**DOKUZ EYLÜL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**INTRODUCTION TO NATURAL LANGUAGE PROCESSING ASSIGNMENT REPORT**

2014510048

Melda Karadağ

# CHAPTER 1

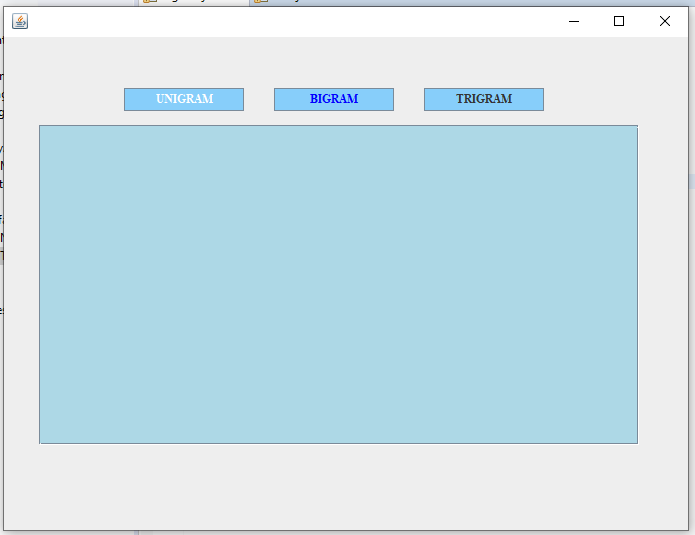
# JAVA AND PYTHON ALGORITHMS

The scope of this assignment, to develop n-gram algorithms working (for 1,2 and 3-grams of Markov Model) on a folder which has different texts. Based on this, I have tried to apply approaches based on the relationship between 1,2 and 3 words using java and python software language.

**1.1 Java Project**

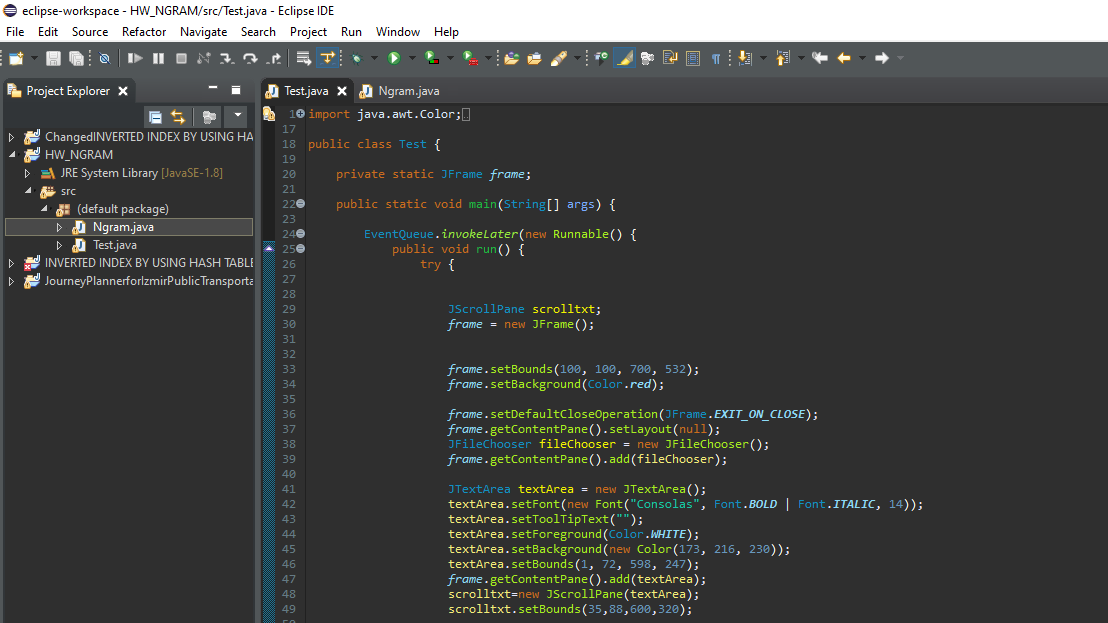
Java project includes two classes named Ngram and Test. The "ngram" class is a class where unigram, bigram and trigram relationships are calculated and holds the corresponding result and decimal values. The variable newArray used in this class was created to be used as the result value to be printed on the screen. The separate file read operations for unigram, bigram and trigram operations are designed so that different files can be selected at different times since different buttons are designed for each operation. The "uniGram", "biGram", "triGram" functions are designed as void and designed as master class. In this type of Ngram class results such as "bigram" in accordance with the variables defined in the process were obtained.

The initial interface designed for all operations is as follows:

