**Cmpe300: Analysis of Algorithms Fall’22 Project2**

**Student Names:**

* **Abdullah Susuz**
* **Umut Demir**

**Submitted Person: Umut Demir**

**Title of the Project: MPI Programming Project**

**Submission Date: 20/12/2022**

**Introduction**

In this project, we are expected to calculate the data for a bigram and unigram language model. For calculation, we use MPI framework and mpi4py python library for our calculations in a parallel way.

MPI:“MPI is a standardized and portable message-passing standard designed to function on parallel computing architectures.”, (from [wikipedia](https://en.wikipedia.org/wiki/Message_Passing_Interface))

**Program Interface**

In order to run this program, one has to have a computer that runs python3 including mpi4py library and mpi framework. This program can be run with mpiexec command. Also there are some parameter like the level ol parallelism, name of the input files and name of the merge method. If there is any error, the program ends itself. In addition, command+c or ctrl+c can be used to terminate the program in terminal.

**Program Execution**

As stated above, this program can be run with command line interface. The execution is the program is as follows:

mpiexec -n <num> python3 main.py --input\_file <input-file> --merge\_method <method> --test\_file <test-file>

For example:

Text

Description automatically generated

<num>: It is the number of ranks minus one. That is, <num> states the number of the parallel processes.

<input-file>: This file includes sentences. It is used to calculate the number of unigrams and bigrams.

<method>: There are two different methods for merging the operations. This part states the name of the method, which is etiher WORKERS or MASTER.

<test-file>: This file includes some bigrams. These bigrams are used for probability calculations.

**Input and Output**

As I stated above, one has to have a number for stating the number of parallel processes, a method name that is etiher WORKERS or MASTER, an input file for calculating bigrams and unigrams, finally a test file for probability calculations.

An example for executions is as follows:

Text

Description automatically generated

It means that I have 4 parallel “worker”s for calculating unigrams and bigrams, sample\_text2.txt file for data file for unigrams and bigrams, use MASTER method for merging the results, and test.txt for calculating probabilities of bigrams from this file.

The result of this execution is as follows:

Text

Description automatically generated

First, sentences from sample\_text2.txt is distirbuted to 4 workers. Then number of unigrams and bigrams are printed to the terminal. Other outputs is probability of bigrams from test.txt file.