**DOKUZ EYLÜL UNIVERSITY**

**DEPARTMENT OF COMPUTER ENGINEERING**

# E-BOOK ANALYSIS AND REPRESENTATION

## Assignment Report

**by**

**Eren Çağlar Erdoğan**

**January 2020**

**İZMİR**

# Contents

# 1 INTRODUCTION

Our main goal in this project was to download books from wikibooks with phyton and count the frequency of words. When we did the project, we had to use webscrapping methods and string operations

In the project, we have to download a book on the website with webscrapinng, count and sort the words using the dictionary structure, and then print these words on the screen.

# 2 METHODOLOGY

The first problem I encountered is I didn't know how to pull information from a website, I did research on the internet to solve this problem, and I found the requests and beautifulsoup methods, and I tried to solve the problem as much as I could. after I pulled the data through the website, all I had to do was calculate the word frequency. I extracted punctuation marks and stopwords from the text I pulled from the website to accurately calculate the number of words. Then I calculated the frequencies of the words in the text with the list and dictionary structures in python and printed them on the screen. If the user wanted to compare the two books, I repeated the same process for the second book, detecting the words of the two books that are the same or discrete, and printing their frequencies on the screen.

## 2.1 Structure of Your Project

I used the requests module to retrieve data from the website. This module downloads the codes of the website whose URL is entered. With the beautifulsoup method, we can find and use a specific tag or class in the page source we get from the website. In addition, I used Python's list and dictionary structures and loops and nested loops when calculating word frequency

## 2.2 Encountered Problems and Solutions

The main problem in the project was that we didn't know some of the structures we had to use. I've tried to solve this problem with Internet Research, but there are still some problems. after downloading the source on the website with the requests modul, I couldn't figure out which tags or classes I should find with beautifulsoup, so there may be some extra words and sentences in the text I pulled out of the site and processed. Because of these extra words and sentences, my word frequencies may be a little high. Lastly, sometimes the process of processing can take too long, I think the reason for this is the images on the website.

## 3 CONCLUSION

As a result, thanks to webscrapping methods in this project, we have learned that we are not only dependent on our own computer, we can pull resources through any website thanks to an internet connection and use these resources according to our purpose. In addition, thanks to the project, we had the opportunity to practice the python language

## APPENDIX A: CODE

Code samples from my project

```
#counting word frequencies for Book 1
    for i in word1_list:
        if i in word1_count:
            word1_count[i] = word1_count[i] + 1
        elif i not in word1_count and i not in stopwords:
            word1_count[i] = 1
```

Instead of using the sort method, I used a nested loop

```
#In this nested loop, I sort the words in the word_rank list
according to their values in the dictionary
    for i in range(0, len(word_rank)):
        for j in range(i + 1, len(word_rank)):
            if word_count[word_rank[i]] < word_count[word_rank[j]]:
                holder = ""
                holder = word_rank[i]
                word_rank[i] = word_rank[j]
                word_rank[j] = holder
```

```
#I take the site resource with the requests module and process it with the beautifulsoup module
#There are 3 different print version url types so I compare the length of the source of all three links and take the longest one
a1 = requests.get("https://en.wikibooks.org/wiki/" + book1_url + "/Print_version")
a2 = requests.get("https://en.wikibooks.org/wiki/" + book1_url + "/print_version")
a3 = requests.get("https://en.wikibooks.org/wiki/" + book1_url + "/Print_Version")

if len(a1.content) > len(a2.content) and len(a1.content) > len(a3.content):
    r = a1
elif len(a2.content) > len(a1.content) and len(a2.content) > len(a3.content):
    r = a2
elif len(a3.content) > len(a1.content) and len(a3.content) > len(a2.content):
    r = a3
soup = BeautifulSoup(r.content, "html.parser")
```

## REFERENCES

https://www.ranks.nl/stopwords

https://www.geeksforgeeks.org/python-remove-punctuation-from-string/

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://www.w3schools.com/python/python_ref_string.asp