Rapport SAM
Modèle multi-modaux

Cléa Han, Yanis Labeyrie et Adrien Zabban

janvier 2024

Table des matières

1 Introduction

Notre projet traite des approches multimodales pour la prédiction de changement de prise de parole dans des conversations naturelles. Les objectifs de ce projet nous permettent d'introduire différentes notions de modalité textuelle, visuelle et auditive, ainsi que de comparer et d'explorer différents modèles de traitement multimodaux, et leur fusion entre eux.

2 Données

Les données proviennent d'un ensemble de données multimodales, en français, composé de 26 dyades de 15 à 20 minutes, corpus de données multimodales Paco-Cheese [1]. Chaque dyade est composée de deux personnes qui discutent d'un sujet donné. Les données sont composées de trois modalités : la vidéo, l'audio et le texte.

Le but du projet est, étant donné un bout de texte, audio et/ou vidéo, d'être capable de détecter le changement de parole. On Peut alors voir ce problème comme un problème de classification de données à deux classes.

Nous avons coupé les données sur les IPUs (Inter-Pausal Units, voir l'article [2] pour plus d'information), qui représentent des segments de parole séparés par des pauses de plus de 200ms. À partir de chaque IPUs, nous allons récupérer les fins du flux vidéo, audio et de transcription du discours. On a choisi de prendre les 20 derniers mots, les 2 dernières secondes de l'audio et les 10 dernières images du flux vidéos (appartenant aux deux personnes pour l'audio et la vidéo).

Les données du flux vidéo ont été pré-process par un modèle permettant d'extraire les coordonnées de 709 points d'intérêts du visage, appelé landmarks. On a alors pour chaque image du flux vidéo, une liste de coordonnées représentent les landmarks. Cependant, du fait au grand nombre des données, leur analyse a été assez compliquée [1]

Chaque IPU est alors labellisé par un 0 ou un 1, selon s'il y a un changement de parole après ce moment. Il est important de noter que, comme on peut le voir dans la Table 1, le dataset est fortement déséquilibré.

dataset	nombre d'IPUs	pourcentage de 0	pourcentage de 1
entraînement	8861	80.51	19.49
validation	3340	80.60	19.40
test	2974	82.85	17.15
tout	15175	80.99	19.01

TABLE 1 – Nombre d'IPU et distribution des labels sur les différents dataset.

1. En effet, il y a 15 csv contenant chacun 30000 lignes et 700 colonnes. Ce qui fait que dans le dataloader, la récupération d'un item (récupération des 2 fois 10 images) nécessité d'ouvrir, parcourir le csv pour récupérer les bonnes lignes et le refermer à la fin de l'itération (du au fait qu'on prend chaque item de manières aléatoire). On a quand même pu accélérer le processus en utilisant la fonction *skiprows* de *pandas* qui permet de récupérer des lignes spécifiques dans un gros csv sans lire toutes les lignes mais cela n'a pas été suffisant.

3 Traitement unimodale

3.1 Traitement du texte avec CamemBERT

Pour parvenir à détecter le changement de tour de parole dans les données textuelles, nous avons choisi d'utiliser un modèle considéré comme l'état de l'art pour la tâche de classification de texte : le modèle BERT (Bidirectional Encoder Representations from Transformers [3]). Ce modèle utilise notament les Transformers [4] (voir son architecture Figure 1).

Par ailleurs, BERT étant un modèle comprenant un nombre important de paramètres (environ 100 millions) il est inenvisageable avec nos moyens de l'entraîner "à partir de 0". Nous avons donc opté pour l'utilisation d'un modèle pré-entrainé sur la langue Française appelé CamemBERT et nous avons gelé les poids de CamemBERT.

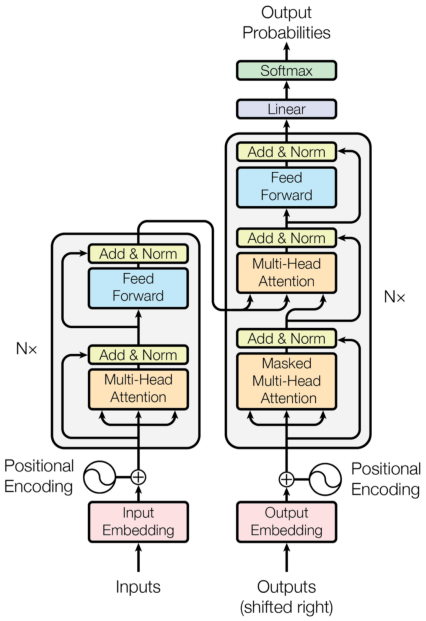

Figure 1: The Transformer - model architecture.

