Chef d'oeuvre : Identifier des marqueurs langagiers de démence

Neang Dany (IAI) Tran Tuan Vu (IAI) Verrier Thomas (IAI) Fouilhé Guilhem (IMA)

29 octobre 2025

Encadrants projet : Braud Chloé et Muller Philippe
Encadrante UE : Benamara-Zitoune Farah

Table des matières

1 Introduction

1.1 Objectifs et lien avec l'étude bibliographique

Nous avons pour but de réaliser un modèle permettant de détecter la présence d'alzheimer chez un patient en ayant pour donnée un enregistrement audio de la personne décrivant une image.

L'objectif principal n'étant pas de faire compétition ou de rivaliser avec les différents participants de l'ADReSSo challenge [Luz+21], mais surtout d'explorer ce que nous pouvons réaliser grâce aux anciens travaux réalisés et aux différents outils d'aujourd'hui sans forcément chercher à avoir le modèle le plus performant. Le plus important est de déterminer les potentielles variables qui jouent un rôle important sur la tâche donnée, à l'aide de différentes expériences.

Nous voulons dans un premier temps comparer différentes sorties d'ASR sur nos données audio, puis souligner les défauts et les qualités potentiels de chacun pour finalement en utiliser certaines sur l'ensemble de nos audio et ainsi générer nos *datasets*.

Par la suite nous avons pour objectif d'atteindre les performances de la *baseline* [Luz+21] dont l'*accuracy* sur le test de leur modèle a atteint les 79%. Nous voulons utiliser BERT comme modèle pré-entraîné et le fine-tuner sur notre tâche afin de récolter les différentes performances de notre modèle suivant nos différents *datasets*.

Différents chemins s'offrent à nous afin d'améliorer, ou trouver des pistes d'améliorations, les performances de notre modèle. Nous voulons explorer l'ajout de certaines *features* à notre modèle et ainsi s'essayer au multimodale avec l'extraction de *features* linguistiques et acoustiques sur nos données.

2 Données à disposition

Nous avons les 166 fichiers audio mis à disposition par le challenge, du dataset train : soit 87 fichiers audio étiquetés AD (*Alzheimer Disease*) et 79 fichiers audio étiquetés CN (*Cognitively Normal*). D'après le challenge, ces catégories sont équilibrées en genre et en âge afin d'éviter tout biais basés sur la composition du *dataset*.

Cependant le *dataset test* du challenge composé de 71 audio nous est fourni non étiqueté. Nous avons donc décidé d'évaluer nos différents modèles sur une proportion de 25% du *dataset train* à disposition. Cela nous ampute d'une certaine partie des données pour entraîner nos modèles mais nous aurons tout de même des métriques de performances qui seront comparables entres nos modèles. De ce fait les performances présentés ne seront pas totalement comparables à ceux des travaux soumis lors du challenge ADReSSo.

3 Travail réalisé

3.1 Recherche d'outils d'ASR adaptés

En premier travail nous avons testé plusieurs ASR dont DeepSpeech, Whisper et WhisperX [Han+14][Pan+15][Bai+23].

3.1.1 DeepSpeech

DeepSeech[Han+14], le modèle d'ASR de Mozilla, a été testé ici dans le but de réaliser la transcriptions des fichiers audio. Ce modèle était cité dans des travaux soumis pour le challenge [Che+21]. L'outil est libre d'accès et gratuit, nous avons trouvé intéressant de l'implémenter et l'éprouver face à nos données. La dernière version disponible et utilisée dans nos travaux est la 0.9.3 publiée en décembre 2020. Dans un premier temps il faut échantillonner les données à la bonne fréquence pour DeepSpeech. Originalement d'une fréquence d'échantillonage de 44.1 kHz, ils sont passées en 16 kHz.

Après une analyse manuelle des transcriptions, nous avons conclu que le modèle n'était pas satisfaisant. En effet la sortie ne possède pas de ponctuation et les résultats en terme d'erreurs de transcription sont trop variables.

3.1.2 Whisper

Nous avons testé Whisper qui fait beaucoup parler de lui récemment. Les sorties des transcriptions de Whisper sont découpées en plusieurs parties sous forme de liste de dictionnaires. Les composantes de la liste sont centrées sur les différentes phrases détectées et transcrites par le système. On retrouve plusieurs propriétés pour chaque phrase :

— id : un identifiant unique pour chaque segment de la transcription. Cela peut être utile pour faire référence à des segments spécifiques lors de l'analyse de la sortie de l'ASR.

— seek : l'heure de début du segment dans la source audio (en millisecondes).

— start : l'heure de début du segment dans la transcription texte (en millisecondes). Cela peut différer de seek si le modèle ASR a besoin de quelques secondes pour traiter l'audio avant de commencer à produire la transcription.

— end : l'heure de fin du segment dans la transcription texte.

— text : le texte transcrit pour ce segment.

— token : une liste de jetons (mots ou sous-mots) associés au texte transcrit. Les modèles ASR peuvent diviser le texte en sous-mots pour améliorer la précision.

— temperature : la température utilisée pour générer cette transcription. Cela fait référence à une technique d'échantillonnage stochastique utilisée par certains modèles ASR pour générer des transcriptions plus diversifiées.

— avg_logprob : la probabilité logarithmique moyenne de ce segment. Cela peut être utilisé pour évaluer la qualité de la transcription.

— compression_ratio : le rapport de compression appliqué à l'audio avant le traitement par l'ASR. Cela peut être utilisé pour comparer la qualité de la transcription pour différents niveaux de compression audio.

