# R Notebook

```
#Corpus Import & Syntax
#fileencoding needs to be set to "latin1"
MentionsURL <-"https://raw.githubusercontent.com/Misterresearch/CUNY-Projects/master/App%20Mentions.csv

MentionsTable <- read.csv(file = MentionsURL, header = TRUE, sep = ",", strip.white = TRUE, na.strings =
MentionsTable$description = as.character(MentionsTable$description)

#Create data frame
mentiondesc <- data.frame(MentionsTable$description)

#makes each row in data frame a document, required for subsequent statistical analysis.
mentiondesc <- Corpus(DataframeSource(mentiondesc))

#Corpus Loading, filtering and stemming code. See "Basic Text Mining" source in end notes.
mentiondesc <- tm_map(mentiondesc, removePunctuation)
#for(j in seq(mentiondesc))
#{
  #mentiondesc[[j]] <- gsub("/", " ", mentiondesc[[j]])
  #mentiondesc[[j]] <- gsub("@", " ", mentiondesc[[j]])
  #mentiondesc[[j]] <- gsub("\\|", " ", mentiondesc[[j]])
#}
mentiondesc <- tm_map(mentiondesc, removeNumbers)
mentiondesc <- tm_map(mentiondesc, tolower)
mentiondesc <- tm_map(mentiondesc, removeWords, stopwords("english"))
mentiondesc <- tm_map(mentiondesc, removeWords, c("none","the", "and", "or" , "http\\w*"))
mentiondesc <- tm_map(mentiondesc, stemDocument)
