Lab 5

Team Members:

Name: Divyank Sharma
Person No:50207091

Name: Vijit Singhal
Person No:50207212

Environment Setup:

We are using PySpark with Jupyter notebook for this Lab.

The input files are placed in a folder: /home/hadoop/input_folder.

We will be using the following environment variables which need to be set in :

1.bashrc
2.spark-env.sh

alias python=/usr/bin/python3
export PYSPARK_PYTHON=/usr/bin/python3
export PYSPARK_DRIVER_PYTHON=ipython3
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"

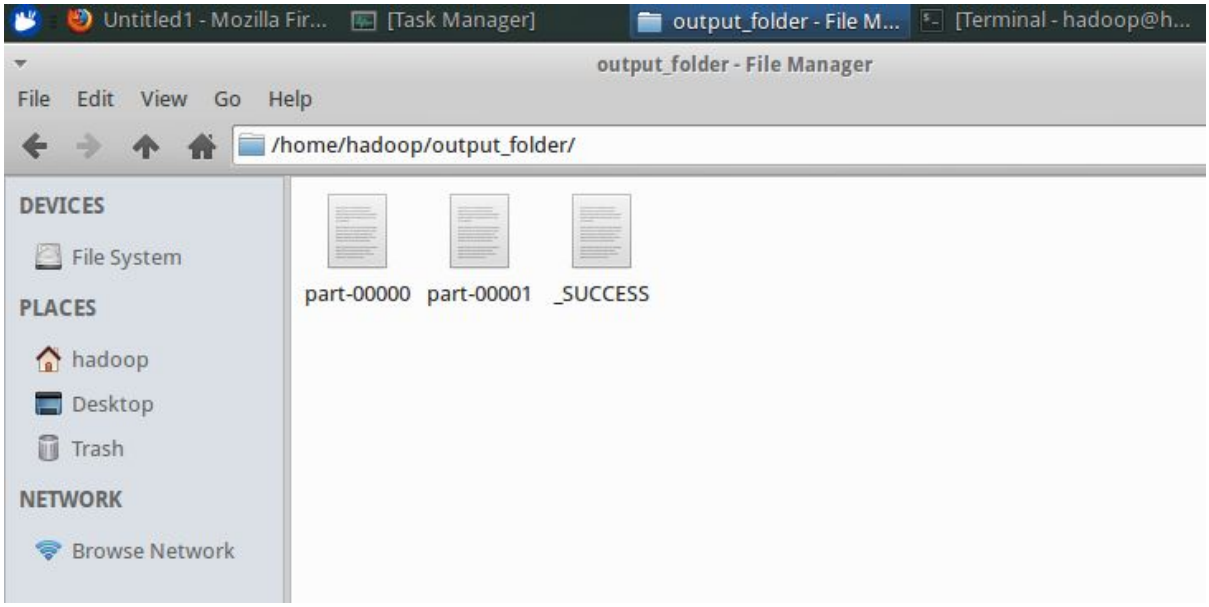
PySpark needs to be started using the command:

```
hadoop@hadoop-VirtualBox:~$ spark/bin/pyspark
[TerminalIPythonApp] WARNING | Subcommand `ipython notebook` is deprecated and w
ill be removed in future versions.
[TerminalIPythonApp] WARNING | You likely want to use `jupyter notebook` in the
future
[I 23:27:48.941 NotebookApp] Serving notebooks from local directory: /home/hadoo
p
[I 23:27:48.941 NotebookApp] 0 active kernels
[I 23:27:48.941 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/?token=433c037c4a5ba7fa7e38992c081edff1e828389ef39cf874
[I 23:27:48.941 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 23:27:48.944 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=433c037c4a5ba7fa7e38992c081edff1e828389ef39
cf874
[I 23:27:51.275 NotebookApp] Accepting one-time-token-authenticated connection f
rom 127.0.0.1
```

