Université Sorbonne Paris Nord
Sup Galilée
Spécialité Informatique

Conduite et gestion de projet

Spécialité : INFO2

Sprint 4

Interface de Génération sonore par IA

Étudiants :
Seyfeddine JOUINI
Yosra SASSI
Sirine TLILI
Hasna ELGARANI
Kaoutar BRAHIMI

Client :
Vittascience

5 mars 2025

Table des matières

Introduction

Ce rapport présente les activités réalisées durant le Sprint 4 du projet de développement d'un module de génération musicale par intelligence artificielle. Il détaille les progrès réalisés pour chaque cas d'utilisation, en se concentrant sur la synthèse vocale, la reconnaissance vocale, la conversion de la parole, la synchronisation labiale et la génération musicale. Pour chaque cas d'utilisation, les parties réalisées sont exposées, ainsi que les défis rencontrés et les limitations actuelles identifiées. Ce rapport présente également les axes d'amélioration pour chaque fonctionnalité, en vue des prochaines étapes du projet. Enfin, un diagramme de Gantt est inclus pour illustrer l'avancement du projet dans son ensemble.

1. Rappel sur les besoins fonctionnels du projet

Le projet de génération sonore par intelligence artificielle vise à fournir une plateforme éducative innovante destinée aux enseignants et aux élèves. Les fonctionnalités du projet permettent de développer des compétences en synthèse vocale, reconnaissance vocale, et génération musicale, tout en favorisant une approche interactive et pédagogique.

Les besoins fonctionnels du projet incluent les cas d'utilisation suivants :

1. **Synthèse vocale** : Permettre à l'utilisateur de convertir un texte écrit en parole, en choisissant des paramètres tels que la voix, la vitesse et la hauteur.
2. **Reconnaissance vocale** : Convertir l'entrée vocale de l'utilisateur en texte écrit, avec une prise en charge locale sans connexion Internet.
3. **Conversion de la parole** : Assurer une interaction fluide entre l'utilisateur et le système, en permettant une communication naturelle via la reconnaissance et la synthèse vocale.
4. **Synchronisation labiale** : Synchroniser les mouvements de la bouche d'un personnage virtuel avec la parole générée, pour créer des vidéos réalistes.
5. **Génération musicale** : Générer automatiquement de la musique (des paroles et une mélodie) à partir d'un thème ou d'un prompt donné.

Ces besoins fonctionnels forment la base des développements réalisés durant ce sprint et guideront l'évolution du projet vers un système complet et intégré.

2. Cas d’utilisation 1 : Synthèse vocale

Ce cas d’utilisation a pour objectif de permettre à un utilisateur de générer un son synthétique réaliste à partir d’un texte saisi, en utilisant l’API Web Speech (SpeechSynthesis). Cette solution fonctionne entièrement en local, sans dépendance à une connexion Internet, répondant ainsi à des besoins en accessibilité et en utilisation hors ligne.

2.1. Parties réalisées

Les principales parties réalisées au cours de ce sprint pour ce cas d’utilisation sont :

- **Mise en place d’une application** permettant de tester l’API Web Speech et de générer une voix synthétique à partir d’un texte.
- **Conception d’une interface utilisateur intuitive**, offrant une expérience fluide et accessible.
- **Ajout de paramètres personnalisables**, incluant :
    - La hauteur de la voix (pitch) pour ajuster le timbre et apporter des variations émotionnelles.
    - La vitesse de lecture (rate) permettant de ralentir ou d’accélérer l’énonciation.
    - Le volume sonore (volume) ajustable selon les besoins.
    - Le choix de la langue (Français ou Anglais) avec un filtrage des voix disponibles.
    - La sélection du genre de la voix (homme ou femme).
- **Ajout d’une fonctionnalité d'enregistrement** permettant de télécharger l’audio généré sous un format adapté.

