

Live Subtitles : Assistant vocal en Java

Présenté par Hssaine Mohammed Amine

3A - IATD-SI

19 mai 2025

Plan de la présentation

- 1 Introduction et problématique
- 2 Objectifs du projet
- 3 Technologies et bibliothèques
- 4 Architecture et fonctionnement
- 5 État d'avancement
- 6 Perspectives d'évolution
- 7 Conclusion

- **Contexte actuel** : Omniprésence des assistants vocaux et des solutions de transcription
- **Limites des solutions existantes** :
 - Forte dépendance aux services cloud
 - Questions de confidentialité des données
 - Solutions souvent propriétaires et coûteuses
- **Problématique** : Comment proposer une solution de reconnaissance vocale accessible, performante et adaptée aux ordinateurs personnels ?

Objectifs du projet

- Développer une application desktop en Java :
 - Proposant des sous-titres en **temps réel**
 - Avec une interface utilisateur intuitive
 - Fonctionnant sur des configurations matérielles standard
- Utiliser des technologies **open source** et **multiplateformes**
- Poser les bases d'un système évolutif et adaptable

- **Vosk API**

- Moteur de reconnaissance vocale open source
- Utilise des modèles acoustiques pré-entraînés
- Support multilingue
- Traitement du signal optimisé pour Java

- **Avantages principaux :**

- Flexibilité d'utilisation
- Intégration facilitée dans les projets Java
- Communauté de développement active
- Modèles adaptés à plusieurs cas d'usage

- **DL4J et ND4J**

- Bibliothèques de calcul scientifique pour Java
- Optimisation des opérations matricielles
- Support pour l'intégration de deep learning

- **Maven**

- Gestion structurée des dépendances
- Organisation du cycle de vie du projet
- Facilitation du déploiement

- **Interfaces utilisateur**

- Swing : robuste et éprouvé
- JavaFX : moderne et flexible

Composants principaux :

① Module de capture audio

- Acquisition du signal via les APIs Java standards
- Prétraitement du signal pour optimisation

② Module de reconnaissance

- Intégration de Vosk pour l'analyse audio
- Traitement par segments optimisés

③ Module d'affichage

- Interface graphique réactive
- Affichage des sous-titres synchronisés

- ① Capture audio via le microphone du système
- ② Segmentation et prétraitement du signal
- ③ Analyse par le moteur de reconnaissance Vosk
- ④ Optimisation avec les bibliothèques ND4J
- ⑤ Affichage en temps réel dans l'interface graphique

Principes de conception :

- Architecture modulaire et découplée
- Traitement par flux pour optimiser les performances
- Gestion efficace des ressources système

Fonctionnalités implémentées

- Système de capture audio
- Intégration du moteur Vosk
- Interface utilisateur de base
- Transcription en temps réel
- Support pour français et anglais

Performance générale

- Latence acceptable pour utilisation en temps réel
- Bonne précision de reconnaissance en environnement contrôlé
- Consommation de ressources raisonnable
- Stabilité sur différentes configurations

Points forts actuels

Fiabilité de la reconnaissance vocale
Adaptabilité à différents environnements sonores
Facilité d'utilisation de l'interface

- **Migration vers une architecture Spring Boot**

- Création d'une API REST dédiée
- Séparation backend/frontend
- Facilité de déploiement et évolutivité

- **Interface web moderne**

- Développement frontend avec React ou Angular
- Expérience utilisateur améliorée
- Accessibilité multi-plateforme

- **Avantages attendus**

- Flexibilité d'accès accrue
- Maintenance simplifiée
- Possibilité d'intégration à d'autres services

- **Intégration avancée avec DL4J**

- Développement de modèles adaptés au contexte
- Amélioration continue de la précision
- Adaptation aux spécificités linguistiques

- **Architecture distribuée**

- Traitement modulaire et scalable
- Options de synchronisation entre local et cloud
- Équilibre entre performances et protection des données

- **Enrichissements fonctionnels**

- Support multilingue étendu
- Adaptation au vocabulaire spécifique
- Options d'export et partage des transcriptions

- **Contribution du projet**

- Solution de reconnaissance vocale accessible en Java
- Alternative aux services propriétaires existants
- Démonstration de la faisabilité technique

- **Défis relevés**

- Optimisation des performances sur environnement Java
- Intégration d'outils de reconnaissance vocale avancés
- Conception d'une architecture évolutive

- **Impact potentiel**

- Amélioration de l'accessibilité numérique
- Meilleure maîtrise des données utilisateurs
- Base pour des développements communautaires

Merci de votre attention !

Questions et discussions