International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

To implement the Speech Synthesis in the Process of Web Scraping

Virag Umathe[1], Shubham Chopade[2]

[1,2]Department of Electronics and Telecommunication, Pimpri Chinchwad college of Engineering Pune, Maharashtra Pune.

Abstract: Most of the real-life data is available in the analog form, data being in the form of speech, light, and electric pulses. The data that we generated in the last 2 years is more than the total data generated in human history[10]. There are search engines that let us access publicly available data. Web scraping is another possible solution that lets the user extract the relevant data from websites[18] by just running a python script[12]. However, the users still need to rewrite the Python script every time they want to extract a new form or structure of data. This paper intends to automate this process of re-writing the script by using speech synthesis as a mediator. This process introduces a smart voice-operated assistant that can interact and search for the relevant data demanded by the user. Additionally, by employing this it will make the user save on time on writing the script and spend the time analyzing the data obtained. Essentially, the data collection process will become faster and automated. This research is of primitive importance because automation leads to productivity which is the basic need of the new industry.

Keywords: web scraping, web collection, web data extraction, World Wide Web, Hypertext Transfer Protocol, Speech to text (STT)

Introduction

The amount of data available on the internet is humongous. There is a need to develop a system that is helpful in getting the data efficiently and maintaining the relevancy up to the mark. Although there are several other manual techniques of accessing the available information on web pages, the time required for an individual to find the correct link to get the accurate information is substantially large. This follows in the wastage of a lot of time in finding the best possible results. However, search engines like Google have helped in ameliorating this issue by prioritizing the pages with PageRank Algorithm [17]. PageRank is the way of deciding the importance of web pages for that specific keyword entered on the Google search engine. It is a link analysis algorithm that assigns numerical weights to all the hyperlinked nodes within a network and further decides the rank in the search results based on its relevance with the keyword entered. Despite such efficient algorithms, the searching part needs to be done by an individual and by visiting every web page to collect the information. Furthermore, the information collected is not organized initially, it needs to be organized and then can be put into use. This approach is notably inefficient and could be made simpler.

Application Programming Interface (API) is another automated way of collecting the data from the web[8]. But it has some inherent limitations when compared with the method of web scraping. API acts as a mediator between two software and helps in establishing communication between them. Basically, the data that is needed is coded into JSON format and is passed to the API and in return, another JSON format is obtained. However, there are restrictions while using JSON files and that allows access to the only exclusive type of data fields. This is a major limitation of using API for extracting data from web pages because the data formats are variable across the internet and it demands dynamic code that can be customized to access various formats of data fields. Since API is not capable of doing so, it cannot be used for efficiently extracting data from web pages.

Web scraping is an automated process of extracting the data from different web pages [21] present on the internet. It provides an ideal solution in terms of flexibility it offers for extracting different formats of data available. A Python script is fetched to a web scraper which can be customized as per the user needs. Initially, the XML/HTML code format of the web-page is scanned and the relevant text, links, or images are extracted separately—which is decided by the script provided to the scraper [23]. There are multiple libraries like Beautifulsoup[1], Scrapy[2], Selenium[3], and urllib[4] for scraping the web pages. The above-mentioned libraries are based on the Python programming language [12]. Other languages like Java can also be used for this purpose, however, the programming complexity is relatively more than that of Python.

This research intends to automate the initial stage of writing the Python script for the scraper manually and essentially accepting speech input from the user for the same. For that, there is a library in Python "SpeechRecognition"[5][6] which is an excellent open-source alternative to use for this purpose. It needs the user to set the microphone as an input source for speech and fetch the result to the recognizer. It ultimately converts the

speech into text which is understood by the scraper. The format of the data required is identified and a corresponding Python script is generated for the same. The script is provided as an input to the scraper that assists in finding the appropriate results and gathering the information for the respective data fields. *Further, Text To Speech (TTS)[14]is an advanced API that is used as an artificial speech synthesizer and converts the speech input to text format and can be implemented in software as well as hardware.*

Objective

To design a web scraping system that can get the inputs from the user by using speech synthesis and convert that into the relevant python script.

To describe the rules and regulations while using web scraping on web pages.

Methodology

The process consists of multiple steps. The first step is Speech Synthesis that accepts voice as the input and converts it into equivalent text using Python [11][12]. Speech Synthesis [16] in itself is divided into three stages: the initial stage focuses on reducing the ambiguity of the input given; the next stage converts the interpreted signal into text and finally, the text is extracted to extract some information out of the string to take further action.

Introducing Speech Synthesis in web scraping

Speech synthesis is basically the generation of spoken language by machine on the basis of written inputs and generation of the voice as we feed the output as an input to the synthesizer [14].