— no_speech_prob : la probabilité que ce segment ne contienne pas de parole. Cela peut être utilisé pour détecter les segments silencieux ou les segments où le modèle ASR n'a pas détecté de parole.

```
[{'start': 0.0,
  'end': 4.0,
  'text': " Tell me everything that's going on.",
  'tokens': [14026, 502, 2279, 326, 338, 1016, 319, 13],
  'avg_logprob': -0.6751615084134616,
  'no_speech_prob': 0.046610601246356964},
 {'start': 4.0,
  'end': 16.0,
  'text': ' Well, I guess there are four people in the corner.',
  'tokens': [3894, 11, 314, 4724, 612, 389, 1440, 661, 287, 262, 5228, 13],
  'avg_logprob': -0.6751615084134616,
  'no_speech_prob': 0.046610601246356964},
 {'start': 16.0,
  'end': 30.0,
  'text': ' We are grading and they are going to get some cookies from the cookie jar.',
  'tokens': [775,
   389,
   43165,
   290,
   484,
   389,
   1016,
   284,
   651,
   617,
   14746,
   422,
   262,
   19751,
   17379,
   13],
  'avg_logprob': -0.3991618394851685,
  'no_speech_prob': 1.171174426417565e-05},
 {'start': 30.0,
```


FIGURE 1 – Exemple de transcription avec Whisper de l'audio adrso025.wav

Après avoir comparé manuellement l'audio avec sa transcription, nous avons constaté que même si la température fixée est à 0, un phénomène d'hallucination du système est pourtant présent. Celui-ci, après un certain temps de silence, ajoute plusieurs mots estimés très plausibles. Dans l'image ci-dessus 1 le deuxième segment n'a pas été prononcé par le locuteur. Cela nous amène à questionner la pertinence de Whisper pour la tâche à réaliser.

Effectivement Les segments de silence peuvent jouer un rôle crucial dans la détection de la maladie d'Alzheimer à partir d'un texte [Sye+21]. La maladie d'Alzheimer peut affecter la capacité d'une personne à communiquer de manière cohérente, et cela peut se manifester sous la forme de pauses prolongées ou de segments de silence dans leur discours. Les segments de silence peuvent également être utilisés comme des indicateurs de progression de la maladie, car les patients atteints d'Alzheimer ont tendance à avoir des périodes de silence plus longues à mesure que leur maladie s'aggrave.

La sortie de transcription par Whisper n'est pas alignée sur les mots, ce qui rend impossible la détection de ces segments silencieux.

3.1.3 Whisperx

Nous avons recherché une alternative à Whisper et nous avons trouvé Whisperx. Whisperx est une variante considérée comme une version plus performante de Whisper. Elle reprend le modèle de Whisper et propose une transcription au niveau des mots (*word alignment*). De plus son paramètre de température est fonctionnel, nous n'avons pas constaté de phénomène d'hallucination. Grâce à Whisperx nous pouvons alors extraire ce que nous cherchons dans une transcription. A l'aide du *word alignment*, nous pouvons localiser et calculer les temps de silences dans une transcription.

```
[{'text': 'Tell', 'start': 0.7822349570200574, 'end': 0.9426934097421205},
 {'text': 'me', 'start': 0.9627507163323783, 'end': 1.0630372492836677},
 {'text': 'everything',
  'start': 1.1232091690544412,
  'end': 1.4641833810888254},
 {'text': "that's", 'start': 1.504297994269341, 'end': 1.70487106017192},
 {'text': 'going', 'start': 1.7449856733524356, 'end': 1.9255014326647566},
 {'text': 'on.', 'start': 1.9455587392550144, 'end': 1.9856733524355303},
 {'text': 'Well,', 'start': 8.000953288846521, 'end': 8.121067683508103},
 {'text': 'the', 'start': 8.801715919923737, 'end': 8.90181124880839},
 {'text': 'kids', 'start': 8.96186844613918, 'end': 9.182078169685415},
 {'text': 'that', 'start': 9.242135367016207, 'end': 9.46234509056244},
 {'text': 'are', 'start': 9.602478551000953, 'end': 9.722592945662536},
 {'text': 'working', 'start': 11.784556720686368, 'end': 13.606291706387037},
 {'text': 'on', 'start': 14.80743565300286, 'end': 14.967588179218303},
 {'text': 'the', 'start': 16.04861773117255, 'end': 16.26882745471878},
 {'text': 'corner,', 'start': 16.348903717826502, 'end': 17.32983794089609},
 {'text': 'they', 'start': 17.40991420400381, 'end': 17.530028598665396},
 {'text': 'are', 'start': 17.59008579599619, 'end': 17.710200190657773},
 {'text': 'going', 'start': 17.75023832221163, 'end': 18.030505243088655},
 {'text': 'to', 'start': 18.13060057197331, 'end': 18.33079122974261 2},
 {'text': 'get', 'start': 18.490943755958057, 'end': 18.811248808388942},
 {'text': 'some', 'start': 18.871306005719735, 'end': 18.951382268827455},
 {'text': 'cookies', 'start': 20.693040991420403, 'end': 21.193517635843662},
 {'text': 'from', 'start': 21.714013346043853, 'end': 22.274547187797904},
 {'text': 'the', 'start': 25.55767397521449, 'end': 25.717826501429936},
 {'text': 'And', 'start': 28.100125156445557, 'end': 28.7008760951189},
 {'text': 'the', 'start': 28.84105131414268, 'end': 29.26157697121402},
 {'text': 'mother', 'start': 32.926157697121404, 'end': 33.36670838548185},
 {'text': "doesn't", 'start': 33.50688360450563, 'end': 34.04755944931164},
 {'text': 'see', 'start': 35.289111389236545, 'end': 35.529411764705884},
 {'text': 'it', 'start': 35.56946182728411, 'end': 35.66958698372966},
 {'text': 'because', 'start': 35.729662077597, 'end': 36.13016270337923},
 {'text': "she's", 'start': 36.19023779724656, 'end': 36.45056320400501},
 {'text': 'inside', 'start': 36.530663329161456, 'end': 37.07133917396746},
 {'text': 'of', 'start': 37.11138923654568, 'end': 37.15143929912391},
 {'text': 'drying', 'start': 40.85607008760951, 'end': 41.29662077596996},
 {'text': 'the', 'start': 41.33667083854819, 'end': 41.436795994993744},
 {'text': 'clothes.', 'start': 41.516896120150186, 'end': 42.01752190237797},
 {'text': 'Think', 'start': 44.02003091190108, 'end': 44.20030911901082},
 {'text': 'about', 'start': 44.2203400309119, 'end': 46.62404945904173},
 {'text': 'it,', 'start': 46.784296754250384, 'end': 46.844389489953635},
```