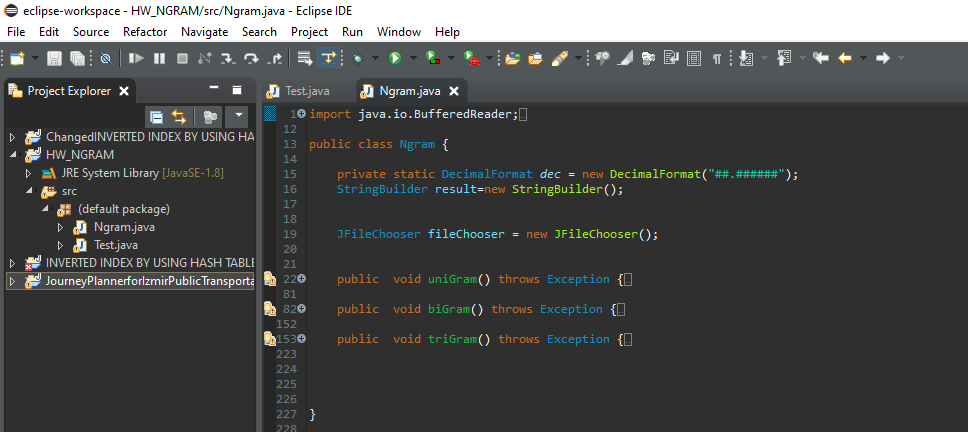


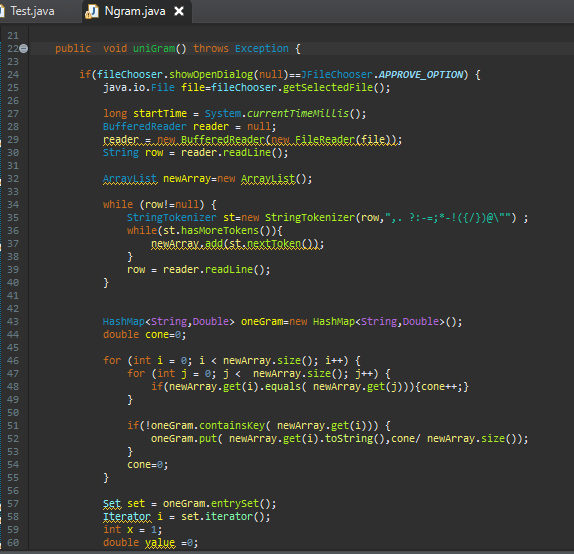
In the "Test" class, the interface design was performed and the return results of the transaction results made in the Ngram class were realized. This process results are shown on the screen by using button, textArea and scrollPane java swing components.

The code developed for the overall structure of the page is as follows:

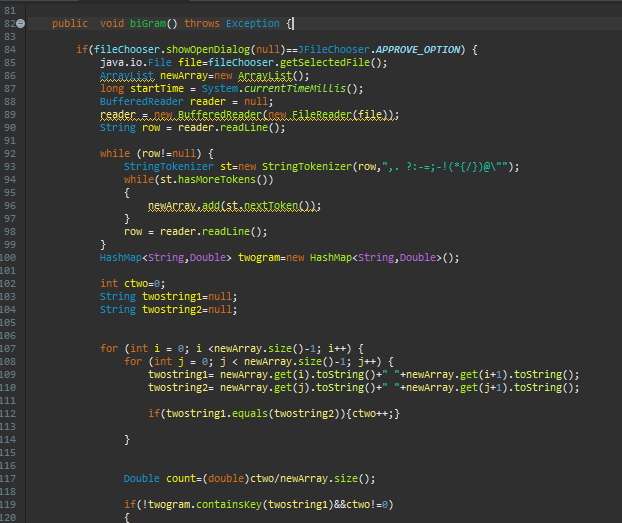


For each analysis, functions are designed that perform the same operation as the name. The general structure of the "ngram" class is as follows:

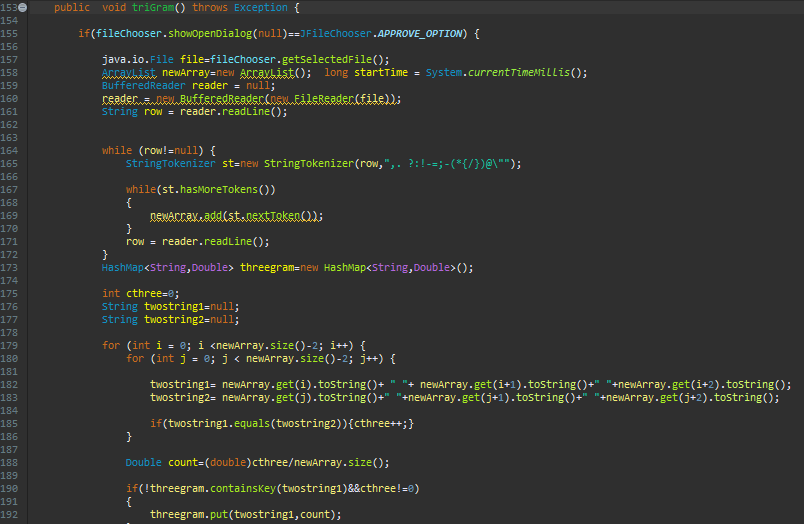




Code segment of unigram analysis is shown above.



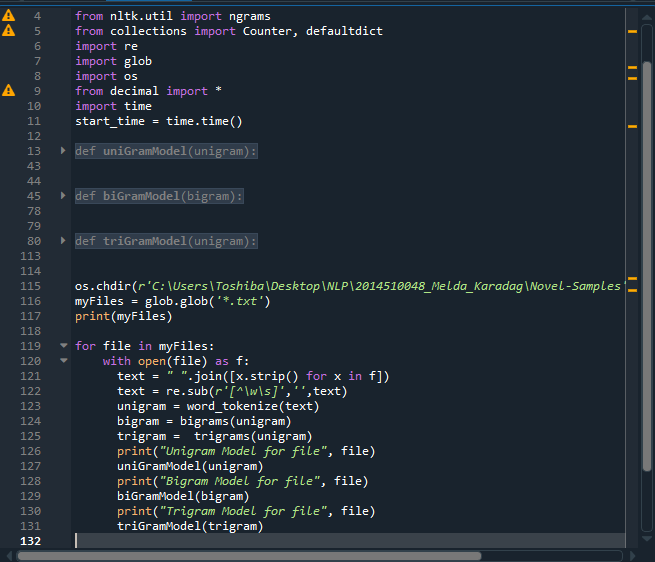
Code segment of bigram analysis is shown above.



