Rapport SAM
Modèle multi-modaux

Cléa Han, Yanis Labeyrie et Adrien Zabban

janvier 2024

1 Introduction

2 Données

Les données proviennent d'un ensemble de données multimodales, en français, composé de 26 dyades de 15 à 20 minutes. 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. Dans le cas des vidéos, les données sont déjà pré-traitées et sont composées de landmarks du visage des deux personnes en conversation. Ces données sont des séquences de 10 images par personne, soit 20 images au total. Les données audio sont des enregistrements audio des deux interlocuteurs. Enfin, les données textuelles sont des scripts de conversation des deux interlocuteurs. En effet, les données sont annotées de deux manières : avec de la segmentation de la parole basée sur les silences en utilisant les IPUs et avec des scripts de conversation, qui servent eux le traitement de texte. Les IPUs (Inter-Pausal Units) représentent des segments de parole séparés par des pauses de paroles et sont utilisés comme unité de parole dans l'analyse de conversations. Utiliser l'unité du IPU permet d'analyser la structure et le rythme de la dynamique conversationnelle.

Au-delà des fichiers audios et des fichiers textes qui sont sous leur format traditionnel respectif, les données des vidéos seront traitées à travers de fichier csv qui contiennent les landmarks du visage des deux personnes en conversation. Ces fichiers csv sont composés de colonnes indicant les caractéristiques des différentes prises de paroles.

Ces données "vidéos" sous forme de fichiers csv causent une significative augmentation du temps de chargement de ces données étant donné qu'ils donnent le lien vers des ressources à charger, et ce pour chaque ligne du fichier csv. Pour palier ce problème, nous avons utilisé des *skiprows* afin d'en accélérer le processus et de sélectionner les données pertinentes à notre apprentissage.

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 [2]). Ce modèle utilise notament les Transformers [3] (voir 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.

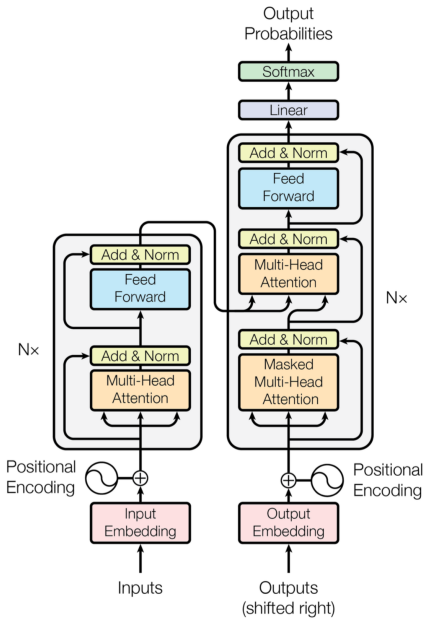


Figure 1: The Transformer - model architecture.

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 768. 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*.

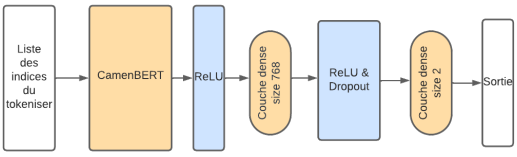


FIGURE 2 – Architecure du modèle *TEXT*.

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 [1]. 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*.

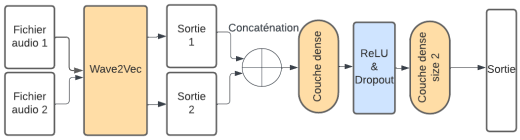


FIGURE 3 – Architecure du modèle *AUDIO*.

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*.

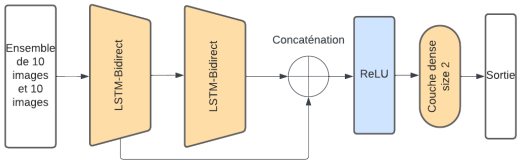


FIGURE 4 – Architecure du modèle *VIDEO*.

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 présenté 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 Maximum de vraisemblance

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 *LIKELIHOOD* du fait que le but est de maximiser la log-likelihood. La Figure 5 résume l'architecture de ce modèle.

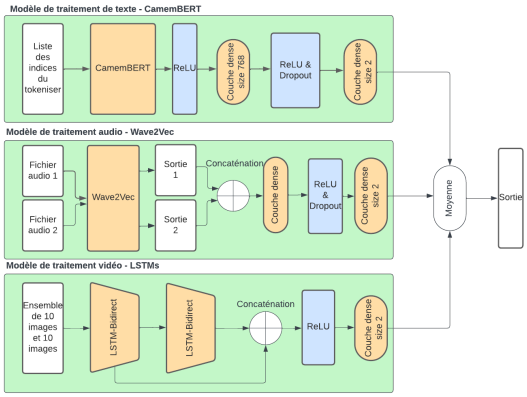


FIGURE 5 – Architecure du modèle *LIKELIHOOD*

4.2 Apprentissage multi

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 appelera *MULTI*.