FIGURE 2 – Exemple de transcription avec Whisperx de l'audio adrso025.wav

3.2 Réalisation de scripts pour extraire les transcriptions

3.2.1 Augmentation des données

Nous savons que BERT encode un texte avec un vecteur de taille 768. Celle-ci étant limitée, nous avons eu pour idée de segmenter les données. Cette technique nous permet aussi d'augmenter considérablement la quantité de données disponibles pour le futur entraînement, ce qui peut améliorer les performances de notre modèle d'apprentissage automatique. Bien que la longueur des segments audio peut se discuter nous avons choisi de générer des segments de 30 secondes, ce qui nous amène à 513 audio. Nous gardons les audio de longueur originale afin de pouvoir constater ou non l'impact de la segmentation d'audio au niveau transcription comme au niveau des performances de notre modèle.

3.2.2 Génération des transcriptions

Bien qu'ayant tout d'abord essayer de transcrire avec le modèle large de Whisper, nous avons vite rebroussé chemin car le temps d'exécution prenait un temps colossale (à peu près 1h par audio). Nous avons donc utiliser Whisper et WhisperX avec leur modèle de langage*medium* entraîné avec 769 millions de paramètres pour transcrire nos deux types d'audio. Avant le processus des transcriptions, nous avons élaboré la structure de nos données : chaque transcriptions seront composées d'un dictionnaire comportant deux clés :

— une clé "text" associée à l'ensemble du texte généré par la transcription.

— une clé ”label” pour étiqueter nos transcriptions avec :
  — ”1” : pour AD
  — ”0” : pour CN

3.2.3 Temps silencieux (*whitespaces*)

Dans les transcriptions des audio générés par whisperX, nous avons ajouté dans le champ ”text” les expressions des temps silencieux (WS) en s'inspirant du travaille de Syed [Sye+21]. Nous avons ajouté les tokens suivant :

— ”.” : pour un silence entre 2 et 4 secondes.
— ”..” : pour un silence entre 4 et 6 secondes.
— ”long silence” : pour un silence de plus de 6 secondes.

L'ajout des ces tokens sont insérés entre les mots en calculant simplement le temps entre la fin d'un mot et le début du prochain mot.


```
{"text": "What's going on in the picture?
In here? Look at this. This goes away. .
I really don't know because I haven't
gotten things turned up. Right. .. Have a
look at the picture. See if you can tell
me what's going on. .. Well, this one. I
think it's right here. What is it?
It's... ... my girl. But it isn't. long
silence This is the kind of... ... thing
that... ... whoever... ... really... ..
What is it that's happening here? What's
happening here? She's cleaning. long
silence And what are they doing? . I'm
looking to see what they have so they can
get it ready for bed, I guess. I don't
know. tears down it's falling",
"label": 1}
```


FIGURE 3 – Exemple d'une donnée générée avec Whisperx et de l'ajout des token WS de l'audio adrso033.wav

3.3 Statistiques sur le nombre de tokens par enregistrements

Pour mieux comprendre l'impact des segmentations et comparer les deux classes nous avons réalisé des statistiques sur les transcriptions obtenues avec les outils d'ASR WhisperX et Whisper.

FIGURE 4 – Histogrammes de la distribution du nombre de tokens par enregistrement pour WhisperX.

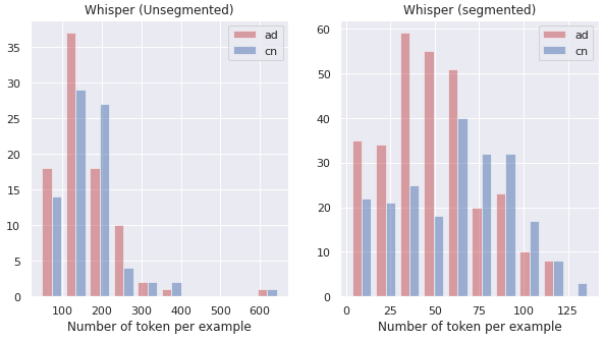

FIGURE 5 – Histogrammes de la distribution du nombre de tokens par enregistrement pour Whisper.

ASR Model	**Mean (ad)**	**Mean (cn)**
WhisperX	163.586	165.886
WhisperX (seg)	50.000	61.294
WhisperX (seg + ws)	50.420	60.583
WhisperX (ws)	172.034	169.354
Whisper	161.195	166.165
Whisper (seg)	49.088	61.078

TABLE 1 – Moyenne du nombre de token par enregistrement

On remarque que les distributions de la classe AD sont davantage centrés vers la gauche, c'est à dire que ces individus s'expriment en moyenne plus lentement.

3.4 Modèle Baseline

Nous avons utilisé le modèle préentraîné bert-en-uncased de BERT. L'architecture de notre premier modèle est très simple avec une première couche de 786 paramètres utilisant BERT connecté à une couche de dropout puis à une couche dense de taille 1 avec la fonction d'activation sigmoïd pour avoir en sortie une valeur entre 0 et 1 pour notre classification binaire. La couche de *dropout* sert à désactiver différents neurones lors de l'apprentissage afin de limiter le sur-apprentissage (*overfit*) de notre modèle. De ce fait, notre modèle prend en entrée un texte et retourne une valeur comprise 0 et 1. Notre modèle atteint une *accuracy* de 78% au test ce qui est proche de la *baseline* du challenge.

