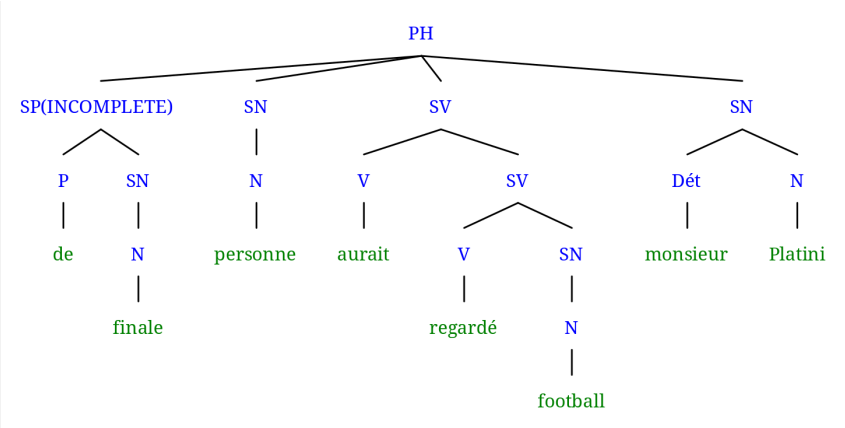

Figure 2 – Arbre syntaxique de la première phrase du Tour1« de final, personne aurait regardé footboll, Monsieur Platini. »

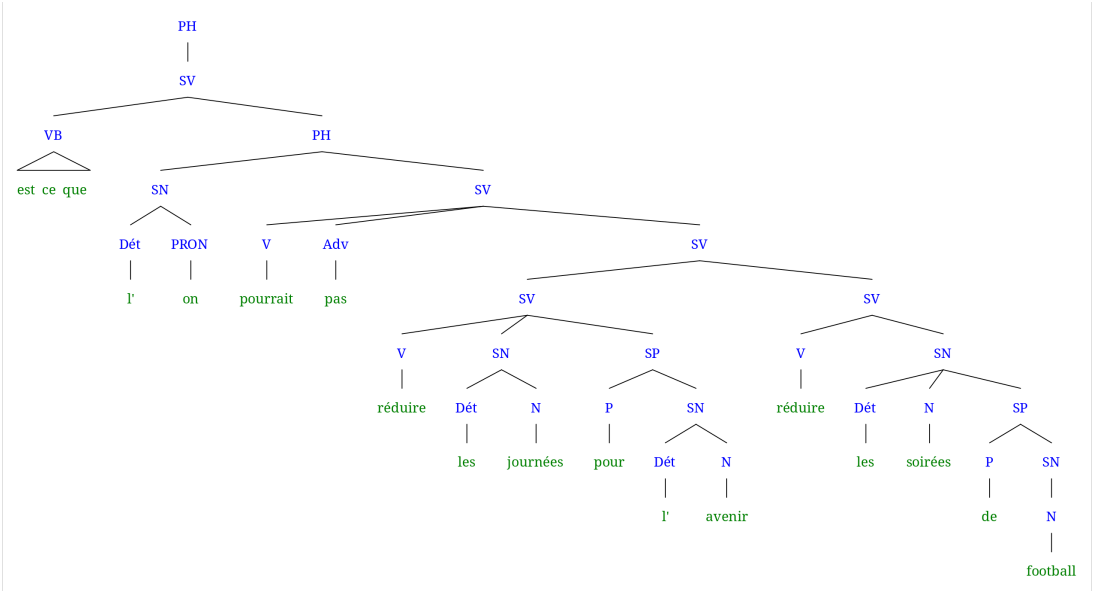

Figure 3 – Arbre syntaxique de la deuxième phrase du Tour1 « Est-ce que l'on pourrait pas réduire les journées pour l'avenir, réduire les soirées de football ? »

