CSM 6404 Data Mining

A Report on

Word Count From a Text Without Sorting and Using Sorting (Descending and Ascending Order)

SUBMITTED TO:	**SUBMITTED BY:**
Dr. Md. Rakib Hasan Professor Department of Computer Science and Mathematics Faculty of Agricultural Eng. And Technology Bangladesh Agricultural University	**Md. Shafiur Rahman Khan** Ms. in Computer Science Roll no.-19CSJJ1M Reg. no.-50527 Session:2019-20

BANGLADESH AGRICULTURAL UNIVERSITY

MYMENSINGH- 2202

CONTENTS

1. Introduction

Data mining is the process of discovering pattern in large data sets involving methods at the intersection of machine learning, statistics and database management systems. It is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information.

Data mining involves six common classes of tasks. They are given below-

1. Anomaly Detection: The identification of unusual data records, that might be interesting or data errors that require further investigation.

2. Association Rule Learning: Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

3. Clustering: is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

4. Classification: is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".

5. Regression: is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".

6. Summarization: providing a more compact representation of the data set, including visualization and report generation.

   However there are also some open source software for mining a data but we use here google colab/ jupyter notebook(anaconda) to complete the tasks. Both of this software have some features for data mining.

   We generally use def word_count(str) to count the occurrences of word in a sentence from a text with out using sorting and we use word_list and counter to count occurrences of word from a text.(Descending and Ascending Order)

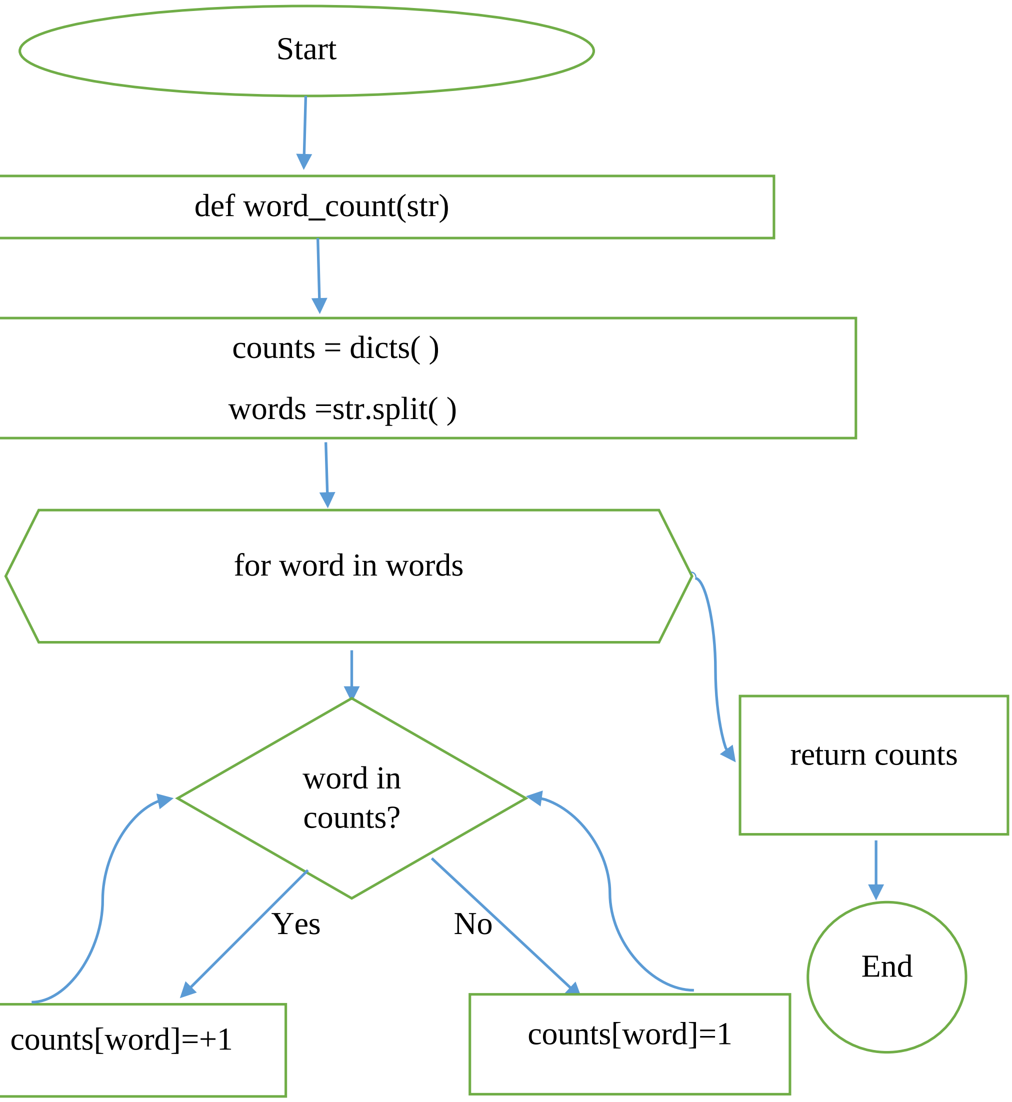
   In both case we have to count the occurrences of word from a text without sort and with sort(Descending and Ascending Order) respectively.