3.5 Expériences sur la tâche de classification

3.5.1 Les différents datasets

Lors de la partie précédente nous avons évoqué la génération de plusieurs types de données. Ces différents types de données forment nos différents datasets :

— whisper : Dataset provenant des transcriptions des audio dans leur taille originale par whisper.

— whisper_seg : Dataset provenant des transcriptions des audio segmentés de longueur 30 secondes et moins par whisper.

— whisperX : Dataset provenant des transcriptions des audio dans leur taille originale par whisperX.

— whisperX_seg : Dataset provenant des transcriptions des audio segmentés de longueur 30 secondes et moins par whisperX.

— whisperX_seg_ws : Dataset provenant des transcriptions des audio segmentés de longueur 30 secondes et moins par whisperX avec l'ajout des tokens de temps de silence (*whitespaces*)

— whisperX_ws : Dataset provenant des transcriptions des audio dans leur taille originale par whisperX avec l'ajout des tokens de temps de silence.

— whisper_debut : Dataset provenant des transcriptions des premiers segments de 30 secondes de chaque audio par whisper

— whisperX_debut : Dataset provenant des transcriptions des premiers segments de 30 secondes de chaque audio par whisperX

— whisperX_ws debut : Dataset provenant des transcriptions des premiers segments de 30 secondes de chaque audio par whisperX avec l'ajout des tokens de temps de silence.

Nous avons généré plusieurs modèles qui se différencient uniquement par leur *dataset* d'entraînement (l'architecture reste la même sauf pour notre modèle multimodale).

3.5.2 Segmentation

En comparant les performances des modèles utilisant les données du *dataset* whisper et whisperX respectivement avec les performances des modèles utilisant les données du *dataset* whisper_seg et whisperX_seg, il semble que la segmentation audio contribue positivement à notre tâche. Les résultats montrent que les modèles entraînés sur les données segmentées obtiennent une meilleure performance que ceux entraînés sur les données non segmentées, ce qui suggère que la segmentation audio aide les modèles à mieux comprendre les différentes caractéristiques des textes transcrits avec ces audio. Il est probable que BERT soit en mesure de mieux distinguer les caractéristiques des textes d'une certaine longueur ou du moins n'excédant pas une certaine taille.

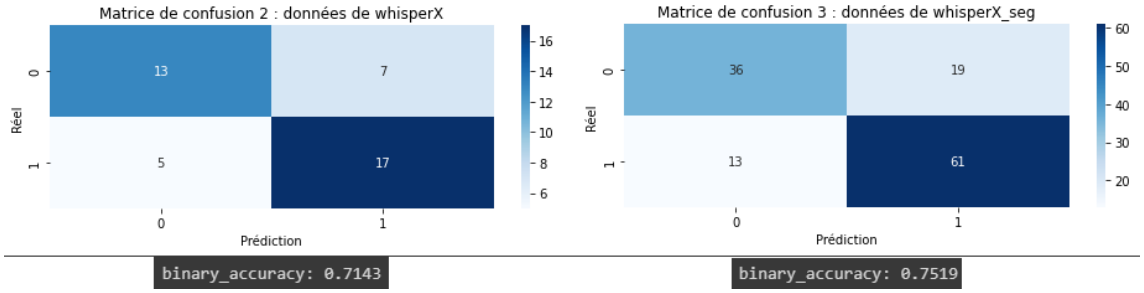

FIGURE 6 – Comparaison des Matrices de confusion et accuracy sur le test du modèle utilisant les données de whisperX et du modèle utilisant les données de whisperX_seg

3.5.3 Les Whitepaces

On remarque un gain de 2% d'*accuracy* avec l'ajout des tokens WS 3.5.3. Ce résultat nous laissent interpréter que ces représentations de silences directement insérées dans le texte, influencent favorablement la capacité du modèle à traiter notre tâche de classification.

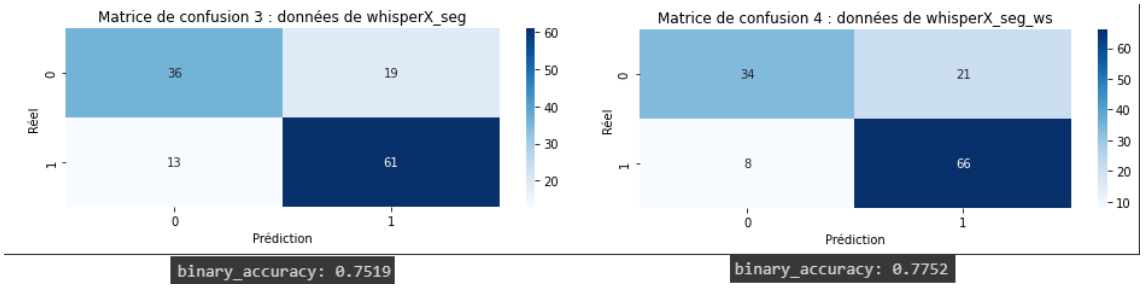

FIGURE 7 – Comparaison des Matrices de confusion et *accuracy* sur le test du modèle utilisant les données de whisperX_seg et du modèle utilisant les données de whisperX_seg_ws (modèle *baseline*)

Cependant, avec une expérience différente ces résultats nous portent à confusion. Dans la figure ci-dessous 3.5.3, nous avons comparé l'impact des WS sur les audio non segmentés. La tendance est inversée par rapport aux résultats précédent sur la Figure 5 3.5.3. De ce fait nous ne pouvons pas tirer de conclusion sur cet aspect. On pourrait expliquer ce phénomène car nous avons gardé les derniers segments de chaque audio dont la longueur est de moins de 30 secondes. Il est donc possible que certains segments soient très courts et nuisent à l'apprentissage.

