CMPE 300 - Analysis of Algorithms
Fall 2022
MPI Programming Project

Due Date: 20.12.2022 Tuesday 23:59

ÜMMÜ SENA ÖZPINAR – 2019400279
ELİF KIZILKAYA – 2018400108

TABLE OF CONTENTS

INTRODUCTION 3

PROGRAM INTERFACE 3

PROGRAM EXECUTION 4

INPUT AND OUTPUT 4

PROGRAM STRUCTURE 5

EXAMPLES 7

IMPROVEMENTS AND EXTENSIONS 11

DIFFUCULTIES ENCOUNTERED 11

CONCLUSIONS 12

INTRODUCTION

In this project, our goal was to calculate the probability data for an n-gram, specifically a bigram, language model. We would be presented with two separate documents: an input file consisting of separated sentences and a test file consisting of bigrams (two words separated by a single space). We were asked to calculate the conditional probabilities of the bigrams given in the test file by finding their number of occurrences in the input file. In order to do this, we were to make use of the MPI framework which is a master-slave/worker process-based system. Our problem consisted of such steps:

1) **Reading the input file and distributing the data:** Each sentence in the input file were to be separated and distributed between the worker processes as evenly as possible to provide balance. This task would be handled by the master process.
2) **Calculating the frequencies:** After the data is sent by the master and received by the worker process, each worker should count the bigrams and unigrams for the sentences it has received. This calculation should be done concurrently.
3) **Merging:** After the counting of the bigrams and the unigrams by each worker process, these results will be merged by summing. There will be two different ways of merging. In the first flow, after the worker processes have finished their calculation, they will send their result to the master process to be merged. In the second flow, merging should be done sequentially. Each worker process should receive the previous worker process's data, sum it with its own and send it to the next worker process. The result will be handed off to the master process. The flow picked, will depend on the argument provided to the program.
4) **Calculating the probabilities:** Master process should calculate the conditional probabilities with the data that is passed to it.

To solve the problem defined above, we used Python language. Our solution consists of if-else statements for the master process and worker process. In each part, the necessary requirement of the project is satisfied which will be explained in detail at the following parts.

PROGRAM INTERFACE

In this project, we used Python as our programming language. To run our code, the command given in the project description file may be used by changing the directories of the input and test files accordingly. Merge method must be specified. The workflow of the program (whether the merging will be done sequentially by the worker processes or at once by the master process) will depend on this. Here is an example:

```
mpiexec -n 5 python3 main.py --input_file data/sample_text.txt --
merge_method MASTER --test_file data/test.txt
```

If user receives an error about oversubscribe, it can add "-oversubscribe" option to the command.

```
mpiexec -n 5 -oversubscribe python3 main.py --input_file
data/sample_text.txt --merge_method MASTER --test_file data/test.txt
```

It is necessary to write input file path right after “—input_file” statement is indicated. This is also necessary for the “--merge_method” and “—test_file”. The order of the parameters can change.

PROGRAM EXECUTION

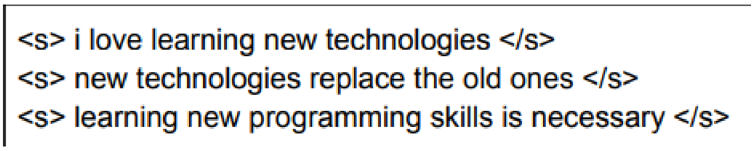
For this program, user will supply two text files. First one is a preprocessed text file of sentences, one for each line, consisting of only words. These sentences are modified to start and end with special tokens, “<s>” and “</s>” respectively. Although this version of the program requires these modifications, the program could be easily adapted to working for non-processed versions of text documents. The second document is another text document consisting of bigrams that the user wants searched in the first document.

The main purpose of our program is to find the conditional probabilities of the given bigrams. This is done by finding the frequency of their occurrence in the first document and calculating their conditional probability. This information is then printed onto the screen as output and the user can see each requested bigram along with its probability of occurrence in the given text. The program provides some further information to the user. Each worker process’s rank as well as the number of sentences it has received for inspection, is also printed on the screen.

Other than the files, the user also has to specify the merge method of the algorithm in the running command either as MASTER or WORKERS.

INPUT AND OUTPUT

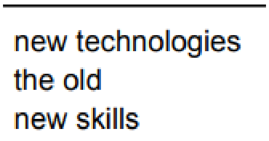
The program takes two “txt” files, both made up of string characters. Here is an example snippet from the first input file:



```
<s> i love learning new technologies </s>
<s> new technologies replace the old ones </s>
<s> learning new programming skills is necessary </s>
```

As can be seen from the picture, each line consists of a sentence, however long or its words separated by as many space characters as there may be, each beginning and ending with special tokens “<s>” and “</s>” to symbolize sentence beginnings and ends as well as for ease of calculation.

The second input file is test file containing the list of bigrams (sequence of 2 words), each bigram in a separate line. Apart from the condition that a bigram consists of two words, there are no other specifications for the format of the bigrams. Here is a snippet of the file content, showing its structure:



```
new technologies
the old
new skills
```

PROGRAM STRUCTURE

The overall structure of the program can be divided into 3 parts and 2 subparts for 2&3:

1. Taking the parameters from the command line
2. Master part
   2.1. Merge method MASTER
   2.2. Merge method WORKERS
   2.3. Test part
3. Worker part
   3.1. Merge method MASTER
   3.2. Merge method WORKERS
4. Probability Calculation

1. Taking the parameters from the command line

To read from the command line we used built-in “sys”. All other parts of the project are done with built-in data structures. First, we take the number of processors by using Get_Size(). The number of workers is “this value-1” since one of them is the master.

