

8INF887 - Apprentissage Profond

## ***SOUS-TITRAGE ET TRADUCTION AUTOMATIQUE DE FILMS/VIDÉOS***

Salim Mansouri, Olwen Martin et  
Samuel Madrigal

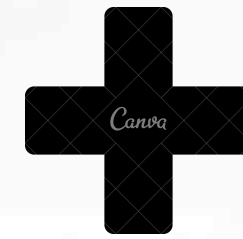


# ***SOMMAIRE***

- 1. CHOIX DES TECHNOLOGIES***
- 2. TRAITEMENT DES SOUS-TITRES***
- 3. PREMIÈRES ARCHITECTURES***
- 4. AJOUT DES SOUS-TITRES***
- 5. ARCHITECTURE FINALE***
- 6. RÉSULTAT FINAL***
- 7. CONCLUSION***

# ***CHOIX DES TECHNOLOGIES***

MoviePy



# ***CHOIX DES TECHNOLOGIES***

## **Transcription/Segmentation**



## **Traduction**



# ***CHOIX DES TECHNOLOGIES***



- + Haute précision et robustesse pour la transcription
- + Support multilingue étendu
- Segmentation imprécise
- Ressources computationnelles élevées



- + Haute précision et robustesse pour la segmentation
- + Léger et performant sur des appareils limités
- Précision parfois inférieure pour la transcription
- Nécessité de télécharger plusieurs modèles pour les langues souhaitées

# ***TRAITEMENT DES SOUS-TITRES***

## **Formatage du texte**

```
def process_entries(self, entries: List[SRTEntry]) -> List[SRTEntry]:
    # Fusionner avec référence si fournie
    if self.reference_entries:
        entries = merge_srt_files(self.reference_entries, entries)
    # Filtrer les entrées vides
    entries = [entry for entry in entries if entry.text.strip()]
    for entry in entries:
        # Traduction unique si un traducteur est configuré
        if self.translator:
            # On traduit et récupère le texte traduit
            result = self.translator(entry.text, max_length=512)
            entry.text = result[0]['translation_text']
        # Correction de la ponctuation et formatage
        entry.text = fix_punctuation(entry.text)
        entry.text = format_text(entry.text, self.max_chars)
        # Vérification finale pour les majuscules après virgule
        entry.text = re.sub(r',\s+([A-Z])', lambda m: f", {m.group(1).lower()}", entry.text)
    # Réindexer les entrées
    for i, entry in enumerate(entries):
        entry.index = i + 1
    return entries
```

- Gestion de la langue
- Uniformisation de la ponctuation
- Détection des accents
- ...