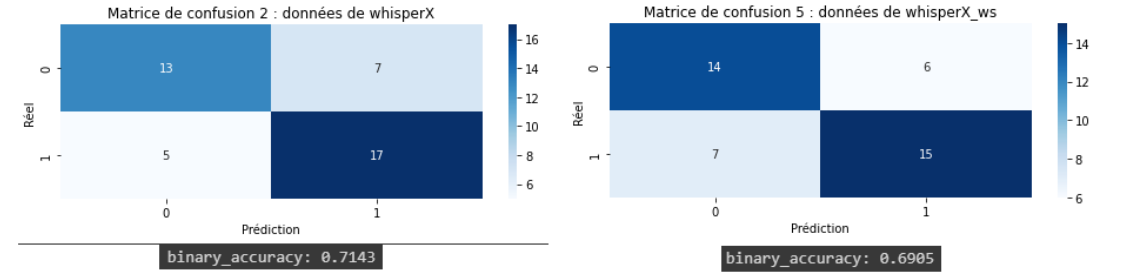

FIGURE 8 – Comparaison des Matrices de confusion et *accuracy* sur le test du modèle utilisant les données de whisperX et du modèle utilisant les données de whisperX_ws

3.5.4 Utilisation du premier segment de chaque audio

Nous avons essayé d'utiliser seulement les premiers segments de 30 secondes de chaque audio, afin de savoir si ces premières informations sont suffisantes et si elles peuvent capturer les caractéristiques importantes pour notre tâche. Nous avons effectué plusieurs expérimentations sur les différentes configurations des train et des test possibles. Par la suite je nomme PS : les premiers segments de 30 secondes de chaque audio.

— train uniquement sur les transcriptions des PS, puis test sur des transcriptions des audio non-segmentés.

— train uniquement sur les transcriptions des PS, puis test sur des transcriptions des PS.

— train sur les transcriptions de l'ensemble des segments audio, puis test sur des PS.

— train sur les transcriptions des audio non segmentés, puis test sur des PS

Bien que les résultats puissent être discutables en termes d'*accuracy*, il semblerait que train sur les PS soient bénéfiques à l'entraînement du modèle et amène une performance au test sur des transcriptions des audio non-segmentés. Dans le cas avec la transcription avec whisper comme dans le cas avec whisperX ces deux entraînements sur les PS puis d'une évaluation sur les transcriptions des audio non-segmenté ont donné les meilleurs résultats jusqu'à présent.

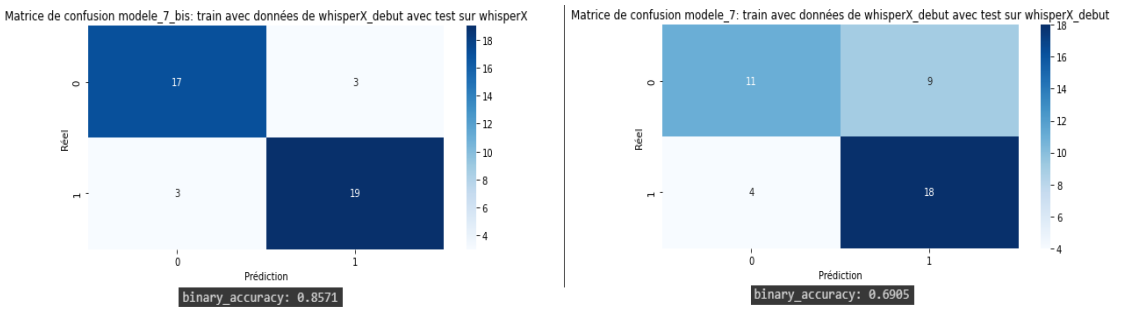

FIGURE 9 – Comparaison des Matrices de confusion et *accuracy* sur 2 évaluations au test différentes (whisperX et whisper_debut) du modèle entraîné sur whisperX_debut

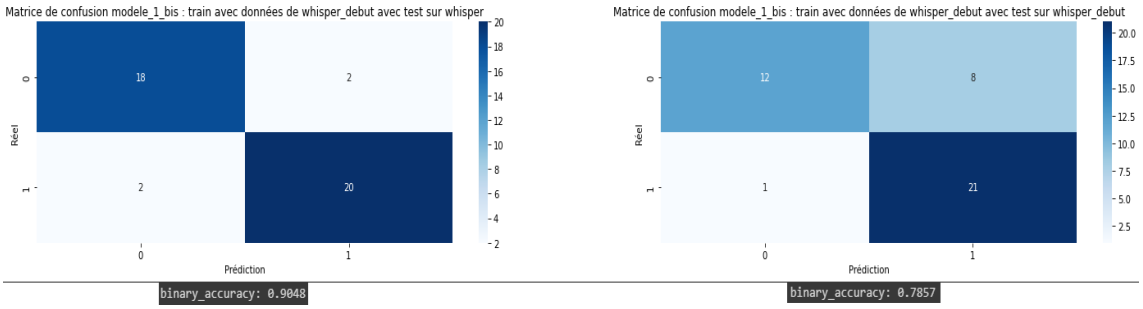

FIGURE 10 – Comparaison des Matrices de confusion et *accuracy* sur 2 évaluations au test différentes (whisper et whisper_debut) du modèle entraîné sur whisper_debut

Les deux figures 3.5.5 3.5.4 ci-dessus nous suggèrent que l'entraînement sur les PS est une bonne solution, mais que les évaluer sur des PS n'est pas une bonne méthode. Il est probable que s'entraîner sur des données plus petites améliore la phase d'apprentisage en limitant l'*overfit* sur nos modèles. En combinaison avec l'entraînement sur les PS, les évaluer ensuite sur un *dataset* de PS semble ne pas être bénéfique.

