FIT5196 Task 1 in Assignment 1

Name: Peiyu Liu

SID: 31153291

Date: 15/04/2021

Libraries used:

- re (Use regular expression matching specific data)
- os (From path to read and write files)
- langid (langid.classify() check english data)

1. Introduction

For assignment 1 and task 1, I am provided one package files. I need to understand data's components. Each file has four main parts: uuid, author, published time, text. Read all txt files, gather each txt file together, divide uuid, author, published, and text individually. Gather each part of data into one dictionary. their keys and values need to be matched. Use the langid package to check and delete all data that do not belong to English. Use regular expression to compile special characters and handle Unicode. Write the result to one CSV file and one XML file according to sample format.

Details in coding sections

2.Assignment coding

import Libraries- os,re,langid

- download package: pip install package_name
- import package: import package_name

In [1]:

```
import os
import re
import langid
```

Read files from directory

Use os.listdir(path_name)

endswith() function to limit filename.txt

for loop to read all txt files in directory

file.read() function to store all text into one doc

- Iterator read all txt files in path directory
- Store all content of files in string doc

In [2]:

```
txt_collections = " "
path = './31153291'
for file_each in os.listdir(path):
    if file_each.endswith('txt'):
        file_read = open(path + '/' + file_each, 'r', encoding='utf-8')
        txt_collections += file_read.read()
```

Pre-process raw data

Matching data in files to regular expression

- Compile regular expression: re.compile() function to compile expression
- replace unwanted data: re.sub to replace data

In [3]:

```
unicode_reg = re.compile(r'\\u\S*')
clear_collection = re.sub(unicode_reg, '', txt_collections)
unicode_regx = re.compile(r'\?\?')
clear_collection = re.sub(unicode_regx, '', clear_collection)
```

Process data to identify uuid, author, published time, text

According to analysis data, get each part regular expression format

Use re.findall() function to find all matching data from files' contents

- uuid: 63d2b01f75221bbc4517c594789a4e2ab3001e74
- author: WILL GRAVES, AP Sports Writer
- published: 2021-01-20T10:03:00.000+02:00
- text: A Trump fan's best-case scenario for the Biden era Trump children emotional...

Notice: after find all data. Use len(id_collections) and len(other) to check length is matching each other.

In [4]:

```
id_collections = re.findall('uuid": "(.*?)"', clear_collection)

author_collections = re.findall('author": "(.*?)"', clear_collection)

text_collections = re.findall('text":\s"(.*?)", "', clear_collection)

time_collections = re.findall('published": "(.*?)"', clear_collection)
```

Contents dictionary

Use dicitionary to combine all information, key to value.

- Dictionary: {key:value}
- Combine each personal information as a group.
- Format: uuid: value, author: value, text: value, published: value.
- append() function add dictionary to list store space.

In [5]:

```
info_combine = []
id_check = []
for flag1 in range(len(id_collections)):
    if id_collections[flag1] not in id_check:
        id_check.append(id_collections[flag1])
        info_dic = {'uuid': id_collections[flag1],
                    'author': author_collections[flag1],
                    'text': text_collections[flag1],
                    'published': time_collections[flag1]
                    }
        info_combine.append(info_dic)
```

Check English contents and replace symbol to computer language.

Use langid.classify() function to check English contents and store them.

Notice: langid.classify() will reture a tuple, the first place is language type, so I need to write index 0 to check the language.

- Throw all non-English contents away.
- Read contents from text part to process symbols.
- Replace specific symbols or code to language processing format.

In [ ]:

```
english_check = []
for flag2 in info_combine:
    content_text = flag2['text']
    content_text = re.sub('\\\\n', '\n', content_text)
    content_text = re.sub('&', '&amp;', content_text)
    content_text = re.sub('\\\\"', '&quot;', content_text)
    content_text = re.sub('<', '&lt;', content_text)
    content_text = re.sub('>', '&gt;', content_text)
    flag2['text'] = content_text
    if langid.classify(flag2['text'])[0] == 'en':
        english_check.append(flag2)
```

CSV output processing

file.write() to output file

- Output name should be 31153291.csv, format should be according to sample.csv
- Write excel title first: uuid, author, published, text
- Use ',' to divide each part

I learn how to output csv in tidy way from below website: resource： https://docs.python.org/3/library/csv.html (https://docs.python.org/3/library/csv.html)

In [ ]:

```
csv_output = open('31153291.csv', 'w')
csv_output.write('uuid, author, published, text\n')
for flag3 in english_check:
    part_content = flag3['text']
    part_content = re.sub('"', '""', part_content)
    csv_output.write(flag3['uuid'] + ',"' + flag3['author'] + '",' + flag3['published'] + ',"' + part_content + '"'
+ '\n')
csv_output.close()
```

XML output processing

file.write() to output file

- If author is empty, change author /author to author/
- According to sample.xml

In [ ]:

```
xml_output = open('31153291.xml', 'w', encoding='utf-8')
xml_output.write('<?xml version="1.0" ?>' + '\n')
xml_output.write('<sample>' + '\n')

for flag4 in english_check:
    xml_output.write('    <item>' + '\n')
    xml_output.write('            <uuid>' + flag4['uuid'] + '</uuid>' + '\n')
    if flag4['author'] == '':
        xml_output.write('                <author/>' + '\n')
    else:
        xml_output.write('                <author>' + flag4['author'] + '</author>' + '\n')
    xml_output.write('            <published>' + flag4['published'] + '</published>' + '\n')
    xml_output.write('            <text>' + flag4['text'] + '</text>' + '\n')
    xml_output.write('    </item>' + '\n')
xml_output.write('</sample>' + '\n')
xml_output.close()
```

3. Summary

Task 1 is not a difficult assignment, but it will practice your analysis ability and your capacity to handle data. It is an excellent practice to learn how to observe the data format, organize and classify them through the structure of the data. The efficiency of using python to process data is very convenient.

4. Reference

CSV output: https://docs.python.org/3/library/csv.html (https://docs.python.org/3/library/csv.html)