Using speech-to-text recognition, text summarization and web scraping to detect websites for blind people

Apu Islam , MD Abir Anam Bhuiyan , Pratick Bhowmick , Annajiat Alim Rasel , MD. Mustakin Alam ,
MD. Farhadul Islam , and Md Humaion Kabir Mehedi

Department of Computer Science and Engineering
BRAC University
66 Mohakhali, Dhaka - 1212, Bangladesh
{*apu.islam, md.abir.anam.bhuiyan, humaion.kabir.mehedi*
pratick.bhowmick, md.mustakin.alam, md.farhadul.islam}*@g.bracu.ac.bd*
annajiat@bracu.ac.bd

***Abstract*—The modern world has seen much advanced technology to provide assistance to people with special disabilities. A giant leap forward came with the development of voice recognition in popular search engines such as Google. However, this function is limited to some extent as keywords are not always enough for the search engine to know exactly what the person is seeking, hence the top-most link is not always the result being sought. This research applies web scraping technique and text summarization along with the speech-to-text recognition to create an application that will guide the user, especially those who are sight impaired, to their desired result.**

***Index Terms*—Web scraping, voice recognition, text summarization, speech-to-text.**

I. Introduction

The sole goal of engineers and scientists is to make life easier and more convenient for everyone. The popular search engine Google makes approximately 8.5 billion searches a day [1], which shows how useful and crucial its usage is in daily life. The introduction of voice search option further makes the availability of this engine to those who find it difficult to type and search. Yet a category remains- those who are sight impaired or with low visions. We have implemented several techniques to allow them to search in Google smoothly using only voice.

In the modern era, speech technologies play a crucial role as assistive technology for those who are blind or visually impaired in removing the obstacles such as accessing written information, operating devices with rather complex user interfaces and others [2]. A website may contain a lot of information, some that are external and not needed by the user, thus it becomes time consuming for users to find the information they are looking for; which is why condensing and summarizing the text resources becomes significantly more crucial [3]. Furthermore, web scraping is more quicker and more efficient and may be used to collect and aggregate data from hundreds or even millions of pages for processing and information extraction. This method is excellent for a range of purposes [4].

Our system initiates by greeting the user and makes a query if the user wants to look up anything. Based on the given input Google will generate several links. This is the main role of the system; utilizing web scraping technique it summarizes the information in each of those links and reads back in audio format to the user. The user can then proceed to whichever link of their preference. For this purpose we have used several libraries (mainly Python, Whisper) and transformers (Happy Face).

Problem Statement: Due to the lack of presence of a free system that enables people with sight disabilities or low visions to operate digital applications, it is crucial to find such a process and make it accessible to everyone.

II. Related Works

The fundamental element that has to be emphasized is accessibility-aware design and development, as this will enable solutions to be shared among manufacturers and service providers, allowing accessibility to be consistently given at least in public spaces, facilities, and resources [2].

Web scraping has been shown to have a wide range of applications. Utilizing online scraping tools would cost more up front, but would continually pay for themselves as a type of short-term loss, long-term gain. Additionally, compared to human entry, employing a web scraping technology would produce data that is far more extensive, accurate, and consistent [4].

An automatic-text summarization (ATS) system's primary goal is to provide a summary that condenses the key points of the input material into a smaller amount of space while minimizing repetition. without having to read the complete content, the ATS systems assist users in understanding the key aspects of the original material. Users will gain from the

978-1-6654-6316-4/22/$31.00 © 2022 IEEE

automatically generated summaries and will spend less time and effort as a result. [3]

III. Methodology

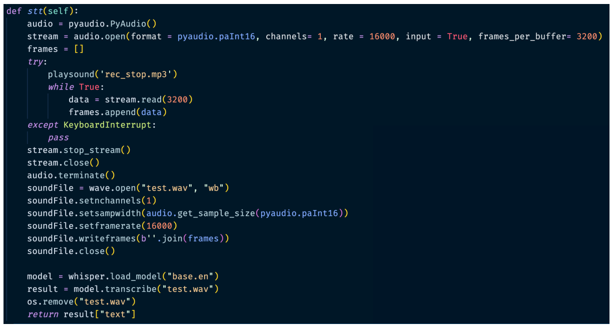
The approach of our system is that it obtains an audio input from the user, and then among the generated links, applying web-scraping and text summarization techniques it prepares a summary. The summary is then converted back into speech and read aloud to the user. In this paper, we decided to use whispers and transformers for speech recognition to get decent accuracy to capture the speech using the recognition.

