Beautifulsoup4 for Data Scraping in Building Lyrics-guessing Bot API

Muhammad Gumilang - 13514092
Informatics and Computer Science Major
School of Electrical Engineering and Informatics
Bandung Institute of Technology, Ganesha Street 10, Bandung 40132, Indonesia
13514092@std.stei.itb.ac.id

***Abstract* — Building a bot for applications is trending right now, because they serve their own API and we can also involve another APIs to collaborate and make a functional bot. A bot needs a data to be processed, which can be gathered from web, specifically its page resource, by a tool for data scraping. One of the tools for data scraping is Beautifulsoup4.**

***Keywords* — Beautifulsoup4, Python, Data Scraping, musicaLine, Line Bot API, Lyrics**

I. Introduction

An application needs inputs to be processed. After that, the application will yield the result. In the process of the application, there might be data to support the progression. It can also alter the result later. Some inputs and the supporting data are collected from a user. But what if there is no user for needed input? What if we need to prepare a lot of data in a short time?

Well, we can use a program to collect a lot of data quickly. There's a method called *data scraping*. It gathers data from web, specifically its page resource. One of the tools for data scraping is Beautifulsoup4. Beautifulsoup4 let us filter which part of the webpage we want to obtain. It also comes with a lot of feature to help us sort the data we need.

One of the programs that uses data scraping is musicaLine, which is a Line Bot API. The data scrapped from the web is lyrics needed to be given to user to be guessed the title or the artist of the lyrics. The webpage that is scrapped for its lyrics is Metrolyrics. Besides just gathering raw data from Metro Lyrics, we still need to do data cleansing, which is get rid of or add any additional part we need in order to make a better usable data.

II. Literature Study

A. *Line Bot API*

Line Bot API is a bot API that lets us interact with individual user through either Line Offcial or Line@ account. We can customize our own response that will be sent directly and automatically to users that add our bot as their friends. We can also send interactive message to users from our bot API at any given time. The requests are sent using JSON-based APIs.[4]

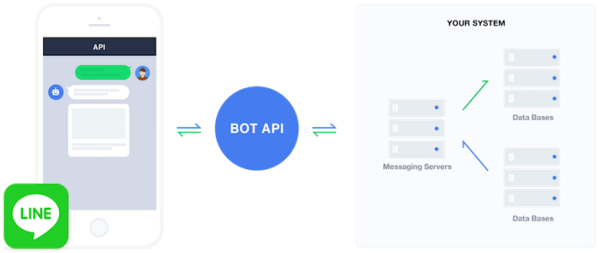

Figure 1. Schema of Line Bot API

Information is sent from the LINE platform to the user when the user either sends a message (message) or performs such action such as adding our account as a friend (operation). The information is sent via HTTP to our registered URL. Then, a JSON string is generated in the request body for operations and it varies depending on the type of operation.

B. *Data Scraping*

Data scraping is technique to gather data from human-readable output from another program. The key element that differs data scraping from any regular parsing is that the output being scraped was intended for display to an end-user, rather than input to another program. Data scraping usually ignores scraping binary data such as images or multimedia data, display formatting, rendundant labels, and other information which is irrelevant or hinders automaed processing.[1]

There are three types of data scraping techniques; screen scraping, web scraping, and report mining. Screen scraping is a practice of reading text data from a display screen. Web scraping is extracting data from web through DOM parsing which we will go deeper about in this paper. Report mining is extracting from human-readable computer reports.

C. *Web Scraping*

Web scraping software access the World Wide Web directly using Hypertext Transfer Protocol or web browser. It can be done manually by user software.

There are some techniques that are considered as web scraping. Sometimes the best web-scraping technology is human's manual examination and copy-and-paste. It is

good at filtering what we need but takes a lot of time to finish the task. We can also use text pattern matching to find what we're looking for with the help of such as UNIX grep or regular-expression pattern matching. Static and dynamic webpages can also be extracted by posting HTTP requests to the web server using socket programming. Another technique is DOM parsing which retrieving data based on the DOM (Document Object Model) tree that is controlled and parsed by browsers.

