CS 638 Project Stage 2 Report

Zhiwei Fan zfan29@wisc.edu
Lingfeng Huang lhuang58@wisc.edu
Fang Wang fwang64@wisc.edu

October 25, 2016

Source Websites and Relevant Documentation

1. Source Websites:
   Barnes&Noble: *http://www.barnesandnoble.com*
   Goodreads: *https://www.goodreads.com*

2. Corresponding Tables:
   tableA.csv: table transformed from **Barnes&Noble**'s data
   tableB.csv: table transformed from **Goodreads**'s data

3. Original Schema:
   tableA(id: integer, title: string, year: integer, pages: integer, day: integer, publisher: string, month: integer, authors: string, ISBN13: string)
   tableB(id: integer, title: string, isbn: string, pageCount: integer, author: string, publisher: string, date: string)

Schema Transformation

According to *Original Schema* in previous section, tableA and tableB originally have different schemas. After careful consideration, we decide to transform both tables to be in the consistent schema shown as the following:

table(id: integer, title: string, authors: string, ISNB13: string, pages: integer, publisher: string, publishedYear: integer, publishedMonth: integer, publishedDay: integer,)

Specific Steps of Transformation

Most attributes in *tableA* remain the same. The only change made to *tableA* was that *ISBN3* was renamed to be *isbn3*, *year* was renamed to *publishedYear*, *month* to *publishedMonth*, *day* to *publishedDay*. For *B* the attribute *isbn* was renamed to *isnb3*, *pageCount* was renamed to *pages*; attribute *date* has been properly parsed and split into three separated attributes *publishedYear*, *publishedMonth* and *publishedDay* by running appropriate python scripts on the original csv files *tableA.csv* and *tableB.csv* (additional *reordering* has also been done to further strengthen the consistence and make it more convenient when coming to the data analysis and cleaning stage).

Set of Attribute: S

According to the transformation steps we have described in the previous part, the attributes of the consistent schema in *set S* is:
id, title, authors, ISBN13, pages, publisher, publishedYear, publishedMonth, publishedDay.

Attributes Analysis (Table A)

We take Table A as example to show our analysis of attributes. The same process and ideas have been applied to Table B as well. Due to the space limitation, we omit the analysis of Table B here and only display our results for Table A.

Missing Values

In Table A, there are only three attributes we have detected with *missing value* issue and there are: *pages*, *publishedDay*, *authors*. The corresponding missing value statistical data is reported in the following subsection.

Relevant Statistical Data

pages:
number of missing values: 466
missing percentage: 8.83%
missing fraction: 466/5279

publishedDay:
number of invalid values: 4
missing percentage: 0.75% **missing fraction** 4/5279

authors:
number of missing values: 2
missing percentage: 0.39%
missing fraction: 2/5279

Instead of using *missing values*, we use *invalid values* for attribute in *publishedDay* for missing-value report. We do have found that every tuple in table A has attribute value *publishedDay*, however, it's very obvious that some of those values are *out of range* (less than 1 or greater than 31). We regard these *invalid values* equivalent to *missing values*.

Discussion about Possible Solutions for Missing Values

For those values are missing in *pages*, the most naive way we could think is to fill '0' for every entry that *page* is missing. However, this simple solution might not be feasible if *pages* is considered as an important attribute when doing entity matching. Then there is high possibility we will *miss* the match for those books that have '0' pages. The more complicated and tedious solution would be going to a few other websites, fetching the pages information again for those books that have missing values. Taking this solution could give us much more reliable "data fixing" but will bring much more tedious work and put us under the risk of "new data cleaning" task for those new data (which is only used for fixing the *real data* actually). For those values are missing in publishedDay and authors, since the number of values is very small, we could easily fix them by searching the information on the source website we used for crawling data and fill those missing values manually in a few minutes. The decision of our final solution for missing values needs to be carefully considered based on both of the easiness of work and its influence on entity matching quality.

Attributes Type Identification

As shown in the schema of table in *schema transformation* section, the attributes we have selected are all quite easy to determine their types:

id: integer, *title*: string, *authors*: string, *ISNB3*: string, *pages*: integer, *publisher*: string, *publishedYear*: integer, *publishedMonth*: integer, *publishedDay*: integer