Code segment of trigram analysis is shown above.

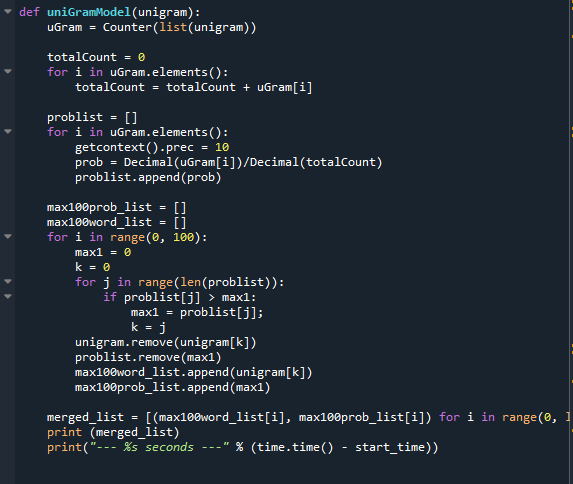
**1.2 Python Project**

Python project is one class project named NGRAM. Like the java project it has three functions that does n-gram analysis which are “uniGramModel”, “biGramModel”, “triGramModel”. Also each fuction is calculating their time score. There is one more function which opens the txt files folder and read them one by one. Python project does not have interface, we see the result on console.

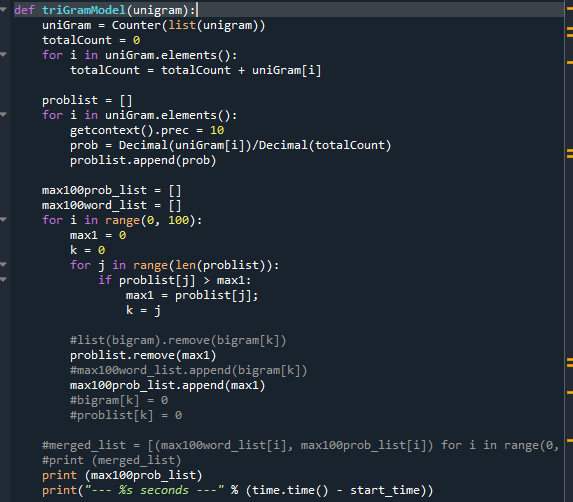




Unigram’s function is shown above.



Bigram’s function is shown above.



Trigram’s function is shown above.

Unlike the java project, in Python when I create bigram and trigram structures can not be sliced, that is why however I can find each structures scores but I can’t match them with their own words. So bigram and trigram fuctions can return only probabilities not with the names of that probabilities.

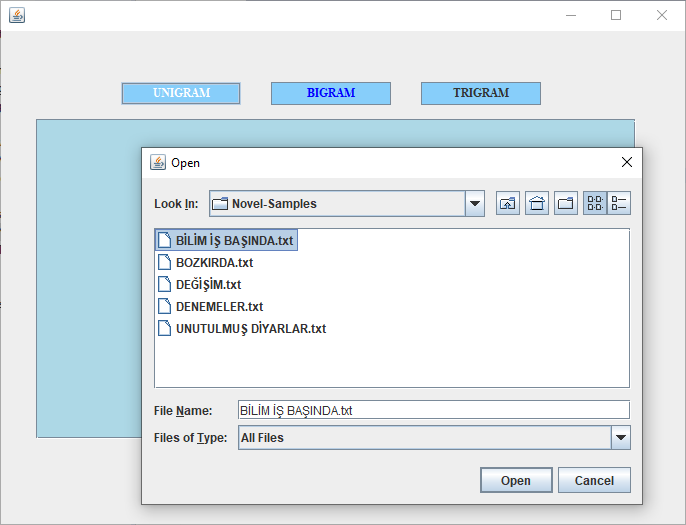
# CHAPTER 2

# TEST RESULTS

**2.1 Unigram Results**

**a. Java**

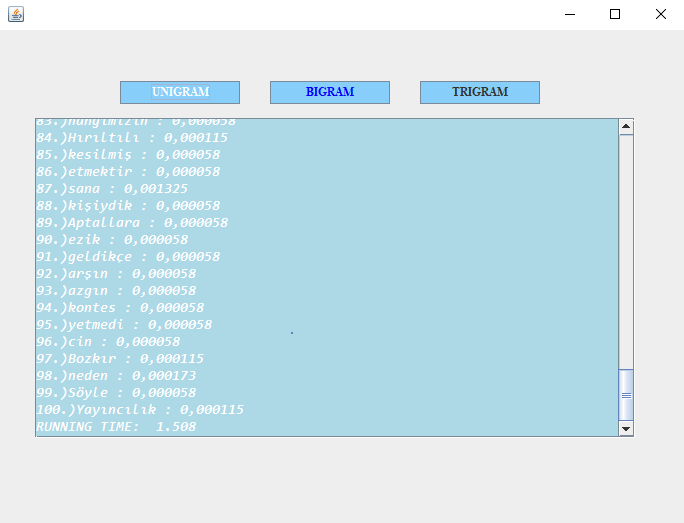
After pressing the button designed for 3 different applications, pop-up screen appears as text file selection. The screen that appears after pressing the Unigram key is shown as follows:



The results obtained from the unigram function applied for 5 different text are as follows:



Results of ”BİLİM İŞ BAŞINDA”



Results of ”BOZKIRDA”