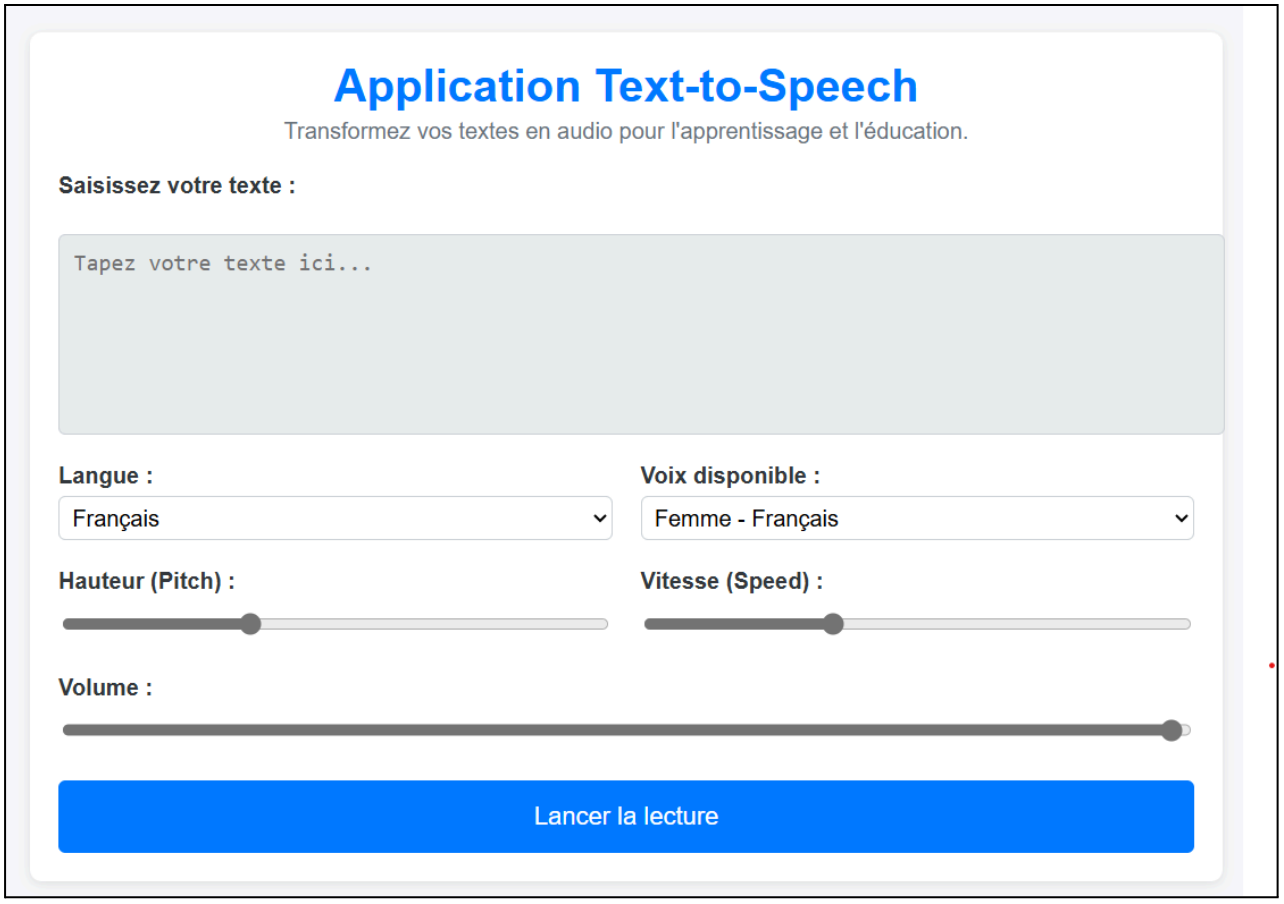


Figure : Interface utilisateur de la Synthèse vocale

Comme illustré ci-dessus, l'interface permet d'entrer un texte, de choisir une voix, d'ajuster les paramètres essentiels tels que la hauteur, la vitesse et le volume, et d'enregistrer la voix générée. Cette conception garantit une utilisation fluide et accessible.