In a for loop, we are searching for the parameter names which start with “--”. Those are: merge method, input file path and test file path. After finding these parameter names, we are taking the next argument for the actual parameter values.

2. Master part

If the rank of the processor is 0, then this indicates that the process is master. Thus, the if check returns true. Inside we are opening the input file with the “encoding="utf-"8”, since there can be Turkish characters inside the input text. Then by using for loop, we are dividing this text into lines and clear it from spaces by using strip().

To indicate each share of the works for each worker, we used share_array list. We find the quotient and remainder. We add the remainder to the first workers 1 by 1 and then, master sends the input lines to the workers with tag = 1.

2.1. Merge method MASTER

If the MASTER is chosen as merge_method, the if condition returns true. In this method, the master receives the frequency of unigrams and bigrams from each worker with tag = 2. Then in each iteration, it merges the received frequency with the previous frequencies to the result dictionary. To do this, we write additional function named merge(dict1,dict2).

Merge Function:

It takes 2 parameters named dict1 and dict2. First, it searches the key list of the dict1 and if there is same key in the dict2, it adds the total frequencies and updates the dict1 dictionary. It also deletes the same key in the dict2, in order to prevent overriding. At last, it uses built-in update() function to add dict2 to the dict1. Then, it returns the final dictionary, dict1.

2.2. Merge method WORKERS

If WORKERS chosen as merge_method, then master does not need to do anything to the result it received. It uses as it is. It received the data with tag = <the rank of the last worker>.

2.3. Test part

It reads the test_file from command line with the same procedure in the input_file. Then in a for loop, it checks if the bigram is contained in the result dictionary and takes its frequency if there is. It uses the built-in split() function and takes its first element to take the first word of the bigram. Then, we check if the result dictionary contains this unigram frequency. Then it takes this frequency if there is one. By the method we are given in the project description, we divide these 2 frequencies and find the conditional probability of the bigram. Additionally, we check if the unigram frequency is not 0 in order to prevent 0 division.

3. Worker part

Workers use dictionary structure to hold the frequencies for each unigrams and bigrams. They receive the input lines from the master with tag = 1, then it writes its rank and number of lines it received as required. For each line in its data, worker divides it into its tokens. If token is included in the word_dictionary, it increases it by 1; if not, it makes it 1. With this method, the unigram frequencies are calculated.

For the bigrams, it uses the statement:

[tokens[i:i+2] for i in range(0, len(tokens)-1)].

In this, it produces tokens[0:2], tokens[1:3]… as bigrams. Since this is a list and it is not hashable, we are turning this into string by the statement:

bigram = ' '.join([str(s) for s in bi_token]).

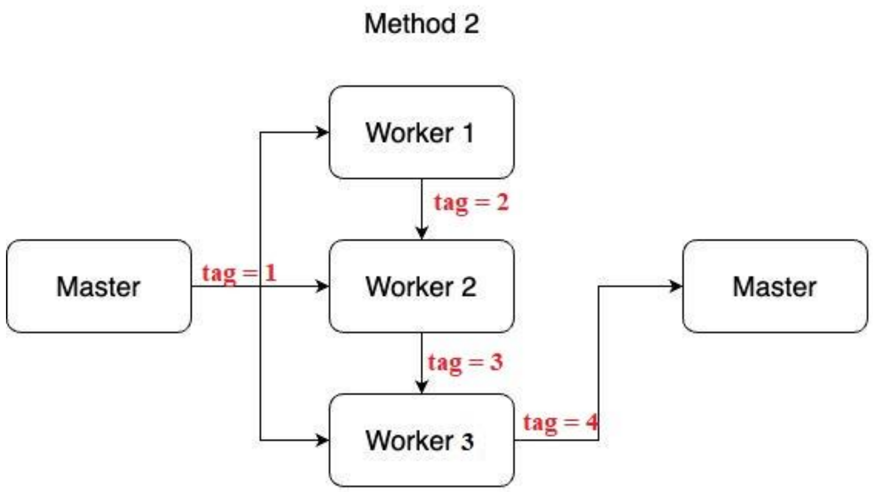
Then, it checks if the bigram is contained in word_dictionary key list. If there is one, it increases it by 1; if not, it makes it 1. Then it send the data to the master or to the next worker according to the merge_method.

3.1. Merge method MASTER

If this merge method is chosen, then the worker sends its data directly to the master with tag = 2.

3.2. Merge method WORKERS

If this merge method is chosen, then the worker sends its data to the next worker. Thus, they should have the same tag number. Thus, while receiving the data, we send it with the tag number = rank of the worker. Then it merges its data with the received data by using the merge function explained above. While sending the data, it uses the tag number = rank of the worker + 1 since it sends to the next worker. If the worker is the last processor; then it sends the data to the master with tag = <rank of the worker +1> which is equal to the number of processors. We tried to summarize the procedure by using the figure given in the project description below. In this example number of processors is 4, so there are 3 workers.



The tag numbers:

We used tag numbers as if they represent the priority. In MPI, the sending and receiving part should have the same tag number in their comm.send() and comm.recv() functions. In our structure, first, master sends its data to the workers. So in this, we used tag=1 to send the data from the master and receive the data from the master in workers part. In MASTER merge method, it uses tag=2 with the same idea. The same idea is applied to the merge method WORKERS.