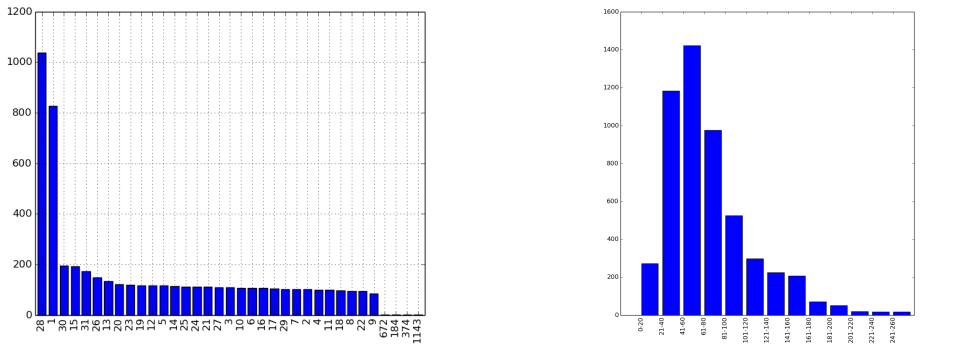
publishedYear , *publishedMonth* and *publishedDay* all have clear meaningful numerical domains and thus are identified as *integer*. Every books's *ISBN3* is composed of 13 numbers, which is not in the integer type value range -2,147,483,648 to 2,147,483,647. Thus we identify *ISBN3* as string type instead. The number of total books could perfectly fit in the integer type value range, so we identify *id*, the attribute we generated during the data crawling stage, as integer type. *title* and *authors* are intuitively identified as string type and *pages* is identified as integer type.

Textual Attribute Value Length Statistical Data Report

There are *four* textual attributes: *title*, *authors*, *publisher*, *ISBN3*

title:
maximal length: 255
minimal length: 4
average length: 67.078
authors:
maximal length: 193
mininimal length: 2
average length: 25.366
publisher:
maximal length: 60
minimal length: 3
average length: 19.64
ISBN3:
maximal length: 12
minimal length: 13
average length: 13

Possible Outliers and Anomalies Detection and



(a) Histogram on publishedDay values

(b) Histogram on length of title values

Figure 1: Histograms for Anomaly Attribute Values Detection

We choose to use the analysis of attribute values of *publishedDay* and *title* to show our analysis on outliers detection. From figure 1a, extremely dense distribution on value 1 and 28 has been detected. Although we admit this *extremely* high dense distribution is somehow different from the normal conception of *outliers* (not attribute value anomaly), we think this observation is important since there might be high possibility of data corruptions (those publishedDays filled with *1* and *28* might not be exactly correct). In addition, we have also observed several *outliers* according to the normal definition: *672, 184, 374, 1143* are definitely *out of range*. One possibility is that attribute values of other attributes have been incorrectly fetched and filled in the *publishedDay* field. *pages* is very likely to be in those attributes. One possible reason could be that the source websites we used to crawling data from have accidentally mis-placed those values in the wrong place in their website structure. From figure 1b, we have observed that there are very few attribute values reside in the range 200 to 260, thus we conclude those values as possible outliers due to their low frequency and the big gap between those values and the average value of title length (67.078 according to *Textual Attribute Value Length Statistical Data Report*).

Attribute Standard Discussion

We intend to represent the published date of every book in *year, month, day* separately. Originally, the published date is represented as a whole string in table B. We have already transform it into *publishedYear, publishedMonth, publishedDay* as described in *Specific Steps of Transformation*. All other attributes have intuitive and simple standard attribute value format as shown in the schema.

Synonyms Among Attribute Values Discussion

Fortunately, *synonyms* is unlikely to happen among attribute values of any attribute in our final schema. *synonyms* are more likely to be seen in *categorical attributes* and there is no categorical attributes in our schema.

Attribute Values Sprinkle Issues

We do have found that for some tuples, attribute values are *sprinkled* all over the item. For example, we have found that some tuples missed *pages*, *publisher*, *publishedYear*, *publishedMonth* and *publishedDay* and those missing values actually show in *title*. The result of this issue result in both *value missing* and *outliner* (e.g., *title* becomes long since it includes other *wrong positioned attribute values*), which will further influence the *entity matching* stage. The reason for this issues might be attributed the possibility that the source website we crawled data from have mis-positioned some attributes for some books or the lack of consideration of data structure and organization when crawling the data.

Additional Data Quality Problems

Due to the *data missing* and *attribute values sprinkle issues*, some other data issues have also arisen and detected. For example, some attribute values of some tuples have been detected to be placed in the *wrong position*. This is either because of some attribute values before the *misplaced attribute* have been missing or some previous values have been *sprinkled* in other attributes.

Software Tools Used in Analysis and Data Cleaning

We mainly used *pandas* for data analysis (e.g., collecting information about missing values, length of title attribute, etc). In addition, we have used *csv package* in *python* to do relevant schema transformation and partial data cleaning.