**Q2.** Write a short description of how the sentiment analysis was done and what the outcome is. Make sure your answer is no longer than three paragraphs, and should at minimum answer these questions:

* How did your processing affect the sentiment assignment, if at all?
* What measure did you use to determine the sentiment label? Why? Do any of the label assignments surprise you?
* Include a few specific examples of label assignment and how it was determined and why it does or does not make sense. Audience: general – management or non-technical staff.

I want to use a different dataset for this question: Amazon grocery and gourmet food review data gathered by Stanford Network Analysis Project (SNAP). This review dataset contains 151,254 user reviews from May 1996 to July 2014. Dictionary are basically list of words and a sentiment score on each word (positive or negative). For each document, first I converted text to lower cases, then I matches the positive or negative words from dictionary word list to sum up the total sentiment score. I tried three different dictionary word list. The results of all three-sentiment analysis are input into a voting system to determine final sentiment analysis result. Each method returns a positive, negative or neutral. Which sentiment has the highest voting results is the final sentiment results. The voting system increased accuracy and reduced variance. This approach is like an ensemble method.

I was surprised by some of the discrepancies in results. I found out list of reviews where review is negative but user rating is 5/5 (false negative), and another list of reviews where sentiment is positive but user rating is 1/5 (false positive). There are total of 9907 out 151,254 reviews or 6.5% of the reviews are in false negative category, and there are 4067 or 2.7% of the reviews are in false positive category. For example, the review 96 list below [1] is a false negative. The user gives a rating of 5 out of 5 and anyone reads it see writer has a positive sentiment to the product. But there are more negative words than positive thus the negative sentiment result. Similarly review 1334 [2] is a false positive example. In this example, “pretty” was count as a positive sentiment. My algorithm did not get it right the first time. In the following section I will discuss in detail how I fixed this problem.

[1] "This bottle is a steal, at Whole Foods (at least where I live) a bottle half this size costs twice as much. Before I was also using imitation vanilla, it smells okay but you can smell the fillers. This bottle smells like fresh, pure vanilla. I used it in a lemon poppy seed cake and it was delicious, mild and aromatic! My kitchen smelled WONDERFUL!"

[2] I have had other stevia products before and liked them. This stuff, however, is incredibly nasty. It tastes chemically and toxic. Twice I tried to get myself to drink it and both times I opted to throw out my coffee rather than tolerate the horrid taste. Awful product.

**Q3.** Write a short description of the exercise and the outcome. Make sure your answer is no longer than three paragraphs, and should at minimum answer these questions:

* What outcome did you choose? Why?
* How did you change the dictionary to achieve that outcome?
* How would you explain (justify, rationalize) those changes if necessary? Audience: general – management or non-technical staff.

The HL function gives a negative sentiment on review 96. However, the user's product rating is 5/5 so this should be a positive review. The review is positive from reading it. But computer does not understand that. So, I want to change the algorithm to fix it. It turns out words "smell" "smells", "smell", "smells", "smelled" are all negative words in the HL dictionary. But this is a food review therefore "smell" is a normal word. And in actual text "smelled wonderful", "smelled okay" are positive or at least neutral. First I realized “smelled wonderful!” did not count because the punctuation was not removed. By removing the punctuations from "delicious," "aromatic!" and "wonderful!", these words will be matched with positive words in the dictionary, this improves review sentiment to more positive. Then, we would not miss some important positive or negative words. But the text still has more negative words than positive words. At first I wanted to remove "smell" all other "smells", "smell", "smells", "smelled" from the HL dictionary but I decided to not to do it. This is because "smelled" can still be a negative sentiment in other cases. For example, "something smelled" like "something stinks" has a negative connotation. I do not want to change the rule here for one text.

I thought we were already looking at previous word why not look at next word too. By looking at both previous and next words we can get more accurate results. For words that are both positive and negative sentiment, if the next word is opposite sentiment, then we do not count it. I modified the function to look at the next word. After making this change this review’s sentiment result changed from negative to positive. But this is just one example, how can I justify making a change without testing it on more data? I can justify the changes after seeing the reductions in false negative and false positive numbers from validation on all 151,254 reviews. It reduced the false negative from 6.9% to 5.3%, and false positive from 2.5% to 2.2%. The improvement is because false positive term “smelled wonderful” is positive even though “smelled” is a negative word; false negative term “incredibly nasty” is now negative even though “incredible” is a positive word. The improvement on false positive is less because there are less terms like “incredibly nasty”, “pretty dreadful”, and “super disgusting” (positive word with next word is negative).

We only look at one previous word, in the future I want to look at two previous words: “no longer”, “no more”. “no longer good” should be a negative sentiment. There are many two words term we can consider. I did not include them in my function but it certainly can improve result even more.