4. Probability Calculation

We first find the frequency of the first word in the bigram in the sample text. Then we find the frequency of the bigram itself in the sample text. Their division gives us the conditional probability of the bigram in the given text. We also check for "division by 0" to not encounter any errors in case the bigram does not exist in the text. The formula we used for our calculation can be found below:

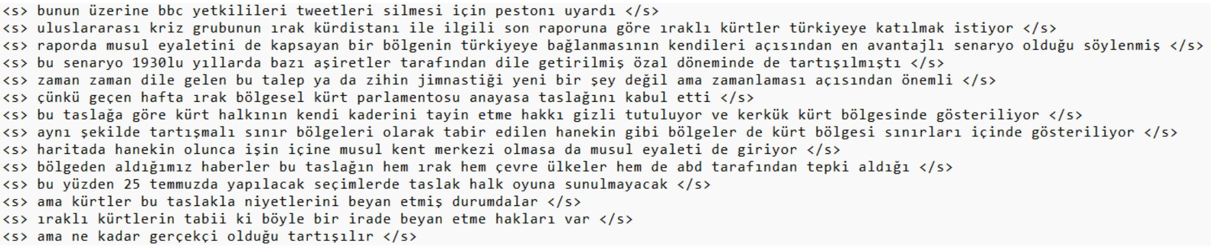
$$P(transparent|its\ water\ is\ so) = \frac{Freq(its\ water\ is\ so\ transparent)}{Freq(its\ water\ is\ so)}$$

P(its water is so transparent that) = P(its) * P(water|its) * P(is|water) * P(so|is) * P(transparent|so) * P(that|transparent)

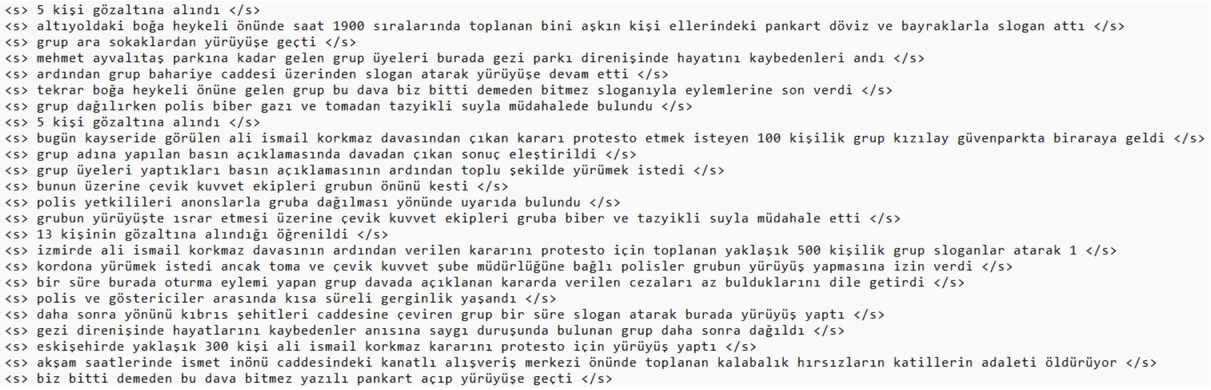
EXAMPLES

Our sample text that we've used to test our code consists of 236434 lines and 4010649 words. Each line starts with the "<s>" character and ends with the "</s>" character. Each word is seperated by one or more space characters.

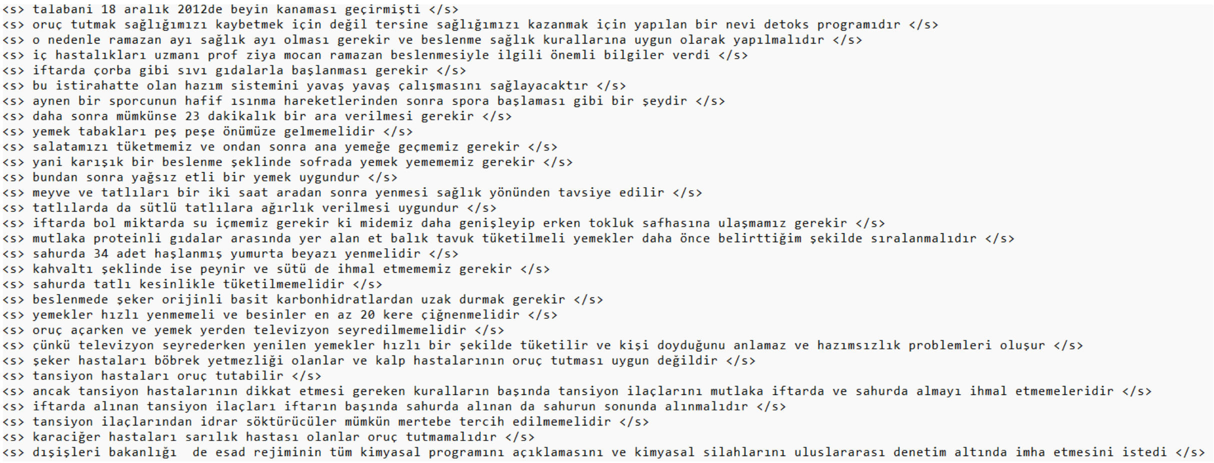
When we tested our code manually to check if we've found the right results with our code, we concluded that there were more than enough scenarios included in this sample text for different possible cases. The size of the document was appropriate for testing the efficiency of our code. Words were at times seperated by more than 1 space and they included affix and prefixes which did not exist in our searched word. This allowed us to test our program thoroughly. Here is a snippet of a few lines of the sample file:



```
<s> bunun üzerine bbc yetkilileri tweetleri silmesi için pestonı uyardı </s>
<s> uluslararası kriz grubunun ırak kürdistanı ile ilgili son raporuna göre ıraklı kürtler türkiyeye katılmak istiyor </s>
<s> raporda musul eyaletini de kapsayan bir bölgenin türkiyeye bağlanmasının kendileri açısından en avantajlı senaryo olduğu söylenmiş </s>
<s> bu senaryo 1930lu yıllarda bazı aşiretler tarafından dile getirilmiş özal döneminde de tartışılmıştı </s>
<s> zaman zaman dile gelen bu talep ya da zihin jimnastiği yeni bir şey değil ama zamanlaması açısından önemli </s>
<s> çünkü geçen hafta ırak bölgesel kürt parlamentosu anayasa taslağını kabul etti </s>
<s> bu taslağa göre kürt halkının kendi kaderini tayin etme hakkı gizli tutuluyor ve kerkük kürt bölgesinde gösteriliyor </s>
<s> aynı şekilde tartışmalı sınır bölgeleri olarak tabir edilen hanekin gibi bölgeler de kürt bölgesi sınırları içinde gösteriliyor </s>
<s> haritada hanekin olunca işin içine musul kent merkezi olmasa da musul eyaleti de giriyor </s>
<s> bölgeden aldığımız haberler bu taslağın hem ırak hem çevre ülkeler hem de abd tarafından tepki aldığı </s>
<s> bu yüzden 25 temmuzda yapılacak seçimlerde taslak halk oyuna sunulmayacak </s>
<s> ama kürtler bu taslakla niyetlerini beyan etmiş durumdalar </s>
<s> ıraklı kürtlerin tabii ki böyle bir irade beyan etme hakları var </s>
<s> ama ne kadar gerçekçi olduğu tartışılır </s>
```

