Stage5 Data Analysis

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

December 15, 2016

Term explanation

tableA: book data obtained from Barnes and Noble.
tableB: book data obtained from Good reads.
scheme of both tables: id, title, authors, ISBN13, pages, publisher, publishedYear, publishedMonth, publishedDay
tableC: the table after blocking stage.
tableE: the table after merging stage. It has the same schema as tableA and tableB, but its id is not obtained from tableA nor tableB. Its id is new and in increasing order start from 0.

Description of table merging

During the project stage 4, we chose *random forest* as our final selected model because of its high F1 score. Thus, in this final stage, we first obtained the prediction result from running *random forest* to the blocking tableC from stage4 (matching tuple pairs with information from tableA and tableB). By looking at the prediction boolean array generated from random forest, we generate a intermediate table containing tuple pairs that are predicted to be matched. We call this filtered-table. Our goal of merging is to ensure we find all unique tuples from tableA and tableB so we need to use this filtered-table in the next stage.

how did we generate merged table?

To generate tableE, we first keep all the data from tableA (always select the values from the tuple from table A) because tableA has well formatted data and there are lots of missing data in tableB. Thus, when there is a match, tableA's data has higher priority than data in tableB. After this step, we still have't added the tuples that are not present in tableA but present in tableB. Then, using the filtered-table from the last step, we are able to know which tuples in tableB have already presented in the tableA. We don't need these data so we take tableB and find all tuples which IDs are not present in the filtered-table. These are the tuples that are not present in tableA but present in tableB. By adding these tuples to tableE as well, we have a complete tableE. During the process of adding tuples from tableB, we encountered several value misplacement issues: tableB has lots of dirty data in pages, publishedYear and publishedMonth field. For example, value in pages is a publisher string instead of a number or value in publishedYear is misplaced by the author strings etc. For cases like this, we choose to discard these tuples since these tuples cannot be populated by finding matches in tableA. We also cleaned tableE so that in column id, title, pages, publishedYear, publishedMonth are having no misssing values (while ISBN13 may has missing value which is listed below in one of the sample from tableE). In summary, the step of merging is as follows: put all tuples from tableA to tableE, add tuples from tableB that has not yet been presented in the tableA into tableE as well. Perform some data cleaning while adding.

Statistics on Table E

schema of table E: id, title, authors, ISBN13, pages, publisher, publishedYear, publishedMonth, publishedDay
number of tuples in table E: 5922

id	title	authors	ISBN13	pages	publisher	publishedYear	publishedMonth	publishedDay
0	The Java Tutorial: A Short Course on the Basics / Edition 6	Raymond Gallardo,Scott Hommel,Sowmya Kannan,Joni Gordon,Sharon Biocca Zakhour	9780134034089	864	Addison-Wesley	2014	12	26
3691	Database System: Design and Development	M. M. Kashyap	9788120712638	382	Apt Books, Inc.	1993	6	1
5848	Conceptual Schema And Relational Database Design: A Fact Oriented Approach	G.M. Nijssen,Terry Halpin	9780131672635	342	Prentice Hall	1989	4	1
5920	Concepts of Database Management System	Shefail Naik		348	Pearson	2013	4	1

Table 1: 4 samples from tableE

Data analysis

Since our schema is quite simple and unfortunately we did not obtain the data of book prices nor user reviews, there are limited tasks we can do. We finally chose two data analysis tasks we would like to perform. First, we would like to know if the number of published books are increasing over the periods. Second, we would like to know if books are getting thicker (more pages) over the time periods.

Were there any problems with the analysis process and with the data?

To conduct data analysis, we performed clustering based on grouping all tuples by its published periods (10 year per period), then we counted the number of books published in the periods and also calculate the average pages for each period. In the first iteration, we found that the average of pages are abnormally high in some periods. Thus, we perform anomaly detection by first sorting fields including pages, publishedYear and publishedMonth, we detected books that have over 20,000 pages which are obviously wrong and we also detected publishedMonth that has value of 1996 which seems to be a duplication of its publishedYear. We cleaned the abnormal data and luckily these unusual cases are small, so deleting these tuples is unlikely to greatly influence our data analytics due to our analysis setting. (the 5922 tuples in tableE are the tuples survived after this process)
After anomaly detection and further data cleaning. We performed clustering again. This time the result are as follows (Note that the time periods are represented by their first year, here 1960 means the period from 1960 to 1970 instead of a single year):