Néanmoins il est notable de souligner la performance des modèles testés et évalués sur des PS en comparaison à ceux évalués sur des transcriptions des audio non-segmentés.

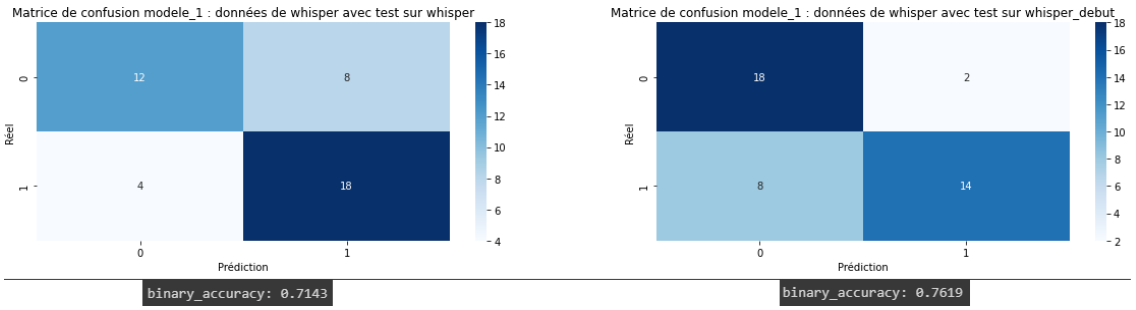

FIGURE 11 – Comparaison des Matrices de confusion et *accuracy* sur 2 évaluations au test différentes (whisper et whisper_debut) du modèle entraîné sur whisper

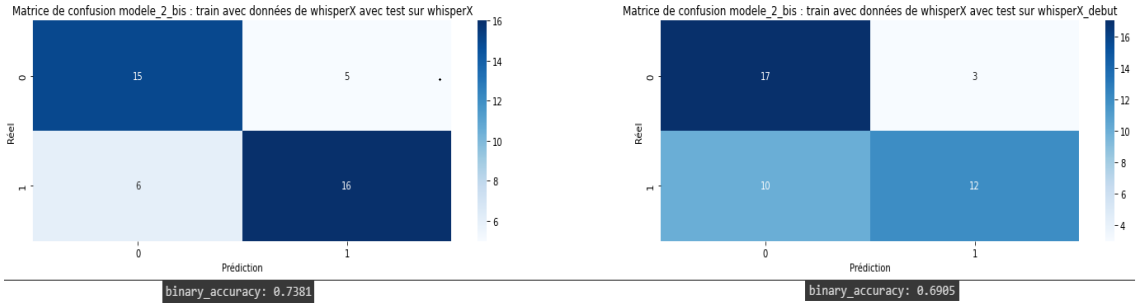

FIGURE 12 – Comparaison des Matrices de confusion et *accuracy* sur 2 évaluations au test différentes (whisperX et whisperX_debut) du modèle entraîné sur whisperX

3.5.5 Modèle Multimodal

Il est ressorti de notre étude bibliographique que les modèles les plus performants proposés par les participants au challenge ADReSSo utilisaient presque tous à la fois les modalités textuelles (basées sur les transcriptions automatiques) et audio. Nous avons exploré ce type d'architecture sur le modèle de [Che+21] en traitant séparément les deux modalités avant de fusionner les vecteurs de *features* en utilisant une régression logistique (LR).

Pour la modalité textuelle, nous avons gardé exactement le modèle *baseline*. Pour la modalité audio, nous avons utilisé les *features Extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS)* qui donnent pour chaque enregistrement un vecteur de taille 88. Ce modèle est résumé sur le diagramme 3.5.5.

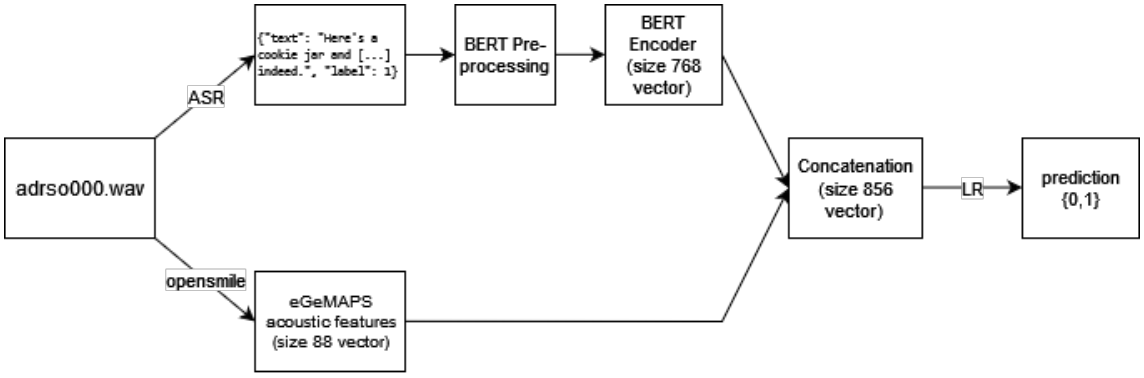

FIGURE 13 – Diagramme de notre premier modèle multimodal

Nous n'avons malheureusement pas trouvé de configuration de l'entraînement de ce réseau qui fournissait un résultat satisfaisant et avons à la place considéré une autre architecture où on ne concatène qu'un seul score qu'on obtient à partir d'une régression logistique sur les *features* acoustiques. Ce modèle est résumé sur le diagramme 3.5.5.