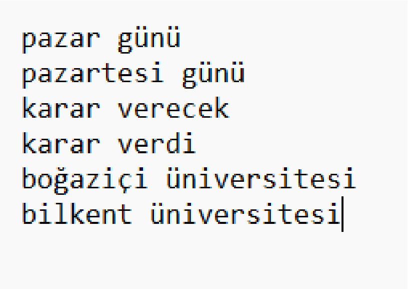


```
<s> 5 kişi gözaltına alındı </s>
<s> altıyoldaki boğa heykeli önünde saat 1900 sıralarında toplanan bini aşkın kişi ellerindeki pankart döviz ve bayraklarla slogan attı </s>
<s> grup ara sokaklardan yürüyüşe geçti </s>
<s> mehmet ayvalıtaş parkına kadar gelen grup üyeleri burada gezi parkı direnişinde hayatını kaybedenleri andı </s>
<s> ardından grup bahariye caddesi üzerinden slogan atarak yürüyüşe devam etti </s>
<s> tekrar boğa heykeli önüne gelen grup bu dava biz bitti demeden bitmez sloganıyla eylemlerine son verdi </s>
<s> grup dağılırken polis biber gazı ve tomadan tazyikli suyla müdahalede bulundu </s>
<s> 5 kişi gözaltına alındı </s>
<s> bugün kayseride görülen ali ismail korkmaz davasından çıkan kararı protesto etmek isteyen 100 kişilik grup kızılay güvenparkta biraraya geldi </s>
<s> grup adına yapılan basın açıklamasında davadan çıkan sonuç eleştirildi </s>
<s> grup üyeleri yaptıkları basın açıklamasının ardından toplu şekilde yürümek istedi </s>
<s> bunun üzerine çevik kuvvet ekipleri grubun önünü kesti </s>
<s> polis yetkilileri anonslarla gruba dağılması yönünde uyarıda bulundu </s>
<s> grubun yürüyüşte ısrar etmesi üzerine çevik kuvvet ekipleri gruba biber ve tazyikli suyla müdahale etti </s>
<s> 13 kişinin gözaltına alındığı öğrenildi </s>
<s> izmirde ali ismail korkmaz davasının ardından verilen kararını protesto için toplanan yaklaşık 500 kişilik grup sloganlar atarak 1 </s>
<s> kordona yürümek istedi ancak toma ve çevik kuvvet şube müdürlüğüne bağlı polisler grubun yürüyüş yapmasına izin verdi </s>
<s> bir süre burada oturma eylemi yapan grup davada açıklanan kararda verilen cezaları az bulduklarını dile getirdi </s>
<s> polis ve göstericiler arasında kısa süreli gerginlik yaşandı </s>
<s> daha sonra yönünü kıbrıs şehitleri caddesine çeviren grup bir süre slogan atarak burada yürüyüş yaptı </s>
<s> gezi direnişinde hayatlarını kaybedenler anısına saygı duruşunda bulunan grup daha sonra dağıldı </s>
<s> eskişehirde yaklaşık 300 kişi ali ismail korkmaz kararını protesto için yürüyüş yaptı </s>
<s> akşam saatlerinde ismet inönü caddesindeki kanatlı alışveriş merkezi önünde toplanan kalabalık hırsızların katillerin adaleti öldürüyor </s>
<s> biz bitti demeden bu dava bitmez yazılı pankart açıp yürüyüşe geçti </s>
```



```
<s> talabani 18 aralık 2012de beyin kanaması geçirmişti </s>
<s> oruç tutmak sağlığımızı kaybetmek için değil tersine sağlığımızı kazanmak için yapılan bir nevi detoks programıdır </s>
<s> o nedenle ramazan ayı sağlık ayı olması gerekir ve beslenme sağlık kurallarına uygun olarak yapılmalıdır </s>
<s> iç hastalıkları uzmanı prof ziya mocan ramazan beslenmesiyle ilgili önemli bilgiler verdi </s>
<s> iftarda çorba gibi sıvı gıdalarla başlanması gerekir </s>
<s> bu istirahatte olan hazım sistemini yavaş yavaş çalışmasını sağlayacaktır </s>
<s> aynen bir sporcunun hafif ısınma hareketlerinden sonra spora başlaması gibi bir şeydir </s>
<s> daha sonra mümkünse 23 dakikalık bir ara verilmesi gerekir </s>
<s> yemek tabakları peş peşe önümüze gelmemelidir </s>
<s> salatamızı tüketmemiz ve ondan sonra ana yemeğe geçmemiz gerekir </s>
<s> yani karışık bir beslenme şeklinde sofrada yemek yemememiz gerekir </s>
<s> bundan sonra yağsız etli bir yemek uygundur </s>
<s> meyve ve tatlıları bir iki saat aradan sonra yenmesi sağlık yönünden tavsiye edilir </s>
<s> tatlılarda da sütlü tatlılara ağırlık verilmesi uygundur </s>
<s> iftarda bol miktarda su içmemiz gerekir ki midemiz daha genişleyip erken tokluk safhasına ulaşmamız gerekir </s>
<s> mutlaka proteinli gıdalar arasında yer alan et balık tavuk tüketilmeli yemekler daha önce belirttiğim şekilde sıralanmalıdır </s>
<s> sahurda 34 adet haşlanmış yumurta beyazı yenmelidir </s>
<s> kahvaltı şeklinde ise peynir ve sütü de ihmal etmememiz gerekir </s>
<s> sahurda tatlı kesinlikle tüketilmemelidir </s>
<s> beslenmede şeker orijinli basit karbonhidratlardan uzak durmak gerekir </s>
<s> yemekler hızlı yenmemeli ve besinler en az 20 kere çiğnenmelidir </s>
<s> oruç açarken ve yemek yerden televizyon seyredilmemelidir </s>
<s> çünkü televizyon seyrederken yenilen yemekler hızlı bir şekilde tüketilir ve kişi doyduğunu anlamaz ve hazımsızlık problemleri oluşur </s>
<s> şeker hastaları böbrek yetmezliği olanlar ve kalp hastalarının oruç tutması uygun değildir </s>
<s> tansiyon hastaları oruç tutabilir </s>
<s> ancak tansiyon hastalarının dikkat etmesi gereken kuralların başında tansiyon ilaçlarını mutlaka iftarda ve sahurda almayı ihmal etmemeleridir </s>
<s> iftarda alınan tansiyon ilaçları iftarın başında sahurda alınan da sahurun sonunda alınmalıdır </s>
<s> tansiyon ilaçlarından idrar söktürücüler mümkün mertebe tercih edilmemelidir </s>
<s> karaciğer hastaları sarılık hastası olanlar oruç tutmamalıdır </s>
<s> dışişleri bakanlığı  de esad rejiminin tüm kimyasal programını açıklamasını ve kimyasal silahlarını uluslararası denetim altında imha etmesini istedi </s>
```