1. Initial Stage

Reading words sounds easy, but if you've ever read a book to a young child [15], understanding it might not be as trivial as it is to us. The main obstacle is that written text is ambiguous and can be interpreted to understand in different contexts. For instance, consider numbers, dates, times, abbreviations, acronyms, and special characters that need to be turned into words. The number 1843 might refer to a number of items ("one thousand eight hundred and forty-three"), a year or a time ("eighteen forty-three"), or a padlock combination ("one eight four three"), each of which is read out slightly discordantly. While humans follow the sense of what's written and figure out the pronunciation that way, computers generally don't have the ability to do that, so they engage statistical probability techniques or neural networks to arrive at the most likely pronunciation instead. Similarly, if the word "year" occurs in the same sentence as "1843," it might be reasonable to guess this is a date and pronounce it "eighteen forty-three." So the initial stage in speech synthesis, which is generally called pre-processing or normalization, is all about reducing ambiguity and narrowing down the many different ways you could read a piece of text into the one that's the most suited.

2. Input Stage

Having figured out the words that need to be known; the speech synthesizer now has to generate the sounds that make up those words. In the input part, we give the voice as input and the synthesizer converts it to text and initiates the search or process. The search is performed in a set of predefined data sets. For example, consider a dictionary of words consisting of the pronunciation of each word and its possible usage in the sentence. This will help in determining the accurate phonics of each word that is received as input to this stage. A final pronunciation of the input speech signal is obtained and is passed onto the next stage.

3. Output Stage

The final stage of Speech Synthesis focuses on converting the clear signals from the previous stages into a string of characters [16] which is understandable by the machine. Essentially, the signal is fed to the synthesizer for it to convert into the text string. This string is fed to the web scraper to perform further operations. The main advantage of including speech synthesis in this process is, the machine will take care of writing and running the Python script without much interference of the user. Therefore, saving the net time required to scrape the data will be substantially reduced. Further, the text is processed and analyzed and it is matched with different formats of data entries on different web pages. The type of data that is to be inserted into the data fields is decided by the structure of the web page available on the internet. The structure of a web page is explained further.

Structure of a typical web page

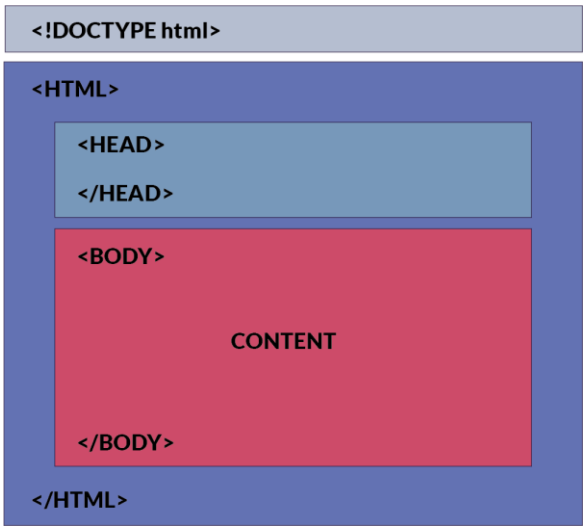

Fig. 1 Structure of a webpage

The head tag consists of the headings on any given webpage. From that, we can extract the title of the data which is further explained in the body [22]. All the metadata (data about data basically all links style sheets etc.) corresponding to the website comes in the head tag it is basically the index of the website and all the text, media comes in the body text contain code and text in website and media contain links, photos, sound, locations, videos, etc.

Inspecting website

Inspect is a browser-specific feature that gives access to the source code of the website and gives us the exact plot of that website that is in which tab in which attribute specific element is written.

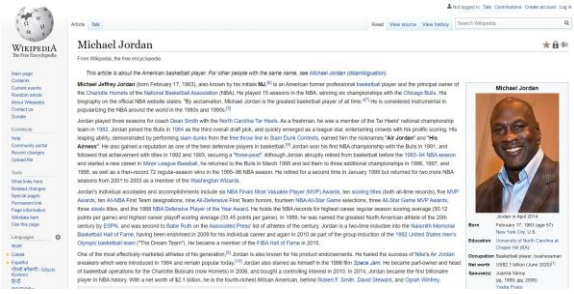

Fig. 2 Wikipedia Page

The above image shows a Wikipedia page having multiple elements [9] such as images, headings, descriptions, lists, etc. To extract the information from the page, we need to see the source HTML code of the page. That can be done by turning on inspect the website. As shown below, the source code is displayed with respective tags.

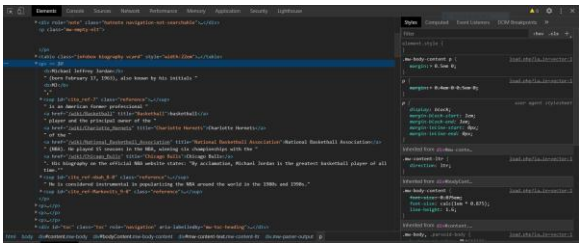

Fig. 3 Inspect a webpage

HTTP (Hypertext Transfer Protocol)

HTTP is the set of rules for transferring text, image, sound, video and other media files on the world wide web, As one opens the internet browser starts the use of the protocol [19][20]. HTTP is the protocol that runs on top of the TCP/IP (Transmission Control Protocol / Internet Protocol) which is used for the interconnection between devices on the internet.