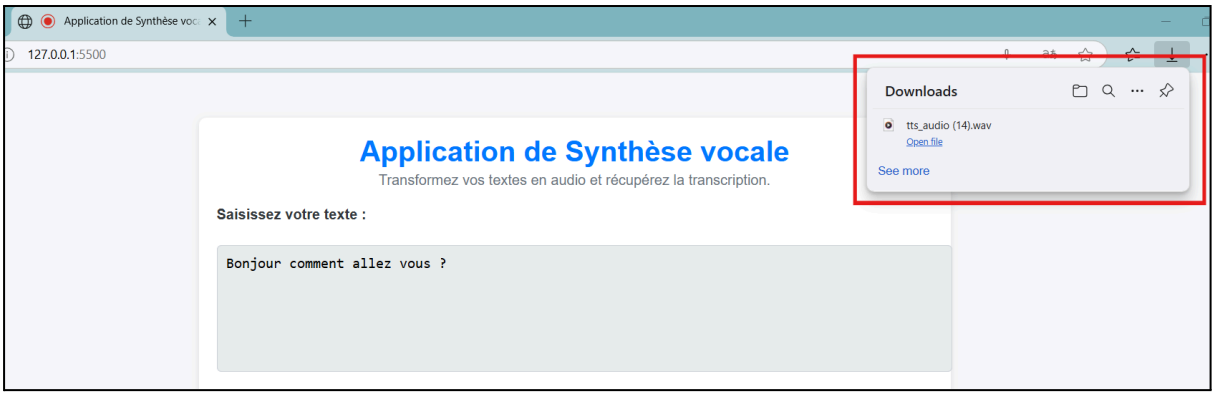


Figure : Enregistrement automatique de l'audio généré

La figure ci-dessus montre le téléchargement de l'enregistrement généré, offrant ainsi à l'utilisateur la possibilité de conserver et réutiliser le fichier audio selon ses besoins.

2.2. Défis et limitations actuelles

Malgré les avancées réalisées, certaines difficultés et limitations persistent :

- **Qualité des voix disponibles** : L'API Web Speech dépend des voix préinstallées sur le navigateur et le système d'exploitation, ce qui limite le choix et la qualité des voix synthétisées.
- **Format de l'enregistrement** : La possibilité de télécharger l'audio généré est implémentée, mais le format et la qualité du fichier peuvent varier selon les navigateurs.

Ces limitations ouvrent des perspectives d'amélioration pour le prochain sprint, notamment l'exploration de solutions alternatives pour enrichir les voix disponibles, optimiser la compatibilité multi-navigateurs et améliorer la gestion des fichiers audio exportés.

3. Cas d’utilisation 2 : Reconnaissance vocale

Ce cas d’utilisation vise à permettre à un utilisateur de convertir une entrée vocale en texte écrit en utilisant Whisper, un modèle de reconnaissance vocale performant développé par OpenAI. Contrairement aux solutions basées sur des API de navigateur, Whisper offre une meilleure précision et peut être exécuté en local pour une utilisation hors ligne.

2.1. Parties réalisées

Les principales parties réalisées au cours de ce sprint pour ce cas d’utilisation sont :

- **Intégration du modèle Whisper** pour la reconnaissance vocale en local.
- **Développement d’une interface utilisateur** permettant à l’utilisateur d’activer la reconnaissance vocale d’un simple clic.
- **Affichage en temps réel du texte reconnu**, permettant une interaction fluide avec le système.

Figure : Interface utilisateur de la Reconnaissance vocale

Comme montré ci-dessus, l’interface permet à l’utilisateur de déclencher la reconnaissance vocale d’un simple clic. Elle affiche en temps réel le texte reconnu, offrant ainsi une interaction instantanée. Cette conception garantit une utilisation intuitive et efficace, avec un retour immédiat du texte dicté.

2.2. Défis et limitations actuelles

Bien que Whisper offre une reconnaissance vocale performante, plusieurs défis et limitations ont été identifiés :

- **Consommation de ressources** : L'exécution locale de Whisper peut nécessiter une puissance de calcul élevée, en fonction du modèle utilisé.
- **Temps de traitement** : Bien que performant, le traitement de l'audio peut être légèrement plus lent qu'une API en ligne, en particulier pour de longs enregistrements.
- **Gestion de la ponctuation** : Whisper insère automatiquement de la ponctuation, mais elle peut nécessiter des ajustements selon le contexte.
- **Limitations sur les durées d'enregistrement** : Certains navigateurs imposent des restrictions sur la durée maximale de reconnaissance vocale continue.

Ces limitations constituent des axes d'amélioration pour le prochain sprint, notamment l'optimisation des performances, l'amélioration de l'interface utilisateur et l'intégration d'outils facilitant l'édition du texte reconnu.