Our test file included bigrams (sequence of 2 words) that was included in our sample file many times. These bigrams existed in the sample text in different forms (such as with or without affixes). Here is the contents of the test file:



```
pazar günü
pazartesi günü
karar verecek
karar verdi
boğaziçi üniversitesi
bilkent üniversitesi
```

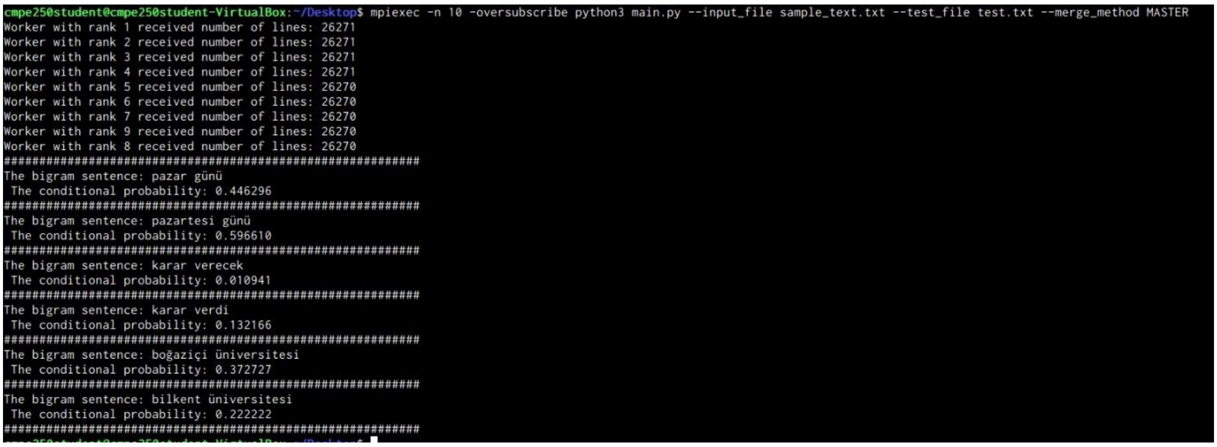
While running our program, we were able to change two parameters:

First was the number of processes (1 master and varying number of worker processes) which was minimum 2 (1 master, 1 worker process). This parameter also affected our output. Since we print the number of lines received by each worker process, by changing the number of worker processes, we changed how many lines they each receive.

Second parameter we could change was the merge method (MASTER or WORKER). This change did not affect our output, just the inner workings of the program.

