Lab 2: Search Terms

Submitted By: Julian Singkham
Date: 12/17/2020

Abstract

The purpose of this lab is to familize ourselves with data cleaning, search term analytics, and spellchecking.

- The first objective was to derive search terms from a csv files and clean the data.
- The second objective was to create a frequency dictionary of the search terms.
- The final objective was to spellcheck the search terms using spellchecker and create a new spellchecked frequency dictionary

Parameters

In [1]:

```
from spellchecker import SpellChecker
import pattern.en
import csv

csv_freq_dict = {}
csv_freq_dict_spellchecked = []
```

Functions

Imports a CSV file and creates a list of the first item of each row.

Param csv: Name of the CSV file

Return:A list of the first item of each row of the csv

In [2]:

```
def import_csv_list_first_col(csv):
    temp = []
    csv_raw_data = []
    with open(csv) as file:
        for line in file:
            temp.append((line.rstrip('\n').split(',')))
            csv_raw_data = [str(row[0]) for row in temp]
    file.closed
    return csv_raw_data
```

Given a list of strings, create a new list where each string is split by spaces.
EX: "Spicy Bacon" would be ["spicy", "bacon"]

Param original_list: List to split **Return**: A list of single word strings

In [3]:

```python
def split_tokens(original_list):
    new_list = []
    for item in original_list:
        new_list.extend(item.split(' '))
    return new_list
```

Removes web spaces from a string token

Param token: String token

Return: A string without web spaces

In [4]:

```python
def remove_web_spaces(token):
    token.replace("%20", "")
    return token
```

Removes non-alphabet characters from a string token

Param token: String token

Return: A string with only alphabet characters

In [5]:

```python
def remove_non_alphabet(token):
    fixed_token = ""
    for char in token:
        if char.isalpha():
            fixed_token = fixed_token + char
    return fixed_token
```

Creates a frequency dictionary given a string list where the key is a string and the key-value is how many times the string appeared in the list.

Param input_list: String list

Return: A frequency dictionary

In [6]:

```python
def list_to_freq_dict(input_list):
    word_frequency = [input_list.count(i) for i in input_list]
    return dict(list(zip(input_list,word_frequency)))
```

Creates a sorted frequency list given a frequency dictionary

Param freq_dict: Frequnecy dictionary

Return: A 2d list where the first row is frequency and the second row is the string

In [7]:

```python
def sort_freq_dict(freq_dict):
    sorted_list = [(freq_dict[key], key) for key in freq_dict]
    sorted_list.sort()
    sorted_list.reverse()
    return sorted_list
```

Creates a spellchecker dictionary where the key is the misspelled word and the key-value is the most likely corrected word

Param input_list: List of misspelled words

Return: A spellecheck dictionary

In [8]:

```python
def spellcheck_dict_init(input_list):
    spell = SpellChecker(distance=1)
    spellchecked_list = []
    for word in input_list:
        spellchecked_list.append(spell.correction(word))
    return dict(list(zip(input_list,spellchecked_list)))
```

Given a misspelled string token, return the most likely corrected word

Param token: Misspelled token

Return: A correctly spelled word

In [9]:

```python
def spellcheck_token(token):
    fixed_token = csv_spellcheck_dict[token]
    return fixed_token
```

In [10]:

```python
def list_to_csv(input_list):
    fields = ["Frequency", "Word"]
    with open("Frequency of search terms", "w") as file:
        write = csv.writer(file)
        write.writerow(fields)
        write.writerows(input_list)
```

In [*]:

```python
csv_raw = import_csv_list_first_col("searchTerms.csv")
csv_fixed = split_tokens(csv_raw)
csv_spellchecked = []
for i in range(len(csv_fixed)):
    csv_fixed[i] = remove_web_spaces(csv_fixed[i])
    csv_fixed[i] = remove_non_alphabet(csv_fixed[i])
    i += 1
csv_freq_dict = list_to_freq_dict(csv_fixed)
csv_freq_list = sort_freq_dict(csv_freq_dict)
csv_spellcheck_dict = spellcheck_dict_init(csv_fixed)

for word in csv_fixed:
    csv_spellchecked.append(spellcheck_token(word))
csv_spellcheck_dict = list_to_freq_dict(csv_spellchecked)
csv_spellcheck_freq_list = sort_freq_dict(csv_freq_dict)

list_to_csv(csv_spellcheck_freq_list)
```

Conclusion

References

(1) Used the information in this link for removing non alphabet characters. https://stackoverflow.com/questions/43023795/removing-all-numeric-characters-in-a-string-python (https://stackoverflow.com/questions/43023795/removing-all-numeric-characters-in-a-string-python)