

Real-time Subtitling for Live Events

Introduction :

In an increasingly global and inclusive environment, real-time subtitling has emerged as a powerful tool for enhancing accessibility and audience engagement during live events. Whether for individuals with hearing impairments or non-native language speakers, real-time subtitles provide an immediate textual representation of spoken content, bridging communication gaps.

This project focuses on developing a real-time subtitling system using automatic speech recognition (ASR) and natural language processing (NLP) technologies. The system captures live audio from events, transcribes it instantly, and displays synchronized subtitles to the audience.

Problem Statement :

Live events often lack accessibility features such as subtitles, limiting participation for diverse audiences. Manual transcription is slow, expensive, and prone to delays, making it unsuitable for dynamic, real-time environments. This project addresses the need for a scalable, accurate, and automated solution for live subtitling.

Objectives :

Enhance Accessibility: Provide real-time subtitles to support hearing-impaired users and multilingual audiences.

Automate Transcription: Replace manual efforts with AI-driven speech recognition for faster and consistent results.

Improve Audience Engagement: Increase inclusivity and comprehension during live broadcasts, conferences, and public events.

Methodology:

Problem Analysis: Assessed the challenges in live event accessibility and the limitations of manual subtitling.

Audio Capture and Processing: Integrated real-time audio feed from live events for speech input.

Speech Recognition Engine: Utilized tools like Google Speech-to-Text API or DeepSpeech to transcribe spoken words into text.

Subtitle Formatting: Applied NLP techniques for punctuation, sentence segmentation, and delay minimization.

Display and Sync: Designed a user interface for displaying synchronized subtitles on screens or personal devices.

Testing and Validation: Conducted pilot tests at mock events to measure latency, accuracy, and user feedback.

Sample Code:

```
from google.colab import files
import zipfile, os, time
import speech_recognition as sr

# Upload and extract ZIP uploaded =
files.upload() zip_file =
list(uploaded.keys())[0] extract_path =
"live_audio" with zipfile.ZipFile(zip_file, 'r')
as zip_ref:

    zip_ref.extractall(extract_path)

recognizer = sr.Recognizer() subtitle_file =
"live_subtitles.txt"

# Find all wav files audio_files = [] for root, _,
files_in_dir in os.walk(extract_path):    for file in
sorted(files_in_dir):

    if file.lower().endswith(".wav"):

        audio_files.append(os.path.join(root, file))

with open(subtitle_file, "w", encoding="utf-8") as f:    f.write("===
Live Subtitles ===\n")
```

```

for filepath in audio_files:    with
sr.AudioFile(filepath) as source:
audio = recognizer.record(source)    try:

    text = recognizer.recognize_google(audio)
timestamp = time.strftime('%H:%M:%S')
print(f"[{timestamp}] {text}")    with open(subtitle_file,
"a", encoding="utf-8") as f:        f.write(f"[{timestamp}]
{text}\n")    except sr.UnknownValueError:

    print(f"[{time.strftime('%H:%M:%S')}] Could not understand.")    except
sr.RequestError as e:

    print(f"[{time.strftime('%H:%M:%S')}] API error: {e}")    time.sleep(1)

```

Input Format:

A ZIP file containing one or more .wav audio files representing audio segments to be transcribed.

Output Format:

A plain text file named 'live_subtitles.txt' containing timestamped subtitles in the following format:

[HH:MM:SS] Transcribed speech text

Result :

The system successfully transcribed each audio segment and appended the recognized text with timestamps to the subtitle file, simulating live subtitle generation.

Sample Output:

```
14 May, 10.34 am(2).wav: with Mom this morning we talked about her childhood she smiled the whole
time I should do this more often
file 10.wav: so hot today when I try to buy her hair these are the moments that stay
file 8.wav: someone paid for the woman's best for ahead of me she smile at like her day was just
saved
14 May, 10.34 am_.wav: my niece said I'm her favorite person that made my whole week
nm-file-8.wav: for the first time it did not look great but tasted amazing proud of myself
nm-file-4.wav: let alone in the cafe just watching people go by everyone has their own story it made
me feel both small and connected
14 May, 10.30 am_(2).wav: Sunset the color was Andrea
nm-file-1.wav: rainy day stained made hot chocolate and watch it on old movie I feel peaceful and
cozy
nm-file-9.wav: definition of the book I have been putting off the last line stay with me you are
enough even when you feel empty
memo.wav: cup of tea felt like the world slow down for the bit
14 May, 10.34 am.wav: my niece said I'm her favorite person that made my whole week
```

Testing and Validation:

The system was tested with multiple WAV files of varying lengths and speech clarity.

Recognition accuracy was validated by comparing transcribed text to the original speech. Error handling was tested by including noisy and silent audio clips.

Conclusion:

The implementation of a real-time subtitling system significantly enhances the inclusivity and accessibility of live events. By leveraging speech recognition and NLP, this solution offers a reliable and scalable approach to live transcription. Future improvements could involve multilingual translation and speaker identification for even broader applications.