

ST - TRANSLATION

Wentworth institute of Technology

Impact of AI on Society

By: Robert Sprowl

Asmeret Jafarzade

Fengnan Zhang

Contents

01

Background

02

Objectives

03

Demo

04

Reference

Background

What is speech-to-text?

Speech to text is a speech recognition software that enables the recognition and translation of spoken language into text through computational linguistics. It is also known as speech recognition or computer speech recognition.



Speech to Text

The IBM Watson Speech to Text service uses speech recognition capabilities to convert Arabic, English, British English, Spanish, Brazilian Portuguese, Japanese, and Mandarin speech into text.

Objectives

- Create a program that can transcribe and translate speech from, or to, seven different languages
 - You can either upload a previously recorded audio file or record one in real time
 - Make the user record audio first in an interactive format
 - Record the audio using a command line audio processing tool called SoX (Sound exchange)
 - Save the recording to a file
 - Output the recorded audio then perform transcription -> translation
-

API Usage (Speech-to-Text, Translation)

For this project, we used two APIs:

- IBM Watson Speech-to-Text -> Takes in speech through an audio file or real-time recording -> Outputs into a text file for further processing.
- IBM Language Translator -> Takes the transcript output file and converts it to another language of choice. Can also utilize direct text input for general translation purposes.

We also use the speech-to-text API to calculate the transcription confidence.

The translation API is also used to detect the input's spoken language and the confidence value associated with it.

Usage

- Convert an audio book in another language to a text book in another language
- If you're programming a commercial product and you want to set a baseline in another language without native speaking developers
- You can use it to translate any audio or text
- Increases accessibility options

Script: Real-time Recording

- Script will allow a user to record an audio file in real-time.
- This audio will then be transcribed into text and passed into the translation API.
- User gets a selection for the desired language they want to translate to.

```
# Function that handles microphone recording
def record_mic():
    # Ask user if they want to record until silence is detected or for a set amount of seconds.
    print("\n-----\n")
    print("\n\n1 Record until silence\n")
    print("\n2 Record for amount of seconds\n")
    mic_choice = input("Choice: ")

    # Depending on the choice, it will use SoX to record directly to a file named: 'mic_audio.flac'.
    if mic_choice == "1":
        input("Press enter to begin recording...\n")
        subprocess.run('rec mic_audio.flac silence 1 0.1 3% 1 3.0 3%', shell=True, check=True, executable='/bin/bash')
        audio_playback = input("\nPlay audio back? (Y/N): ")

        # If user wants to hear what they just recorded, play audio file.
        if audio_playback.upper() == "Y":
            subprocess.run('play mic_audio.flac', shell=True, check=True,
                           executable='/bin/bash')
        elif audio_playback.upper() == "N":
            print("\nSkipping audio playback...")

    # Set static amount of seconds to record for.
    elif mic_choice == "2":
        record_seconds = input("\nHow many seconds of recording?: ")
        input("Press enter to begin recording...\n")
        subprocess.run('rec mic_audio.flac trim 0 ' + record_seconds, shell=True, check=True, executable='/bin/bash')
        audio_playback = input("\nPlay audio back? (Y/N): ")

        # If user wants to hear what they just recorded, play audio file.
        if audio_playback.upper() == "Y":
            subprocess.run('play mic_audio.flac', shell=True, check=True,
                           executable='/bin/bash')
        elif audio_playback.upper() == "N":
            print("\nSkipping audio playback...")

    print("#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$")

    # Headers for the upcoming API call.
    stt_headers = {
        'Content-Type': 'audio/flac',
    }

    file_input = 'mic_audio.flac'

    print("\nLOADING....\n")

    # Speech to text audio input file
    stt_data = open(file_input, 'rb').read()
    audiorresponse = requests.post(
        'https://api.us-east.speech-to-text.watson.cloud.ibm.com/instances/547a7142-216a-448c-9c5a-9ff698316466/v1/recognize',
        headers=stt_headers, data=stt_data, auth=('apikey', 'nhPSxINUql-C6Gdy8MPwChLxNLzX1MkNzNACrMRv1Eor'))

    # Set encoding for json output and serialize data.
    audiorresponse.encoding = 'utf-8'
    audiojson = audiorresponse.json()
```