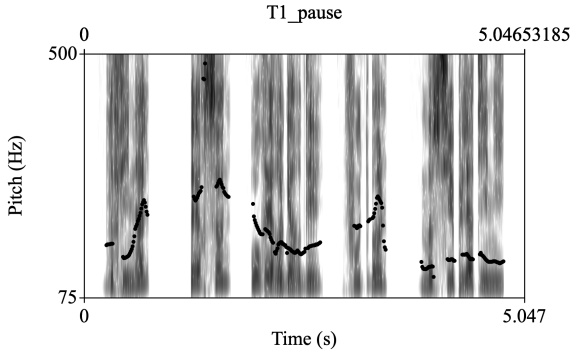

Figure 4 – Contour mélodique de la 1er phrase du Tour1

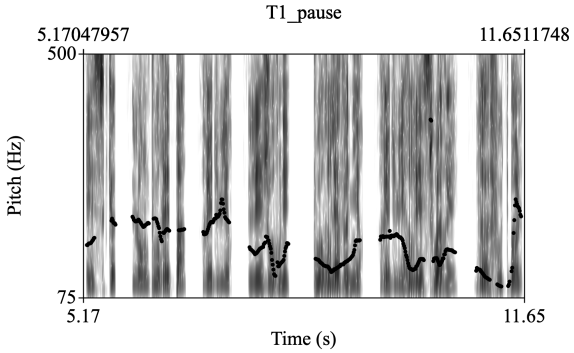

Figure 5 – Contour mélodique de la 2er phrase du Tour1

4.2 Deuxième tour de parole : structures de performance

Pour analyser des noyaux d'unité prosodique et des pauses, j'ai utilisé les structures de performance. D'abord, il faut aller de gauche à droite pour trouver les noyaux. Les mots lexiques et grammaticales seront rattachés à un noyau. Ensuite, les unités non-verbales sont regroupé au niveau supérieur.

« Non, ça se discute pas comme ça. La Coupe du Monde à vingt-quatre équipes durerait trente-un jours, à trente-deux équipes, elles durent trente-trois jours. » Cette phrase de Tour 3 contient plusieurs propositions et éléments.

```
De gauche à droite,  identification des noms, verbes, adjectifs postposés,
et prépositions.
Noms : Coupe du Monde, équipes, jours.
Verbes : se discute, durerait, durent.
Adjectifs : vingt-quatre, trente-un, trente-deux, trente-trois.
Prépositions : à, à.
```

La Coupe du Monde est une unité prosodique centrée autour du nom Coupe. Vingt-quatre équipes et trente-deux équipes sont des unités centrées autour du nom équipes. Trente un jours et trente-trois jours sont des unités centrées autour du nom jours. Les adjectifs qualificatifs *vingt-quatre, trente-un, trente-deux, trente-trois* sont rattachés aux noms équipes et jours. L'articles *la* est rattaché à *Coupe du Monde* et les prépositions *à* sont rattachées aux unités *vingt-quatre équipes et trente-deux équipes*.

Figure 6 – Structures de performance de la phrase 2 du Tour3

4.3 3er à 5er tour de parole

La prosodie des trois dernières tours doit exprimer une émotion spécifique (joie, colère, tristesse) pour chacun. Le logiciel *Emofilt* a été utilisé pour modifier les fichiers avec extension pour ces trois tours de parole.

5 Analyse Paralinguistique

5.1 Adaptation du logiciel à MacOS : pas de solution trouvée

Pour adapter le logiciel *auDeep* à MacOS arm64 M2, j'ai essayé plusieurs méthodes, de la machine virtuelle à Docker, en passant par la compilation de la source. J'ai finalement passé à Docker qui ne faisait pas fonctionner le logiciel *auDeep*.

L'installation du logiciel *auDeep* sur la machine virtuelle a échoué à cause de l'architecture arm64 de Mon PC. Plus précisément, il n'y a pas de fichier wheel de tensorflow supérieur à la version 1.15.2 et inférieurs à la version 2.x dans l'architecture arm64 (MacOS). J'ai trouvé une seul fichier wheel *tensorflow-1.15.5-cp37-cp37m-linux_ aarch64.whl*[2] qui est compatible avec l'architecture arm64 dans la machine virtuelle Linux sur mon PC, toutefois il a généré plusieurs conflits de dépendances (numpy particulièrement dans ce cas).

