DOKUZ EYLÜL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

E-BOOK ANALYSIS AND REPRESENTATION

Assignment Report

by
Yavuz Yılmaz

January 2020
İZMİR

Contents

1 INTRODUCTION

I first searched the libraries and codes required for the project and found the bs4 library and the requests module. After I made the correct version of the book, I converted it to text and printed it into a file, then I prepared a stop word list and took care of the parts such as uppercase and lowercase letters.Finally, I completed the counting parts.

2 METHODOLOGY

I used requests and beautifulsoup to get the book from internet. I encountered some technical problems related to encoding, except that I had trouble removing stop words while writing the code. When counting words, I converted all uppercase letters to lowercase to provide case sensitivity. Then I had an alignment problem while outputting, but I was able to solve it partially. I had a problem in the 2nd book while printing different words, I fixed this problem by editing the variables I used.

2.1 Structure of Your Project

I used beautifulsoup and request to get data from the internet. I prepared the stop words as a list. While I was counting words, I again created a new list and counted words using the for loop. I used the place .replace code for the signs and capital letters that need to be removed. While extracting the stop words, I returned 2 for nested and put 1 condition. I used for again while outputting, and I used the code I found for alignment and adjusted it to look good. The part of collecting the words in 2 books was easy, I did not do different things, just the code I used for alignment caused a problem here. I used 2 forms and 2 lists for different words again, I put 1 conditional in it and pulled different words out of the words.

.

2.2 Encountered Problems and Solutions

I had a lot of trouble with aligning and extracting stop words, and I could not solve some of them, but I managed many successfully

2.3 Improvements

I did not make any additional enhancements or improvements

3 EXPERIMENTATION

I have experimented with using different, asking for more word lists. I tried to approach the outputs given in the example.

4 CONCLUSION

I learned a lot about extracting data from the internet and realized that it is much more convenient and useful than other languages we learned even while doing research on python. I found and implemented codes with very different functions, especially the spacing and .replace commands.

APPENDIX A: CODE

I used code like .remove .replace .pop while creating the project. I also got help from requests and beautifulsoup.

```
import requests

from bs4 import BeautifulSoup


flag = 1

while flag == 1:
```

```
    number_of_books = int(input('How many book do you want to be listed (1 or 2): '))

    if number_of_books == 1:

        BOOK_1 = input('Please Enter The Book Name: ')

        user_request = str(input("Would you like to set a frequency (Y/N): "))

        if user_request == "Y" or user_request == "y":

            number = int(

                input(

                    'How many word frequency would you like to see: '))  # the part where the necessary inputs are taken

        else:

            number = 20

        splitted_book1 = BOOK_1.replace(" ", "_")

        url = 'https://en.wikibooks.org/wiki/' + splitted_book1 + '/Print_version'  # address of the printable version of the book

        r = requests.get(url)

        soup = BeautifulSoup(r.content, "html.parser")

        book1_text_version = soup.get_text()  # the part we convert the book to text

        f = open("book1.txt", "w", encoding='utf-8')

        f.write(book1_text_version)  # the part we printed the book in the text

        f.close()
```

stop_word_list = {"i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours",

"yourself",

"yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its",

"itself",

"they", "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this",

"that",

"these",

"those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had",

"having",

"do", "does",

"did", "doing", "a", "an", "the", "and", "but", "if", "or", "because", "as", "until", "while",

"of",

"at", "by",

"for", "with", "about", "against", "between", "into", "through", "during", "before", "after",

"above",

"below", "to",

"from", "up", "down", "in", "out", "on", "off", "over", "under", "again", "further", "then",

"once",

"here",

"there", "when", "where", "why", "how", "all", "any", "both", "each", "few", "more", "most",

"other",

"some",

"such", "no", "nor", "not", "only", "own", "same", "so", "than", "too", "very", "s", "t",