API : Language Detection

- We use the language-translator API to detect the current language found in the transcription.
- With this information we can correctly translate+transcribe back and forth between multiple languages.

```
# Just make a new variable for the detected language of the audio
CURRENT_LANGUAGE = str(detectedLang_json["Languages"][0]["language"])

# Initial language selection screen
while True:
    print("Select language to translate to: \n")
    print("1) Spanish\n")
    print("2) French\n")
    print("3) Portuguese\n")
    print("4) German\n")
    print("5) Simplified Chinese\n")
    print("6) Japanese\n")
    print("7) English\n")
    print("-----")
    language_choice = input("\nChoice: ")

    # Do a check to see if current language == the language you are trying to translate to.
    # If so, deny this and repeat until valid choice.
    if CURRENT_LANGUAGE == "es" and language_choice == "1" or language_choice.upper() == "SPANISH":
        print("\nAudio is already in Spanish!\n\n")
        pass
    elif language_choice == "1" or language_choice.upper() == "SPANISH":
        translationLang = CURRENT_LANGUAGE + "-es"
        return translationLang
    elif CURRENT_LANGUAGE == "fr" and language_choice == "2" or language_choice.upper() == "FRENCH":
        print("\nAudio is already in French!\n\n")
        pass
    elif language_choice == "2" or language_choice.upper() == "FRENCH":
        translationLang = CURRENT_LANGUAGE + "-fr"
        return translationLang
    elif CURRENT_LANGUAGE == "pt" and language_choice == "3" or language_choice.upper() == "PORTUGUESE":
        print("\nAudio is already in Portuguese!\n\n")
        pass
    elif language_choice == "3" or language_choice.upper() == "PORTUGUESE":
        translationLang = CURRENT_LANGUAGE + "-pt"
        return translationLang
    elif CURRENT_LANGUAGE == "de" and language_choice == "4" or language_choice.upper() == "GERMAN":
        print("\nAudio is already in German!\n\n")
        pass
    elif language_choice == "4" or language_choice.upper() == "GERMAN":
        translationLang = CURRENT_LANGUAGE + "-de"
        return translationLang
    elif CURRENT_LANGUAGE == "zh" and language_choice == "5" or language_choice.upper() == "CHINESE" or language_choice.upper() == "SIMPLIFIED CHINESE":
        print("\nAudio is already in Chinese!\n\n")
        pass
    elif language_choice == "5" or language_choice.upper() == "CHINESE" or language_choice.upper() == "SIMPLIFIED CHINESE":
        translationLang = CURRENT_LANGUAGE + "-zh"
        return translationLang
    elif CURRENT_LANGUAGE == "ja" and language_choice == "6" or language_choice.upper() == "JAPANESE":
        print("\nAudio is already in Japanese!\n\n")
        pass
    elif language_choice == "6" or language_choice.upper() == "JAPANESE":
        translationLang = CURRENT_LANGUAGE + "-ja"
        return translationLang
    elif CURRENT_LANGUAGE == "en" and language_choice == "7" or language_choice.upper() == "ENGLISH":
        print("\nAudio is already in English!\n\n")
```