D. *Beautifulsoup4*

According to Beautifulsoup4's documentation webpage, Beautifulsoup4 is a python library for pulling out data out of HTML or XML files and it works with wny parser we choose to provide idiomic ways of navigating, searching, and modifying the parse tree. Beautifulsoup4 saves a lot of hours of work.

Beautifulsoup4 reads the parse tree and users can filter the data needed by specifying the DOM element. For intance, if we want to get the title of the webpage, we can just tell Beautifulsoup4 to get the element *<title>*. Beautifulsoup4 will retrieve the first specified element it finds. Moreover, we can have a list of the DOM element needed by telling Beautifulsoup4 to find all the DOM element specified.

III. METHODS AND TOOLS

A. *Scraping Tools*

For the data of the lyrics, we will be scraping using Python with the help of Beautifulsoup4. We also need to import 'request' library in order to HTTP request the link we want to scrape. The data we received will be stored in JSON form. There are some methods to scrape the lyrics; by link, by artist, or by text file (a list of artists). There are also some methods made in separated files to involve API from Genius to get proper artists' data, while the lyrics are scraped from Metrolyrics.

Before building the scraping methods, first, the Python file receives input from user through the 'argv' by importing 'sys' and read/write external file by importing 'os'. Because the data will be stored in JSON form, we also need to import 'json' to the Python file. Therefore, we need to create .json file first to store the data and specify the filename in our program of that .json file.

To scrape by one link, the file need an input from the user which is the link from the Metrolyrics. For instace, when we want to scrape 'Hello' lyrics by Adele, we can insert "http://www.metrolyrics.com/hello-lyrics-adele.html" as the input. So when we execute the program we add two input, which is the .json file to store the result and the link, it goes like this:

```
C:\Users\MuhammadGumilang\Documents\GitHub\LINE-dev-huwalaumba\scrape>python scrape_m
 test http://www.metrolyrics.com/hello-lyrics-adele.html
b'Hello'
```

Figure 2. Command Line to Execute Scraping by Link


```
[
    {
        "feat_artist": null,
        "title": "Hello",
        "alt_name": [
            "Adele Adkins"
        ],
        "pyongs_count": 173,
        "genius": true,
        "artist": "Adele",
        "lyrics": "Hello, it's me\nI was wondering if
    }
]
```


Figure 3. Result of the Scraping in JSON Form

To scrape by artist, we need to be smart about the pattern the website uses to draft the artists. If we take a look to the artists' profiles, they have common pattern for their website links.

```
http://www.metrolyrics.com/adele-lyrics.html
http://www.metrolyrics.com/ed-sheeran-lyrics.html
http://www.metrolyrics.com/coldplay-lyrics.html
http://www.metrolyrics.com/rihanna-lyrics.html
http://www.metrolyrics.com/taylor-swift-lyrics.html
http://www.metrolyrics.com/lady-gaga-lyrics.html
```

Figure 4. List of Some Links to Artists' Lyrics

We see that all of them starts with the Metrolyrics domain, artist's name (with stripe replacing space), stripe, lyrics, and '.html'. We can use this pattern by inserting the the input to the pattern. The program then executes the method to scrape by link, because in the webpage of the artist's lyrics, there are links to the lyrics to be scraped, which can be used as the input to scrape by link. It goes like this:

```
C:\Users\MuhammadGumilang\Documents\GitHub\LINE-dev-huwalaumba\scrape>python scrape_metro_ar
tist.py test arctic-monkeys
b'Do I Wanna Know?'
b'Fluorescent Adolescent'
b'R U Mine?'
b'When The Sun Goes Down'
b'I Bet You Look Good On The Dancefloor'
b'505'
b'Cornerstone'
b'I Wanna Be Yours'
b'Brainstorm'
b'Arabella'
```

Figure 5. Command Line to Execute Scraping by Artist

To scrape by file, we list the name of the artists and use the file path as the input of the program. It will then executes program to scrape by artist one by one.