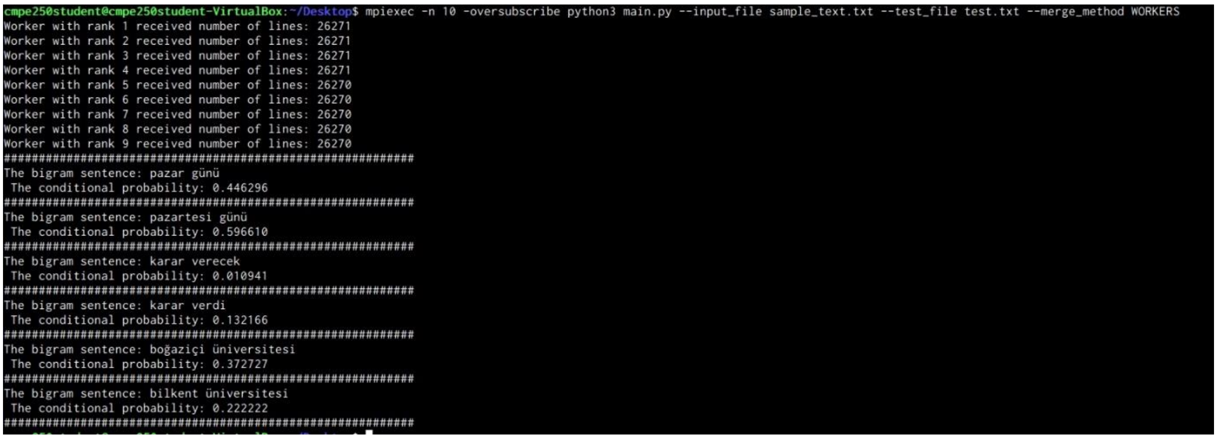
Both parameter changes did not affect the final findings of probability since these values depend only on the sample and test files. They only affected the time it took to find the results. Below are different outputs for different values of the parameters:

10 processes (1 master, 9 workers) – Merge method: MASTER:



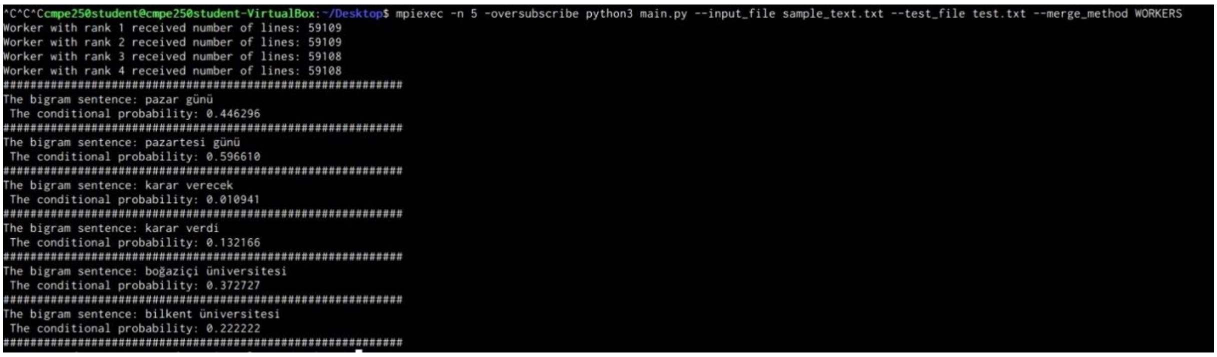
```
cmpe250student@cmpe250student-VirtualBox:~/Desktop$ mpiexec -n 10 -oversubscribe python3 main.py --input_file sample_text.txt --test_file test.txt --merge_method MASTER
Worker with rank 1 received number of lines: 26271
Worker with rank 2 received number of lines: 26271
Worker with rank 3 received number of lines: 26271
Worker with rank 4 received number of lines: 26271
Worker with rank 5 received number of lines: 26270
Worker with rank 6 received number of lines: 26270
Worker with rank 7 received number of lines: 26270
Worker with rank 9 received number of lines: 26270
Worker with rank 8 received number of lines: 26270
###########################################################
The bigram sentence: pazar günü
 The conditional probability: 0.446296
###########################################################
The bigram sentence: pazartesi günü
 The conditional probability: 0.596610
###########################################################
The bigram sentence: karar verecek
 The conditional probability: 0.010941
###########################################################
The bigram sentence: karar verdi
 The conditional probability: 0.132166
###########################################################
The bigram sentence: boğaziçi üniversitesi
 The conditional probability: 0.372727
###########################################################
The bigram sentence: bilkent üniversitesi
 The conditional probability: 0.222222
###########################################################
```

10 processes (1 master, 9 workers) – Merge method: WORKERS:



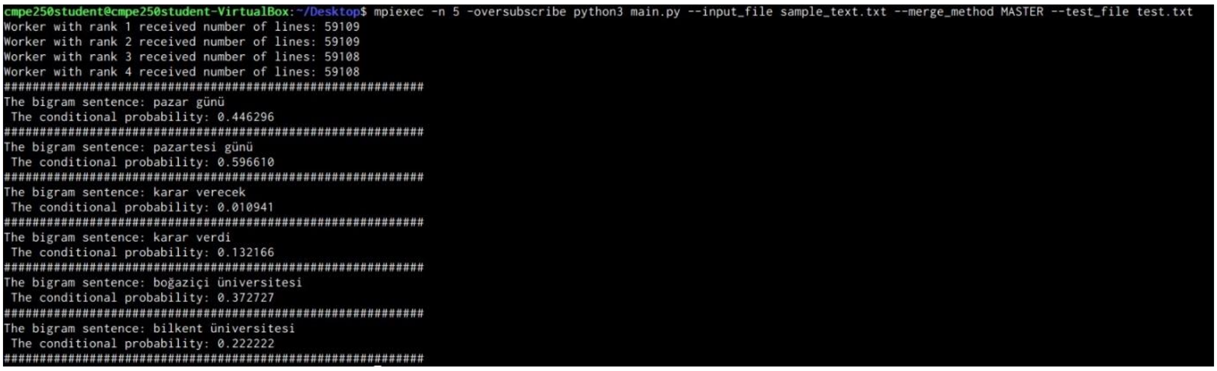
```
cmpe250student@cmpe250student-VirtualBox:~/Desktop$ mpiexec -n 10 -oversubscribe python3 main.py --input_file sample_text.txt --test_file test.txt --merge_method WORKERS
Worker with rank 1 received number of lines: 26271
Worker with rank 2 received number of lines: 26271
Worker with rank 3 received number of lines: 26271
Worker with rank 4 received number of lines: 26271
Worker with rank 5 received number of lines: 26270
Worker with rank 6 received number of lines: 26270
Worker with rank 7 received number of lines: 26270
Worker with rank 8 received number of lines: 26270
Worker with rank 9 received number of lines: 26270
###########################################################
The bigram sentence: pazar günü
 The conditional probability: 0.446296
###########################################################
The bigram sentence: pazartesi günü
 The conditional probability: 0.596610
###########################################################
The bigram sentence: karar verecek
 The conditional probability: 0.010941
###########################################################
The bigram sentence: karar verdi
 The conditional probability: 0.132166
###########################################################
The bigram sentence: boğaziçi üniversitesi
 The conditional probability: 0.372727
###########################################################
The bigram sentence: bilkent üniversitesi
 The conditional probability: 0.222222
###########################################################
```

