Sami Emre Erdogan
Student ID: 2019280513

Big Data Final Project

How to run the code for **wiki-Vote** and **com-lj.ungraph** : (1st command line is for wiki-Vote, 2nd command line is for com-lj.ungraph respectively)

I had some server pyspark error. I fixed it by downgrading the pyspark version to 2.4.0 and it worked.

```
pip install pyspark==2.4.0
```

Preprocessing data:

```
python pre_process.py --path /nfs/share/data/wiki-Vote.txt

python pre_process.py --path /nfs/share/data/com-lj.ungraph.txt
```

Copying the preprocessed file to hdfs:

```
hdfs dfs -copyFromLocal wiki-Vote_pre_mapped.txt   .

hdfs dfs -copyFromLocal com-lj_pre_mapped.txt   .
```

Pagerank calculation:

```
python pagerank.py --path hdfs://bd2:9000/user/2019280513/wiki-
Vote_pre_mapped.txt --num_iter 20 --k_top 5 --dump_factor 0.8

python pagerank.py --path hdfs://bd2:9000/user/2019280513/com-
lj_pre_mapped.txt --num_iter 20 --k_top 5 --dump_factor 0.8
```

Postprocessing Pagerank:

```
python post_process.py --input_path wiki-Vote_pre_mapped_pagerank_top_5.txt -
-mapping wiki-Vote_map.txt

python post_process.py --input_path com-lj_pre_mapped_pagerank_top_5.txt --
mapping com-lj_map.txt
```

Trustrank computation. Nodes and white list used for 100 nodes:

```
python trustrank.py --path hdfs://bd2:9000/user/2019280513/wiki-
Vote_pre_mapped.txt --num_iter 20 --dump_factor 0.8

python trustrank.py --path hdfs://bd2:9000/user/2019280513/com-
lj_pre_mapped.txt --num_iter 20 --dump_factor 0.8
```

Postprocessing data:

```
python post_process.py --input_path wiki-Vote_pre_mapped_trustrank_top-
all.txt --mapping wiki-Vote_map.txt

python post_process.py --input_path com-lj_pre_mapped_trustrank_top-all.txt -
-mapping com-lj_map.txt
```

Postprocessing white list:

```
python wl_postprocess.py --input_path wiki-Vote_trustrank_white_list.txt --
mapping wiki-Vote_map.txt

python wl_postprocess.py --input_path com-lj_trustrank_white_list.txt --
mapping com-lj_map.txt
```

Sami Emre Erdogan
Student ID: 2019280513

Implementation

Since com-jl data is too large I haven't included all of the files in the results folder however I have included the terminal results as proof but I have included everything from wiki-Vote data.

The implementation consisted of 5 files: Pagerank, TrustRank, Prepocess, postprocess and whitelist post process. Of course before using Pyspark we had to set parameters for such as iterations,dump factor k-top in our code.

Preprocessing: Pre-processed to convert the raw identifier of vertices to the integer ID.

Postprocesing: It re-mappes the operation and converts node ID after preprocess and computes the results to raw identifier.

Pagerank: initializes edges of source_id and list of out-nodes gets and updates ranks contributions to the rank of other nodes.

TrustRank: initializes edges of source_id and list of out-nodes gets and updates ranks contributions to the rank of other nodes and also for whitelist for later use. Updates node ranks based on their in-nodes contributions. Sorts nodes by rank and takes the k best ranks.

Speed and Thoughts

For the **wiki-Vote** it took around 23 seconds each for trustrank and Pagerank. Whereas in **com-lj.ungraph** it took around 58.33 minutes each for PageRank and TrustRank because of its ungraphed big data. I used python for this task maybe if I used C++ It would have been faster but I am more comfortable with Python.

Screenshot terminal results of **com-lj.ungraph:**
RageRank:

```
2019280513@bd1:~$ python pagerank.py --path hdfs://bd2:9000/user/2019280513/com-lj_pre_mapped.txt --num_iter 20 --k_top 5 --dump_f
actor 0.8
2020-06-19 05:03:30,542 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c
lasses where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Time: 3529.93 seconds
Top-5 nodes: [('2790787', 4.8059011236131154e-05), ('3573262', 2.7780714622310063e-05), ('3301462', 2.5771399058494316e-05), ('198
7747', 2.5476388090837087e-05), ('2181580', 2.4796263086789944e-05)]
```

Sami Emre Erdogan
Student ID: 2019280513

TrustRank:

```
2019280513@bd1:~$ python trustrank.py --path hdfs://bd2:9000/user/2019280513/com-lj_pre_mapped.txt --num_iter 20 --dump_factor 0.8
2020-06-19 06:15:16,574 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c
lasses where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Total program time: 3508.77 seconds
Top-all nodes: [('2790787', 4.8060288471146686e-05), ('3573262', 2.7786584986332085e-05), ('3301462', 2.5782744886103877e-05), ('1
987747', 2.5481655938654144e-05), ('2181580', 2.47982664463133e-05)]
```