11-641 Home Work 3

Name: NIKHIL MALIK
AndrewID:nmalik1

Statement of Assurance
All the material submitted for this assignment is my own work.

Experiments

(a) Describe the custom weighing scheme that you have implemented. Explain your motivation for creating this weighting scheme.

The custom weighting scheme: 0.9*log(Page Rank Value) + 0.1*(Search Relevance Score)
The distribution of the the raw values of the Page Rank and Search Relevance score suggest that the page rank value are exponentially decreasing. Therefore a more equitable weighting between the two requires a log of the Page Rank Value.

See Table2, highlighted values. These indicate improved mean average precision at 0,10 and 20% recall values as a result. However the precision remains the same or worse at higher recall levels when compared to the original linear weighting scheme.

(b) Report the performance of the 9 approaches as described above.

Method	map	Wall-clock time (secs)	Dampening Factor	Weighting Method
NS_GPR	0.0472	0.16	0.8	1000*rank value
NS_QTSPR	0.045	2.2	0.8	1000*rank value
NS_PTSPR	0.0463	2.2	0.8	1000*rank value
WS_GPR	0.1123	0.16	0.8	0.9*rank value + 0.1*search relevance scc
WS_QTSPR	0.1069	2.2	0.8	0.9*rank value + 0.1*search relevance scc
WS_PTSPR	0.1107	2.2	0.8	0.9*rank value + 0.1*search relevance scc
CM_GPR	0.143	0.16	0.8	0.9*log(rank value) + 0.1*search relevance
CM_QTSPR	0.1122	2.2	0.8	0.9*log(rank value) + 0.1*search relevance
CM_PTSPR	0.1127	2.2	0.8	0.9*log(rank value) + 0.1*search relevance

Table1: For each of the 9 methods – map score, wall clock time, dampening factor

	NS_GPR	WS_GPR	CM_GPR
0	0.1811	0.2851	0.6749
0.1	0.0824	0.2168	0.3858
0.2	0.0781	0.2084	0.237
0.3	0.0733	0.1945	0.1712
0.4	0.0698	0.1704	0.1322
0.5	0.0644	0.1469	0.1092
0.6	0.0523	0.1118	0.08043
0.7	0.0294	0.0618	0.0476
0.8	0.0116	0.0379	0.012
0.9	0.0072	0.0268	0.0076
1	0.0041	0.0118	0.0041

Table2: For each of the 3 GPR methods: comparison of 11 point map scores averaged over all the queries

NS_QTSPR	NS_PTSPR	WS_QTSPR	WS_PTSPR	CM_QTSPR	CM_PTSPR
ircl_prn.0.00 all 0.1657	ircl_prn.0.00 all 0.1768	ircl_prn.0.00 all 0.2689	ircl_prn.0.00 all 0.2802	ircl_prn.0.00 all 0.5741	ircl_prn.0.00 all 0.5915
ircl_prn.0.10 all 0.0786	ircl_prn.0.10 all 0.0804	ircl_prn.0.10 all 0.2051	ircl_prn.0.10 all 0.2125	ircl_prn.0.10 all 0.2713	ircl_prn.0.10 all 0.2663
ircl_prn.0.20 all 0.0729	ircl_prn.0.20 all 0.0758	ircl_prn.0.20 all 0.1983	ircl_prn.0.20 all 0.2049	ircl_prn.0.20 all 0.1681	ircl_prn.0.20 all 0.1803
ircl_prn.0.30 all 0.0702	ircl_prn.0.30 all 0.0728	ircl_prn.0.30 all 0.1857	ircl_prn.0.30 all 0.1888	ircl_prn.0.30 all 0.1323	ircl_prn.0.30 all 0.1357
ircl_prn.0.40 all 0.0663	ircl_prn.0.40 all 0.0689	ircl_prn.0.40 all 0.1620	ircl_prn.0.40 all 0.1679	ircl_prn.0.40 all 0.1108	ircl_prn.0.40 all 0.1202
ircl_prn.0.50 all 0.0618	ircl_prn.0.50 all 0.0642	ircl_prn.0.50 all 0.1407	ircl_prn.0.50 all 0.1459	ircl_prn.0.50 all 0.0961	ircl_prn.0.50 all 0.1104
ircl_prn.0.60 all 0.0497	ircl_prn.0.60 all 0.0515	ircl_prn.0.60 all 0.1080	ircl_prn.0.60 all 0.1101	ircl_prn.0.60 all 0.0666	ircl_prn.0.60 all 0.0581
ircl_prn.0.70 all 0.0276	ircl_prn.0.70 all 0.0290	ircl_prn.0.70 all 0.0593	ircl_prn.0.70 all 0.0614	ircl_prn.0.70 all 0.0323	ircl_prn.0.70 all 0.0329
ircl_prn.0.80 all 0.0113	ircl_prn.0.80 all 0.0115	ircl_prn.0.80 all 0.0379	ircl_prn.0.80 all 0.0372	ircl_prn.0.80 all 0.0120	ircl_prn.0.80 all 0.0129
ircl_prn.0.90 all 0.0073	ircl_prn.0.90 all 0.0074	ircl_prn.0.90 all 0.0261	ircl_prn.0.90 all 0.0269	ircl_prn.0.90 all 0.0080	ircl_prn.0.90 all 0.0077
ircl_prn.1.00 all 0.0040	ircl_prn.1.00 all 0.0041	ircl_prn.1.00 all 0.0101	ircl_prn.1.00 all 0.0105	ircl_prn.1.00 all 0.0041	ircl_prn.1.00 all 0.0042