4. Cas d'utilisation 3 : Conversion de la parole

4.1. Parties réalisées

Dans le cadre de notre projet, nous avons développé un module de conversion de la parole (*Speech-to-Speech*) permettant de capturer une voix humaine via un microphone et de la restituer sous forme d'une voix synthétisée par une intelligence artificielle. L'objectif principal était d'obtenir une voix de sortie aussi naturelle que possible, tout en garantissant un fonctionnement hors ligne et une compatibilité commerciale, conformément aux exigences du client.

- **Acquisition et enregistrement de l'audio :**

  La première étape consiste à capter la voix de l'utilisateur via un microphone. Nous utilisons la bibliothèque **sounddevice** pour enregistrer un signal audio avec une fréquence d'échantillonnage de 16 kHz et un canal unique (mono). L'audio enregistré est ensuite sauvegardé temporairement sous format WAV pour faciliter son traitement ultérieur.

- **Extraction des caractéristiques vocales :**

  Une fois l'audio capturé, nous utilisons la bibliothèque Librosa pour extraire des caractéristiques acoustiques permettant de mieux modéliser la voix de l'utilisateur. Ces caractéristiques comprennent :

  - ➢ **La fréquence fondamentale (*f0*)**, qui représente le ton de la voix et permet d'adapter la synthèse vocale pour maintenir une intonation naturelle.
  - ➢ **Les coefficients cepstraux MFCC (*Mel-Frequency Cepstral Coefficients*)**, qui capturent les aspects spectraux de la parole.
  - ➢ **Le tempo et l'énergie RMS**, qui renseignent sur le rythme et l'intensité de la voix d'entrée. Ces éléments sont cruciaux pour ajuster la voix synthétisée afin de reproduire des variations naturelles de la parole humaine.

- **Transcription de la parole en texte :**

  L'enregistrement vocal est ensuite transcrit en texte grâce au modèle Whisper, développé par OpenAI. Ce modèle, basé sur l'apprentissage profond, est capable de convertir l'audio en texte de manière précise, même en présence de bruit de fond ou d'accents variés. La transcription permet d'obtenir une version textuelle fidèle de ce qui a été dit par l'utilisateur.

- **Génération de la voix de sortie :**

  Une fois le texte transcrit, nous utilisons la bibliothèque VITS (Variational Inference Text-to-Speech) pour générer une voix de sortie. Le choix de VITS s'est imposé après plusieurs recherches, car il s'agit d'un modèle avancé de synthèse vocale qui :

  - ➢ Fonctionne hors ligne, répondant ainsi à l'une des principales contraintes du projet.
  - ➢ Offre une qualité de voix réaliste, en intégrant des inflexions naturelles et une fluidité améliorée par rapport aux modèles classiques de synthèse vocale.

Le modèle VITS prend le texte en entrée et génère un fichier audio (output.wav) contenant la voix de sortie. Nous avons également ajusté la vitesse de parole en fonction du tempo détecté afin d'obtenir une intonation plus fluide et naturelle.

- **Restitution et sauvegarde de la voix synthétisée :**

  Enfin, le fichier audio généré est stocké sous format WAV et restitué à l'utilisateur. Ce fichier peut être utilisé pour diverses applications telles que la conversion de voix, l'assistance vocale ou encore la lecture audio personnalisée.

Grâce à ce processus, notre solution permet une transformation fluide et réaliste de la parole, tout en restant performante et adaptée à une utilisation commerciale hors ligne.

Nous avons conçu une interface graphique initiale permettant d'avoir une vue globale du fonctionnement du projet et d'interagir avec les différentes étapes du processus. Cette interface a été développée en Python et sert principalement à tester et valider l'intégration des différentes fonctionnalités du module.

