task1_31282016

September 17, 2020

1 FIT5196 Task 1 in Assessment 1

Student Name: Nabilah Anuwar

Student ID: 31282016 Date: 13 Sept 2020

Environment: Python 3 and Jupyter notebook

Libraries used: please include the main libraries you used in your assignment here, e.g.,: * re (for regular expression) * os (for directories and files) * langid (for language management)

1.1 1. Import Libraries

```
[ ]: import os
     import re
     import langid
```

1.2 2. Converting to XML

1.2.1 Set up variables

Here put the file just above the directory of the tweets files. If there is a need to change the file name then it can be changed in the directory variable below.

```
[ ]: directory = './31282016'

     # set up possible patterns for regex
     pattern1 = "(?:{\"text\":\")(?P<text>.*?)(?:\",)(?:\"created_at\":\")(?P<date>.
      ↪*?)(?:\",)(?:\"id\":\")(?P<id>\d{19})(?:\"})"
     pattern2 = "(?:{\"text\":\")(?P<text>.*?)(?:\",)(?:\"id\":\")(?P<id>\d{19})(?:
      ↪\",)(?:\"created_at\":\")(?P<date>.*?)(?:\"})"
     pattern3 = "(?:{\"id\":\")(?P<id>\d{19})(?:\",)(?:\"text\":\")(?P<text>.*?)(?:
      ↪\",)(?:\"created_at\":\")(?P<date>.*?)(?:\"})"
     pattern4 = "(?:{\"id\":\")(?P<id>\d{19})(?:\",)(?:\"created_at\":\")(?P<date>.*?
      ↪)(?:\",)(?:\"text\":\")(?P<text>.*?)(?:\"})"
     pattern5 = "(?:{\"created_at\":\")(?P<date>.*?)(?:\",)(?:\"id\":\")(?
      ↪P<id>\d{19})(?:\",)(?:\"text\":\")(?P<text>.*?)(?:\"})"
     pattern6 = "(?:{\"created_at\":\")(?P<date>.*?)(?:\",)(?:\"text\":\")(?P<text>.
      ↪*?)(?:\",)(?:\"id\":\")(?P<id>\d{19})(?:\"})"
```

```
# language type
lang = ['en']

# overwrite files if any exist
# create files is none exist
out = open("31282016.xml", 'w')
out.write("<?xml version=\"1.0\" encoding=\"UTF-8\"?>")
out.write('\n')
out.close()

# opened to write
out = open("31282016.xml", "a")
out.write("<data>")
out.write('\n')

# function to set up tweets format
def tweet(idp, t):
    tweet = "<tweet id=\"{i}\">{txt}</tweet>".format(i=idp, txt=t)
    return tweet
```

1.2.2 Convert to XML

```
[ ]: # put here to ensure when starting this would be empty
     # set up list and dictionary to store tweets
     dates = []
     text = {}
     # set up for id duplicate check
     ids = []

     for filename in os.listdir(directory):
         # ensure that file is .txt
         # http://carrefax.com/new-blog/2017/1/16/draft
         if filename.endswith(".txt") :
             f = open(directory + "/" + filename, "r", encoding="UTF-8")
             lines = f.read()
             pattern = (pattern1, pattern2, pattern3, pattern4, pattern5, pattern6)

             # for each file we test pattern
             for p in pattern:
                 m = re.finditer(p, lines)

                 # check all pattern found
                 for n in m:
                     # assuming that all dates in the correct format
                     date = n['date'][:10]
                     t = n['text']
```

```
                idp = n['id']

                # check if id is duplicate
                if len(ids) == 0:
                    pass

                else:
                    for x in ids:
                        if idp == x:
                            id_check = False
                            break
                        else:
                            id_check = True

                    if id_check is True:
                        pass

                    else:
                        continue

                # prevent errors when importing files as multiple backlashes␣
↪may get recorded
                t = t.replace('\\\\', '\\')

                # make characters emojis
                r = re.finditer(r"(?:.?)(\\u\w{4})+", t)
                if r == None:
                    pass
                else:
                    for i in r:
                        u = i.group()
                        u = u.encode().decode("unicode_escape").
↪encode('utf-16', 'surrogatepass').decode("utf-16")
                        t = t.replace(str(i.group()), u)

                # replace command with backlashes accordingly
                t = t.replace('\\n', '\n')
                t = t.replace('\\r', '\r')

                # replace necessary values with xml values
                t = t.replace('&', '&amp;')
                t = t.replace('<', '&lt;')
                t = t.replace('>', '&gt;')
                t = t.replace(r'\"', '&quot;')
                t = t.replace(r"'", "&apos;")

                # check language
                lang_check = False
                for l in lang:
                    text_lang = langid.classify(t)[0]
                    if text_lang == l:
                        lang_check = True
                        break

                    else:
                        lang_check = False

                if lang_check is True:
                    pass

                else:
                    continue

                # append the ids and dates accordingly to the list
                # append id for duplicate id checking
                ids.append(idp)
                # append dates for tweet reference when importing to document␣
 ↪later
                dates.append(date)

                # append tweet to dictionary list
                # try to see if date exist
                if text.get(date, False):
                    text[date].append(tweet(idp, t))

                # if date doesn't exist then make a new one with a list as its␣
 ↪value and date as key
                else:
                    text[date] = []
                    text[date].append(tweet(idp, t))
            continue
        else:
            continue

print("Files are ready to be converted")
```

1.2.3 Append to file

[ ]:
```
# make sure dates value doesnt repeat
dates = set(dates)
# loop through the dates to import from dictionary
for d in dates:
    # insert beginning tweet tag with id
    out.write("<tweets date=\"{dd}\">".format(dd=d))
    # ensure tweets start in a new line like the sample format
    out.write('\n')
    for tt in text[d]:
        # write in tweets from list
        out.write(tt)
        # ensure tweets start in new line like sample
        out.write('\n')
    out.write("</tweets>")
    out.write('\n')
out.write("</data>")
out.close()

print("Files has been converted to xml")
```

1.3 3. Summary

Basically I go through the files lines to search for those with the correct pattern. To make sure I have the correct pattern there are 6 patterns possible and we test each files for it. There are many assumptions for this: * assuming that date format is correct when pattern is found * trusting that the language detected by the package langid is indeed english

Some concerns of mine are: * I have successfully created a loop and encode and decode accordingly yet the emojis don't all get converted. Some of the emoji's issue is that they do not have a separator between them. Yet adding a separator changed other emojis thats already been correctly converted to code which is incorrect