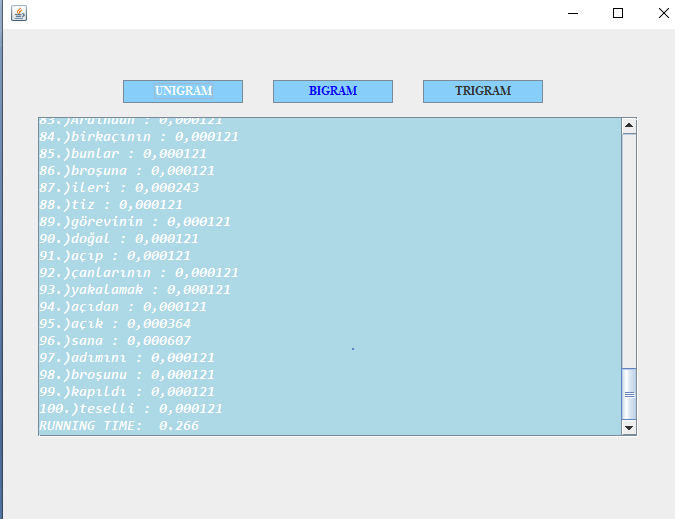
# 

# 

Results of ”DEĞİŞİM”



Results of ”DENEMELER”



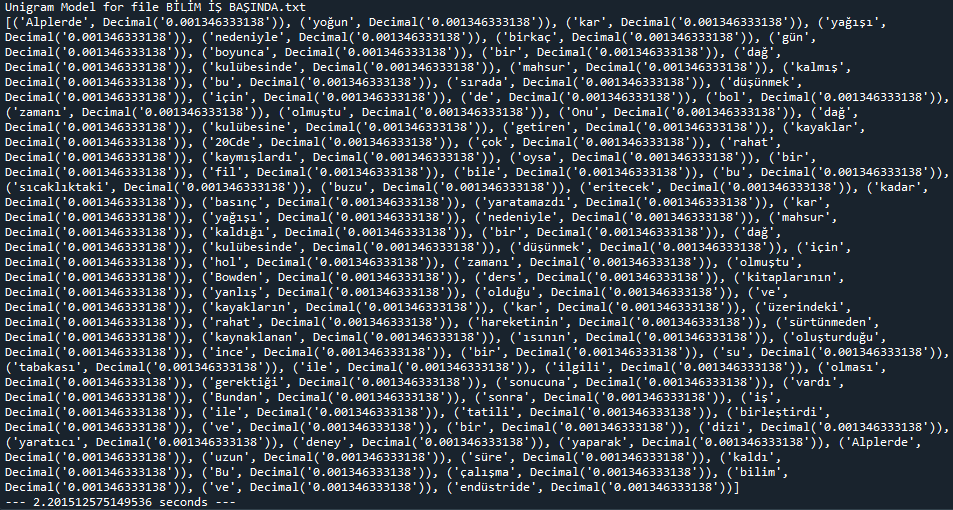
Results of ”UNUTULMUŞ DİYARLAR”

**b. Python**

After to run the program, each file unigram, bigram, trigram values are shown on the console with in each others time.



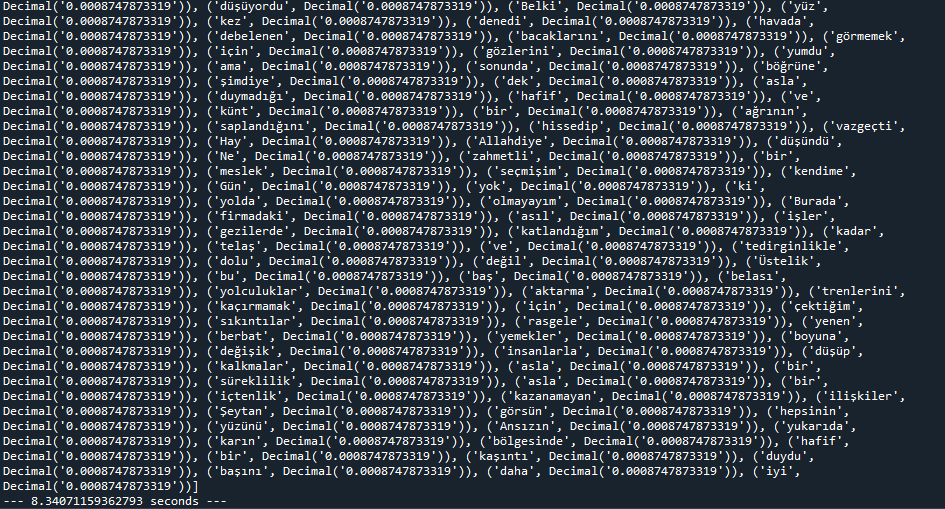
Results of ”BOZKIRDA”



Results of ”BİLİM İŞ BAŞINDA”



Results of “DENEMELER”



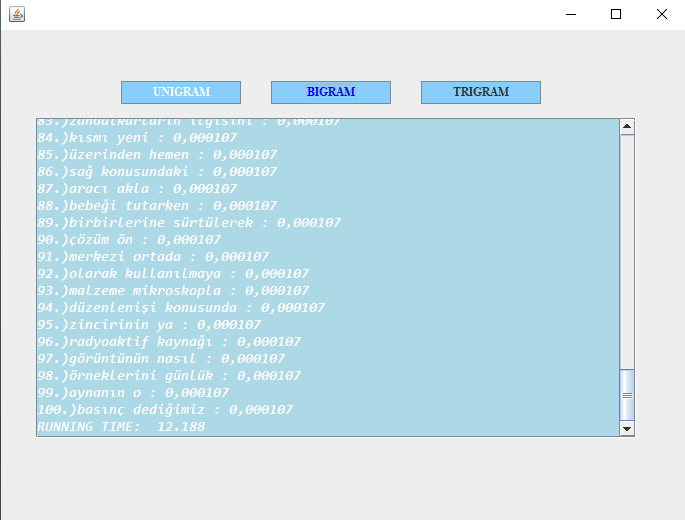
Results of “DEĞİŞİM”