La figure ci-dessous illustre cette première version de l'application :

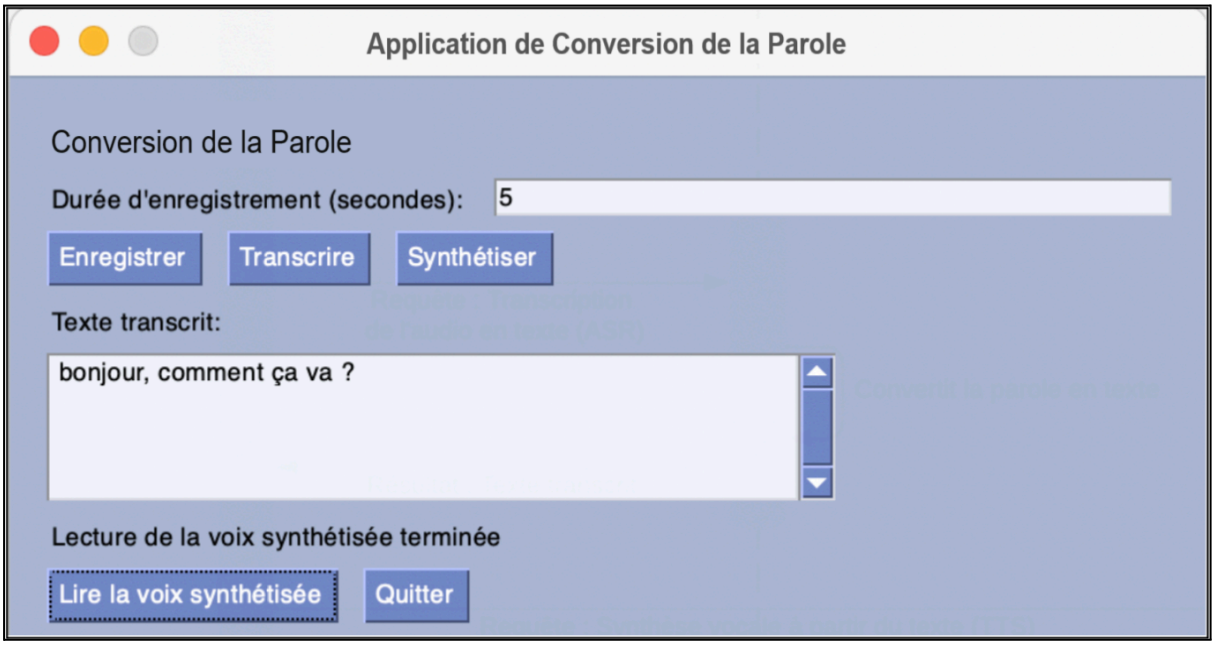


Figure : Interface utilisateur de la Conversion de parole

L'interface actuelle permet de :

- Enregistrer la voix via un microphone.
- Transcrire automatiquement le discours à l'aide du modèle Whisper.
- Générer la synthèse vocale grâce au modèle VITS.
- Lire le fichier audio généré pour vérifier la qualité du résultat final

4.2. Défis et limitations actuelles

Bien que la solution développée réponde aux exigences initiales du projet, plusieurs défis et limitations ont été rencontrés :

- **Qualité et naturalité de la voix générée :** Même si VITS offre une synthèse vocale avancée, la voix générée peut encore présenter des aspects artificiels, notamment dans les transitions entre les phonèmes et les intonations.

- **Sensibilité aux conditions d'enregistrement :** Le système dépend fortement de la qualité du signal audio en entrée. Les bruits de fond, la réverbération et les variations de prononciation peuvent affecter l'extraction des caractéristiques vocales et la transcription par Whisper.

- **Personnalisation de la voix :** Actuellement, la synthèse vocale repose sur un modèle générique pré-entraîné, limitant la personnalisation et l'adaptation à la voix spécifique de l'utilisateur.

Pour dépasser ces limitations, plusieurs axes d'amélioration peuvent être envisagés :

- **Optimisation des modèles** : ajuster les hyperparamètres de VITS et explorer des variantes plus avancées pour une meilleure expressivité.
- **Personnalisation de la voix** : développer une approche permettant d'adapter la voix synthétisée en fonction des caractéristiques spécifiques de l'utilisateur.

Cette version initiale a permis de valider les choix technologiques et d'assurer la bonne intégration des différents modules du système. Dans les prochaines phases du projet, nous prévoyons d'améliorer cette interface en développant une version web plus ergonomique et accessible.

Les évolutions prévues incluent :

- Un design amélioré pour faciliter l'expérience utilisateur.
- L'ajout de nouvelles fonctionnalités comme la personnalisation de la voix.
- Une optimisation des performances pour garantir un temps de réponse plus rapide.