FIGURE 1 – Architecture de Transformers

On prend alors les données de textes qui sont une liste d'entiers, correspondant à l'indice de chaque mot du tokenizer de CamemBERT. Ces données passent alors dans CamemBERT, elles ressortent avec une dimension de 64. Elles passent alors dans une activation ReLU puis une couche dense de taille 768, puis une couche de dropout (avec un taux d'oublie de 10%), pour enfin passer un dernier ReLU et une dernière couche dense à 2 dimensions. La Figure 2 résume l'architecture de ce modèle, qu'on appelera *TEXT*.

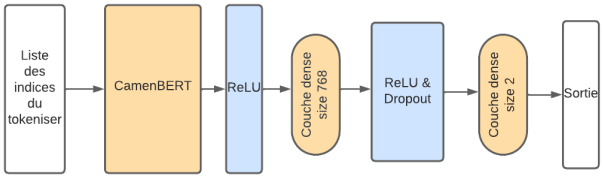

FIGURE 2 – Architecure du modèle *TEXT*.

3.2 Traitement de l'audio

Afin de détecter les changements de parole dans des données audio, nous avons choisi d'exploiter une approche similaire en utilisant un modèle avancé dans le domaine de la représentation audio : le modèle Wave2Vec [5]. Wave2Vec, basé sur les Transformers, s'est établi comme une référence pour la tâche de traitement du signal audio, en particulier pour la détection des variations dans la parole. Grâce à sa capacité à encoder de manière bidirectionnelle les représentations des signaux sonores, Wave2Vec excelle dans la compréhension des nuances acoustiques et des transitions subtiles entre les locuteurs.

On possède deux données en entrée qui sont les enregistrements audio des deux interlocuteurs. On fait alors passer leurs audio dans Wave2Vec puis l'on concatène la sortie. On fait alors passer la concaténation dans une couche dense pour n'avoir qu'une sortie de taille 2. Eventuellement, ce modèle initial pouvait être amélioré en terme d'architecture, nous avons donc à l'issue de la concaténation, ajouter une couche dense, puis une couche de ReLU et de dropout, et enfin finir sur une couche dense de sortie à 2 neurones. La Figure 3 résume l'architecture de ce modèle, qu'on appelera *AUDIO*.

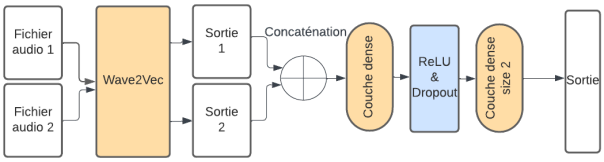

FIGURE 3 – Architecure du modèle *AUDIO*.

3.3 Traitement de la vidéo

Pour traiter les données issues de la vidéo qui sont sous la forme des landmarks du visage des deux personnes en conversation, nous avons choisi d'utiliser un réseau de neurones adapté à cette tâche d'analyse de séries temporelles : le réseau LSTM (Long-Short Term Memory). En effet, ce modèle est pertinent pour la détection du changement de locuteur, car il est capable de conserver des informations sur de longues séquences temporelles, ce qui est essentiel pour analyser les variations subtiles dans les mouvements des landmarks faciaux.

Comme pour l'audio, la vidéo contient un ensemble de deux fois 10 images pour chacun des interlocuteurs. On les fait alors passer dans deux couches LSTM et on concatène les sorties pour ensuite les faire passer dans du relu et une couche

dense de taille 2. La Figure 4 résume l'architecture de ce modèle, qu'on appelera *VIDEO*.

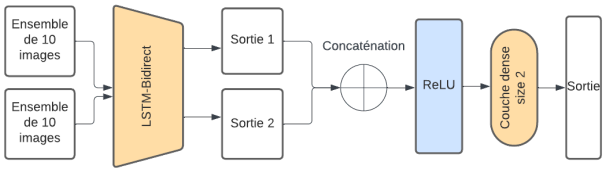

FIGURE 4 – Architecure du modèle *VIDEO*.

4 Traitement multimodal

Notre approche vise à maximiser l'utilisation des dépendances et complémentarités entre différents types de données (vidéo, texte, audio). Pour effectuer des prédictions en utilisant ces diverses modalités, nous avons opté pour l'utilisation de chaque réseau de neurones présentés précédemment afin d'extraire des représentations compressées (ou features) des différents types de données.