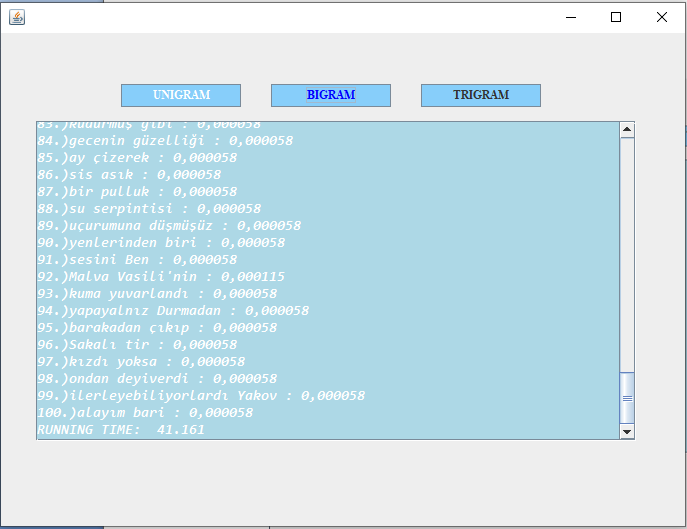
Results of “DİYARLAR”

**2.2 Bigram Results**

**a. Java**



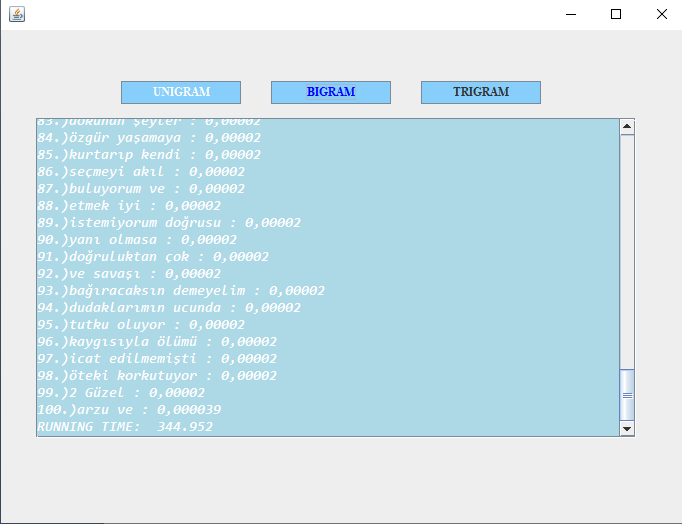
Results of ”BİLİM İŞ BAŞINDA”



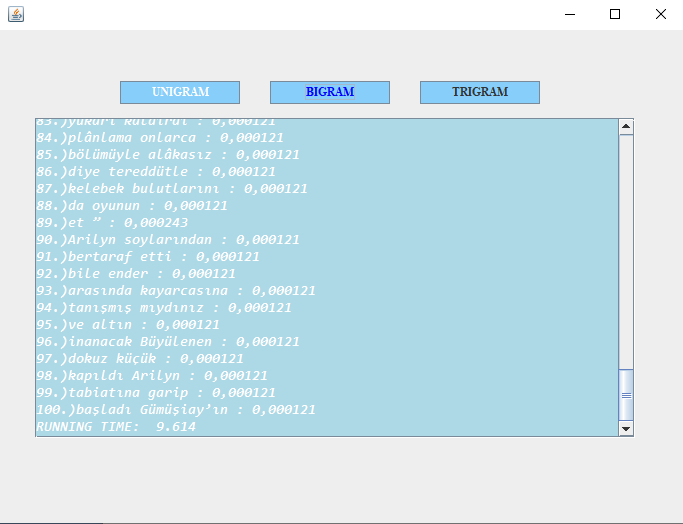
Results of ”BOZKIRDA”



Results of ”DEĞİŞİM”