5 processes (1 master, 4 workers) – Merge method: WORKERS:



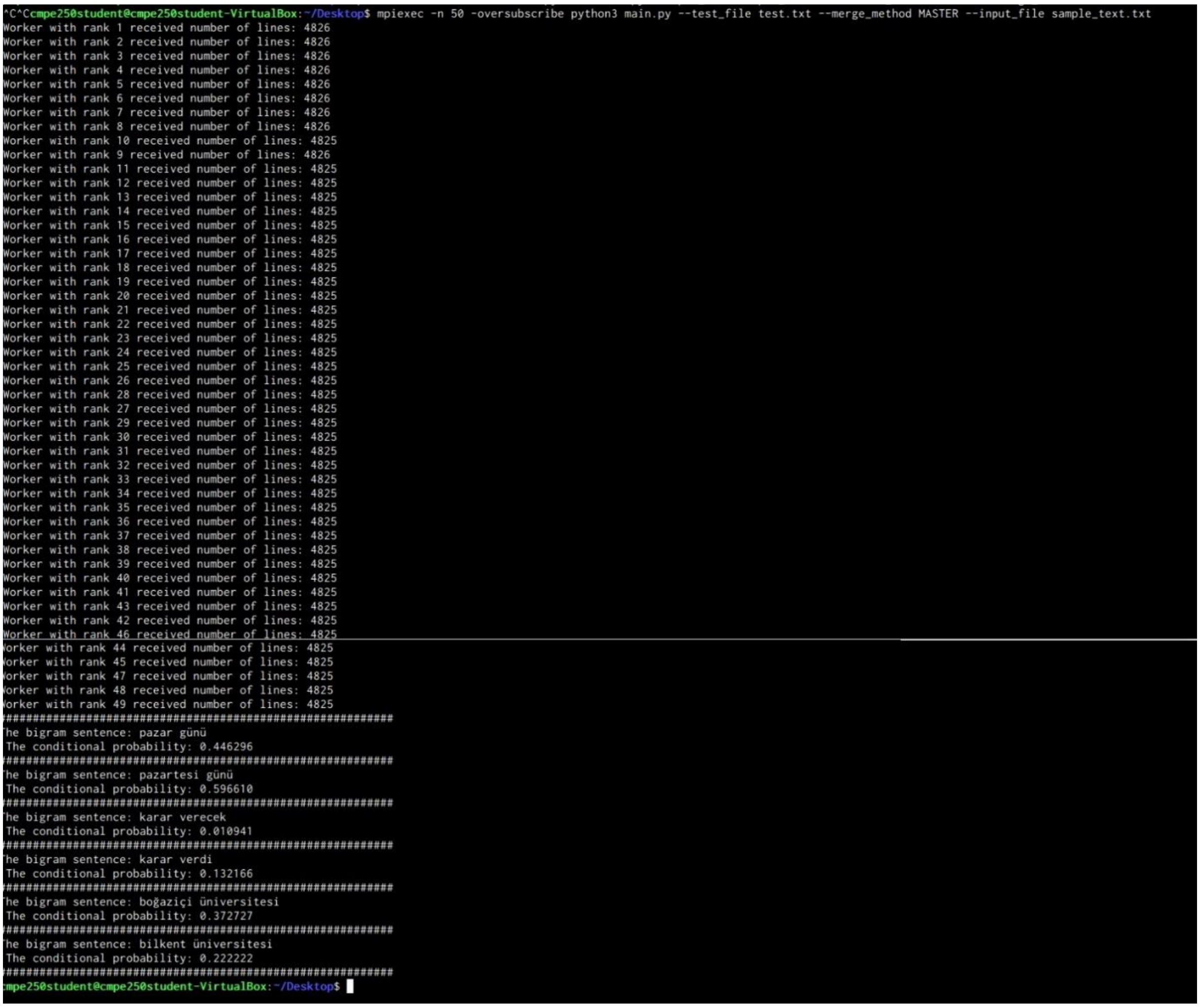
```
^C^C^Ccmpe250student@cmpe250student-VirtualBox:~/Desktop$ mpiexec -n 5 -oversubscribe python3 main.py --input_file sample_text.txt --test_file test.txt --merge_method WORKERS
Worker with rank 1 received number of lines: 59109
Worker with rank 2 received number of lines: 59109
Worker with rank 3 received number of lines: 59108
Worker with rank 4 received number of lines: 59108
###########################################################
The bigram sentence: pazar günü
 The conditional probability: 0.446296
###########################################################
The bigram sentence: pazartesi günü
 The conditional probability: 0.596610
###########################################################
The bigram sentence: karar verecek
 The conditional probability: 0.010941
###########################################################
The bigram sentence: karar verdi
 The conditional probability: 0.132166
###########################################################
The bigram sentence: boğaziçi üniversitesi
 The conditional probability: 0.372727
###########################################################
The bigram sentence: bilkent üniversitesi
 The conditional probability: 0.222222
###########################################################
```