Ensuite, compiler le code source a été bloqué par les outils de build. Pour créer TensorFlow, il faut d'abord installer Bazel[3]. Je n'ai pas trouvé la version correcte de Bazel

2. https ://github.com/noahzhy/tf-aarch64
3. https ://www.tensorflow.org/install/source ?

pour TensorFlow compatible.

Enfin, j'ai passé à Docker avec QEMU émulation. J'ai testé un fichier *Dockerfile* basique pour construire une image Docker avec le logiciel *auDeep*. Dans cette première étape, python3.7 et tensorflow==1.15.2 ont été demandés à installer. Après avoir réussi, j'ai complété les demandes de l'environnement dans l'image Docker et j'ai ajouté le code source de *auDeep* dans l'image. Cette image a été construite et un conteneur a été lancé avec cette image. J'ai réussi à lancer le logiciel *auDeep* dans le conteneur. Cependant, *Cannot convert a symbolic Tensor (model/encoder/initial_states_tuple/strided_slice :0) to a numpy array.* a été retourné. J'ai essayé de résoudre ce problème en vérifiant le code, mais sans succès.

5.2 Analyse des données d'apprentissage

Pour comprendre les états paralinguistiques extraits avec le logiciel, j'ai essayé de résumer un peu le contexte. Les challenges paralinguistiques sont liés aux caractéristiques personnelles du locuteur, qui se regroupe en 3 sub-challenges : personnalité (OCEAN cinq personnalités avec classification binaire), ressemblance(classification binaire) et Pathologie. Le corpus qu'on va utiliser dans cet atelier est un extrait du *Speaker Personnality Corpus (SPC)* qui se concentre sur la sub-tâche de « Agreeableness » [4]. Les features dans la classification viennent de diverses catégories dont une partie est basique et une autre partie est developpée avec différentes méthodes.

Dans le cadre du défi INTERSPEECH 2012 sur les traits de voix des locuteurs, l'ensemble de caractéristiques de base englobe une large gamme de caractéristiques acoustiques pour reconnaître automatiquement des traits de locuteurs tels que la personnalité, la ressemblance et les conditions pathologiques. Ces caractéristiques peuvent être résumées en plusieurs catégories : Descripteurs de bas niveau (LLDs) qui sont des caractéristiques de base extraites directement du signal audio, incluant l'énergie, les caractéristiques spectrales et vocales, etc. Ils capturent les propriétés physiques et psychoacoustiques fondamentales du son ; Descripteurs fonctionnels qui sont dérivées des LLDs par le calcul de valeurs statistiques (par exemple, moyenne, écart-type) ou d'autres quantités dérivées (par exemple, distances entre les pics), ces caractéristiques de niveau supérieur visent à capturer la variation et la dynamique du signal audio sur de plus longues périodes.

L'ensemble de caractéristiques global comprend environ 6 125 caractéristiques. Cet ensemble de caractéristiques est conçu pour soutenir l'évaluation automatique de la personnalité des locuteurs, la ressemblance des voix et les états pathologiques, malgré les défis posés par leur complexité et leur sensibilité aux nuances subtiles. L'approche de base emploie une stratégie d'extraction de caractéristiques et de classification directe destinée à assurer la cohérence entre les sous-défis, mais ne s'optimise pas spécifiquement pour les exigences uniques ou les particularités de chaque sous-tâche. Par conséquent, explorer des méthodes plus personnalisées et spécifiques aux tâches, ainsi que tirer parti des caractéristiques uniques à chaque tâche, pourrait encore améliorer les performances.

4. 256 audios pour l'apprentissage, 183 pour le développement et 201 pour le test. Parmi eux, 323 sont réellement agréables par contre 317 sont désagréables.