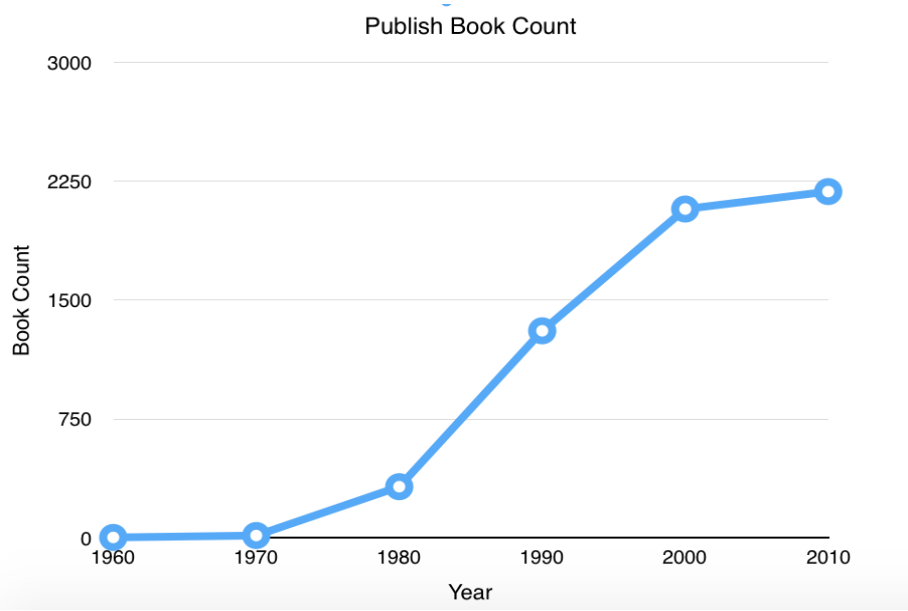

Table 2: relation between book counts and published periods

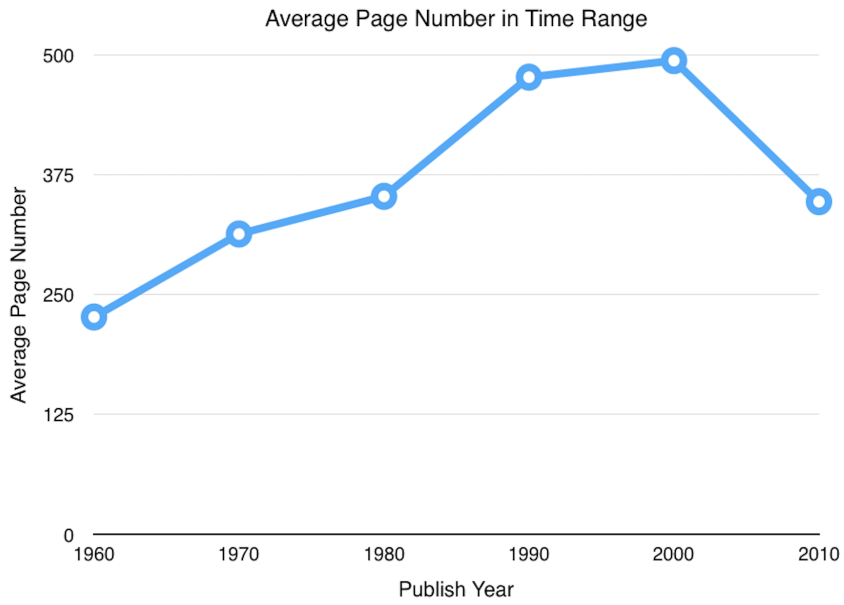

Table 3: relation between pages and published periods

Discussion

What did we learn/conclude from our data analysis?

From table2, we can clearly see a rise in pages of books from time periods started from 1960-1970 period. Our initial natural assumption was that with the increase of time and the development of different fields including science and technology, more new knowledge domains kept emerging and the needs for accessing these knowledge are needed by both academia and industries. Thus, we expect in the future the number of published books will continue to increase. However, from table3, although the average book pages are steadily increases from 1960s to 2000s, the average book pages decrease since 2010. There could be many possible explanations: after our discussion, we think many authors might have some *common habits* (although without explicit explanations) to only write one book around 500 pages. Authors who want to write more about one thing is likely to split up the contents and cover all the contents in separate books instead of one for better organization purposes. The increasing available online resources could also be a factor that decreases the motivation for a author to write a thick book with the increase of the timeline. The power of media and more convenient transportations give people more and easier access to all different kinds of knowledge and information compared to the past, in which time period people are more limited to get information and knowledge from books. For example, abundant information could be got from the internet and people could travel around to attend different academic conference to gain the latest knowledge and information via interesting and active communications with other people rather than plainly reading the book. In addition, the reduction of book pages could be out of the consideration of environment protection as well. Since our data is limited, we could only make these assumptions while more rigorous conclusion could be only be achieved by collecting more data and performing more varied analytic methodologies.

Possible future work

The fact that our schema is very simple has constrained us from doing more complicated analysis. Also, when trying to form the problem setting and conducting corresponding data analytics, we have found that those *dirty* data either with *missing values* and *wrong values* (e.g., anomaly data) both hindered us in exploring more interesting insights from the data and conducting current data analytics. Due to the time limits, we are only able to address these problems for the current stage. In the future, we will be more careful with data cleaning. And pre-determining the specific goal or scope of data analytics should help a lot in data collection stage including the data domain and data schema.