Software Design

Mini-Project 1: Mining Text from the Web

YeongHwa, Kim

1. Overview

I like listening to music; not only Korean music but also Pop songs. I like the almost genres of music. My standards to choose songs are melody and the lyrics. But actually, when I listen to pop songs, I can't catch the lyrics because it's too fast and the words have a note. I hum the songs without understanding the meaning of that. So, I want to make the program that tells us the topic of the songs. The programs will analysis the lyrics, and find the most frequently using words. Then, although the listener is not good at English, he can roughly know the theme that the singer wants to say.

2. Implementation

To collect the lyrics, I used "Google" for the data source. At first, it has no matter. But it forbad me. So I changed the data source to "Bing". When I search with Bing, I can get a lot of webpages. But, I only need the lyrics in any webpage. So I choose just the first website and download it.

```
url = URL(string=listurl[0])
lyrics = url.download()
```

When I download the url, there are so many unnecessary data. So I used "plaintext" function. And also, there was an Unicode problem. I solved that problem with ".encode('ascii', 'ignore')" by googling. After doing that, I got a list that split by the "\n". I want only the lyrics part, so I find the words which located at the front of the lyrics and the end of the lyrics. In this process, I used the song, "Reflection". The front word was "Get "Reflection" Ringtone", and the last word was "Songwriters". With using ".index()",

```
a = lyricslist.index('Get  "Reflection" Ringtone') + 1
list1 = lyricslist[a:]
b = list1.index('Songwriters')
list2 = list1[:b]
```

Finally, I could get the pure lyrics. But, there was a problem still. I wanted to know the most frequently words, but the list2 was the list that components were sentences. So I break the sentences to the words with using "while".

```
nm = 0
while nm < len(list2):
    string = list2[nm]
    list2[nm] = string.split()
    nm = nm + 1
```

But I used ".split()", so nested list was made. To change nested list to normal list, I make a new list called "listnew". After that, I counted the numbers of the words in listnew by using "{}". And then, I import the Counter and used ".most_common()" so I can get the most common words in the lyrics of "Reflection".

3. Results

I got the list like this:
[('I', 20), ('my', 8), ('am', 7), ('a', 6), ('show', 5), ('that', 5), ('will', 5), ('who', 5), ('reflection', 5), ('When', 4)……]

So, we can guess the singer want to say that "I want to show my reflection."

Also, we can find other song's the most frequently words. For, "Lost Stars", *[('the', 19), ('I', 14), ('And', 9), ('are', 8), ('you', 8), ('up', 7), ('on', 7), ('light', 6), ('we', 6), ('a', 6), ('thought, 6), ('all', 5), ('trying', 5), ('But', 5), ('to', 5), ('is', 5), ('out', 5), ('dark?', 5), ('stars', 5), ('lost', 5), ('there', 4), ('crying', 4), ('and', 4), ('heard', 4), ('see', 4)……]*

So, we can guess the singer want to say that "I and you, we are lost stars, trying to light up the dark."

4. Reflection

At first, I just wondering that I can make the pure lyrics words list. But I did it! Also, I made the words finding code. But there are some problems:

1) The lyrics website which is searched for the first time is not always same. So, we have to find the words which located at the front of the lyrics and the end of the lyrics.
2) After finding the frequently using words, we have to guess the theme of the songs because it just shows the list of the words.

To make more convenient program, some analysis process should be added to find the theme directly. And also, only use the one lyrics site.