Applied Intensive Research(B9898)
Final Report

Professor: Andrey Simonov
PhD Advisor: Uliana Evseeva
Savannah Wang(rw2840)

Tips of working with Research Grid

Connecting with Research Grid (Mac):
Research Grid is a computing source at CBS, which works especially well for running very large programs and collaborating and sharing code and data with teammates. The following way I the most common way I use:
Step1: Connecting to the vpn through Cisco Anyconnect with UNI, passcode, and second passcode (which you can get from Duo Mobile.
Step2: Entering following commands in the terminal: ssh my_uni@research.gsb.columbia.edu and entering password used for log in columbia websites.

Accessing data and files:
Data and scripts are saved mainly both on the Grid and in the dropbox.
A recommended way to upload/ download: click Finder -> Go -> connect to Server, entering *'smb://researchFiles.gsb.columbia.edu'*. Uploading/downloading can also be done through terminals.

Working with scripts:
There are two ways to run a script on the Grid: batch and interactive mode. I highly recommend to run scripts under the interactive mode on the grid, especially when you are not sure about the scripts. Interactive mode allows you to see the outlet of each step, while the results only show up after the work is done, which is very inconvenient for debugging. While running the scripts on the Grid, two ways that I used most commonly are:
(accessing interactive mode for python): anapy3 --grid_mem=50g --grid_submit=interactive
(accessing batch mode for python): anapy3 --grid_mem=50g python_script_name.py
(accessing interactive mode for R): R --grid_mem=50g
(accessing batch mode for R):
Rscript --grid_submit=batch --grid_email='my_email@columbia.edu' name_of_my_r_scripts.r
[here, I add my email here, so after the work is done, I can get a notification email.]

My Takeaways:

1) Always clarify the pwd in the very beginning of the scripts so others know the paths for input and output files, that can save lots of time for people not familiar with the files.
2) While working with large amounts of data, try to do the task in the most efficient way, since that makes a huge difference in running time when the dataset is large. Before running the scripts, estimate the memory usage and try the scripts with a smaller subset of it first to make sure it works fine.
3) If stucking on any technical issues for a long time, email the research support to reach for help and ccing professor and uliana. They are very helpful and respond to email very quickly. Contacting to Research Support Team: ResearchSupport@gsb.columbia.edu

Weekly Progress

Week 1:
Met with Professor Simonov.
Had a general idea about the research approach, and read through notes from Youngkeun about script debugging.
Received Dropbox.

Week 2:
Met with Professor Simonov and Uliana. Received access to the Grid, and ran python scripts on it.
Transformed the file propers into a human readable format.

Week 3:
Met with Professor Simonov and Uliana: Ran and debugged for python scripts 6,7 and got familiar with the res flies.
Collaborated with Uliana: Investigated what info we have in res/propers; Coded for try_load_news.py to reorga-nized Prepared news and save it as a CSV file named prepared_news_df.csv in res for later use.

Week 4:
Met with Professor Simonov and Uliana: Created a sparse matrix with all proper nouns and the corresponding vocabularies. (in process), and potentially split it by years to do the robustness check.
(I constructed build_s-parse_ matrix.py, works fine on a smaller dataset, but took a lot of memory while running it (in-formed by the research tech support team, I am trying to work in another way).

Week 5:
Met with Professor Simonov and Uliana: Constructed another way to build the sparse matrix, still took lots of memory, (I.e. through a rough calculation, need (number of column:total number of propers)*(number of rows: total number of articles)*datatype(assumed use 32 bits) number of column: (at most) total number of propers = sure proper+one_proper+two proper+three words = 790143+1426139 + 1285723 + 8945076 = 12447081 => need (at most) 12447081 * 21697904 * 32 / (1024**4) ~ 200 Tb to save the sparse matrix (at least, roughly assumed only 30000 propers are used as columns), sLll need ~ 19Tb If I calculated correctly, it will take lots of memory to store it anyway.

Week 6:
Met with Professor Simonov: Constructed proper & counts dataset. Constructed build the sparse matrix.py, and solved the memory issues
Apply analysis to set the threshold for frequency of words.

Week 7:
Met with Professor Simonov and Uliana:
Constructed outlet & counts dataset.
Constructed the construct_outlet.py
Apply analysis of the outlet_ count matrix into main_analysis.r locally on a subset of data, since grid is done this week.

Week 8:
Run the main_analysis.r on the full dataset. finish detection of sensitivity of proper nouns & outlets and sensiLve proper nouns & types of outlets detection of sensitive proper nouns & classification of topic .

Week 9:
Generated outlet_counts matrix and df_types. Running compute_rank_metric func on grid with full dataset. (It's been run for almost an hour, hopefully it can be down below our meeting) Drafted for the final report.

Week 10:
Prepared for the presentation.
Presentation.

Week 11:
Organized all previous scripts.
Write a README file and a final report.