COMP20008 Data Linkage Project Report

**Scoring method: How product comparisons works**

Pre-processing

I dropped the duplicates from "buy\_small.csv" and "abt\_small.csv" and kept the first occurrence of the duplicates from the data frames I created. df1 corresponding to the "buy\_small.csv" and df2 corresponding to "abt\_small.csv"

I stored the name columns of both data frames and stored the IDABT into different series objects.

Comparison Function

I used the fuzzy wuzzy library and compared the abt\_small\_name to all of the buy\_small\_names using fuzz.token\_set\_ratio.

I used token set ratio as it separates each string into words, turns both lists into sets so it disregards repeating words then sorts them before doing the ratio.

Threshold

I used two threshold values, I checked if the token\_set\_ratio was higher than 95 and if it was I would remove it from the data frame so it not match to any other words as I was pretty certain that it was a correct match

I used a threshold value of 72 to make sure that my precision and recall were balanced. If I had a higher threshold value, my precision would improve but my recall would be lowered. This was the best result as of trial and error.

Final Scoring Function

I did not implement a final scoring function, my final scoring function was the same as my comparison function so the token set\_ratio defined my results.

**Evaluation of the overall performance of the product comparison**

Overall performance of my product comparison is very average, the recall and precision are above 0.6 but below 0.7 which means there are a lot of false negatives and false positives. Opportunities to increase the precision and recall would be to use more comparison functions, such as n gram comparison.

**How your blocking implementation works**

I used the brand names from manufacturer column of buy.csv as my block\_key, I then pre processed the block\_key by making them all lower case and only including the first word of each string. I then made sure that my data frame for buy.csv and abt.csv were all lower case.

Then I went through each of the block keys and checked whether the “name” column of data frame contained the block\_key in the string, and returned the index as a string. Then I returned the product id from list of indexes formed a dictionary of block\_keys as keys and list of product id for my values. After returned the dictionary into the data frame and formed a csv file.

**Evaluation of the blocking method**

Pair Completeness: 0.938 Reduction Ratio: 0.926

Overall performance of blocking method is very high. Pair completeness being high indicates that there is a small amount of false negatives. Reduction ratio being high indicates that there is a lot less comparisons that were made with blocking than without blocking. A improvement that could be made to the block keys also add in range of prices for each brand.

The time complexity is O(n^2)