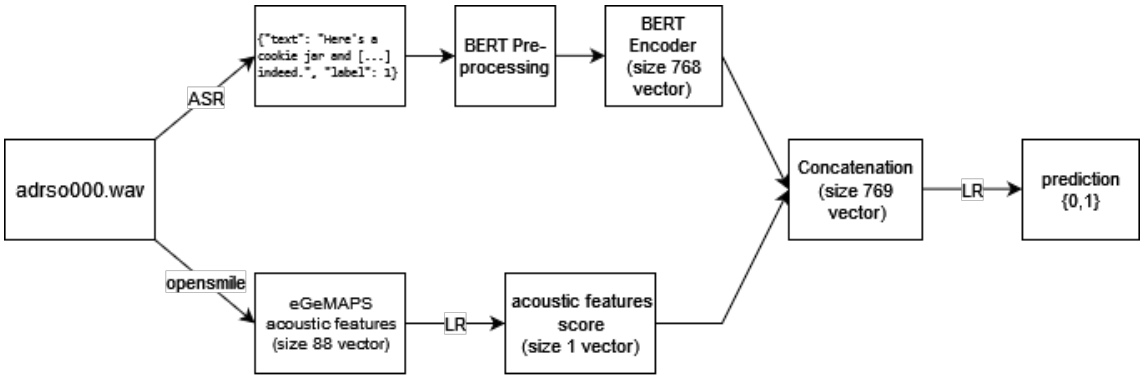

FIGURE 14 – Diagramme de notre modèle multimodal final

Ce modèle donne des résultats sensiblement plus faibles que la *baseline*, comme on peut le voir sur la matrice de confusion. 3.5.5.

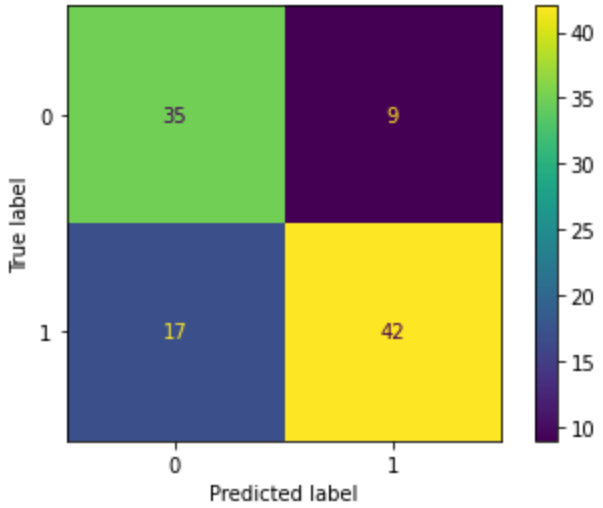

FIGURE 15 – Résultats de notre modèle multimodal sur l'ensemble de test. On obtient une précision de 0.7476

Ce résultat suggère surtout un moins bon entraînement que le modèle *baseline*. En effet il aurait suffi de fixer certains poids à 0 pour retrouver le modèle *baseline*. Il est donc impossible de tirer de conclusion, d'autant que certains articles du challenge ADReSSo comme [Che+21] montrent une amélioration nette des modèles avec l'ajout de ces *features*.

4 Environnement de travail et gestion

4.1 Organisation du travail

Le travail s'est organisé autour de différents *notebooks* sur la plateforme *Google Colab* et du stockage des données sur un drive. L'avantage de ces outils réside dans l'utilisation aisé du GPU et de la vitesse de partage des travaux effectués. Chaque membres pouvaient expérimenter en rendant visible ses résultats.

4.2 Gestion du projet

Pour cette deuxième partie de travail, l'objectif était d'implémenter un modèle de classification suite aux informations recueillies dans l'état de l'art précédemment réalisé.

Plusieurs étapes étaient attendues :
— transcription audio vers texte
— pré-traitement des données
— développement d'un premier système de classification (baseline)
— réalisation d'expériences avec divers datasets générés
— réalisation d'expériences avec un système multimodal
— réalisation d'expériences en ajoutant des features supplémentaires

Après l'étape de pré-traitement des données réalisée, nous avons défini un jalon pour le **26/02/2023** avec pour objectif d'atteindre la *baseline* au niveau du modèle de classification. L'objectif a été réalisé et validé avec les expériences de la partie 3.4.

A partir de cette date et jusqu'au dernier jalon de rendu fixé pour le 19/03/2023, l'objectif a été d'essayer d'améliorer notre modèle et d'expérimenter de nouvelles pistes.

Objectif	**Date fixée**
Transcription et pré-traitement	01/02/2023
Baseline	26/02/2023
Rédaction rapport	19/03/2023
Oral	24/03/2023

TABLE 2 – Dates et objectifs des différents jalons

A chaque nouvelles étapes du projet nous faisions des réunions avec notre encadrante pour décider de la suite du projet en nous donnant des idées d'expérimentations.

5 Difficultés rencontrées

5.1 Temps de calcul

La phase de génération des données était assez fastidieuse. La transcription de l'ensemble des audio prenait un temps conséquent (environs 10 heures avec un GPU). De ce fait la moindre erreur d'exécution était une perte importante de temps. Il nous est arrivé d'être confronté à des segments d'audio ne pouvant être transcrits car WhisperX ne prenait pas en compte les audio de petite longueur. Nous les appelons nos *left-over* car les audio ne pouvaient être divisés parfaitement en segments de 30 secondes, ce qui nous laissait avec des audio de moins de 30 secondes. Nous avons tout de même réussi à transcrire ces *left-over* grâce à une version antérieure de WhisperX qui permettait cette exécution.

Il en était de même pour la partie d'entraînement de nos différents modèles. Nous avons mis beaucoup de temps à faire entraîner correctement notre modèle multimodale, en essayant des changements de couches, des insertions de couches puis en utilisant les stratégies de late fusion et l'early fusion [Hua+20]. Nous n'avons donc pas pu accomplir l'ensemble des objectifs fixés faute de ressources manquantes.

