CS 638 STAGE III

Data blocking

Team Members:

Shixuan Fan

Yuting Liu

Zhenyu Zhang

REPORT

1. Explain the development process, from the first blocker all the way to the final blocker (that you submit in the Jupyter file).

There are several rules we implemented during our blocking processes.

At first we tried to block the tuple pairs by using the same ZIP code (if they didn’t have the same ZIP code, they are blocked out). However, when we tried to use the debugger, we found that there are some pairs that are the same restaurant (same name, same phone number, same address), but with different ZIP code. There are several reasons leading to this situation, as it could be dirty data, chain restaurants, or their ZIP code might change due to some historical reason.

For the second version, we tried to block pairs with similar ZIP code (tuple pairs with greater than 1 edit distance in ZIP code are blocked out). But the same issue still exists (for example, 10003 vs 10017).

Then we added another rule so that only pairs with greater than 1 edit distance in ZIP code and telephone number will be dropped. This solved the ZIP code problem. However, since some restaurants in one data source are using the same phone number, while another source might use separate numbers, we still need other rules to make sure that we could include them.

Finally, we added two new rules (if the tuple pairs share the same name or the same address, they should not be dropped). After this, we found that in the debugger, no potential matches were dropped.

Our final blocker is to keep tuple pairs if (Edit distance of ZIP code <= 1) OR (Edit distance of Phone # <= 1) OR (Same name) OR (Same address)

2. If you use Magellan, then did you use the debugger? If so, where in the process? And what did you find? Was it useful, in what way?

We used debugger to check if our current rule is good enough to include all potential pairs.

We found that this is really useful because it would let us know that our current blocker might rule out some correct matches, so that we could add more rules to try to avoid that. It helps us to make more correct decision on the rules for matches on our blocker.

3. How much time did it take for you to do the whole blocking process.

It takes roughly 5 min to finish the whole blocking process

4. Report the size of table A, the size of table B, the total number of tuple pairs in the Catersian product of A and B, and the total number of tuple pairs in the table C.

Table	Table A	Table B	Total number	Table C
#tuple pairs	3075	3478	10694850	1048576

5. Did you have to do any cleaning or additional information extraction on tables A and B?

When we use Jupyter, it caused us a lot of trouble when the crawled data from the website takes UTF-8 encoding. There are many restaurant names embedded with several special characters. We had to clean our yelp.csv data to avoid some UTF-8 decoding issue.

6. Did you run into any issues using Magellan (such as scalability?). Provide feedback on Magellan. Is there anything you want to see in Magellan (and is not there)? If you do not use Magellan, you can skip this question.

There are several problems we met during our using Magellan.

(1) There are several problems when we setting the system environment firstly. Our group members have PCs with three different operating systems: Linux, IOS, and Windows. We had to say that the setting processes are really hard for all of us.

(2) The python version really matters. We switched to python 2 because python 3 might cause some issues and we could not find the solution from the installing documentation to those problems.

(3) We would like to see a clearer documentation about RuleBasedBlocker. There are really many problems coming out continuously when we run the code and set the system environment. There are no enough Q&A involved to solve those problems we met in practice. All what we could is to try very hard to find any possible solutions and we really hope that the software developer could provide more hints about the solution to those problems.

(4) Debugger sometimes will miss some potential matches, but when we restarted the kernel, it worked out just fine.