Table3: For each of the remaining 6 methods: comparison of 11 point map scores averaged over all the queries

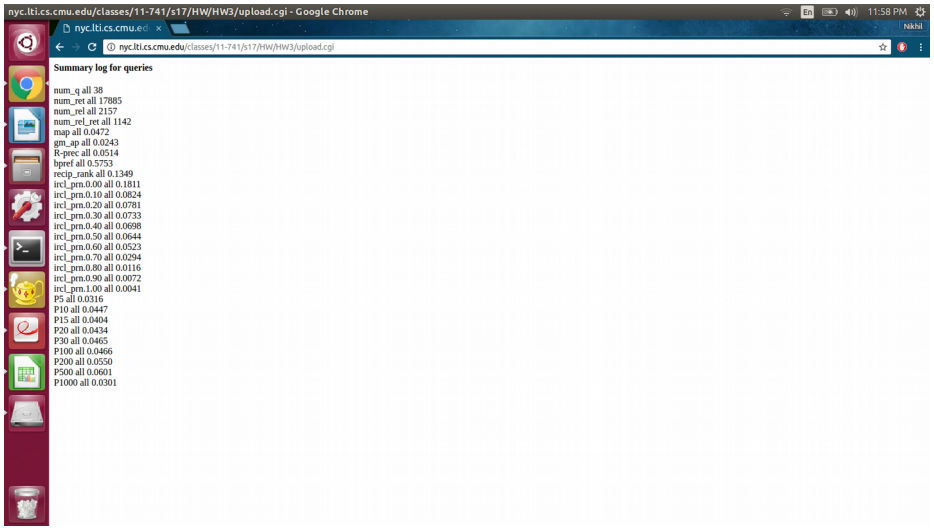

Figure1: A sample screenshot of the trec-eval summary output.

(c) Compare these 9 approaches based on the various metrics described above.

Wall Clock time: The majority of the time is spent in Page Rank calculations, while the combination of search relevance and page rank step is extremely small.
The GPR method requires a single Page Rank calculation and takes ~0.16 seconds.
However QTSPR and PTSPR both reuse the same topic sensitive page rank module needs to run the Page rank algorithm 12 times i.e. once for each topic. Resulting in 0.16*12 ~ 2.2 seconds.

Additionally the time to generate the evaluation files has not been reported. This amounts to an additional ~15-20 seconds.

```
In [14]: (r,r_topics,r_queries,r_users) = LA.Run_Ranking()
Time for GPR:0.142735004425

Writing GPR-10.txt

Time for TSPR:1.93227410316

Writing QTSPR-U2Q1-10.txt

Time for QTSPR:0.204296112061

Writing PTSPR-U2Q1-10.txt

Time for PTSPR:0.219743013382

In [15]: df = LA.Generate_Evaluation_Files(r,r_queries,r_users,PrTgU,PrTgQ)
Time for NS_GPR:0.00685596466064
Time for WS_GPR:0.0107820034027
Time for CM_GPR:0.00883507728577
Time for write 3 GPR Evaluation files:5.11466503143
Time for NS_QTSPR:5.12014293671
Time for WS_QTSPR:0.00942087173462
Time for CM_QTSPR:0.00764608383179
Time for write 3 QTSPR Evaluation files:4.92961597443
Time for NS_PTSPR:4.93502688408
Time for WS_PTSPR:0.00933003425598
Time for CM_PTSPR:0.00748491287231
Time for write 3 PTSPR Evaluation files:4.89636397362
```

Figure2: Show a snapshot of the output with the wall clock times printed out.