(i) Whisper: It is an automatic speech recognition system, sometimes referred to as automatic speech recognition (ASR). This system, which is used to gather data from the web, has been trained on 680,000 hours of multilingual and multitask supervision. To produce more robust accents, background noise, and technical jargon, large and diverse datasets are used. We demonstrate how using a dataset of size and variety increases resilience against accents, background noise, and technical terminology. Additionally, it facilitates both multilingual transcription and English translation from those languages. Since it is an open-source project, more contributions will strengthen the system [5].

(ii) Transformers: It is a product that offers API and tools for training cutting-edge pretrained models. Its models are capable of supporting a variety of tasks, including text categorization, named entity recognition, question answering, language modeling, summarization, translation, multiple choice, and text creation.

These are the resources that were utilized in the construction of a system that aids in providing text summaries to the blind using voice recognition in several languages.

- Implementation + description of codes-



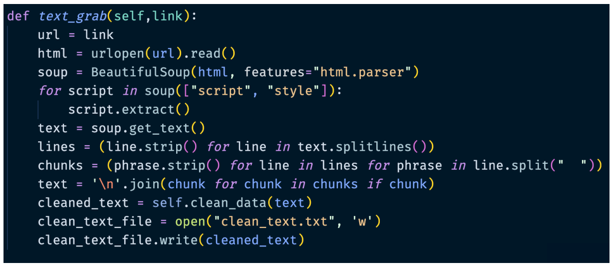
```
def stt(self):
    audio = pyaudio.PyAudio()
    stream = audio.open(format = pyaudio.paInt16, channels= 1, rate = 16000, input = True, frames_per_buffer= 3200)
    frames = []
    try:
        playsound('rec_stop.mp3')
        while True:
            data = stream.read(3200)
            frames.append(data)
    except KeyboardInterrupt:
        pass
    stream.stop_stream()
    stream.close()
    audio.terminate()
    soundFile = wave.open("test.wav", "wb")
    soundFile.setnchannels(1)
    soundFile.setsampwidth(audio.get_sample_size(pyaudio.paInt16))
    soundFile.setframerate(16000)
    soundFile.writeframes(b''.join(frames))
    soundFile.close()

    model = whisper.load_model("base.en")
    result = model.transcribe("test.wav")
    os.remove("test.wav")
    return result["text"]
```

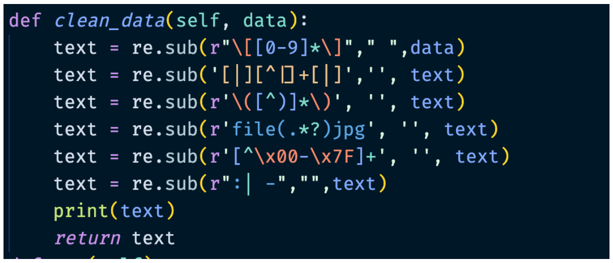
Using the library pyaudio (used for recording audio), input sound is taken into a stream. This stream is placed in a wav file. Using the whisper model described above, speech is extracted from the wav file.

10 links are generated from the query that was provided earlier. The function then goes inside those links and extracts texts using BeautifulSoap (web scraping library). We are using the clean_date() function to remove unnecessary tags.

Clean_data function to remove any unnecessary symbols, attributes and tags.



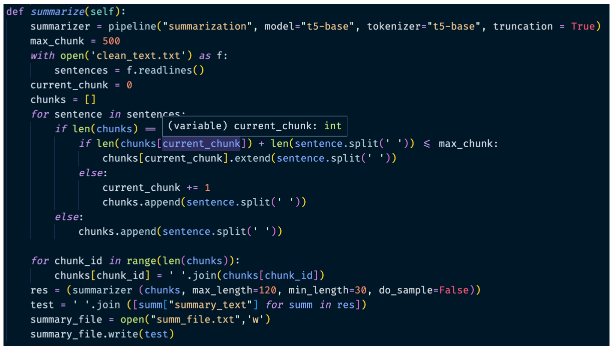
```
def text_grab(self,link):
    url = link
    html = urlopen(url).read()
    soup = BeautifulSoup(html, features="html.parser")
    for script in soup(["script", "style"]):
        script.extract()
    text = soup.get_text()
    lines = (line.strip() for line in text.splitlines())
    chunks = (phrase.strip() for line in lines for phrase in line.split("  "))
    text = '\n'.join(chunk for chunk in chunks if chunk)
    cleaned_text = self.clean_data(text)
    clean_text_file = open("clean_text.txt", 'w')
    clean_text_file.write(cleaned_text)
```



```
def clean_data(self, data):
    text = re.sub(r"\[[0-9]*\]"," ",data)
    text = re.sub('[|][^|]+[|]','', text)
    text = re.sub(r'\([^)]*\)', '', text)
    text = re.sub(r'file(.*?)jpg', '', text)
    text = re.sub(r'[^\x00-\x7F]+', '', text)
    text = re.sub(r":| -","",text)
    print(text)
    return text
```