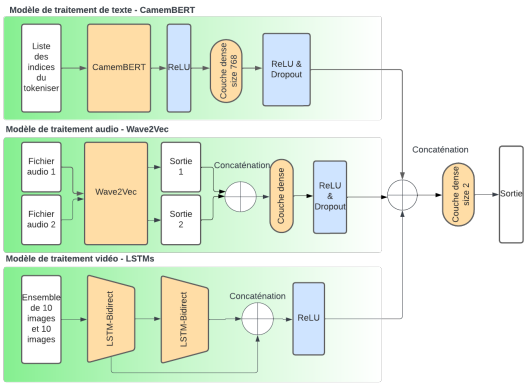


FIGURE 6 – Architecure du modèle *MULTI*.

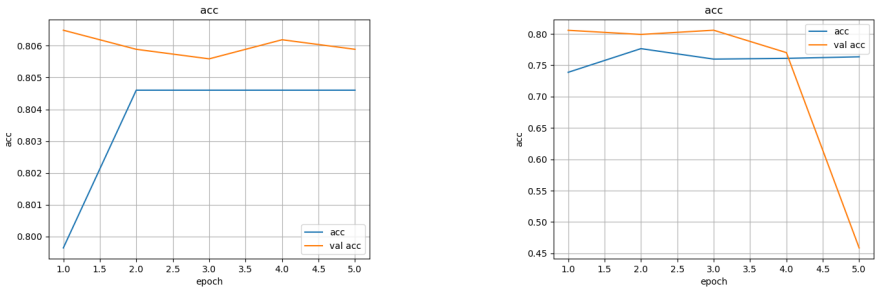
5 Results

5.1 Métriques

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

5.2 Uni-modales

Nous avons entrainé les modèles *TEXT*, *AUDIO* et *VIDEO*. Les deux premier modèles ont été entrainé sur 5 epochs. La Figure 7 montre la coubres d'accuracy d'entraînement et de validation lors de l'entraînement de ces modèles. Cependant, l'entraînement du modèle *VIDEO*, du a la grandeur des données et leurs temps de chargement dans la RAM, nous avons décidé de faire qu'une seule epoch [1].



(a) Accuracy du model *TEXT*

(b) Accuracy du model *AUDIO*

FIGURE 7 – Courbes d'apprentissage des models uni-modales *TEXT* et *AUDIO* sur l'apprentissage.

1. Une epoch durait plus de 2h30 sur google colab avec un GPU Nvidia T4.

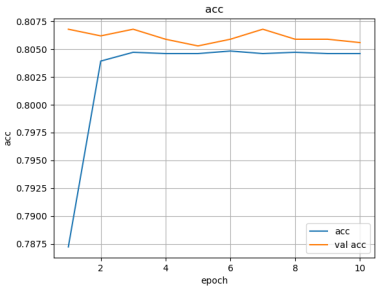
La Table 1 présente les résultats des modèles sur le corpus de test.

Modèle	loss	accuracy	precision	rappel	f_1 score
TEXT	0.539	0.823	0.828	0.816	0.822
AUDIO					
VIDEO	0.773	0.443	0.182	0.032	0.053

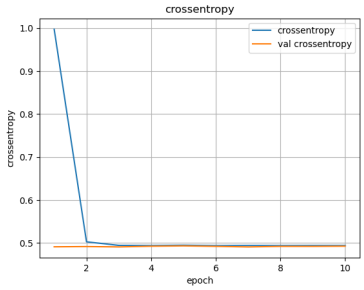
TABLE 1 – Résultats de test des modèles uni-modales

5.3 Multimodal

Voici le résultat obtenu pour le modèle multimodal :



(a) Accuracy of the multimodal model (text+audio) experiment.



(b) Cross-entropy loss of the multimodal model (text+audio) experiment.

FIGURE 8 – Results of the multimodal model (text+audio) experiment.

Références

[1] Alexei BAEVSKI et al. "wav2vec 2.0 : A Framework for Self-Supervised Learning of Speech Representations". In : *CoRR* abs/2006.11477 (2020). arXiv : 2006.11477. URL : https://arxiv.org/abs/2006.11477.

[2] Jacob DEVLIN et al. "BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding". In : *CoRR* abs/1810.04805 (2018). arXiv : 1810.04805. URL : http://arxiv.org/abs/1810.04805.

[3] Ashish VASWANI et al. "Attention Is All You Need". In : *CoRR* abs/1706.03762 (2017). arXiv : 1706.03762. URL : http://arxiv.org/abs/1706.03762.