CS 839 Project Stage 4 Report

Zeping Ren, Stephen Sheen, Shiqi Yang

Data Merging

We first applied the matcher developed in Stage 3 to tables A and B and obtained the matching result, stored in a csv file. We also wrote another crawler that extracts the price information from the websites from which the data were extracted for future analysis. Then, we wrote a Python script that uses the following rules to merge the tables:

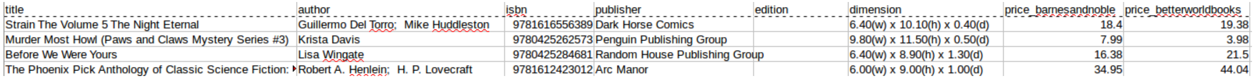
```
A.title = B.title, keep A
A.author = B.author, keep A
A.isbn = B.isbn, keep A
A.publisher = B.publisher, keep A
A.edition = B.edition, keep A
A.dimension = B.dimension, keep A, as B has too many missing entries
keep both the price information from A and B
```

We did run into some issues when merging the tables: there are some duplicate tuples in the matching result. We resolved this by simply eliminating those duplicates. Also, the price information for BetterWorldBooks sometimes shows as "(out of stock)". For those missing values, we simply use the price infomation from Barnes and Noble.

Eventually, the schema of Table E is

```
E(title, author, isbn, publisher, edition, dimension,
  price_barnesandnoble, price_betterworldbooks)
```

There are 835 tuples in Table E. We give four sample tuples:



title	author	isbn	publisher	edition	dimension	price_barnesandnoble	price_betterworldbooks
Strain The Volume 5 The Night Eternal	Guillermo Del Torro; Mike Huddleston	9781616556389	Dark Horse Comics		6.40(w) x 10.10(h) x 0.40(d)	18.4	19.38
Murder Most Howl (Paws and Claws Mystery Series #3)	Krista Davis	9780425262573	Penguin Publishing Group		9.80(w) x 11.50(h) x 0.50(d)	7.99	3.98
Before We Were Yours	Lisa Wingate	9780425284681	Random House Publishing Group		6.40(w) x 8.90(h) x 1.30(d)	16.38	21.5
The Phoenix Pick Anthology of Classic Science Fiction:	Robert A. Henlein; H. P. Lovecraft	9781612423012	Arc Manor		6.00(w) x 9.00(h) x 1.00(d)	34.95	44.04

Appendix shows the python script used to merge the tables.

Data Analysis

Tasks Description

We performed three data analysis tasks.

1. Anomaly detection: find if there exists any extreme price differences, and why.
2. OLAP-style exploration: which store sells cheaper books.
3. Clustering: try to cluster according to title and author attributes.

For task 1, we simply did the following SQL query:

```
SELECT MAX(ABS(price_barnesandnoble - price_betterworldbooks)) FROM E;
```

For task 2, execute the following SQL query:

```
SELECT AVG(price_barnesandnoble) FROM E;
SELECT AVG(price_betterworldbooks) FROM E;
```

For task 3, since we crawled books from five different categories (history, computer, mystery, fiction and science fiction), but the category information is unknown from the table, we would like to try clustering using only title and author information to see if they are capable of recognizing the categories of the books.

Results

Anomaly Detection: Extreme price differences

We did find some books with price difference of over 50, or even 100 dollars. The reason, after investigation, could be one of the following:

- Barnes and Noble sells primarily new books, while a lot of books on BetterWorldBooks are used.
- Some books only have one version (hard or paper copy) on one website.
- Different third-party sellers also give different prices.
- Some books do not have new version. Only marketplace price is given.

OLAP-style exploration: Which store is cheaper?

From the result above, we expect that BetterWorldBooks would be cheaper. And the result is as expected. The average price of books at Barnes and Noble is 29.43, and 27.81 for BetterWorldBooks.

Clustering to find categories

We defined two feature sets: TF-IDF of titles, and term frequency of authors. Then we performed the following clustering methods: k-means, agglomerative hierarchical and Birch. All methods generated similar results.

Here is the result of k-means clustering:

```
distortion: 0.9538190934701668
number of cluster #0: 37
number of cluster #1: 697
number of cluster #2: 39
number of cluster #3: 35
number of cluster #4: 27
```

And the clustering result is somewhat unsatisfactory. We eye-balled the clusters and found that books from every category are kind of scattered across the clusters.

Conclusion, Lessons Learned and Discussion

We managed to draw some conclusion about that price range from both stores, as shown in the last section. For cheaper used books, BetterWorldBooks is definitely a better choice. But, for more complete selection of books, Barnes and Noble is a better choice as some books are not found in BetterWorldBooks.

However, as can be seen from last section, the result of the third task is not satisfactory. The possible reason is that: 1) clustering is not suited for this task, or 2) the feature encoding (TF-IDF) that we are using is not well-defined in this specific task, or, 3) most probably, our crawler accidentally crawled a lot of books from categories other than what is specified in our plan.

Future Work

Given more time, we could explore using classification models to recognize book categories. But then we need to further obtain some label of category information from the web first. Also, we may explore some methods to predict the price range using the information given in the table. What's more, we could find some correlation between the authors and their preferred publishers.

Appendix: Python code that merges tables

The following code takes the output from the py_entitymatching package and write to the standard output the selected columns. We directed the output to a csv file for simplicity.

```
import csv

# write to a redirected output

with open(’Matched.csv’, ’r’) as csvfile:
    line = csv.reader(csvfile, delimiter=’,’)
    for row in line:
        if row[-1] == ’1’:  # if it is a match
            print(’,’.join(row[3:8]))   # keep columns 3-8, which only keeps A

# remove duplicates
isbn = []

with open(’TableE.csv’, ’r’) as csvfile:
    line = csv.reader(csvfile, delimiter=’,’)
    for row in line:
        if row[2] in isbn:
            continue
        print(’,’.join(row))
        isbn.append(row[2])
```