IV. Experimental Result Analysis

Since most of our results are verbal, we are showing an example of text summarization obtained before reading out to the user.



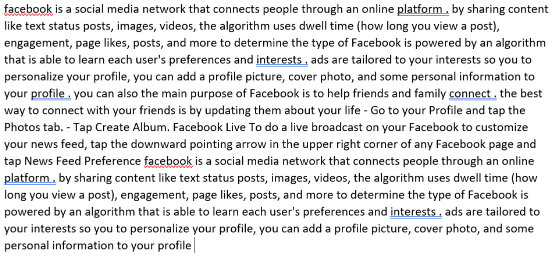
```
def summarize(self):
    summarizer = pipeline("summarization", model="t5-base", tokenizer="t5-base", truncation = True)
    max_chunk = 500
    with open('clean_text.txt') as f:
        sentences = f.readlines()
    current_chunk = 0
    chunks = []
    for sentence in sentences:
        if len(chunks) ==
            if len(chunks[current_chunk]) + len(sentence.split(' ')) ≤ max_chunk:
                chunks[current_chunk].extend(sentence.split(' '))
            else:
                current_chunk += 1
                chunks.append(sentence.split(' '))
        else:
            chunks.append(sentence.split(' '))

    for chunk_id in range(len(chunks)):
        chunks[chunk_id] = ' '.join(chunks[chunk_id])
    res = (summarizer (chunks, max_length=120, min_length=30, do_sample=False))
    test = ' '.join ([summ["summary_text"] for summ in res])
    summary_file = open("summ_file.txt",'w')
    summary_file.write(test)
```

Processed text from the clean_data function is divided into small chunks. These chunks are summarized individually. Finally, all the small summaries are compiled together.

https://blog.hubspot.com/marketing/how-to-use-facebook

Upon receiving the audio input- 'how to use Facebook', the link above is generated which our system summarizes into following-



facebook is a social media network that connects people through an online platform . by sharing content like text status posts, images, videos, the algorithm uses dwell time (how long you view a post), engagement, page likes, posts, and more to determine the type of Facebook is powered by an algorithm that is able to learn each user's preferences and interests . ads are tailored to your interests so you to personalize your profile, you can add a profile picture, cover photo, and some personal information to your profile . you can also the main purpose of Facebook is to help friends and family connect . the best way to connect with your friends is by updating them about your life - Go to your Profile and tap the Photos tab. - Tap Create Album. Facebook Live To do a live broadcast on your Facebook to customize your news feed, tap the downward pointing arrow in the upper right corner of any Facebook page and tap News Feed Preference facebook is a social media network that connects people through an online platform . by sharing content like text status posts, images, videos, the algorithm uses dwell time (how long you view a post), engagement, page likes, posts, and more to determine the type of Facebook is powered by an algorithm that is able to learn each user's preferences and interests . ads are tailored to your interests so you to personalize your profile, you can add a profile picture, cover photo, and some personal information to your profile

V. Conclusion

In this paper, we tried to produce a tool that uses speech recognition then searches for data through web scraping then with the help of the results, a text summarization of the search is provided. We can further work in the future to increase the accuracy of the summarization for making it more convenient for blind people.

References

[1] M. Mohsin, “10 google search statistics you need to know in 2022,” Feb 2022. [Online]. Available: https://www.oberlo.com/blog/google-search-statistics

[2] D. Freitas and G. Kouroupetroglou, “Speech technologies for blind and low vision persons,” *Technology and Disability*, vol. 20, no. 2, p. 135–156, 2008.

[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A comprehensive survey,” *Expert Systems with Applications*, vol. 165, p. 113679, 2021.

[4] M. Khder, “Web scraping or web crawling: State of art, techniques, approaches and application,” *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 3, p. 145–168, 2021.

[5] A. Radford, “Introducing whisper,” Dec 2022. [Online]. Available: https://openai.com/blog/whisper/