Les modèles individuels sont initialement pré-entraînés, puis fusionnés en une seule architecture multimodale qui subit ensuite un nouvel entraînement. Le pré-entraînement des modèles individuels présente l'avantage de réduire le temps de convergence de l'entraînement de cette architecture multimodale comprenant un nombre important de paramètres.

Au cours de ce processus, pour des raisons de contraintes de ressources, on gèle les poids des modèles unimodales.

4.1 Le LATE FUSION

Ce premier modèle multimodal fait passer chaque donnée dans son modèle pré-entraîné associé et fait une moyenne des sorties. Le but est alors de faire une moyenne des probabilités selon chaque canal. Nous appellerons ce modèle *LATE FUSION*, du fair que la fusion se fasse à la fin. Comme ce modèle utilise seulement les modèles unimodaux pré-entrainé, il n'y a pas besoin de faire d'apprentissage sur ce modèle. La Figure 5 résume l'architecture de ce modèle.

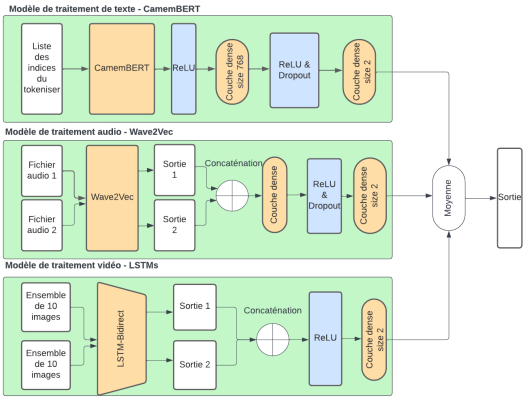

FIGURE 5 – Architecture du modèle *LATE FUSION*

4.2 L'EARLY FUSION

Ce deuxième modèle plus évolué consiste à faire passer chaque donnée dans son modèle pré-entraîné associé et de récupérer l'information avant qu'elle ne passe dans la dernière couche dense (celle de taille deux). On concatène alors toute l'information et l'on la fait passer dans une grande couche dense de taille 2. La Figure 6 résume l'architecture de ce modèle, qu'on appellera *EARLY FUSION*, du fait que la fusion se passe avant les couches de neurones.

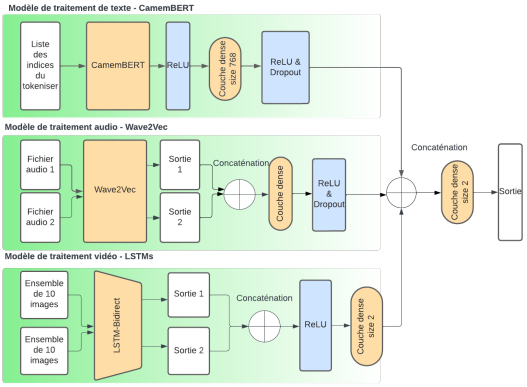

FIGURE 6 – Architecture du modèle *EARLY FUSION*.

5 Entraînements

Nous avons entraîné les modèles unimodaux et le *EARLY FUSION*, sur 2 epochs, car ils utilisent tous (sauf *VIDEO*), un modèle pré-entraîné, comment BERT ou Wave2Vec. Nous avons ensuite gardé les poids de l'epoch qui avait la plus basse loss de validation.

Cependant, l'entraînement du modèle *VIDEO*, dû a la grandeur des données et leurs temps de chargement dans la RAM, nous avons décidé de faire qu'une

seule epoch [2], et de ne pas l'intégrer dans les modèles multimodaux.

5.1 Loss et poids

Nous avons utilisé la fonction de coût *crossentropy* pour l'entrainement de modèles. Cependant, du fait, du fait du grand déséquilibré des classes (voir Table 1), nous avons ajouté des poids sur les classes dans la fonction crossentropy. En effet, sans ces poids, les modèles avaient tendance à prédire toujours des 0, ce qui leurs permetait d'optenir 80% d'accuracy sans dificultés. Nous avons alors cherché des poids à mettre dans la loss de telle sorte que les modèles ne prédisaient pas systématiquement qu'un seul label [3]. Nous avons alors trouvé et appliqué les poids $w_0 = 1$ et $w_1 = 3.9$. La loss se calcule alors avec l'équation 1 (avec des notations standards) :

$$L(x, y) = \frac{-1}{N} \sum_{n=0}^{N} w_0(1 - y_n) \log(1 - x_n) + w_1 y_n \log(x_n) \tag{1}$$

5.2 Métriques

Pour comparer les résultats obtenus, nous avons utilisé la loss, mais aussi d'autres métriques, comme l'accuracy, la précision, le rappel et le f_1-score.

