Data Linkage Report

COMP20008 Project 2
Name: Fu-Sheng(Jeremy) Huang
Student Number: 1046432

Enjoy dealing with data in the world

Linking Data without Blocking

Approach

By observing the data from abt_small.csv and buy_small.csv, it can be deduced that the product names are somewhat like the product description from abt to know product from buy. So, the approach to linkage is to measure the **text similarities of the product names and description** from data sets and match the products that have similarity value higher than a certain threshold. This rather brute-force approach takes $O(n3)$ time to complete the matching including to purify the words. Once product pairs that exceed the threshold have been extracted, further processing is performed. For each product in the buy_small.csv that has multiple matches with product in the abt_small.csv, only the product with highest threshold was chosen to be paired with the product.

Similarity Function and Threshold and Regex

To measure the similarity of two product names, **token_set_ratio from fuzzywuzzy from python library** was used. FuzzyWuzzy is a library of python that is used for string matching. **It uses Levenshtein Distance** to calculate the difference between sequences. Furthermore, token_set_ratio attempts to rule out the differences in the strings. It checks the ratio on three substring sets and returns the max value. Further explanation about it can be found at https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/. **The threshold for name comparison set to 91 and for the model numbers, I set 92 for after pure and 78 for original type**, and anything higher than this was considered as a potential match. This threshold was decided based on trial and error to find the break-even point of fn (false-negative) and fp (false-positive). And the other selection conditions are to filter some correct matches that I believe is right then put them in answer directly and skip some mismatches that I believe is incorrect. The most importantly, I use the **Regex methods** to improve.

Results and Evaluation

The executed program has linked 150 items between abt_small.csv and buy_small.csv with a true positive count of 135, false positive count of 15, false negative count of 14 and true negative count of 61,588. These yields recall of 90.6% and precision of 90.0%.

To further increase the accuracy of data linkage, **additional text matching can be performed**. For each potential match that was found, similarity between two products' name and description can be computed to ensure that the highest similarity for the product name also has highest similarity for the product name and description.

Recall

90.6 %

Precision

90.0%

True-Positive

135

Actual Value

149

Linking Data with Blocking

Approach

Again, by observing the data from abt.csv and buy.csv, it can be deduced that the product names are somewhat similar or the same as manufacturer. For every product in each data set, the product name or manufacturer that was tokenized, the words contained in **stopWords were removed** and the remaining words were placed into names. Then the words in names were **processed by stemming before adding to block** of each data sets. Once two different blocks are created, simply take the intersection of two blocks and resulting blocks are the blocking for two data.

Stop words and Stemming

stopWords(stw) contains words that appear frequently in the English language. Often, these words do not provide useful information. Here, in addition to the original Stw in nltk library, **string.punctuation from string** library those are also common in the English language. For stemming the words in names, **PorterStemmer(ps)** from nltk.stem.porter was used. Ps uses the Porter Stemming Algorithm which removes the commoner morphological and inflexional endings from words

Results and Evaluation

The executed program created 112 unique blocks and linkage with 1,051 true positive, 65483 false positive, 46 false negative count, 1,113,872 true negative count (assuming unique blocking) and 1,180,452 total possible record pairs. This yield to pair completeness of 95.81% and reduction ratio of 94.36%.

To further improve the result, it is possible to reduce the number of false positive pairs by **expanding the stopWords** list. For example, extra statistics from verification shows that largest paired block, “soni” has 182 abt ids and 176 buy ids resulting in many false positive pairs. However, to exclude them will also reduce the number of comparisons required for matching. Therefore**, we can also compare the names in both data set**, which in my experiment I got almost 100 % PC in result and the true positive is about 1092 that has only 5 difference to actual value. Moreover, their descriptions also have good results, but the PC cannot over 95 percent. In my observation, it is because the stopWords are more in descriptions and some of the description data in buy are in float type.

PC

95.81 %

RR

94.36 %

True-positive

1051

Actual Value

1097