5 processes (1 master, 4 workers) – Merge method: MASTER:



```
cmpe250student@cmpe250student-VirtualBox:~/Desktop$ mpiexec -n 5 -oversubscribe python3 main.py --input_file sample_text.txt --merge_method MASTER --test_file test.txt
Worker with rank 1 received number of lines: 59109
Worker with rank 2 received number of lines: 59109
Worker with rank 3 received number of lines: 59108
Worker with rank 4 received number of lines: 59108
###########################################################
The bigram sentence: pazar günü
 The conditional probability: 0.446296
###########################################################
The bigram sentence: pazartesi günü
 The conditional probability: 0.596610
###########################################################
The bigram sentence: karar verecek
 The conditional probability: 0.010941
###########################################################
The bigram sentence: karar verdi
 The conditional probability: 0.132166
###########################################################
The bigram sentence: boğaziçi üniversitesi
 The conditional probability: 0.372727
###########################################################
The bigram sentence: bilkent üniversitesi
 The conditional probability: 0.222222
###########################################################
```

50 processes (1 master, 49 workers) – Merge method: MASTER



```
^C^Ccmpe250student@cmpe250student-VirtualBox:~/Desktop$ mpiexec -n 50 -oversubscribe python3 main.py --test_file test.txt --merge_method MASTER --input_file sample_text.txt
Worker with rank 1 received number of lines: 4826
Worker with rank 2 received number of lines: 4826
Worker with rank 3 received number of lines: 4826
Worker with rank 4 received number of lines: 4826
Worker with rank 5 received number of lines: 4826
Worker with rank 6 received number of lines: 4826
Worker with rank 7 received number of lines: 4826
Worker with rank 8 received number of lines: 4826
Worker with rank 10 received number of lines: 4825
Worker with rank 9 received number of lines: 4826
Worker with rank 11 received number of lines: 4825
Worker with rank 12 received number of lines: 4825
Worker with rank 13 received number of lines: 4825
Worker with rank 14 received number of lines: 4825
Worker with rank 15 received number of lines: 4825
Worker with rank 16 received number of lines: 4825
Worker with rank 17 received number of lines: 4825
Worker with rank 18 received number of lines: 4825
Worker with rank 19 received number of lines: 4825
Worker with rank 20 received number of lines: 4825
Worker with rank 21 received number of lines: 4825
Worker with rank 22 received number of lines: 4825
Worker with rank 23 received number of lines: 4825
Worker with rank 24 received number of lines: 4825
Worker with rank 25 received number of lines: 4825
Worker with rank 26 received number of lines: 4825
Worker with rank 28 received number of lines: 4825
Worker with rank 27 received number of lines: 4825
Worker with rank 29 received number of lines: 4825
Worker with rank 30 received number of lines: 4825
Worker with rank 31 received number of lines: 4825
Worker with rank 32 received number of lines: 4825
Worker with rank 33 received number of lines: 4825
Worker with rank 34 received number of lines: 4825
Worker with rank 35 received number of lines: 4825
Worker with rank 36 received number of lines: 4825
Worker with rank 37 received number of lines: 4825
Worker with rank 38 received number of lines: 4825
Worker with rank 39 received number of lines: 4825
Worker with rank 40 received number of lines: 4825
Worker with rank 41 received number of lines: 4825
Worker with rank 43 received number of lines: 4825
Worker with rank 42 received number of lines: 4825
Worker with rank 46 received number of lines: 4825
orker with rank 44 received number of lines: 4825
orker with rank 45 received number of lines: 4825
orker with rank 47 received number of lines: 4825
orker with rank 48 received number of lines: 4825
orker with rank 49 received number of lines: 4825
############################################################
he bigram sentence: pazar günü
 The conditional probability: 0.446296
############################################################
he bigram sentence: pazartesi günü
 The conditional probability: 0.596610
############################################################
he bigram sentence: karar verecek
 The conditional probability: 0.010941
############################################################
he bigram sentence: karar verdi
 The conditional probability: 0.132166
############################################################
he bigram sentence: boğaziçi üniversitesi
 The conditional probability: 0.372727
############################################################
he bigram sentence: bilkent üniversitesi
 The conditional probability: 0.222222
############################################################
mpe250student@cmpe250student-VirtualBox:~/Desktop$ 
```

IMPROVEMENTS AND EXTENSIONS

While writing the code, we tried to write clean, understandable, and working code. However, at some places such as for search algorithms; we wrote our code without considering its execution time. We may improve our code by changing some of the algorithms we used, so we can get better execution times in larger inputs. Also, we did not do any error check and assumed that user will enter a proper input. We can add error checks to our code as well. However overall, we are satisfied of our work.

DIFFUCULTIES ENCOUNTERED

We had some difficulties while trying to understand how MPI environment works. This was the first time we have encountered with it. First, we tried to understand how it works by writing simple codes. The difficulty is that the order of the execution between master-processor and processor-processor codes. Then, we figured out that we should use “tag” parameter for this purpose and match the tag numbers with receiving and sending part. After that, we could do whole project easily.

CONCLUSIONS

Through this project, we learned about the concepts of mpi framework, master and worker processes and how they work, n-gram language in language processing and probability calculation over textual data.