Asuka Li
CS 470
Dr. Zhao
Feburary 23, 2022

CS 470 HW2 Report

Honor Code Statement
THIS WORK WAS MY OWN WORK. IT WAS WRITTEN WITHOUT CONSULTING WORK WRITTEN BY OTHER STUDENTS OR COPIED FROM ONLINE RESOURCES. Asuka Li

I. Analysis

The threshold I used to generate my output file was 500.

The minimum support value I used to conduct the analysis was 500. Comparing the output results of using minimum support values of 100 and 1000, I've concluded that 500 gives the most useful information. When using 100, I've found that there were multiple overlaps in the itemsets outputted, therefore not entirely serving as useful information. However, when using 1000, too many itemsets were filtered that some valuable information appeared to be missing compared to the result we have by using 1000. From the patterns I have found using the implemented method, "flu" and "got" seem to be some of the most frequently used words in the tweets. A lot of the other frequent itemsets show that people simply report that they had gotten their flu shot that day, and the pain level of their shot. There isn't any significantly important information tweeted in great frequency besides mundanely reporting the status of their flu shot. One thing that caught my eyes was the frequent itemset "first flu shot" with a support value of 591. It was to my surprise that it's the first time getting a flu shot for so many people in 2014.

II. Algorithmic Optimizations

I followed the pseudocode provided in our CS 470 lecture slide to implement the Apriori algorithm. In generating the possible combinations of the keywords to produce different number of itemsets, I utilized the itertools package in Python, specifically the combinations method to easily conduct the process. For the pruning and elimination steps, I used .issubset to make the

identification process easier with the built-in function. To make the get support step more self-explanatory, I utilized hash maps for its constant time insertion and deletion to improve the speed of the algorithm. Besides the packages and built-in functions, I used to write the algorithm, there is no algorithmic optimizations I used in the implementation.

III. Lessons Learned

Implementing the algorithm was definitely a challenging process which pushed me to think critically about how the algorithm looks at each data row and the process of pruning and eliminating itemsets from the frequent itemset list. This homework helped me understand each step of the algorithm beyond the level taught in class. Furthermore, I learned that there are many different approaches towards the same algorithm. Through doing research to get inspirations on implementing the algorithm, I found many different workings and implementations of the same algorithm with different optimizations. One example is that if a person knew that they were working with large amount of textual data, they may have implemented a faster string comparison method, compared to someone working with numerical data.