"can",

"will", "just",

"don", "should", "now"} # stop word list

```
words = book1_text_version.replace("\n", " ")

words = words.replace("{", " ")

words = words.replace("}", " ")

words = words.replace("(", " ")

words = words.replace(")", " ")

words = words.replace("<", " ")

words = words.replace(">", " ")

words = words.replace("#", " ")
```

```
words = words.replace("""", " ")

words = words.replace("", ' ')

words = words.replace("?", " ")

words = words.replace(";", " ")

words = words.replace("-", " ")

words = words.replace("\\", " ")

words = words.replace("[", " ")

words = words.replace("]", " ")

words = words.replace("←", " ")

words = words.replace("*", " ")

words = words.replace("=", " ")

words = words.replace("==", " ")

words = words.replace("_", " ")

words = words.replace("!", " ")

words = words.replace(".", " ")

words = words.replace(",", " ")

words = words.replace("@", " ")

words = words.replace("`", " ")

words = words.replace("/", " ")

words = words.replace(":", " ")

words = words.replace("0", " ")
```

```
words = words.replace("1", " ")

words = words.replace("2", " ")

words = words.replace("3", " ")

words = words.replace("4", " ")

words = words.replace("5", " ")

words = words.replace("6", " ")

words = words.replace("7", " ")

words = words.replace("8", " ")

words = words.replace("9", " ")

words = words.replace("A", "a")

words = words.replace("B", "b")

words = words.replace("C", "c")

words = words.replace("D", "d")

words = words.replace("E", "e")

words = words.replace("F", "f")

words = words.replace("G", "g")

words = words.replace("H", "h")

words = words.replace("I", "i")

words = words.replace("J", "j")

words = words.replace("K", "k")

words = words.replace("L", "l")
```

```
words = words.replace("M", "m")

words = words.replace("N", "n")

words = words.replace("O", "o")

words = words.replace("P", "p")

words = words.replace("R", "r")

words = words.replace("S", "s")

words = words.replace("T", "t")

words = words.replace("U", "u")

words = words.replace("V", "v")

words = words.replace("Y", "y")

words = words.replace("Z", "z")

words = words.replace("W", "w")

words = words.replace("X", "x")

words = words.replace("Q", "q")  # punctuation and alphabet regulations


words = words.split()


for word in words:

    for stop_word in stop_word_list:

        if word == stop_word:

            words.remove(word)  # part where stop words are removed
```

```
        # This part extracts stop words, but it cannot extract all of it. I could not solve this problem.
    allWords1 = {}
    tempword_count = 0
    temp_word = " "
    listed_words = {}

    for word in words:
       if word not in allWords1:
          allWords1[word] = 1
       else:
          allWords1[word] += 1  # count of words

    for a in range(len(allWords1)):
       for key1 in allWords1.keys():
          if allWords1[key1] > tempword_count and key1 not in listed_words:
             tempword_count = allWords1[key1]
             temp_word = key1
       listed_words[temp_word] = tempword_count
       tempword_count = 0
       temp_word = " "  # sorting part from higher to lower
```

```
    print("BOOK 1: ", BOOK_1)
    print("NO WORD          FREQ_1")
    NO = 1

    for key1 in listed_words.keys():
        if NO < 10 and NO <= number:
            print(NO, "{0:12} {1:7}".format(key1, allWords1[key1]))
            NO += 1
        elif NO <= number:
            print(NO, "{0:11} {1:7}".format(key1, allWords1[key1]))
            NO += 1  # output part
    decision = str(input("Do you want to run program again(Y/N): "))
    if decision == "Y" or decision == "y":
        flag = 1
    else:
        print("Good bye!!")
        break
elif number_of_books == 2:  # Skip when 2 books are entered
    BOOK_1 = input('Please Enter The First Book Name: ')
    BOOK_2 = input('Please Enter The Second Book Name: ')
```

```
user_request = str(input("Would you like to set a frequency (Y/N): "))
if user_request == "Y" or user_request == "y":
    number = int(
        input(
            'How many word frequency would you like to see: '))  # the part where the necessary inputs are taken
else:
    number = 20
splitted_book1 = BOOK_1.replace(" ", "_")
splitted_book2 = BOOK_2.replace(" ", "_")
url = 'https://en.wikibooks.org/wiki/' + splitted_book1 + '/Print_version'  # address of the printable version of the first book
url2 = 'https://en.wikibooks.org/wiki/' + splitted_book2 + '/Print_version'  # address of the printable version of the second book
r = requests.get(url)
r2 = requests.get(url2)
soup = BeautifulSoup(r.content, "html.parser")
soup2 = BeautifulSoup(r2.content, "html.parser")
book1_text_version = soup.get_text()  # the part we convert the first book to text
book2_text_version = soup2.get_text()  # the part we convert the second book to text
```

```
f = open("book1.txt", "w", encoding='utf-8')

f.write(book1_text_version)  # the part we printed the first book in the text

f.close()

f1 = open("book2.txt", "w", encoding='utf-8')

f1.write(book2_text_version)  # the part we printed the second book in the text

f1.close()

stop_word_list = {"i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours",

                  "yourself",

                  "yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its",

                  "itself",

                  "they",

                  "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this", "that",

                  "these",

                  "those",

                  "am",

                  "is", "are", "was", "were", "be", "been", "being", "have", "has", "had", "having", "do",

                  "does",
```