5. Cas d'utilisation 4 : Synchronisation labiale

Cette fonctionnalité permet à l'utilisateur de donner vie à une image en animant le visage et en synchronisant les mouvements des lèvres avec un dialogue ou une chanson provenant d'un fichier audio, créant ainsi une illusion réaliste de parole ou de chant.

5.1. Parties réalisées

La réalisation de cette fonctionnalité se fait en deux étapes clé :

1. Transcription de l'audio :
   Afin de transcrire l'audio en un texte on a utilisé :
   - Le modèle **WHISPER** :
     - Ce modèle prend en entrée un audio de type **mp3, mp4, mpeg, mpga, m4a, wav et webm.**
   - Le langage **Python**

La figure suivante montre un texte transcrit par whisper :

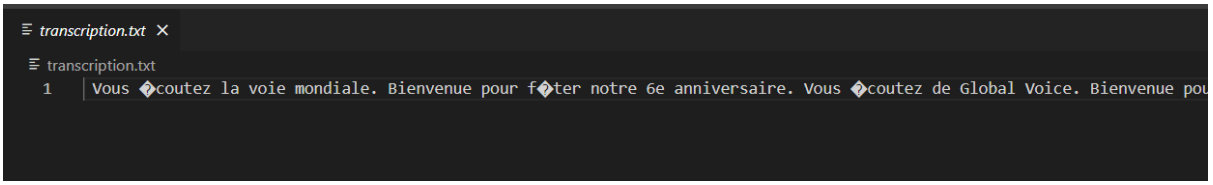
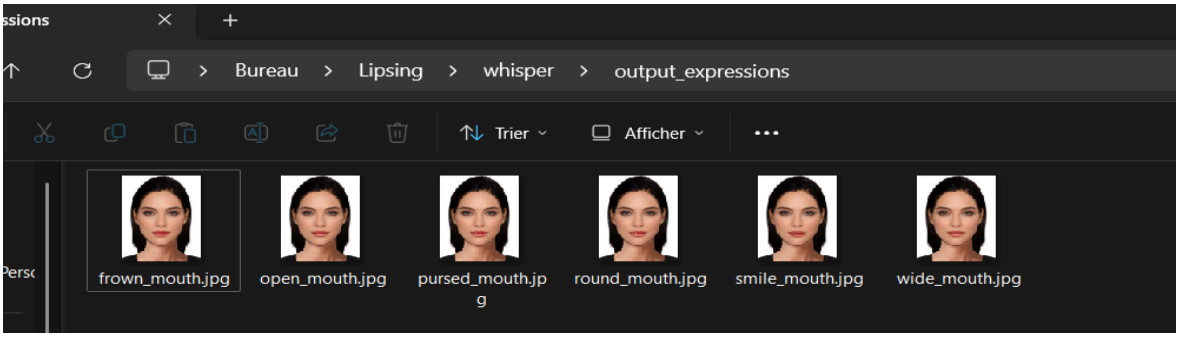


Figure : Interface utilisateur de la Conversion de parole

2. Animation de l'image :
   Afin d'animer les lèvres dans une image, on a utilisé **Numpy** avec **Python** pour faire le traitement des images. Nous avons généré 7 images pour chaque image en entrée :
   - Frown-image
   - open-mouth
   - round-mouth
   - smile-mouth
   - wide-mouth
   - pursed-mouth

La figure suivante montre les images générées par la fonctionnalité d'animation d'image :



5.2. Défis et limitations actuelles

Comme évoqué précédemment, nous avons essayé plusieurs approches tel que les modèles suivants :

- Wav2Lip
- Rhubarb

Après des échanges avec le client, nous avons décidé de ne pas opter pour ces solutions pour des raisons telles que (une licence ne permettant pas la commercialisation d'application utilisant ses modèles). De ce fait on est retourné à notre option de départ.

6. Cas d'utilisation 5 : Génération musicale

Cette fonctionnalité permet à l'utilisateur de générer automatiquement des paroles et une mélodie à partir d'un thème ou d'un prompt donné. En combinant un modèle de langage pour la création des paroles et une bibliothèque de traitement audio pour la composition musicale, l'application produit une chanson complète, incluant à la fois les paroles et la mélodie.

