Hamed Marvi - 610396143 - HW1

the data we are using has 90981words and vocabulary size of 15530.

most used words are:

```
most commone words:
the : 5495
and : 2425
a : 2257
to : 2005
of : 1757
you : 1278
in : 1180
his : 1149
i : 982
he : 883
```

We read the text file word by word and lowercase each word.

After reading each word there are two possibilities:

1- it is not a new words. we store all the unique words (vocabulary) in an array called "vocab" and the repetition of that word in another array called "vocabcount". after reding an already seen word, we will increment its number in vocabcount.

```
for word in line.split():
    n=n+1
    if word in vocab:         #if its not  a new word
        for m in range(len(vocab)):     #find the word and add to its frequency
            if vocab[m]==word:
                vocabcount[m]=vocabcount[m]+1
                m = len(vocab)
```

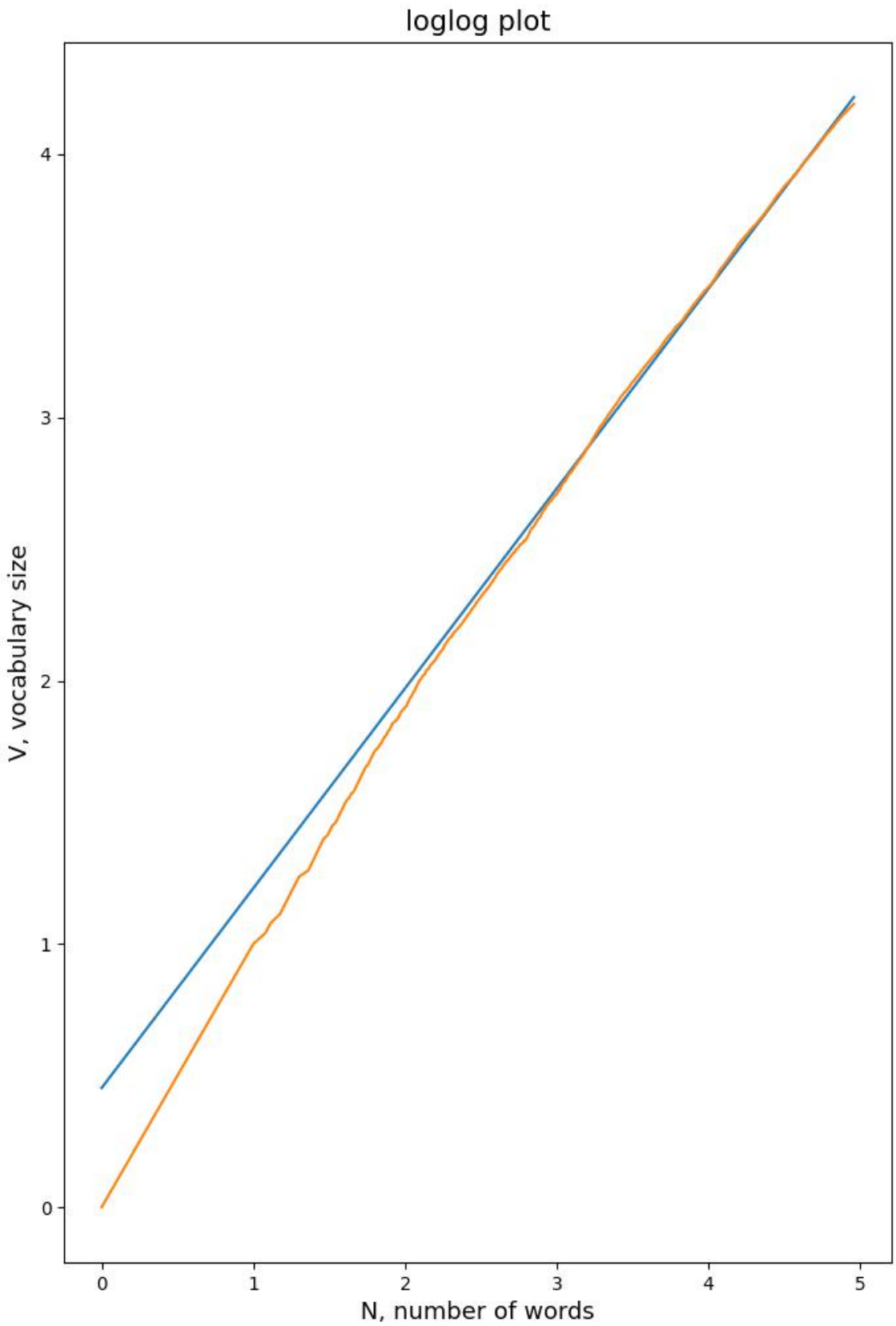
2- if it is a new word (not in vocab), then we will add it to vocab and its repetition so far in vocabcount would be 1.