2. Methodology

There are two methods which are using for without sorting and sorting method to count occurrences of word from a given text. At first we describe the without sorting method to count occurrences of word from a text.

2.1Without Sorting method to count occurrences of word from a text:

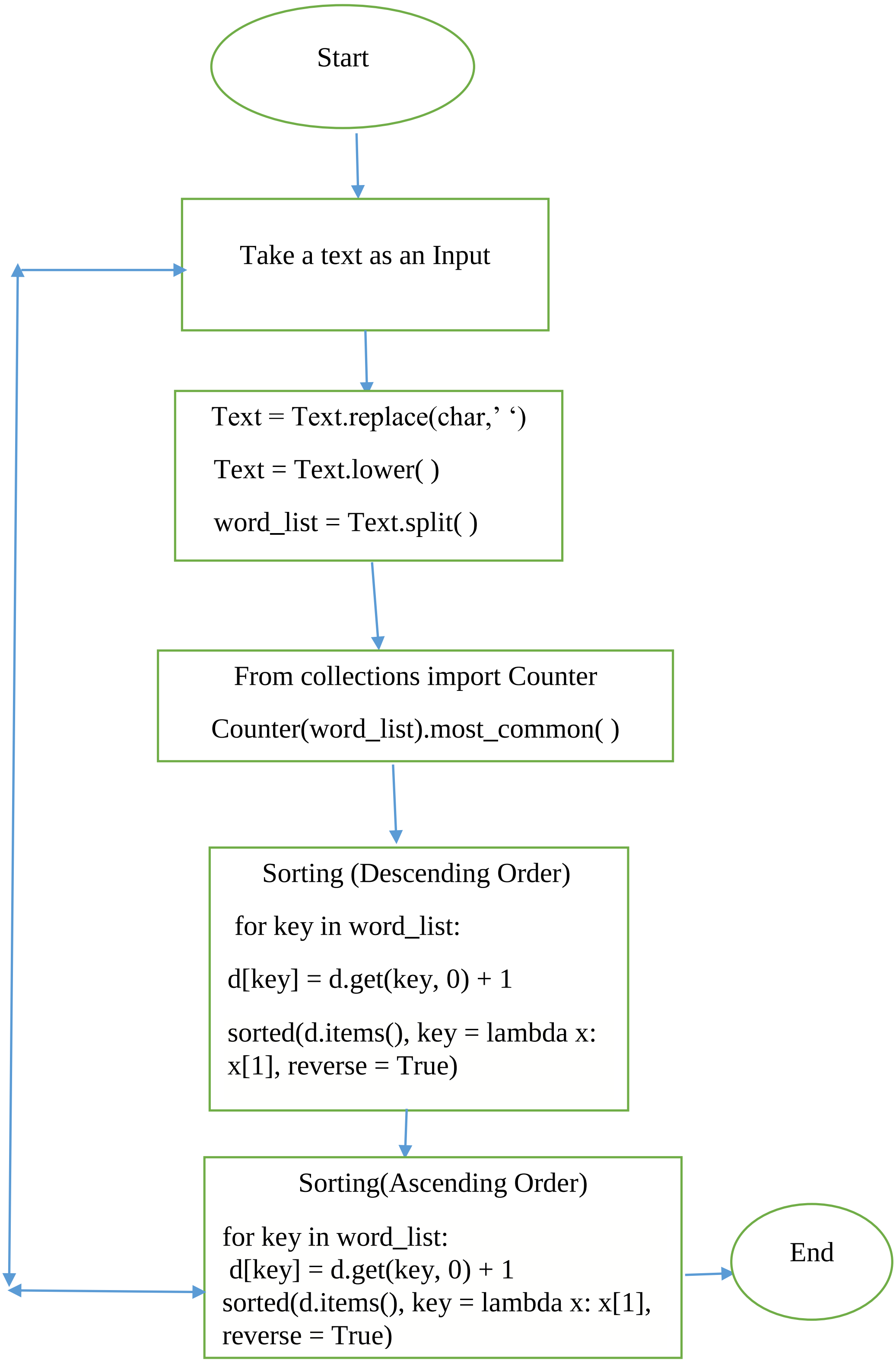
A flow chart is given below for it.