6.1. Parties réalisées

Afin de générer des paroles à partir d'un thème ou d'un prompt donné, on a utilisé :

- **Le modèle GPT-2 (fine-tuned)** :

  Ce modèle prend en entrée un texte (par exemple, "une chanson sur l'amour") et génère des paroles cohérentes et créatives en fonction du thème. Le modèle a été entraîné sur un ensemble de données de paroles de chansons pour améliorer sa capacité à produire des textes poétiques et structurés.

  En plus, ce modèle est sous **Licence MIT** qui permet de l'utiliser dans des produits commerciaux.

  (Lien de la Licence https://github.com/openai/gpt-2/blob/master/LICENSE)

La figure suivante présente l'entraînement du modèle par une dataset contenant **30000 paroles** de chansons déjà existantes :

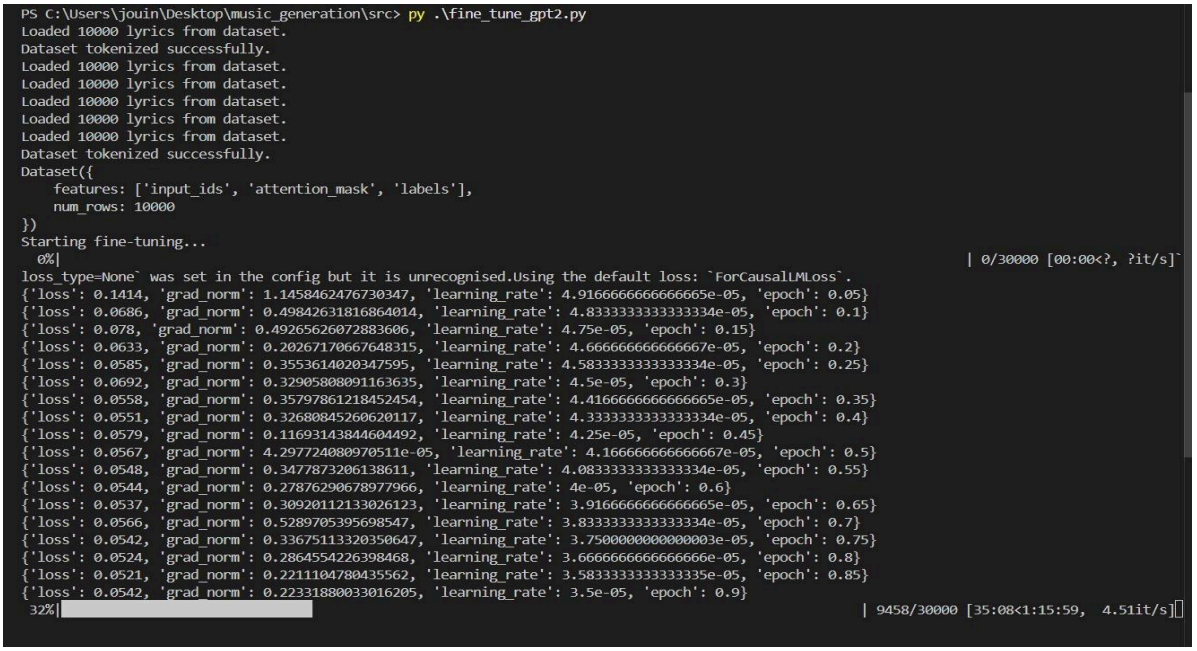


Figure : Entraînement du modèle

Parties à venir :

Afin de générer une mélodie à partir des paroles ou d'un thème donné, on a utilisé :

- **La bibliothèque music21** :

  Cette bibliothèque permet de créer des séquences musicales en définissant des notes, des rythmes et des tempos. Elle génère des fichiers MIDI qui peuvent ensuite être convertis en formats audio (WAV, MP3) pour une écoute immédiate.
- **FluidSynth** : Pour la conversion MIDI en WAV.
- **SoundFont (`.sf2`)** : Pour simuler des instruments réels (par exemple, piano, guitare).
- **pydub** : Pour la conversion du fichier WAV en MP3.