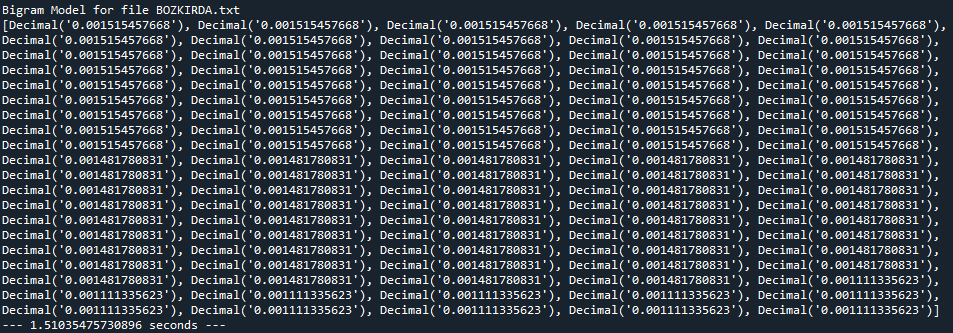


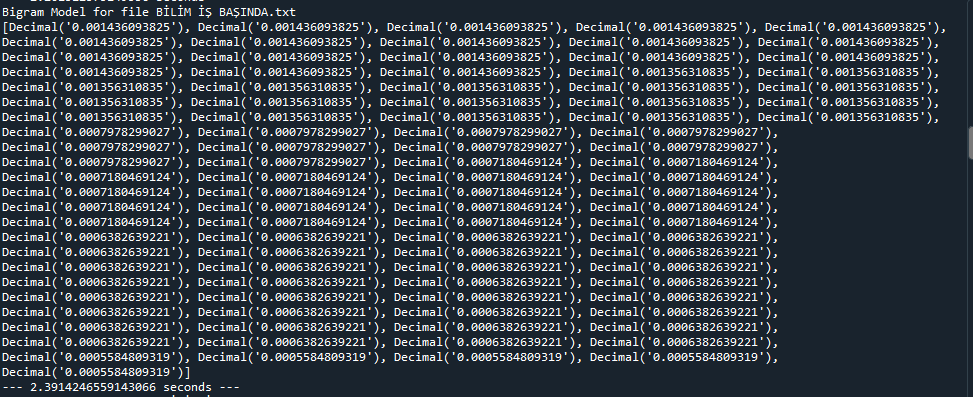
Results of ”DENEMELER”

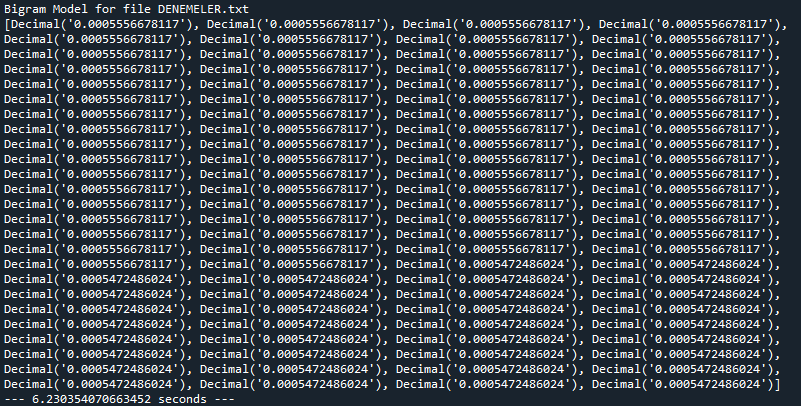


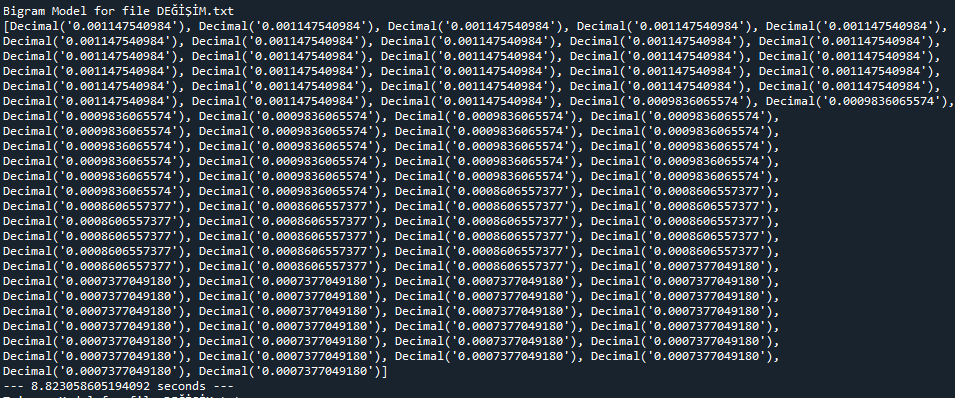
Results of ”UNUTULMUŞ DİYARLAR”

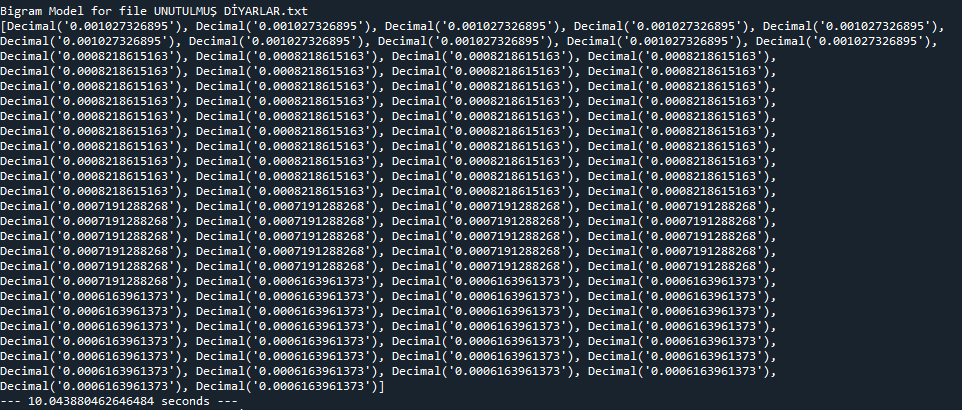
**b. Python**

Results of “BOZKIRDA”

Results of “BİLİM İŞ BAŞINDA”

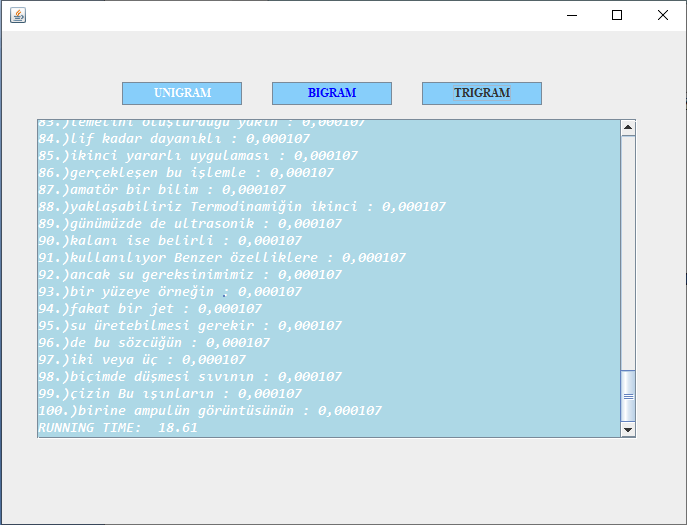
Results of “DENEMELER”