"did",

"doing",

"a", "an", "the", "and", "but", "if", "or", "because", "as", "until", "while", "of", "at",

"by",

"for",

"with",

"about", "against", "between", "into", "through", "during", "before", "after", "above",

"below",

"to",

"from",

"up", "down", "in", "out", "on", "off", "over", "under", "again", "further", "then", "once",

"here",

"there",

"when", "where", "why", "how", "all", "any", "both", "each", "few", "more", "most", "other",

"some",

"such", "no",

"nor", "not", "only", "own", "same", "so", "than", "too", "very", "s", "t", "can", "will",

```
                "just",
                "don",
                "should",
                "now"}  # stop word list

words = book1_text_version.replace("\n", " ")
words = words.replace("{", " ")
words = words.replace("}", " ")
words = words.replace("(", " ")
words = words.replace(")", " ")
words = words.replace("<", " ")
words = words.replace(">", " ")
words = words.replace("?", " ")
words = words.replace(";", " ")
words = words.replace("-", " ")
words = words.replace("\\", " ")
words = words.replace("[", " ")
words = words.replace("]", " ")
words = words.replace("*", " ")
words = words.replace("=", " ")
words = words.replace("==", " ")
```

```
words = words.replace("#", " ")

words = words.replace("""", " ")

words = words.replace("", ' ')

words = words.replace("_", " ")

words = words.replace("!", " ")

words = words.replace(".", " ")

words = words.replace(",", " ")

words = words.replace("@", " ")

words = words.replace("←", " ")

words = words.replace("`", " ")

words = words.replace("/", " ")

words = words.replace(":", " ")

words = words.replace("0", " ")

words = words.replace("1", " ")

words = words.replace("2", " ")

words = words.replace("3", " ")

words = words.replace("4", " ")

words = words.replace("5", " ")

words = words.replace("6", " ")

words = words.replace("7", " ")

words = words.replace("8", " ")
```

```
words = words.replace("9", " ")

words = words.replace("A", "a")

words = words.replace("B", "b")

words = words.replace("C", "c")

words = words.replace("D", "d")

words = words.replace("E", "e")

words = words.replace("F", "f")

words = words.replace("G", "g")

words = words.replace("H", "h")

words = words.replace("I", "i")

words = words.replace("J", "j")

words = words.replace("K", "k")

words = words.replace("L", "l")

words = words.replace("M", "m")

words = words.replace("N", "n")

words = words.replace("O", "o")

words = words.replace("P", "p")

words = words.replace("R", "r")

words = words.replace("S", "s")

words = words.replace("T", "t")

words = words.replace("U", "u")
```

```
        words = words.replace("V", "v")

        words = words.replace("Y", "y")

        words = words.replace("Z", "z")

        words = words.replace("W", "w")

        words = words.replace("X", "x")

        words = words.replace("Q", "q")  # punctuation and alphabet regulations for first book


        words = words.split()


        words2 = book2_text_version.replace("\n", " ")

        words2 = words2.replace("{", " ")

        words2 = words2.replace("}", " ")

        words2 = words2.replace("(", " ")

        words2 = words2.replace(")", " ")

        words2 = words2.replace("<", " ")

        words2 = words2.replace(">", " ")

        words2 = words2.replace("?", " ")

        words2 = words2.replace(";", " ")

        words2 = words2.replace("#", " ")

        words2 = words2.replace("'", " ")
```

```
words2 = words2.replace("", ' ')

words2 = words2.replace("-", " ")

words2 = words2.replace("\\", " ")

words2 = words2.replace("[", " ")

words2 = words2.replace("]", " ")

words2 = words2.replace("*", " ")

words2 = words2.replace("=", " ")

words2 = words2.replace("==", " ")

words2 = words2.replace("_", " ")

words2 = words2.replace("←", " ")

words2 = words2.replace("!", " ")

words2 = words2.replace(".", " ")

words2 = words2.replace(",", " ")

words2 = words2.replace("@", " ")

words2 = words2.replace("`", " ")

words2 = words2.replace("/", " ")

words2 = words2.replace(":", " ")

words2 = words2.replace("0", " ")

words2 = words2.replace("1", " ")

words2 = words2.replace("2", " ")

words2 = words2.replace("3", " ")
```

```
words2 = words2.replace("4", " ")

words2 = words2.replace("5", " ")

words2 = words2.replace("6", " ")

words2 = words2.replace("7", " ")

words2 = words2.replace("8", " ")

words2 = words2.replace("9", " ")

words2 = words2.replace("A", "a")

words2 = words2.replace("B", "b")

words2 = words2.replace("C", "c")

words2 = words2.replace("D", "d")

words2 = words2.replace("E", "e")

words2 = words2.replace("F", "f")

words2 = words2.replace("G", "g")

words2 = words2.replace("H", "h")

words2 = words2.replace("I", "i")

words2 = words2.replace("J", "j")

words2 = words2.replace("K", "k")

words2 = words2.replace("L", "l")

words2 = words2.replace("M", "m")

words2 = words2.replace("N", "n")

words2 = words2.replace("O", "o")
```

```
words2 = words2.replace("P", "p")

words2 = words2.replace("R", "r")

words2 = words2.replace("S", "s")

words2 = words2.replace("T", "t")

words2 = words2.replace("U", "u")

words2 = words2.replace("V", "v")

words2 = words2.replace("Y", "y")

words2 = words2.replace("Z", "z")

words2 = words2.replace("W", "w")

words2 = words2.replace("X", "x")