Demo Time

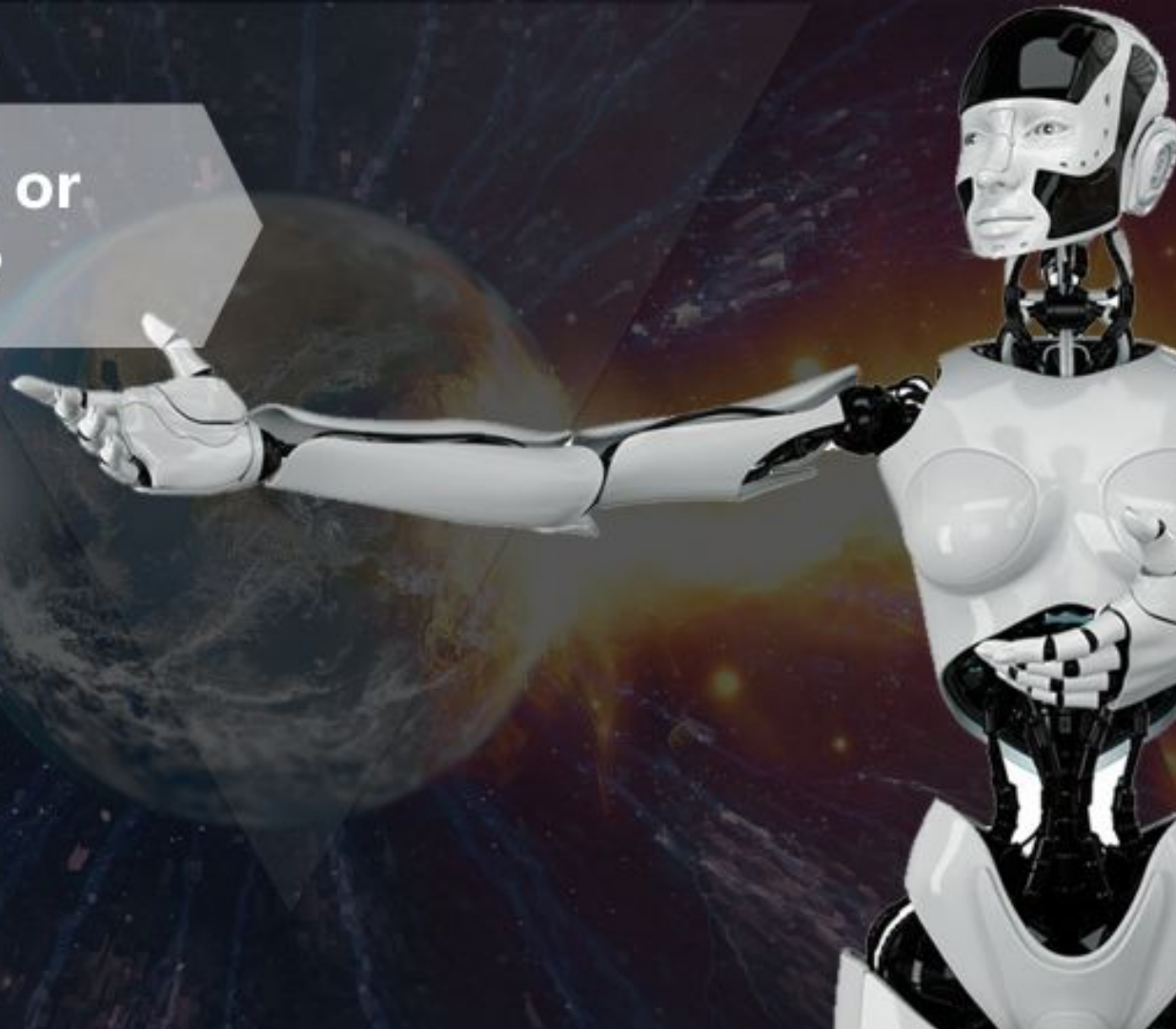
1

Reference

1. <https://cloud.ibm.com/docs/speech-to-text?topic=speech-to-text-science>
2. <https://www.ibm.com/cloud/watson-speech-to-text>
3. <https://cloud.ibm.com/apidocs/speech-to-text?code=python>
4. <https://www.ibm.com/cloud/learn/speech-recognition>
5. https://www.youtube.com/watch?v=A9_00gW1LZU
6. <https://www.youtube.com/watch?v=lwpi1Lm6dFo>
7. <https://cloud.ibm.com/apidocs/language-translator>



Any Questions or
Comments ?





Wentworth
INSTITUTE OF TECHNOLOGY



THANK YOU !