Results of “DEĞİŞİM”

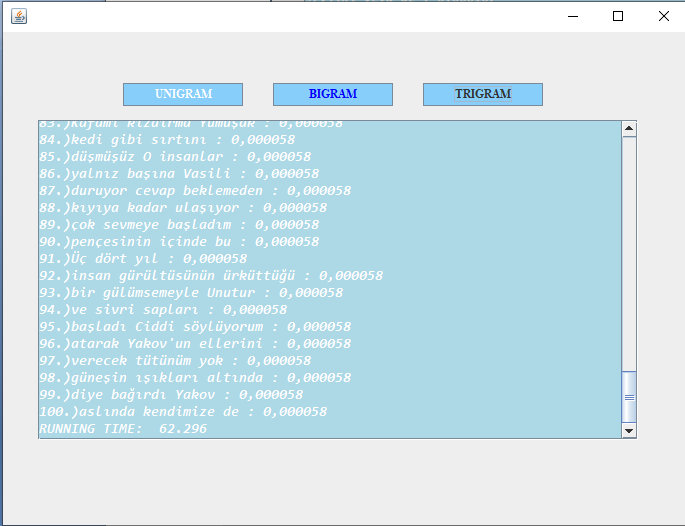
Results of “UNUTULMUŞ DİYARLAR”

**2.3 Trigram Results**

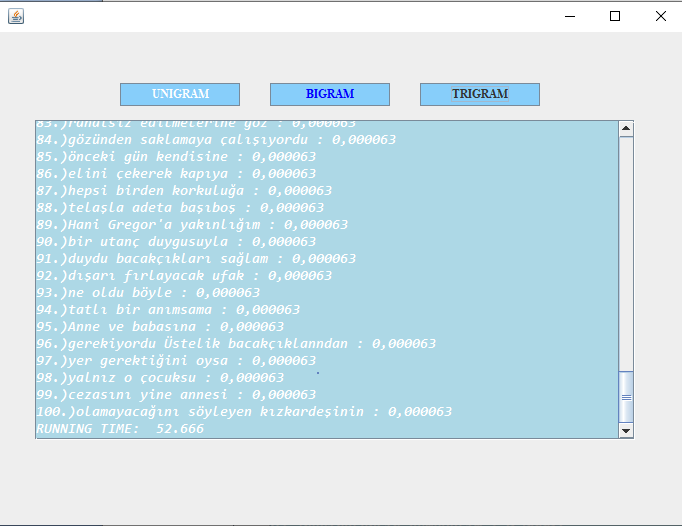
**a. Java**



Results of ”BİLİM İŞ BAŞINDA”



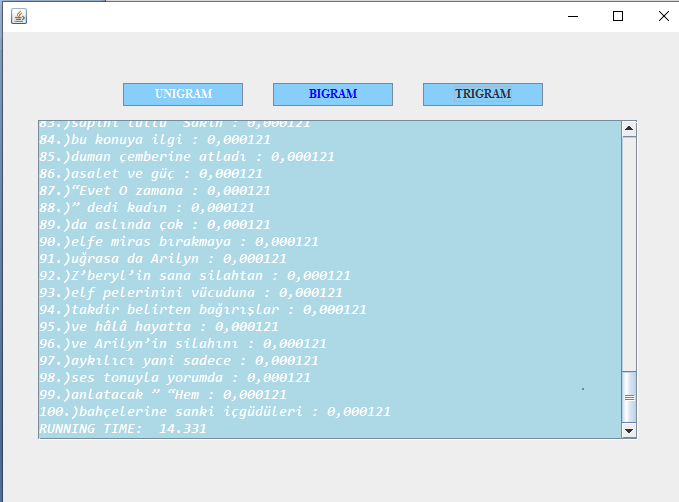
Results of ”BOZKIRDA”



Results of ”DEĞİŞİM”