5.3 Détection paralinguistique avec *auDeep* et SVM

auDeep est un toolkit Python pour l'apprentissage automatique non supervisé qui utilise des encodeurs auto-récurrents pour extraire des caractéristiques à partir de données audio.

Dans ce projet, j'ai utilisé *auDeep* pour extraire des caractéristiques de Mel spectrogrammes des audios. Tous les audios d'apprentissage ont une durée de 10 secondes. Tous les spectrogrammes sont normalisés à 0 dB et ensuite quatre seuils de l'intensité (amplitude) sont testé dans ce cas pour réduire le bruit de fond : inférieur à -30 dB, inférieur à -45 dB, inférieur à -60 dB et inférieur à -75 dB. Après cette étape de filtrage, les sons faibles ne sont pas pris en compte et les extractions seront améliorées avec des caractéristiques plus claires et plus précises.

L'auto-encodeur est mis en place pour construire des représentations des données de spectrogramme. Deux couches et 256 unités par couche sont définies pour ce réseau. Pour différents seuils, on obtient des différentes représentations. L'apprentissage permet également de fusionner les représentations générées par plusieurs modèles pour avoir un ensemble de caractéristiques plus riche. Ce qui est lisible pour l'humain, ce sont les fichiers sous format *.csv* qui contiennent les features.

Un modèle de SVM est construit pour une tâche de classification binaire. Les fichiers *train, développement, test* sont identifiés avec leurs titres. Pour optimiser la performance, plusieurs niveaux de complexité sont testés et leurs matrices de confusion sont également affichées. Avec la meilleure complexité, le modèle sera ré-entraîné avec toutes les données de développement et d'apprentissage et puis faire la prédiction sur les données de test.

```
Complexity 0.000010
UAR on Devel 49.1
Confusion matrix (Devel):
['A', 'PA']
[[ 76   3]
 [102   2]]

Complexity 0.000100
UAR on Devel 54.1
Confusion matrix (Devel):
['A', 'PA']
[[68 11]
 [81 23]]

Complexity 0.001000
UAR on Devel 57.9
Confusion matrix (Devel):
['A', 'PA']
[[52 27]
 [52 52]]
```

```
  Complexity 0.010000
  UAR on Devel 57.2
  Confusion matrix (Devel):
  ['A', 'PA']
  [[47 32]
   [47 57]]

  Complexity 0.100000
  UAR on Devel 62.3
  Confusion matrix (Devel):
  ['A', 'PA']
  [[49 30]
   [39 65]]

  Complexity 1.000000
  UAR on Devel 61.8
  Confusion matrix (Devel):
  ['A', 'PA']
  [[49 30]
   [40 64]]
Optimum complexity: 0.100000, maximum UAR on Devel 62.3
```

Synthèse	**Approche**	**Tours de parole**
STS	Script 1 - F0 moyenne, durée	T1, T3, T4, T9, T11
	Script 2 - F0 initiale et centrale, durée	T1, T3, T4, T9, T11
TTS	Règles de Klatt (durée phonémique)	T1
	Espeak (par formant)	T3, T4, T9, T11
	Arbre syntaxique (pause)	T1
	Structures de performance (pause)	T3
	Emofilt (émotion)	T4, T9, T11

Table 3 – Différentes perspectives entre les types de synthèse de la parole

5.4 Analyse des tours de parole

Tous les 25 tours de parole générés sont étiquetés en état « agréable ». Dans le tableau 1, j'ai noté que tous les locuteurs d'origine sont homme. La voix choisie pour la synthèse est *fr1* de MBROLA, homme aussi. Ce que je n'ai pas mentionné dans les sections passées, c'est que les locuteurs ont l'âge environ 45 ans. Cela pourrait être une raison pour laquelle tous les tours de parole sont étiquetés en état « agréable ». Les audios manquent de

variabilité et sont trop similaires, cela pourrait également conduire à une classification unilatérale.