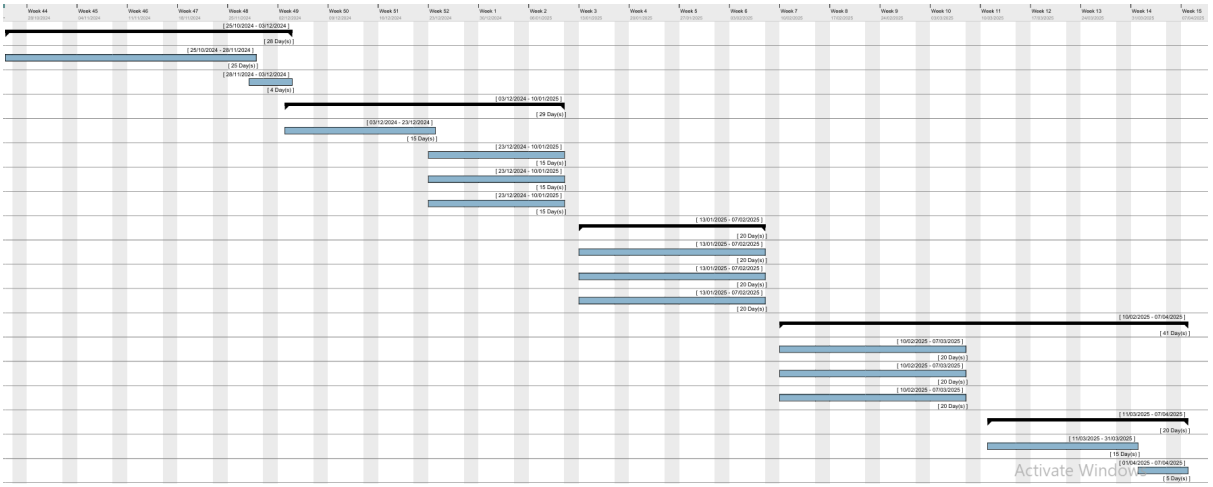
6.2 Défis et limitations actuelles

La conversion MIDI en WAV est une étape délicate car elle dépend totalement de FluidSynth.

Actuellement, des conflits de dépendances et des problèmes de compatibilité empêchent FluidSynth de fonctionner correctement.

7. Diagramme de Gantt

Le diagramme de Gantt ci-dessous illustre la planification de notre projet. Il détaille les différentes tâches, leur répartition dans le temps ainsi que les dépendances entre elles. Ce planning permet d'assurer une gestion efficace des ressources et de suivre l'avancement du projet de manière structurée.



Name	Begin date	End date	C...	Cani...	New c...
Sprint 1	25/10/2024	03/12/2024			
Initialisation du projet et analyse des besoins / cahier d charge	25/10/2024	28/11/2024			
Développement d'une Démo pour le module TTS	28/11/2024	03/12/2024			
Sprint 2	03/12/2024	10/01/2025			
Recherche et analyse technologique MindMap	03/12/2024	23/12/2024			
Développement et intégration du prototype du module TTS	23/12/2024	10/01/2025			
Développement et intégration du prototype du module Lypsing	23/12/2024	10/01/2025			
Recherche et Développement d'une Démo pour le module STS	23/12/2024	10/01/2025			
Sprint 3	13/01/2025	07/02/2025			
Développement et intégration du prototype du module STS	13/01/2025	07/02/2025			
Finalisation du module Lypsinc	13/01/2025	07/02/2025			
Finalisation du module TTS	13/01/2025	07/02/2025			
Sprint 4	10/02/2025	07/04/2025			
Finalisation et optimisation des performances	10/02/2025	07/03/2025			
Finalisation du module STS	10/02/2025	07/03/2025			
Développement du module de génération musicale	10/02/2025	07/03/2025			
Sprint 5	11/03/2025	07/04/2025			
Finalisation du module de génération musicale	11/03/2025	31/03/2025			
Tests d'acceptation et documentation	01/04/2025	07/04/2025			

Figure : Diagramme de Gantt

Conclusion

En conclusion, le Sprint 4 a permis de réaliser des avancées significatives dans le développement des différentes fonctionnalités du projet de génération musicale par intelligence artificielle. Chaque cas d'utilisation a progressé, avec des solutions techniques mises en place pour répondre aux besoins spécifiques des utilisateurs, tout en prenant en compte les défis rencontrés et les limitations actuelles.

Des axes d'amélioration ont été identifiés, et les prochaines étapes du projet visent à optimiser les performances et à enrichir l'expérience utilisateur. Ce projet continue de se développer vers une solution pédagogique innovante et accessible, en mettant l'accent sur l'apprentissage et l'interaction.