# Rapport de travail

Finetuning de **Wav2Vec2** et **Whisper**

pour la traduction du Fon oral vers le

Français écrit

**Espérance AYIWAHOUN**

# SOMMAIRE

# 1. Introduction

**Contexte du projet:**

- Besoin de solutions de transcription et de traduction pour des langues peu représentées comme le Fon.
- Importance d’améliorer l’accessibilité linguistique et l’inclusion numérique.

**Objectif :**

- Adapter Wav2Vec et Whisper pour la tâche

# 2. Présentation des approches

Pour la traduction du Fon vers le français, nous avons expérimenté deux procédés complémentaires :

**Approche 1 :** Pipeline Wav2Vec2 + MBart

- Extraction des caractéristiques audio avec Wav2Vec2-Base (*pré-entraîné et gelé durant l'entraînement*).
- Passage par un adapter pour ajuster les représentations audio aux exigences du décodeur.
- Traduction via MBart-Large-CC25 (*décodeur partiellement gelé, avec les 6 premières couches figées*).

**Approche 2 :** Fine-tuning de Whisper

- Extraction des caractéristiques audio grâce au feature_extractor
- Utilisation du modèle pré-entraîné Whisper.
- Fine-tuning de Whisper sur le dataset pour optimiser la traduction.

# 3. Dataset et prétraitement

- **Les données :**
  - Corpus de discours en Fon avec leurs traductions en français (Fongbe to French Speech Translation Corpus FFSTC 2025).
  - **Qualité hétéroclite :** certains audios sont bruités, annotations parfois approximatives.

- **Prétraitement :**
  - Découpage des audios en segments courts (5–10 sec) pour mieux gérer la mémoire.
  - Nettoyage et normalisation des transcriptions pour éliminer les fautes.
  - Utilisation *sacremoses* pour tokeniser le texte (Pour MBart)

# 4. Finetuning – Approche technique

**Pipeline :**

- **Pour Wav2Vec2 + MBart :** Extraction des features audio, adaptation via un module dédié et les projection pour le décodeur MBart.
- **Pour Whisper :** Finetune direct pour la tâche de traduction.

**Hyperparamètres :**

- Batch size réduit pour éviter les dépassements de mémoire et les crashs de session.
- Gel de certaines couches pour stabiliser l'entraînement et économiser les ressources.

# 5. Résultats

- L'environnement Google Colab (GPU T4 15GB) impose des limitations sévères, ce qui a empêché un entraînement complet des modèles sur les données disponibles.
- Les erreurs OutOfMemory et les crashs de session m’ont amené à geler certaines couches lors de l'entraînement, afin de valider le pipeline, même si j’étais conscient que cela réduirait considérablement les performances du modèle. Cela a néanmoins permis de confirmer que le code d’entraînement fonctionne correctement.

Le code complet est accessible via ce lien et peut être exécuté sur un environnement disposant de plus de ressources :

***https://github.com/TitanSage02/Stage_AI4Innov/tree/main/TP%20Finetune%20ST***

# 6. Conclusion et perspectives

Un pipeline fonctionnel a été mis en place pour le fine-tuning de Wav2Vec2 et Whisper sur la traduction Fon → Français. Les limitations de Google Colab (mémoire insuffisante, crashs) ont empêché un entraînement complet, mais le code est prêt à être exécuté sur un meilleur matériel.

**Perspectives**

- Tester sur des GPU plus puissants (cloud, serveurs dédiés).
- Améliorer le dataset pour une meilleure qualité d’apprentissage.
- Optimiser l’entraînement avec quantification, distillation.

# MERCI