Requests

Before actually starting with web scraping there must be the establishment of the relation between two websites so that the website can be scratched using a scraper. For request, python library Requests [7] can be used

Rules for scraping the web

1. You should check a website's Terms and Conditions before you scrape it[24]. Be careful to read the statements about the legal use of data. Usually, the data you scrape should not be used for commercial purposes.

2. Do not request data from the website too aggressively with your program (also known as spamming), as this may break the website. Make sure your program behaves in a reasonable manner (i.e. acts like a human). One request for one webpage per second is good but not more.

3. The layout of a website may change from time to time, so make sure to revisit the site and rewrite your code as needed.

The process of web scraping using speech synthesis

The process starts with accepting the information input from the user as the voice and then the process of web scraping and the output in the form of the voice using synthesizer, we can understand it clearly using the following flow

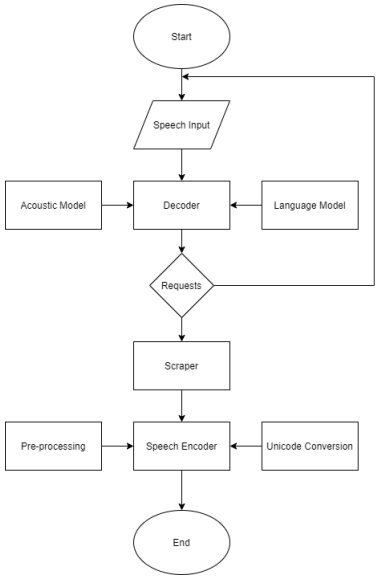

Fig. 4 Flow chart

Flow chart Explanation

Input Text: For taking input as a voice of the user

Speech Processing: Processing of the text and initiate scraping

Web Scraping: Requests from the website and scrape the web surfaces

The output of the web scraping: Result of the web scraping in media

Speech Synthesis: Text to Speech conversion [16] by the speech synthesizer

Output: Output as synthesizer speech

Conclusion

Web Scraping using speech synthesis is the implementation that can scrape the web by just speech and also it can give the output in a speech which will contribute towards the data science, data analysis and in other domains is easy and in a massive amount, which also automates the industry also it will boost up the data retrieval and subject analysis like terms

References

- Documentations and papers

1. Online document

   Beautifulsoup4 https://www.crummy.com/software/BeautifulSoup/bs4/doc/

2. Online document

   Scrapy v2.1

   https://docs.scrapy.org/en/latest/

3. Online document

   Selenium

   https://www.selenium.dev/selenium/docs/api/py/api.html

4. Online document

   urllib3

   https://urllib3.readthedocs.io/en/latest/user-guide.html

5. Online document

   Speech Recognition https://pypi.org/project/SpeechRecognition/

6. Online document GTTS(google talk to speech) https://pypi.org/project/gTTS/

7. Online document

   Requests https://pypi.org/project/requests2/

8. Java Web Scraping Handbook https://www.scrapingbee.com/download/webscrapinghandbook.pdf

9. Wikipedia the free encyclopedia for scraping the data from Michael Jordan page. https://en.wikipedia.org/wiki/Michael_Jordan

10. Online document

    Big Data Statistics 2020 https://techjury.net/blog/big-data-statistics/#gref

11. Pratiksha Ashiwal, S. R. Tandon, Priyanka Tripathi, Rohit Miri Web information retrieval using python and beautifulSoup Volume 4, Issue VI, June 2016, ISSN 2321-9653

12. K.R. Shrinath (2017) Python - The Fastest Growing Programing Language Volume 04, Issue 12, Dec 2017, e-ISSN 2395-0056, e-ISSN 2395-0072

13. Data collection methods on the web for informetric purpose A review and analysis scientometrics 50(1), 7-32

14. Yocheved Levitan TTS and Data Selection improving speech synthesis for low resource language yocheved.levitan@gmail.com

15. Matthew B Hoy (2018) Alexa, Siri, Cortana and more: an introduction to voice assistants

16. Nwakanma Ifeanyi, Oluigbo Ikenna, Okpala Izunna (2014) Text to Speech Synthesis

17. Stanford InfoLab, The PageRank Citation Ranking: Bringing Order to the Web

18. Alexa Skill Builder's Guide

19. Sh. Rajinder Singh, Dr. Satish Kumar (2016) Overview of World Wide Web Protocol Hypertext Transfer Protocol and Hypertext Transfer Protocol Secure

20. Volker Turau, HTTPExplorer: Exploring the Hypertext Transfer Protocol

21. Deepak Kumar Mahto, Lisha Singh A dive into Web Scraper world 31 oct 2016, IEEE

22. Carlos Flavian, Raquel Gurrea, Carlos Orús Web design: A key factor for the website success

23. Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode. An Overview On Web Scraping Techniques And Tools 2018

24. Vlad Krotov, Leiser Silva Legality and Ethics of Web Scraping 2018