# Combinaison

Cycle complet de sous-titrage et traduction

---

**01** Choix de la langue

**02** Génération de fichier .srt  
(via Whisper et/ou Vosk)

**03** Traitement du texte

**04** Création de la vidéo  
avec sous-titres



# Ajout des sous-titres

```
from moviepy.editor import VideoFileClip, CompositeVideoClip, TextClip
from moviepy.video.tools.subtitles import SubtitlesClip
import os

os.environ["IMAGEMAGICK_BINARY"] = "/opt/homebrew/bin/convert"

# Définir une fonction qui génère un clip de texte pour chaque sous-titre
def subtitle_generator(txt):
    return TextClip(txt, font="Arial", fontsize=22, color="white")

# Charger la vidéo d'origine
video = VideoFileClip("ffmpeg-7.1.1/batman.mp4")

# Charger le fichier de sous-titres au format SRT et créer le clip de sous-titres
subtitles = SubtitlesClip("corrected.srt", subtitle_generator)

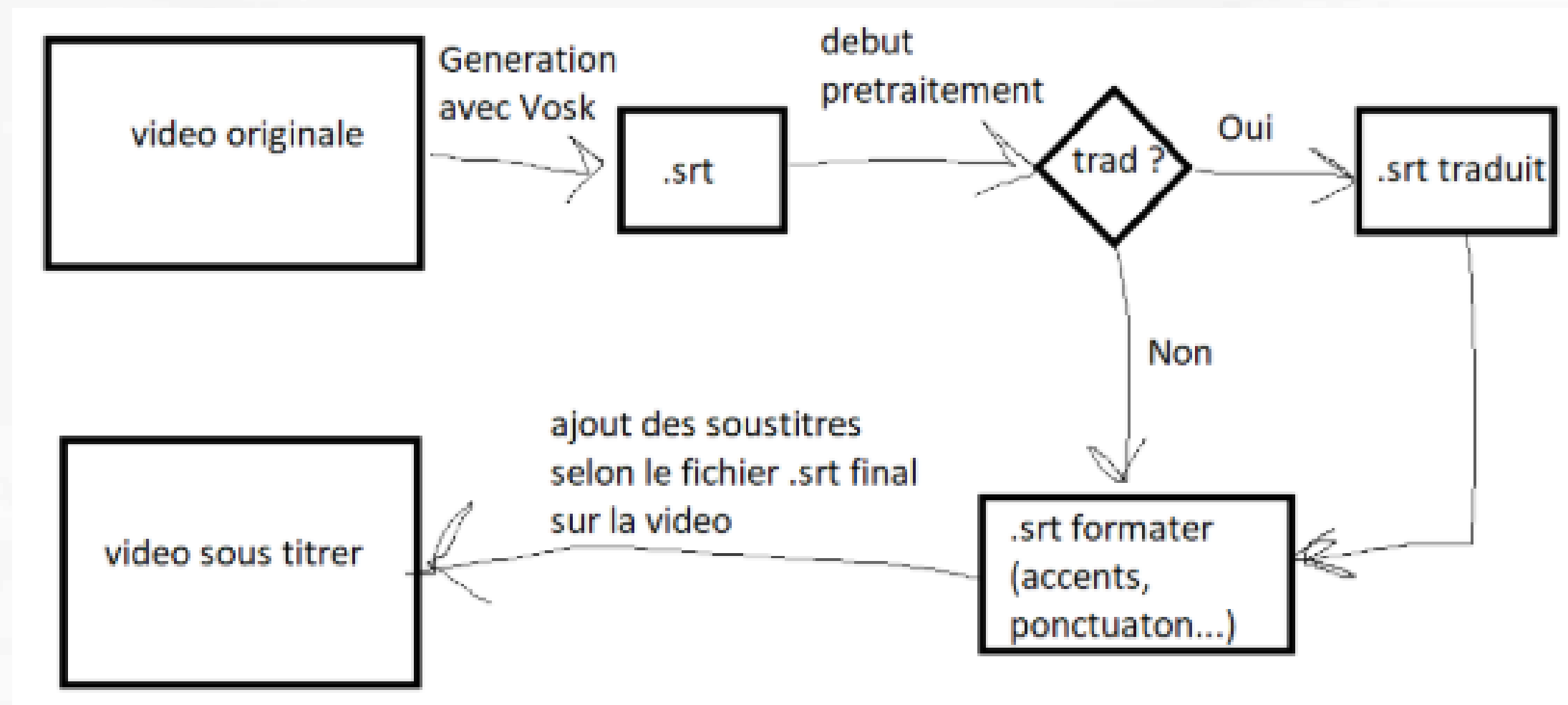
# Superposer les sous-titres sur la vidéo, positionnés en bas au centre
video_with_subs = CompositeVideoClip([video, subtitles.set_pos(('center', 'bottom'))])

# Sauvegarder la vidéo résultante
video_with_subs.write_videofile("video_with_subs.mp4", fps=video.fps)
```

- Générateur de sous-titres
- Combinaison de vidéo et fichier srt
- Écriture de la vidéo



# Architecture finale



# Résultat final



***Merci !***  
***Des Questions ?***