TRANSCRIBEIT

BUILDING TRANSCRIBEIT

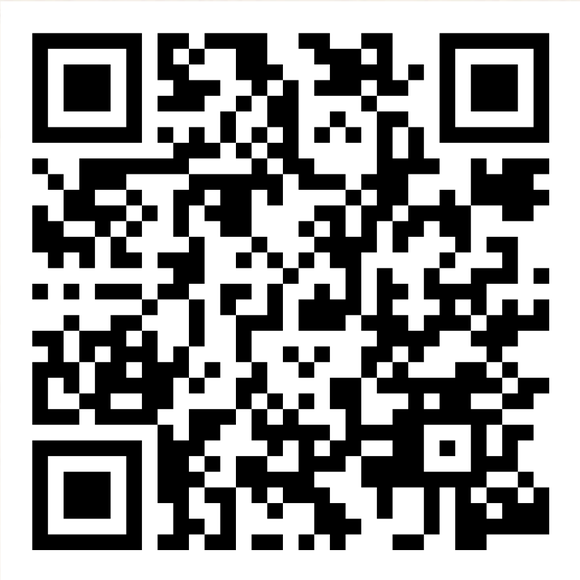
Lessons from building transcription service for local and online multimedia content

PRESENTED BY:

Keerthana Rajesh Kumar
Founder, FOSSIA

Access the slides

https://fossia.org/blog/building-transcribeit



About Me

- A cybersecurity undergraduate
- Free software enthusiast and contributor
- DEIA advocate and accessibility developer
- Founder and Director at FOSSIA
- Founder of InLibre

Agenda

01	Overview
02	Challenges in multimedia accessibility
03	Building TranscribeIt
04	Architecture
05	Challenges faced
06	What next?

Overview

TranscribeIt is a free software transcription application for online and local multimedia content. It’s developed to be robust and cater to needs of hard of hearing, speech and low vision people to improve accessibility.

Features

01 Local, multilingual, timestamped transcriptions

02 Integrated video captioning and description for improved accessibility

03 Customized speaker diarization and transcription export

Challenges in multimedia accessibility

Lack of transcriptions for live streams

Creates challenges for hard of speech and hearing population

Lack of frame-wise video description

Creates challenges for visually impaired people in understanding context

Accessibility concerns with online transcription platforms

Privacy concerns with cloud-based transcription and WCAG non-compliance impacts trust and inclusion

Building TranscribeIt: Philosophy

Local-first

Aid self-hosting with minimal setup for privacy and availability, without losing performance

Accessible

Ensure WCAG compliance and support export mechanisms for easier integration

Robust

Support multiple multimedia formats with customization for simplicity

Building TranscribeIt: Approach

faster-whisper

Local, timestamped and multilingual transcription with improved performance over whisper

FrameStory

Local video description with customization for accessibility for visually impaired people

pyannote

Speaker diarization with customization for improved user experience

Architecture

User Interface

- Customized transcription, description and speaker diarization
- Provide URL or local file
- Requests get processed by backend and responses are retrieved

Backend

Queue-based user request processing

Processing Pipeline

1. Transcription

Timestamped transcription with whisperx

2. Speaker Diarization

Use pyannote for speaker diarization and consolidation of transcription

3. Video Description

Frame extraction and processing for video description using OpenCV and transformers

Challenges Faced

faster-whisper over whisperx, for customization

Resulted in higher development time and complexity, which was an essential tradeoff for planned multilingual transcription enhancement

Lack of multilingual support for FrameStory

FrameStory lacks multilingual support for descriptions, creating a language barrier

High WER with Whisper for Indian languages

Requires usage of fine-tuned models like Whisper-Hindi-v2 (WER from 172% to 14% [v1] to 5% [v2]) with language detection detection by lingua-py

High computational complexity despite optimization

Speaker diarization in addition to video description and transcription results in higher computationa time, requiring optimized asynchronous processing with GPU

Poor results with noisy samples

Noisy multimedia samples have poor transcription accuracy in terms of language detection and segment recognition, requiring usage of denoising libraries (under experimentation)

What Next?

Improve accuracy for non-Latin languages and noisy samples

Support usage of custom fine-tuned models

Multilingual video descriptions

SDK for transcription and description for accessibility

Thank you

REACH OUT

E-mail fossia@riseup.net

Website https://fossia.org

GitHub https://github.com/fossiaorg

VISIT OUR WEBSITE