À l'écoute, les audios synthétisés avec MBROLA à l'aide de *espeak* ont une qualité de parole peu naturelle. Les brèves pauses entre chaque mot ne sont pas naturelles. Dans la figure de spectrogramme, on voit clairement les bandes blanches qui montrent les pauses. J'ai pensé que c'était une raison de « robotique ». Cependant, cette lacune pourrait de quelque sorte ne pas être prise en compte par l'autoencoder. Plus précisément, le filtrage de l'intensité va supprimer les « bruits de fond » et générer automatiquement les bandes blanches dans le spectrogramme avec des seuils d'amplitude. Dans ce cas, lorsque l'on analyse les caractéristiques extraites, l'articulation de l'amplitude faible ne sera pas un facteur essentiel. Si le modèle se concentre principalement sur les parties plus saillantes du signal, telles que celles préservées après le filtrage, alors que les subtilités qui rendent les tours de parole plus naturels sont perdues, qui influent par conséquent sur la classification.

6 Résultats et Discussion

Dans la vue d'ensemble, les résultats de la synthèse de la parole sont satisfaisants. La sélection des tours de parole est bien équilibrée vu les locuteurs. Il est crucial de garantir la diversité des locuteurs pour une analyse paralinguistique. De ce fait, il vaut mieux d'avoir des locuteurs de différents âges et genres. L'extraction de la prosodie se réalise avec les scripts de Praat. Cependant, l'intensité n'est pas prise en compte dans cette phase. Si l'on ajoute cette caractéristique, il est possible que les résultats de la classification soient diversifiée. En total, la synthèse à partir de l'audio fonctionne beaucoup mieux que celle à partir du texte.

Les méthodes TTS sont variées et bien choisies. L'expressivité ne se limite pas à la prosodie, mais aussi à la durée et aux pauses. Avec les règles de Klatt, on a plus de fluidité mais aussi beaucoup plus d'efforts. L'approche de *espeak* est rapide et facile, mais la durée polémique de la parole synthétisée est moins naturelle. L'arbre syntaxique et les structures de performance sont des méthodes plus avancées pour capturer les nuances expressives. L'émotion est également prise en compte dans la synthèse de la parole.

En conclusion, la synthèse de la parole est un domaine complexe qui nécessite une compréhension approfondie de la parole humaine. Les caractéristiques paralinguistiques s'entend aux différentes notions qui concernent la qualité de la parole synthétisée. Pour avoir une synthèse plus expressive, toutes les étapes de la synthèse doivent être prises en compte avec soin.

Références

[1] Dang Khoa Mac. *Génération de parole expressive dans le cas des langues à tons*. Theses, Université de Grenoble ; Institut Polytechnique (Hanoï), June 2012.

[2] Björn Schuller, Stefan Steidl, Anton Batliner, Elmar Nöth, Alessandro Vinciarelli, Felix Burkhardt, Rob Van Son, Felix Weninger, Florian Eyben, Tobias Bocklet, et al. The interspeech 2012 speaker trait challenge. In *INTERSPEECH 2012, Portland, OR, USA*, 2012.

[3] Ali Lewandowski and Amanda I Gillespie. The relationship between voice and breathing in the assessment and treatment of voice disorders. *Perspectives of the ASHA Special Interest Groups*, 1(3) :94–104, 2016.

[4] Daniela Beltrami, Laura Calzà, Gloria Gagliardi, Enrico Ghidoni, Norina Marcello, Rema Rossini Favretti, and Fabio Tamburini. Automatic identification of mild cognitive impairment through the analysis of Italian spontaneous speech productions. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2086–2093, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[5] Fabrice Malfrère, Thierry Dutoit, and Piet Mertens. Un générateur de prosodie "tout automatique". 06 1998.

[6] Michael Freitag, Shahin Amiriparian, Sergey Pugachevskiy, Nicholas Cummins, and Björn W. Schuller. audeep : Unsupervised learning of representations from audio with deep recurrent neural networks. *CoRR*, abs/1712.04382, 2017.