words2 = words2.replace("Q", "q")  # punctuation and alphabet regulations for second book


words2 = words2.split()

for word in words:

    for stop_word in stop_word_list:

        if word == stop_word:

            words.remove(word)  # part where stop words are removed

            # This part extracts stop words, but it cannot extract all of it. I could not solve this problem.

for word2 in words2:
```

```
    for stop_word2 in stop_word_list:
        if word2 == stop_word2:
            words2.remove(word2)  # part where stop words are removed
            # This part extracts stop words, but it cannot extract all of it. I could not solve this problem.
allWords1 = {}
allWords2 = {}
tempword_count = 0
tempword_count2 = 0
temp_word = " "
temp_word2 = " "
listed_words = {}
listed_words2 = {}

for word in words:
    if word not in allWords1:
        allWords1[word] = 1
    else:
        allWords1[word] += 1  # count of words
for word2 in words2:
    if word2 not in allWords2:
```

```
            allWords2[word2] = 1
        else:
            allWords2[word2] += 1  # count of words
    for a in range(len(allWords1)):
        for key1 in allWords1.keys():
            if allWords1[key1] > tempword_count and key1 not in listed_words:
                tempword_count = allWords1[key1]
                temp_word = key1
        listed_words[temp_word] = tempword_count
        tempword_count = 0
        temp_word = " "  # sorting part from higher to lower
    for b in range(len(allWords2)):
        for key1 in allWords2.keys():
            if allWords2[key1] > tempword_count2 and key1 not in listed_words2:
                tempword_count2 = allWords2[key1]
                temp_word2 = key1
        listed_words2[temp_word2] = tempword_count2
        tempword_count2 = 0
        temp_word2 = " "  # sorting part from higher to lower
    print("BOOK 1: ", BOOK_1)
    print("BOOK 2: ", BOOK_2)
```

```
        print("NO WORD          FREQ_1              FREQ_2          FREQ_SUM")
        NO = 1
        for key1 in listed_words.keys():
            if NO < 10 and NO <= number:
                print(NO, '{0:12} {1:7}                    '.format(key1, allWords1[key1]),
allWords2[key1],"\t\t",
                    allWords2[key1] + allWords1[key1])
                NO += 1  # output giving part
            elif NO <= number:
                print(NO, '{0:11} {1:7}                    '.format(key1, allWords1[key1]),
allWords2[key1],"\t\t",
                    allWords2[key1] + allWords1[key1])
                NO += 1  # output giving part
        new_listed_words = listed_words
        new_listed_words2 = listed_words2
        print()
        print("BOOK 1: ", BOOK_1)
        print("DISTINCT WORDS")
        print("NO WORD          FREQ_1")
        for key1 in list(listed_words):
            for key2 in list(listed_words2):
```

```
            if key1 == key2:
                new_listed_words.pop(key1)  # the part where different words are found and removed
        NO = 1
        for key1 in new_listed_words.keys():
            if NO < 10 and NO <= number:
                print(NO, "{0:12} {1:7}".format(key1, allWords1[key1]))
                NO += 1  # output giving part
            elif NO <= number:
                print(NO, "{0:11} {1:7}".format(key1, allWords1[key1]))
                NO += 1  # output giving part

        for a in range(len(allWords1)):
            for key1 in allWords1.keys():
                if allWords1[key1] > tempword_count and key1 not in listed_words:
                    tempword_count = allWords1[key1]
                    temp_word = key1
            listed_words[temp_word] = tempword_count
            tempword_count = 0
            temp_word = " "
        for b in range(len(allWords2)):
```

```
        for key1 in allWords2.keys():
            if allWords2[key1] > tempword_count2 and key1 not in listed_words2:
                tempword_count2 = allWords2[key1]
                temp_word2 = key1
        listed_words2[temp_word2] = tempword_count2
        tempword_count2 = 0
        temp_word2 = " "  # the part where spoiled lists are prepared from the beginning
    new_listed_words = listed_words
    new_listed_words2 = listed_words2
    print()
    print("BOOK 2: ", BOOK_2)
    print("DISTINCT WORDS")
    print("NO WORD          FREQ_1")
    for key2 in list(listed_words2):
        for key1 in list(listed_words):
            if key2 == key1:
                new_listed_words2.pop(key2)  # the part where different words are found and removed
    NO = 1
    for key2 in new_listed_words2.keys():
```

```
            if NO < 10 and NO <= number:
                print(NO, "{0:12} {1:7}".format(key2, allWords2[key2]))
                NO += 1  # output giving part
            elif NO <= number:
                print(NO, "{0:11} {1:7}".format(key2, allWords2[key2]))
                NO += 1  # output giving part
        decision = str(input("Do you want to run program again(Y/N): "))
        if decision == "Y" or decision == "y":
            flag = 1
        else:
            print("Good bye!!")
            break
    else:
        print('You entered wrong input')
        decision = str(input("Do you want to run program again(Y/N): "))
        if decision == "Y" or decision == "y":
            flag = 1
        else:
            print("Good bye!!")
            break
```