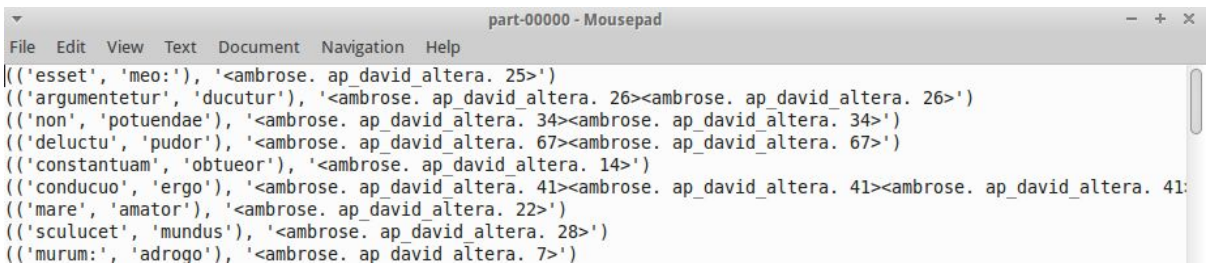
After executing the above command, Jupyter notebook will open up .SparkContext will be available in the Jupyter notebook as an implicit object same as sc object in spark-shell for Scala.

```
In [6]: sc

Out[6]: <pyspark.context.SparkContext at 0xb3cf508c>
```

Output Folder:

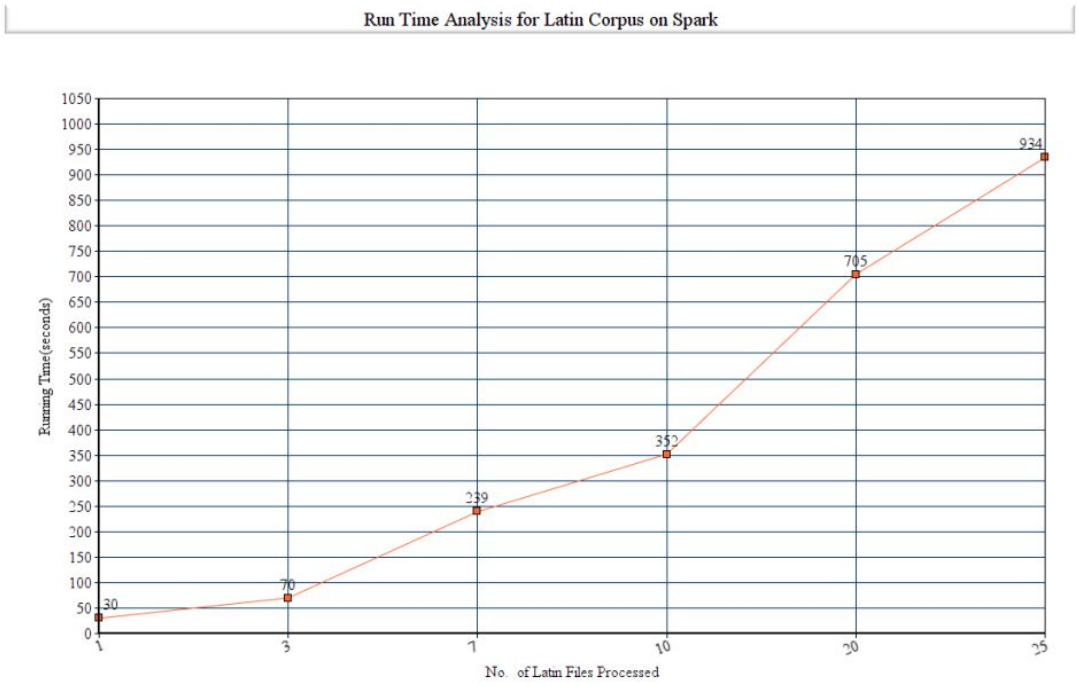


How to interpret the results:



The bigrams are being displayed along with their position in the latin document. The word in the key has been replaced by its lemma if that is present .

Analysis:



Based on the multiple runs on the Latin corpus, we have found the above pattern in the runtime .

The graph clearly shows that as the runtime is directly proportional to number of files processed .