Du fait des déséquilibres de classes, l'accuracy n'est pas les métriques la plus cruciale. On a donc opté pour calculer aussi la précision, le rappel et le f_1-score. Pour ces métriques, on a choisi de les calculer sur chacune des classes et d'en faire la moyenne. Ainsi un rappel de 0.3 signifiera que la moyenne du rappel sur la classe 0 et la classe 1 fasse 0.3. Nous avons choisi de faire cela, car si l'on regardait seulement le rappel sur la classe 1, le TRUE POSITIF (TP) est très souvent égale à 0 et donc le rappel est nul. De même pour la précision et de f_1-score, ce qui nous apportait moins d'information.

6 Résultats

Après avoir entraîné les modèles, nous avons testé leurs performances sur la base de données de test. La Figure 2 nous alors les résultats des modèles en fonctions des métriques.

Modèle	accuracy	precision	rappel	f_1 score
TEXT	82.8	41.3	50.0	45.3
AUDIO	47.1	48.5	47.4	41.5
VIDEO	**82.9**	41.4	50.0	45.2
LATE FUSION	78.5	**50.6**	50.1	**48.8**
EARLY FUSION	**82.9**	43.6	**50.2**	45.7

TABLE 2 – Résultats de test des modèles. Le *LATE* et *EARLY FUSION* n'utilisent pas le modèle *VIDEO*.

2. Une epoch durait plus de 2h30 sur google colab avec un GPU Nvidia T4.
3. Si on met un poids trop grand pour le label 1, même s'il y a que 20% de 1, les modèles vont prédire que des 1.

7 Conclusion

Notre projet a exploré l'utilisation de modèles unimodaux et du multimodaux pour l'analyse dynamique de la conversation en français. Nos modèles unimodaux ont été entraînés sur des données textuelles, audio et vidéo séparément. Leurs performances variaient selon le type de données, avec des résultats prometteurs pour les données textuelles et audio, mais des résultats moins satisfaisants pour les données vidéo.

Nos modèles multimodaux ont été entraînés en combinant les données textuelles et audio.

Comme perspective, on pourrait explorer différentes manières d'intégrer les trois modalités de données (textes, audios et vidéos) dans un modèle multimodal, ainsi que l'utilisation d'autres types de données ou de modèles d'apprentissage automatique. De plus, des recherches supplémentaires pourraient également se concentrer sur l'amélioration des performances du modèle vidéo, peut-être en explorant différentes méthodes d'extraction de caractéristiques ou en utilisant des ensembles de données vidéo plus grands ou plus diversifiés.

Références

[1] S. R. Mary Amoyal Béatrice Priego-Valverde. « PACO : A Corpus to Analyze the Impact of Common Ground in Spontaneous Face-to-Face Interaction. » (2023), adresse : https://paperswithcode.com/paper/paco-a-corpus-to-analyze-the-impact-of-common.

[2] G. Skantze, « Turn-taking in Conversational Systems and Human-Robot Interaction : A Review, » *Computer Speech and Language*, t. 67, p. 101 178, mai 2021. doi : 10.1016/j.csl.2020.101178. adresse : https://www.researchgate.net/publication/347821999_Turn-taking_in_Conversational_Systems_and_Human-Robot_Interaction_A_Review.

[3] J. Devlin, M. Chang, K. Lee et K. Toutanova, « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, » *CoRR*, t. abs/1810.04805, 2018. arXiv : 1810.04805. adresse : http://arxiv.org/abs/1810.04805.

[4] A. Vaswani, N. Shazeer, N. Parmar et al., « Attention Is All You Need, » *CoRR*, t. abs/1706.03762, 2017. arXiv : 1706.03762. adresse : http://arxiv.org/abs/1706.03762.

[5] A. Baevski, H. Zhou, A. Mohamed et M. Auli, « wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations, » *CoRR*, t. abs/2006.11477, 2020. arXiv : 2006.11477. adresse : https://arxiv.org/abs/2006.11477.