

# Rapport explicatif : Fonctionnement de Vosk pour la transcription vocale d'une demande de trajet

## 1. Présentation de Vosk

Vosk est une API de reconnaissance vocale qui fonctionne sur des modèles acoustiques et linguistiques pré-entraînés. Il supporte plusieurs langues et est compatible avec différents systèmes d'exploitation (Linux, Windows, macOS, et Android). Il offre :

- Une faible consommation de ressources,
- Une exécution hors ligne (offline),
- Une intégration facile avec des langages comme Python, Java, ou C++.

Vosk permet de traiter les fichiers audio ou d'écouter un flux audio en temps réel pour transcrire la voix en texte.

---

## 2. Fonctionnement général

Le pipeline de fonctionnement de Vosk comprend les étapes suivantes :

1. **Capture audio :**
    - L'utilisateur fournit une demande vocale via un microphone ou un fichier audio.
    - L'audio est capturé sous un format compatible (e.g., WAV, mono, 16 kHz).
  2. **Traitement préliminaire :**
    - Normalisation des données audio pour réduire le bruit et préparer le fichier pour l'analyse.
  3. **Analyse acoustique :**
    - Vosk utilise des modèles acoustiques pour convertir les signaux audio en phonèmes (unités sonores de base).
  4. **Interprétation linguistique :**
    - Les phonèmes sont associés à des mots en fonction d'un modèle linguistique pré-entraîné.
  5. **Conversion en texte :**
    - Le résultat est une transcription textuelle brute de la demande vocale.
- 

## 3. Mise en oeuvre pour une demande de trajet

Voici comment adapter Vosk pour retranscrire une demande de trajet vocal :

### a) Exemple de scénario

Un utilisateur dit : "Je veux aller de Paris à Lyon demain matin."

### b) Flux de traitement

### 1. Capture de l'audio

- Une application frontale enregistre la voix de l'utilisateur via un microphone.
- Le fichier audio est pré-traité pour être en mono et avec une fréquence d'échantillonnage de 16 kHz.

### 2. Traitement par Vosk

- L'audio est transmis à Vosk pour transcription.
- Le modèle acoustique identifie les phonèmes correspondant à l'énoncé.
- Le modèle linguistique transforme les phonèmes en mots cohérents avec la grammaire de la langue cible (ici le français).

### 3. Post-traitement textuel

- Les erreurs de reconnaissance (e.g., "Lion" au lieu de "Lyon") sont corrigées grâce à un module de correction contextuelle.
- Le texte final est standardisé pour être utilisé par des systèmes tiers.

## c) Conversion en demande de trajet

- Une fois la transcription effectuée, un algorithme d'analyse extrait les entités clés :
    - Point de départ : "Paris"
    - Destination : "Lyon"
    - Date et heure : "demain matin"
  - Les données sont structurées dans un format compatible avec un système de gestion des trajets.
- 

## 4. Intégration technique

Pour implémenter cette solution :

### 1. Installation de Vosk

- Installer la bibliothèque : `pip install vosk`.
- Télécharger un modèle linguistique approprié (e.g., français).

## Code Python de base

```
from vosk import Model, KaldiRecognizer
import wave

# Charger le modèle
model = Model("path_to_french_model")
recognizer = KaldiRecognizer(model, 16000)

# Ouvrir un fichier audio
wf = wave.open("demande_trajet.wav", "rb")

# Transcrire l'audio
while True:
    data = wf.readframes(4000)
    if len(data) == 0:
        break
    if recognizer.AcceptWaveform(data):
        print(recognizer.Result())
```

### 2. Extraction des entités

- Utiliser des outils comme SpaCy ou regex pour identifier les noms de lieux et les dates.

---

## 5. Avantages de Vosk

- **Hors ligne** : Pas besoin de connexion Internet.
- **Efficacité** : Fonctionne bien sur des machines aux ressources limitées.
- **Précision** : Modèles adaptés pour le français et autres langues.

## 6. Limites

- Les erreurs de transcription peuvent survenir si l'audio est bruyant ou si l'accent de l'utilisateur diffère de celui des données d'entraînement.
- Les corrections contextuelles demandent un traitement supplémentaire.

---

## Conclusion

Vosk est une solution adaptée pour convertir des demandes vocales en texte dans un environnement contrôlé. Avec une intégration efficace, il est possible de transformer des demandes vocales de trajet en requêtes textuelles structurées prêtes à être traitées par un système backend.