"v" is the number of unique words seen and now that we have seen a new word, we increment it. "n" is the number of all words seen so far. we will add n to "N" and add v to "V". we will use arrays N and V for heaps plot.

```
elif word not in vocab:  # if it is a new word add it to the vocabulary
    vocab.append(word)
    vocabcount.append(1)
    v=v+1
    N.append(n)    #when a new word is found, put the number of words at that point in N
    V.append(v)    # and put the number of unique points at that point in V
words.append(word)
```

then we will show the heaps plot using log(V) and log(N).

using "numpy.polyfit" we can estimate the fit line for heaps plot.


loglog plot
V, vocabulary size
N, number of words
0
1
2
3
4
5

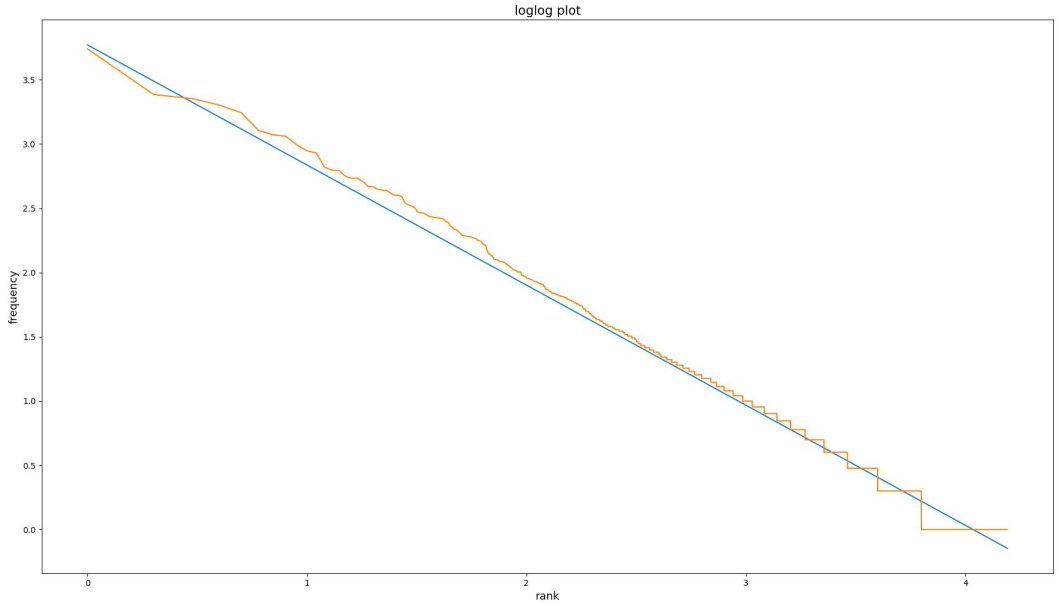
```
fig, (ax2) = plt.subplots(1, 1, figsize=[7, 11])
Nlog , Vlog = np.log10(N) , np.log10(V)
m, b = np.polyfit(Nlog, Vlog, 1)
#plt.plot(Nlog, Vlog, 'o')

plt.plot(Nlog, m*Nlog + b)
ax2.plot(Nlog , Vlog)
#ax2.loglog(N,V)
ax2.set_title('loglog plot', fontsize=15)
ax2.set_xlabel('N, number of words', fontsize=13)
ax2.set_ylabel('V, vocabulary size', fontsize=13)
plt.tight_layout()
plt.show()
print("heaps fit line is: ",m,"*x + ",b)
```

heaps fit line is:

0.7589989701625419 *x + 0.4529860653705679

for zipfs we have to have words in order with their repitition so we will sort vocabcount and vocab together (and then reverse it cause we want it to be in increasing order) . now vocab[i] would be the word with i'th repitition order and voucabcount[i] would be the number of its repitition.

we will show the zipfs plot and with "numpy.polyfit" we will fit the line. the code is same as for heaps.



zipfs fit line is:

-0.9344010963534362 *x + 3.7706637409819908