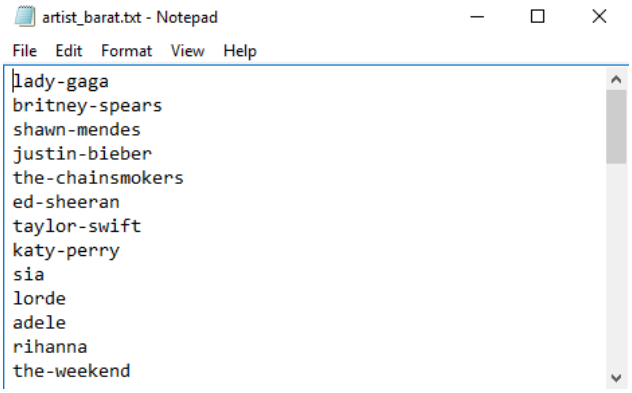

Figure 6. List of the Artists

```
C:\Users\MuhammadGumilang\Documents\GitHub\LINE-dev-huwalaumba\scrape>python scrape_metro_
artist_by_file.py test artist_barat.txt
lady-gaga

b'Bad Romance'
b'Born This Way'
b'The Cure'
b'Million Reasons'
b'Telephone'
b'Paparazzi'
b'Alejandro'
b'Pokerface'
b'You And I'
b'Edge Of Glory'
britney-spears

b'Toxic'
b'Womanizer'
```

Figure 7. Command Line to Execute Scraping by File

The process to get the lyrics from DOM tree is actually simple. First, we need to inspect the page resource of the webpage and find the where the lyrics is and what element that wraps it. For Metrolyrics, all of the lyrics are wrapped in 'p' element with an attribute 'class' called 'verse'.

```
<p class='verse'>Hello, it's me<br>
I was wondering if after all these years you'd like to meet<br>
To go over everything<br>
They say that time's supposed to heal ya<br>
But I ain't done much healing</p><p class='verse'>Hello, can you hea
I'm in California dreaming about who we used to be<br>
When we were younger and free<br>
I've forgotten how it felt before the world fell at our feet</p><p c
And a million miles</p><p class='verse'>Hello from the other side<br
I must've called a thousand times<br>
To tell you I'm sorry<br>
For everything that I've done<br>
But when I call you never<br>
Seem to be home</p><div id="mid-song-discussion" class="js-sd-middle
<div class="author">
<div class="avatar ">
```

Figure 8. Metrolyrics' Lyric Page Resource

Now, in the program, we parse the webpage we have requested. To get the title of the song and artist we use Beautifulsoup4 to find the title element and get its text. The title we retrieved contains the song title and the name of the artist, all white spaces replaced with striped. We slice the text on the part where it separates between the song title and the name of the artist. After that, we remove the stripes.

To obtain the lyrics, we use Beautifulsoup to find the element 'p' that has class 'verse'. All of the lines of the lyrics then accumulated into one variable. In the end, all the data scrapped returned from the method to be stored to .json file specified.

```
def scrape_by_link(link):
    page = requests.get(link)
    page.encoding = 'utf-8'
    html = BeautifulSoup(page.text, "html.parser")

    #Get title and parse
    title = html.find_all('title')[0].get_text()
    title_words = title.split(' ');
    strip_idx = title_words.index("-")
    lyrics_idx = title_words.index("Lyrics")

    #Get artist name
    artist = ""
    for i in range(strip_idx):
        artist += title_words[i]
        if (i != (strip_idx - 1)):
            artist += " "

    #Get song title
    song_title = ""
    for i in range(strip_idx+1,lyrics_idx):
        song_title += title_words[i]
        if (i != (lyrics_idx - 1)):
            song_title += " "
    print(song_title.encode('utf-8'))

    lyrics = html.find_all('p', class_='verse')
    f_lyrics = ""
    for verse in lyrics:
        line = "{0}\n".format(verse.get_text()) + "\n"
        f_lyrics += line
```

Figure 9. Scraping with Beautifulsoup4

B. *API Tools*

In able to make the bot works functionally, we use Node JS to manage requests and responses. The interaction file defines what message the bot needs to receive in order to respond such a specified message. It also use the .json files which is the data of the lyrics that will be shown to the users. Moreover, it will sum the score and manage randomizing songs and lyrics for the user to guess.

