**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

**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.

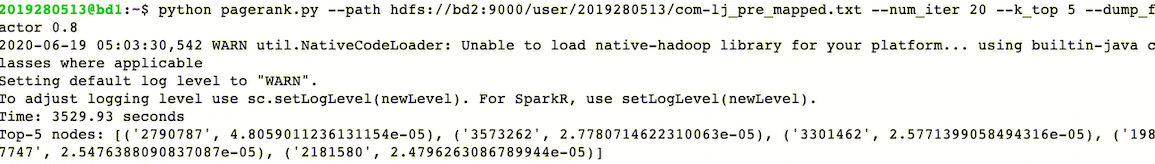
**TrustRank:** initializes edges of source\_id and list of out-nodes gets and updates ranks contributions to the rank of other nodes and also for whitelist for later use. Updates node ranks based on their in-nodes contributions. Sorts nodes by rank and takes the k best ranks.

**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:**



**TrustRank:**