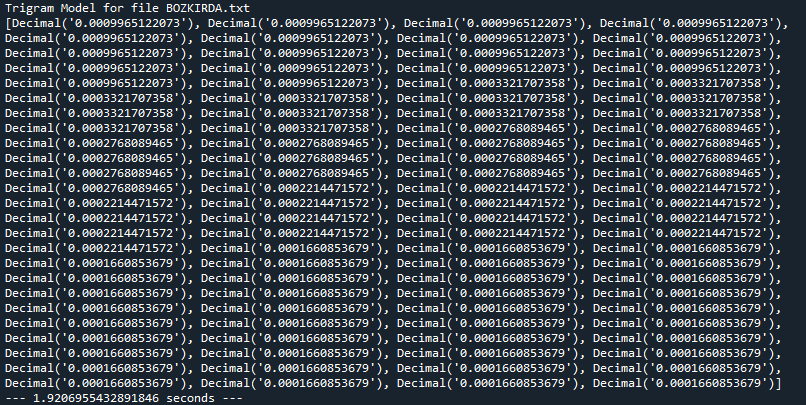


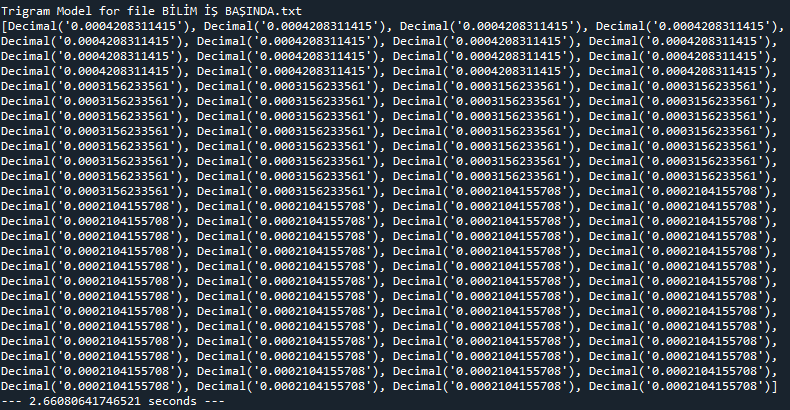
Results of ”DENEMELER”

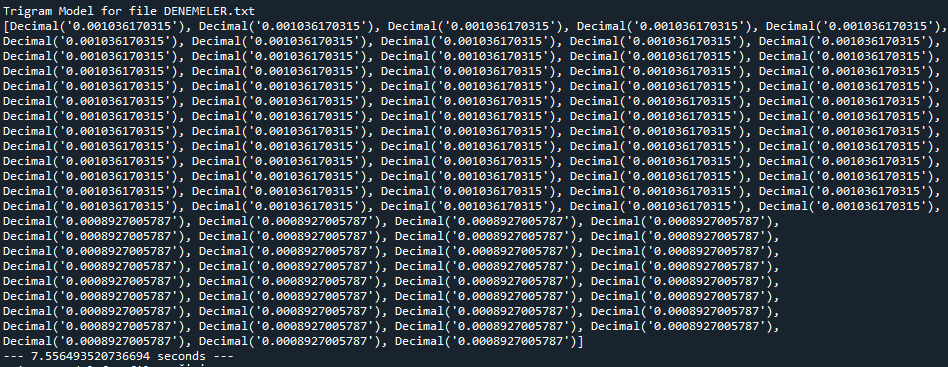


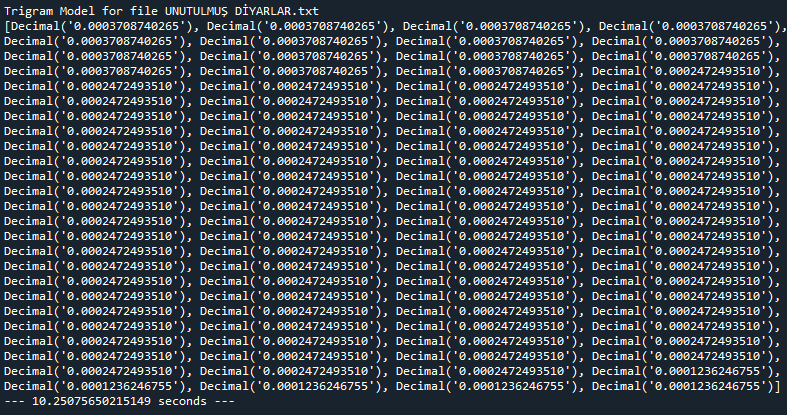
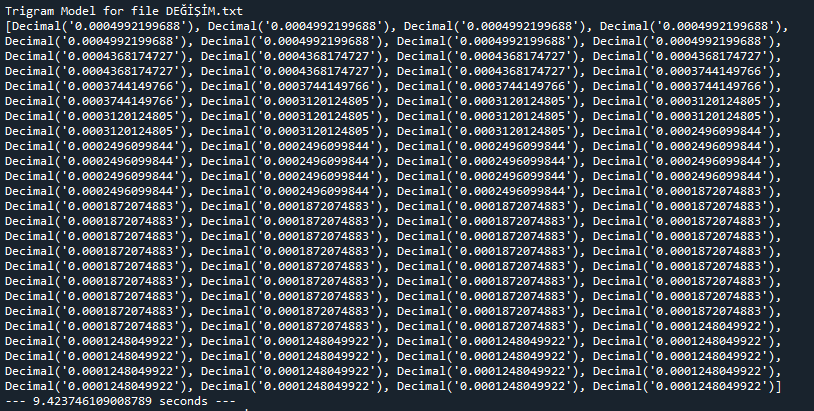
Results of ”UNUTULMUŞ DİYARLAR”

**b. Python**

Results of ”BOZKIRDA”

Results of ”BİLİM İŞ BAŞINDA”

Results of ”DENEMELER”

Results of ”UNUTULMUŞ DİYARLAR”

# CHAPTER 3

# BRIEFLY

# In this project Java program written in Eclipse and Python project has been written inSpyder. In unigram results; we can obviously see that java algorithm is much more faster then python. But in bigram and trigram algorithms Python algorithm is faster then Java even more then unigram results. Probablity results are pretty similar with each program.

# REFERENCES

<https://github.com/hackjutsu/n-gram-demo/blob/master/src/main/java/Ngram.java>

<https://github.com/DanielJohnBenton/Ngrams.java>

https://www2.cs.duke.edu/courses/cps006/fall09/lects/091020/code/NGram.html

<https://github.com/nltk/nltk/wiki/Frequently-Asked-Questions-(Stackoverflow-Edition)>

<https://www.pythonprogramming.in/generate-the-n-grams-for-the-given-sentence-using-nltk-or-textblob.html>

<https://www.tutorialspoint.com/python_text_processing/python_bigrams.htm>

<https://www.geeksforgeeks.org/tf-idf-for-bigrams-trigrams/>