5.2 Interprétations des résultats

Nous nous sommes beaucoup de fois confronté à des problèmes d'interprétations, notamment avec toutes les comparaisons. Il était souvent impossible de tirer des conclusions car nos hypothèses

ne sont pas vérifiées par nos résultats. D'autres paramètres entrent en compte lorsque l'on fait de l'apprentissage supervisé. Un défaut de notre processus d'apprentissage est que notre modèle se heurtait à des problème de sur-apprentissage (*overfit*). Ce problème pourrait s'expliquer sur le manque de données dans le processus d'apprentissage comme le manque de données représentatives pour la phase d'évaluation.

6 Bilan

6.1 Axes d'améliorations

Nombreux de nos choix sont discutables, c'est pourquoi nous proposons dans cette partie des chemins différents ainsi que de potentiels éléments à intégrer au projet dans le futur.

6.1.1 Diarization

Whisperx est en cours d'amélioration sur divers aspects. Il a comme prochain but d'ajouter une fonctionnalité de *diarization*. La *diarization* sert à répondre aux questions de qui parle et quand, afin de différencier les segments de parole par son locuteur. Il serait pertinent de travailler sur cet aspect plus tard. En effet, dans les audio il y a de brèves interventions orales de l'examinatrice afin d'aiguiller le patient (exemple segment 1 dans la figure1 1). On pourrait constater le taux de parole des 2 locuteurs dans nos données et ainsi analyser son importance dans la détection de la maladie lorsque l'on isole la parole du patient.

6.1.2 Choix de la segmentation

Pour générer plus de données nous avons choisi de segmenter l'audio, mais nous n'avons pas explorer une autre possibilité. Nous aurions tout aussi bien pu fractionner, après transcription, le texte en un nombre fixé de tokens. Il serait pertinent d'explorer cette piste pour constater les changements tant bien au niveau performance du modèle d'apprentissage qu'au niveau des différentes données générées.

De plus, nous avons testé seulement la segmentation avec une longueur de 30 secondes. Il est fort probable que la taille que nous avons fixé arbitrairement ne soit pas la plus adaptée au modèle préentrainé de BERT.

6.1.3 Amélioration du modèle

Les choix de l'architecture de notre modèle multimodale qui permet de traiter les 2 type de source d'entrée (linguistiques avec BERT et acoustique avec gemaps) n'est pas optimal car le modèle que nous avons élaboré rencontre un problème d'*overfit* sur l'entraînement. Bien qu'ayant essayer beaucoup d'architecture, les ressources de calcul nous étant limités nous nous sommes contenter du modèle présent. Pour un futur projet il serait perspicace d'essayer d'autre architectures pour capturer le plein potentiel du multimodale.

L'apprentissage de nos modèles sur les données segmentées pouvait être ajusté. En effet nous avons segmenté les audio en redistribuant le label original sur tous les segments. Il aurait été judicieux de réaliser un modèle qui calcule un score global sur la donnée en calculant la moyenne sur l'ensemble de ses segments.

Pour parfaire notre modèles de classification nous aurions pu utiliser les 71 données du test, même si non étiquetées, en utilisant des techniques d'apprentissage semi-supervisé. Le modèle peut utiliser les données étiquetées pour apprendre à associer correctement les entrées aux sorties attendues, et utiliser les données non étiquetées pour découvrir des structures cachées et affiner ses prédictions pour devenir plus robuste.

6.2 Conclusion

Malgré les nombreux obstacles et le manque de résultats concluants, nous avons tout de même fourni un travail qui peut être réutilisé à l'avenir. Nous avons acquis des compétences et de l'expériences significatives dans le domaine de la recherche, tant en ce qui concerne la prise de décisions sur les expériences à mener que de l'analyse des conclusions à en tirer.

Nous tenons à remercier particulièrement notre encadrante Braud Chloé qui nous a accompagner tout au long du projet, en nous apportant soutient et réponses à nos questions. Son expertise nous a été d'une aide précieuse.

7 Bibliographie

Références

[Han+14] Awni Hannun et al. *Deep Speech : Scaling up end-to-end speech recognition*. 2014. DOI : 10.48550/ARXIV.1412.5567. URL : https://arxiv.org/abs/1412.5567.

[Pan+15] Vassil Panayotov et al. "Librispeech : An ASR corpus based on public domain audio books". In : *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, p. 5206-5210. DOI : 10.1109/ICASSP.2015.7178964.

[Hua+20] Shih-Cheng Huang et al. "Fusion of medical imaging and electronic health records using deep learning : a systematic review and implementation guidelines". In : *npj Digital Medicine* 3 (1er déc. 2020). DOI : 10.1038/s41746-020-00341-z.

[Che+21] Jun Chen et al. "Automatic Detection of Alzheimer’s Disease Using Spontaneous Speech Only". In : *Interspeech 2021*. Interspeech 2021. ISCA, 30 août 2021, p. 3830-3834. DOI : 10.21437/Interspeech.2021-2002. URL : https://www.isca-speech.org/archive/interspeech_2021/chen21r_interspeech.html.

[Luz+21] Saturnino Luz et al. *Detecting cognitive decline using speech only : The ADReSS $_O$ Challenge*. preprint. Psychiatry et Clinical Psychology, 26 mars 2021. DOI : 10.1101/2021.03.24.21254263. URL : http://medrxiv.org/lookup/doi/10.1101/2021.03.24.21254263.

[Sye+21] Zafi Sherhan Syed et al. "Tackling the ADRESSO Challenge 2021 : The MUET-RMIT System for Alzheimer's Dementia Recognition from Spontaneous Speech". In : *Proc. Interspeech 2021*. 2021, p. 3815-3819. DOI : 10.21437/Interspeech.2021-1572.

[Bai+23] Max Bain et al. "WhisperX : Time-Accurate Speech Transcription of Long-Form Audio". In : *arXiv preprint, arXiv :2303.00747* (2023).