Amazon Product Reviews Scoring using R
PROBABILITY LAB PROJECT

BY: Vaibhav Jain
IV Semester, B.Tech - ITMI
Cluster Innovation Centre, University of Delhi

Introduction

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing (NLP), text analysis and computational linguistics to identify and extract subjective information from the source materials. In other words, it can also be described as the computational task of automatically determining what feelings a writer is expressing in text. Sentiment is often framed as a binary distinction (positive vs. negative), but it can also be a more fine-grained, like identifying the specific emotion an author is expressing (like fear, joy or anger).

Sentiment analysis is used for many applications, especially in business intelligence. Some examples of applications for sentiment analysis include:

- Analyzing the social media discussion around a certain topic
- Evaluating survey responses
- Determining whether product reviews are positive or negative

However, sentiment analysis is not perfect, and as with any automatic analysis of language, you will have errors in your results. It also cannot tell you why a writer is feeling a certain way. However, it can be useful to quickly summarize some qualities of text, especially if you have so much text that a human reader cannot analyze all of it.

One example of such instance is Reviews on marketing sites like Amazon. Customer's reviews are valuable for sellers as well as producers. But on a huge marketing website like Amazon or Flipkart, monitoring each review is not feasible. This is why, we developed this idea of mapping text reviews to numerical scores so that we can apply mathematical

techniques to more easily find trends in the product sale. For this task, we use sentiment analysis to generate a sentiment score for the reviews. This sentiment score will describe the nature of the review, a higher positive value will signify a favourable review and negative values signify otherwise.

In the rest of the report, we first describe the methodology we followed. Then we will discuss the results we get from our analysis on a dataset of about 400K product reviews from Amazon. In the end, we conclude this report with a note on usefulness of this system.

Methodology

There are many ways to do sentiment analysis. In fact a lot and research and resources has been spent in the last decade since the emergence of Deep Neural Networks. Among many, some popular sentiment analysis techniques are Lexicon-based analysis, Context-dependent analysis, subjectivity summarisation, Statistical-based analysis etc. However, almost every method share a general idea:

- Create or find a list of words associated with strongly positive or negative sentiment.
- Count the number of positive and negative words in the text.
- Analyze the mix of positive to negative words. Many positive words and few negative words indicates positive sentiment, while many negative words and few positive words indicates negative sentiment.

We will also follow the same lines and subsequently discuss each of them in detail. In this project, we choose to use Lexicon-based analysis as it is one of the most popular technique and gives best results when we have a relatively small dataset.

Step#1: Creating a Lexicon

The first step, creating or finding a word list (also called a lexicon), is generally the most time-consuming. While you can often use a lexicon that already exists, if your text is discussing a specific topic you may need to add to or modify it.

The choice of lexicon is the most crucial part of the sentiment analysis. It can make or break your model based on its relevance to the input. For example, the lexicons provided in the R library “tidytext” were constructed via either crowdsourcing (using, for example, Amazon Mechanical Turk) or by the labor of one of the authors, and were validated using some combination of crowdsourcing again, restaurant or movie reviews, or Twitter data. Given this information, we may hesitate to apply these sentiment lexicons to styles of text dramatically different from what they were validated on, such as narrative fiction from 200 years ago. Although, we still can measure the sentiment content for words that are shared across the lexicon and the text, but its accuracy will certainly decrease.

Step#2: Tokenize the Input

This step is one common task in NLP (Natural Language Processing) which in most ways is a superset of sentiment analysis. "Tokens" are usually individual words (at least in languages like English) and "tokenization" is taking a text or set of text and breaking it up into individual its words. These tokens are then used as the input for other types of analysis or tasks, like parsing (automatically tagging the syntactic relationship between words).

```
# tokenize
tokens <- data_frame(text = fileText) %>% unnest_tokens(word, text)
```

Step#3: Removing the stop words

A stop word is a commonly used word (such as "the") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

We now remove the stop word in our code which will help us to analyze the sentiments of the effective words that are essential to the documents

```
# remove stop words
data(stop_words)
tokens <- tokens %>% anti_join(stop_words)
```

Step#4: Get Sentiment Score

Now, we get down to our last task. To find sentiment scores, we need to compare the tokens we parsed from the input in the previous step against a list of positive or negative tokens. A list of words associated with a specific sentiment is usually called a "sentiment lexicon". This is the same lexicon we created in the first step.

```
# get the sentiment from the text:
tokens %>%
  inner_join(get_sentiments("bing")) %>% # pull out only sentiment words
  count(sentiment) %>% # count the # of positive & negative words
  spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
  mutate(sentiment = positive - negative) # # of positive words - # of negative words
```

Step#5: Plotting the sentiments scores

We have the sentiment scores of each review. Now, we would like to plot these reviews with various factor like time and frequency of the scores in order to get a good idea of how the reviews are changing and to know how the customers opinion about the project are changing which can help us to improve the marketing strategies.

```
v<-read.csv("Reviews.csv")
x<-v[204800:205000,c("ProductId","Score","Time")]
y<-x[order(x$Time),]
```

```
g<-subset(y,ProductId=="B000UYIPYY")

i<-g$Time

jpeg("SydneyPretzel.jpg")

plot(as.Date(as.POSIXct(i, origin="1970-01-01")),g$Score,xlab="Years",ylab="Sentiment Score",main="Sydney Pretzel",type="o",cex=1,col="blue")

dev.off()
```

Results

We plotted the graphs depicting the sentiment core of user reviews vs Time(days). This tells us how have the user reviews been over a certain period of time for some of the tested products. Below are some of these plots and we here discuss some of the conclusions we can make from these plots.

- ➔ A plot of Sentiment score of "Canidae Dog Food" over the years of its sale.

- Canidae dog food is an amazon product and we can see that there was a certain dip in popularity among the users between 2008 and 2010.
- It's hasn't been reviewed much and that gives as us an indication that it doesn't have a big user community.

➔ A plot of Sentiment score of "CHAMOMOILE_TEA" over the years of its sale.

- We can see over here that over a span of just two to three years many users expressed their reviews on the website.
- This indicates a certain rise in popularity since the introduction of the tea.
- Majority users have given positive reviews if we consider a four plus rating as positive.
- This tells us that not only the product sale was good, users found it to be of good quality too.

➔ A plot of Sentiment score of "Sydney Pretzel" over the years of its sale.

- Sydney pretzel has been relatively popular over the years and except a few bad sentiment scores it has done very well in the market.

➔ A plot of Sentiment score of "Twistys-Strawberry" over the years of its sale.

- This product was doing very well between 2010 and 2012 and after that it has dipped both in terms of user community and popularity.

Another detail we obtained from our dataset was through plotting a graph between sentiment score and frequency of sentiment scores in the given data set.

The relative frequency histograms obtained above are on a subset of the reviews consisting of two lakh reviews(left) and another fifty thousand reviews(right).

We can see that somehow there's a trend that most people were highly satisfied with their products. Conspiration? Who knows? This either gives us a hint that there are a lot of false reviews posted by non-consumers or a majority of the users end up giving a positive review.

Conclusion

As shown by the results, sentiment analysis can be effectively used to extract a numerical score from text reviews. This sentiment score can more completely define the emotions of the user and can relate to the review when compared to other means like stars, points etc. This can be used to devise more effective marketing strategies and help the industries serve more suitably to the customers. We conclude this report by noting that there still are a lot of possibilities to uncover when it comes to application of sentiment analysis.