Custom Method: Shows significant precision improvement at low recall levels, given the log scale usage.

Dampening Factor: Has been kept at 0.8 across all 9 methods

Mean Average Precision: The map for original and custom weighting scheme are significantly more than the no weighting scheme. Showing the significance of search relevance scores.

(d) Analyze these various algorithms, parameters, and discuss your general observations about using PageRank algorithms

The Page Rank Algorithm stopping criteria is based on convergence of page rank values across iterations. However a simple euclidean distance (or an absolute L1 norm) may lead to early convergence. Measuring the difference between the page rank values across iterations as a ratio of the L2 (or L1) norm of the page rank vector itself is likely to be a better criteria. The attached code was sense-checked for this new criteria, however this calculation has temporarily been commented out.

The initial values of page ranks r = 1/N for all documents does not necessarily change the final convergence of the ranks. An initial value of zeros or 1's also results in the exact same final ranks.

The higher the dampening factor (1-alpha) the higher the weight given to the document linkages.

(e) Discuss some general remarks about
• What could be some novel ways for search engines to estimate whether a query can benefit from personalization?

Query could be classified into “specific” or “non-specific” query based on the sentence construction of the query. A “non-specific” query could benefit from greater personalization to the general topics of interests for the user. However the same personalization might be a nuisance for a “specific” query. This classification can be achieved using a supervised historical query data, tagged based on whether the user subsequently chose from the recommended search results or browsed to find a very specific result of interest.
Examples: “Buy HP Laptop model abc.xyz from Amazon” [Specific]
“What laptops are hot?” [Non-Specific]

Another mechanism could utilize information on: time-of-day, platform(mobile or pc) etc for the user. To differentiate for each user what combination of these features indicate the users interest in personalized results.

• What could be some novel ways of identifying the user’s interests (e.g. the user’s topical interest distribution Pr(t|u)) in general?

- Utilizing past queries by the user.
- Subsequent search results clicked and topics of those pages.
- Collaborative Filtering to identify topics of interest based on similar other users.
- Utilizing information on time-of-day, platform.

Details of the software implementation

(a) Describe your design decisions and high-level software architecture;

Key High level design decisions
- All input files are read in a single go at first.
- The transition matrix is stored as a sparse matrix to avoid memory issues.
- Vector and matrix algebra has been used instead of for loops to improve performance.
- A topic sensitive page rank (1 for each of 12 topics) has been calculated and reused for both the QTSPR and PTSPR methods.

Software Architecture
- Read_Data: Read the 4 files for Page Rank calculation
- Read_indri_files: Read the 5th file for search relevance scores
- PageRank: Runs the core page rank algorithm given a Matrix M and vector p
- TS_PageRank: Runs 12 page ranks for 12 topics
- Online_TS_PageRank: Wrapper Function for QTSPR and PTSPR
- Run_Ranking: Return all three ranks GPR, QTSPR, PTSPR
- Read_PageRank_Output: Converts page rank output to a pandas dataframe
- Write_PageRank_file: Write the 10th iteration files
- Write_treceval_file: Write the trec-eval files
- Generate_Evaluation_Files: Wrapper function that generates all the output files one by one.

(b) Describe major data structures and any other data structures you used for speeding up the computation of PageRank;

- scipy lil_matrix: For sparse matrix storage
- pandas dataframe: For tabular data, before writing the evaluation text files.

(c) Describe any programming tools or libraries that you used;

- scipy: For sparse matrix
- pandas dataframe: For tabular data, before writing the evaluation text files.
- Numpy
- sklearn preprocession: For L1 normalisation of the M matrix
- zipfile: For unzipping the search relevance score files
- time: for calculating wall clock time

(d) Describe strengths and weaknesses of your design, and any problems that your system encountered

Strength

- The entire code including the creation of the evaluation files runs in <1 minute.
- Sparse data structures allow for ~50 iterations of Page rank to complete under 1 second.

Weaknesses

- Naming of the function modules and reusability is lacking at some points.
- Hard coded parameters like alpha and weights.