Flow chart 1: Without Sorting method to count occurrences of word from a text

2.2 With Sorting method to count occurrences of word from a text(Descending and Ascending Order):

A flow chart is given below for it.


Start
Take a text as an Input
Text = Text.replace(char,’ ‘)
Text = Text.lower( )
word_list = Text.split( )
From collections import Counter
Counter(word_list).most_common( )
Sorting (Descending Order)
for key in word_list:
d[key] = d.get(key, 0) + 1
sorted(d.items(), key = lambda x: x[1], reverse = True)
Sorting(Ascending Order)
for key in word_list:
d[key] = d.get(key, 0) + 1
sorted(d.items(), key = lambda x: x[1], reverse = True)
End

Flow chart 2: Sorting method to count occurrences of word from a text(Descending and Ascending Order)

According to flow chart 1 we take a text under word count( ). It uses then str.split( ) and word in counts. Next it uses counts[word]=+1 and counts[word]=1 respectively. Finally from word in words and through return counts it complete the whole tasks without sorting.

According to flow chart 2 it takes a text as an input. Then it uses Text.lower( ), Text.split( ) and word_list. Next it uses Counter(word_list).most_common( ), Sorting (Descending Order) and Sorting(Ascending Order) according to flow chart 2.

The main difference between this two flow chart is first one can count punctuation but second one can not.

3. Result

As far as we concern through this overall method we can count any word occurrences from any pdf or text document. But in order to get appropriate result please use use the following documentation which is attached with Appendix I.

4. Conclusion

This whole tasks is done for making and comfortable easier for big data and data mining scientist and students to increase their research knowledge. Although it is an efficient model for the overall process but further study should be necessary to update the whole procedure.

5. Appendix I

5.1Documentation for without sorting:

Instruction of using Word Count from a text file.ipynb-

1. Open it on google co-lab / jupyter notebook in anaconda.

2. Write any sample text in print(word_count('')) instead of using any text file.This will count the occurences of each word in a text.Some sample texts are given below-

2.1 My name is Rifat . My family lives in Mymensingh . Mymensingh is 120 Km from Dhaka . Dhaka is the capital of Bangladesh . We love Bangladesh .

2.2 My house is beside by a school . Every morning childreen go to to the school . Every day they sing the national anthem . Every school does the same thing . Childreen are gifts from AllAH .

N.B. : If anyone use punctuation(exapmle . , / ;) then it will also count it but please separate them from each word for count. Otherwise it can not count it properly.

Thank you for hearing.

5.2 Documentation for using sorting:

1. Open Word Count From a text Using Sorting (ascending and descending order) . ipynb file in google co-lab(online) or jupyter notebook in anconda(off-line).

2. write any sentence Text=""" """ which is the first cell of code in this ipynb file.

3. run last two cells for geting descending and ascending order of words.

Thank You For Listening.

6. Appendix II

Code of word count from a text without sorting:

```
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

              return counts


print( word_count('The name of the subject is data mining . This is an interesting subject .
People say that the name is interesting . My supervisor sir teaches me the subject .'))
```

7. Appendix III

Code of word count from a text using sorting(descending and ascending order):

```
Text="""
 The name of my country is Bangladesh. The name is beautiful. """

# Cleaning text and lower casing all words
for char in '-.,\n':
    Text=Text.replace(char,' ')
Text = Text.lower()
# split returns a list of words delimited by sequences of whitespace (including tabs, newlines,
etc, like re's \s)
word_list = Text.split()

from collections import Counter
Counter(word_list).most_common()

# initializing a dictionary (descending order)
d = {};

# counting number of times each word comes up in list of words
for key in word_list:
    d[key] = d.get(key, 0) + 1

sorted(d.items(), key = lambda x: x[1], reverse = True)

# initializing a dictionary (ascending order)
d = {};

# counting number of times each word comes up in list of words
for key in word_list:
    d[key] = d.get(key, 0) + 1

sorted(d.items(), key = lambda x: x[1], reverse = False)
```