CS 784 Stage 3: Blocking Explanation

Majid Aksari, Dillon Skeehan

November 1, 2015

1 Datasets

We are matching Barnes and Noble and Amazon Database Management book categories. Even thought we have access to ISBN of the books, we will not be using it for matching because it makes matching trivial. We anticipate that title of the books will play a crucial role in matching. We have other attributes as well such as publication date, number of pages and price. Here are details of the data sets and the candidate set that we obtained:

- Barnes and Noble (table A) size: 3000
- Amazon (table B) size: 3000
- Candidate set (table C) size: 257183

2 Blocking Method

With blocking our goal is to remove obvious non-matches and try not to remove matches. Based on our readings from the book, we anticipated that the attribute equivalence blocker is very strict for our data set and will lead to missing matches. Intuitively, two book titles might refer to the same book but they can have slight differences. Therefore, we decided to use the overlap blocker based on titles of the books.

Initially, we were not sure if we should use word-level or qgram and how many tokens to match. Therefore, we iteratively tried different combinations. We compared different options with regard to the size of the candidate set it produced as well as output of the debugger. For example, with word level blocking, when we asserted that 2 words should match it did not match 1 word books. We detected this issue using the debugger. When we switched to asserting 1 word should be in common, we got too many pairs in our candidate set. We found that using the qgram will also give us unnecessarily large candidate sets as well.

At this point we went back to the data and inspected it. We wrote a block of code to print out one-word. After inspection we found that one-word books

match if they exactly match. Therefore, we decided to use an attribute equivalence blocker to capture these pairs.

Therefore, at the end we combine the results of two blockers to get our candidate set:

1. Overlap blocker on title, remove stop words, word level blocking, at least 2 words in common
2. Attribute equivalence blocker on title

3 Feedback:

- We first tried installing everything on Windows, it took us many hours and it failed. Then we tried on linux it went more smoothly, about an hour.
- At first, one of our tables would not load, so we had to fix the table such that it would load.
- We found that the system could not handle our table 6000 x 5000 and the browser crashed. We kept reducing the size of the tables, and eventually it could handle 3000 x 3000. Even with this size, we when we ran a qgram blocker it would be fine but after that it would crash for the debugger. We think it is a memory constraint, but we could not find out how to increase the memory.
- Also, Debugger forced us to add ("ID","ID") along with other attribute pairs that we wanted it to consider, otherwise, it would produce an error.
- Even though we asked the debugger to use isbn as well as title, the debugger table only included title.
- It took about 10 minutes for the debugger to run.
- For the debugger we used title because we thought that these are most crucial for matching, and when let the system use all attributes it took a very long time to run. It appeared that it will not finish.
- Because the code and the tables are displayed in the same page it is hard to navigate through the page.