The bot can only work in group chat. It will also store the score of the group and will show the leaderboard globally. There are two category to play with this bot, which is western or local songs. Each category will randomize whether the users have to guess the title of the song or the artist of the song. The bot will automatically redact the title in lyrics given.

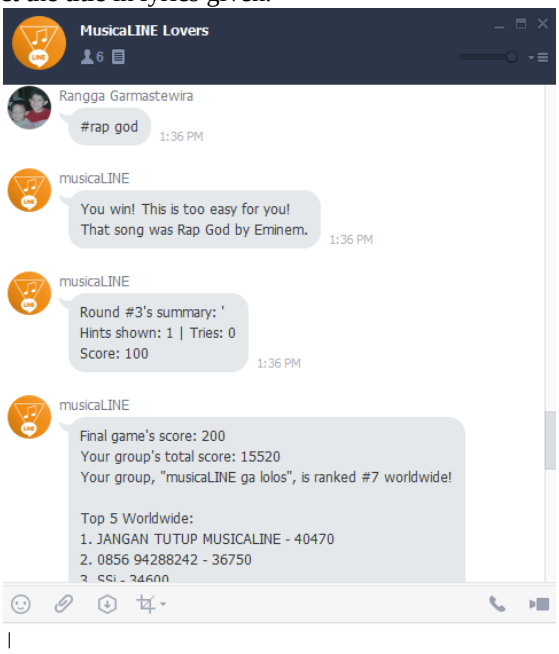

Figure 10. The Interaction Between the Bot and the User

IV. RESULTS

The tools synergized creating a functionally work Line Bot, with hundreds of adders. It has also collected with more than 300,000 score points. Adding more data of lyrics can just be done by running the scraping program, it doesn't have to change a lot of configurations.

With data scraping, we have successfully collected more than 400 lyrics. All of them have been throught data cleansing (to remove unnecessary part of the lyrics or repeating verses). Their other data, such as artist and artist's aliases, has been verified with Genius API, so that users can guess it correctly with more possible options.

V. DISCUSSION

Scraping a webpage to collect its data may be considered illegal. So, we need permission from the organization/individual responsible for the webpage. There

are some webpages that are already restricted, in which the restricted webpages couldn't be parsed.

The idea of having a lot of files per method might be good to manage the files really well, but it might be considered not effecient. It can be redundant. There are three files for scraping, three for editing the artist's profile, and two for data cleansing. All of them separated for the sake of convenience of testing a program. It could be made with lesser files for the easiness on managing and looking file, while also give more efficiency on executing the programs.

VI. Conclusion

Data scraping, or web scraping specifically, is not as hard as what we imagine it would be. By using Beautifulsoup4, we can parse the webpage into a DOM (Document Object Model) tree and filter the data we need. This saves us a lot of hours of work to gather data needed for our application. The data then stored in JSON and will be parsed by the bot to be processed, which is later shown to the users. The bot API use the data that has been through cleansing process and calculate the scores of the users guessing the song title or artist of the lyrics. The hard work has finally paid off to create a working Line Bot API that had a lot of adders.

References

[1] Hemenway, Kevin and Calishain, Tara (2003). *Spidering Hacks*. Cambridge, Massachusetts: O'Reilly. ISBN 0-596-0057-6.

[2] Song, Ruihua, Microsoft Research (September 14, 2007). "Joint Optimization of Wrapper Generation and Template Detection". *The 13th International Conference on Knowledge Discovery and Data Mining.*

[3] Roush, Wade (July 25, 2015). "Diffbot Is Using Computer Vision to Reinvent the Semantic Web". www.xconomy.com Retrieved 2013-03-15

[4] Description about Line Bot API (n.d.). *Overview* (BOT API (Deprecated)). Retrieved May 4, 2017, from https://developers.line.me/bot-api/overview

[5] Overview on Metrolyrics webpages (n.d.). Retrieved May 5, 2017, from http://www.metrolyrics.com/

Declaration

I hereby declare that this paper is my own work, not a copy or translation of others' works, and not an act of plagiarism.

Bandung, 5 May 2017

Muhammad Gumilang
13514092