APPENDIX B: SCREENSHOTS OF YOUR USE CASES

```
for word in words:
    if word not in allWords1:
        allWords1[word] = 1
    else:
        allWords1[word] += 1#count of words
```

```
for word in words:
    for stop_word in stop_word_list:
        if word == stop_word:
            words.remove(word)#part where stop words are removed
```

```
splitted_book1 = BOOK_1.replace(" ", "_")
url = 'https://en.wikibooks.org/wiki/' + splitted_book1 + '/Print_version'#address of the printable version of the book
r = requests.get(url)
soup = BeautifulSoup(r.content, "html.parser")
book1_text_version = soup.get_text() #the part we convert the book to text
f = open("book1.txt", "w", encoding='utf-8')
f.write(book1_text_version)#the part we printed the book in the text
f.close()
```

REFERENCES

1. https://www.programiz.com/python-programming/methods/list/copy
2. https://stackoverflow.com/questions/11941817/how-to-avoidruntimeerror-dictionary-changed-size-during-iteration-error
3. https://realpython.com/python-keyerror/
4. https://python-istihza.yazbel.com/print.html
5. http://www.veridefteri.com/2018/02/23/python-programlamaya-giris-16dize-bicimlendirme/
6. https://www.w3schools.com/python/python_dictionaries.asp
7. https://www.youtube.com/watch?v=WthFJcmGLhY