mentiondesc <- tm_map(mentiondesc, stripWhitespace)
mentiondesc <- tm_map(mentiondesc, PlainTextDocument)

#Single Term Matrices
mdtm <- DocumentTermMatrix(mentiondesc)
mtdm <- TermDocumentMatrix(mentiondesc)
mdtm
```

```
## <<DocumentTermMatrix (documents: 4832, terms: 7152)>>
## Non-/sparse entries: 41757/34516707
## Sparsity           : 100%
## Maximal term length: 218
## Weighting          : term frequency (tf)
```

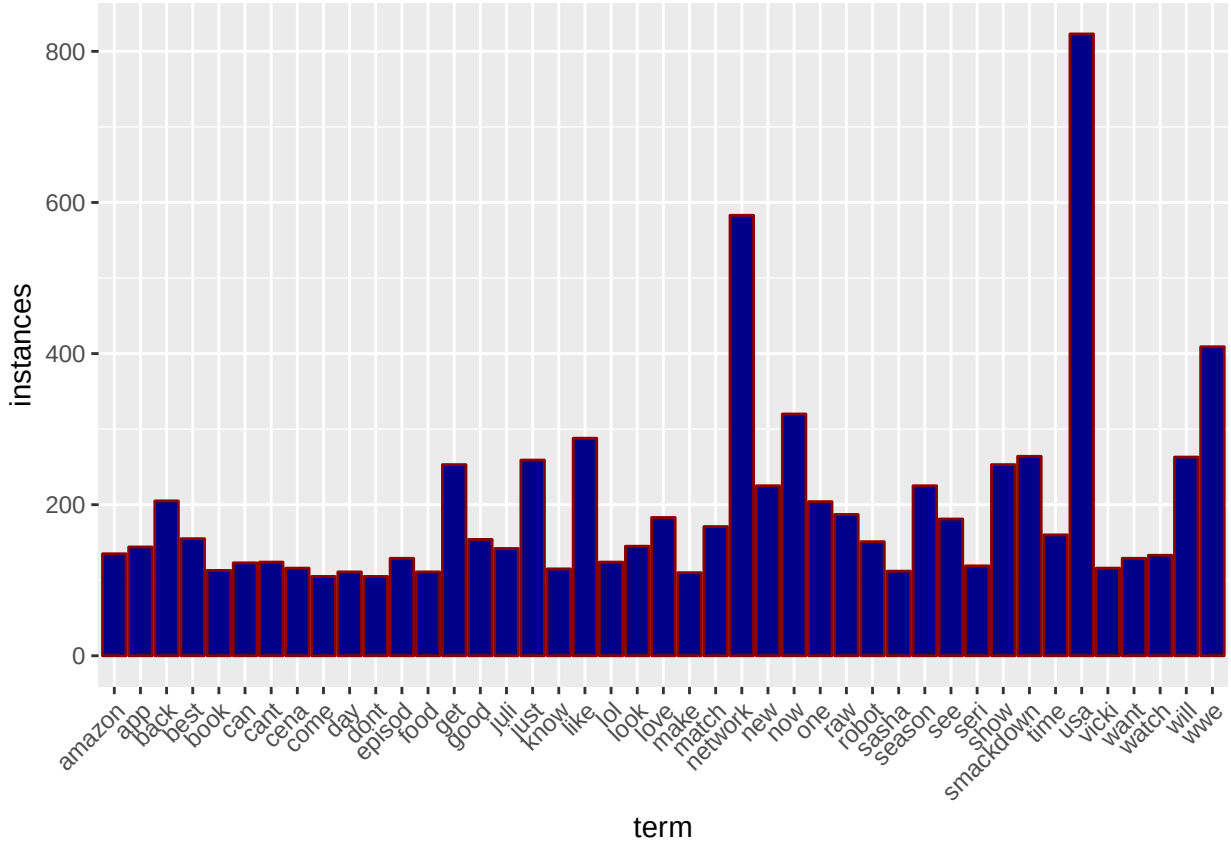
```
mtdm
```

```
## <<TermDocumentMatrix (terms: 7152, documents: 4832)>>
## Non-/sparse entries: 41757/34516707
## Sparsity           : 100%
## Maximal term length: 218
## Weighting          : term frequency (tf)
```

```
#Sparcity settng adjustments
mentionstdm2 <- removeSparseTerms(mtdm, .97)
mentionsdtm2 <- removeSparseTerms(mdtm, .97)
mentionsfreq <- rowSums(as.matrix(mtdm))
```

```
#findFreqTerms(mentionstdm2, lowfreq = 1)
```

```
#Single Term Frequency Charts
tf <- data.frame(term = names(mentionsfreq), instances=mentionsfreq)
subset(tf, mentionsfreq>100) %>%
  ggplot(aes(term,instances)) +
  geom_bar(stat="identity", fill="darkblue", colour="darkred") +
  theme(axis.text.x=element_text(angle = 45, hjust = 1))
```

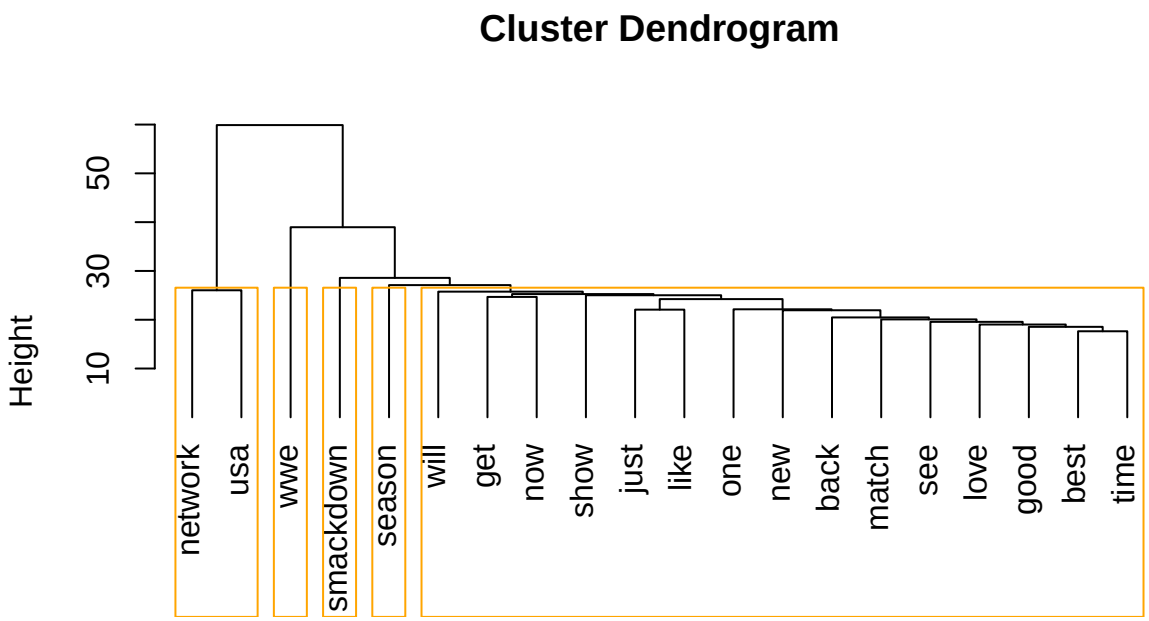



```
#Single Term Word Clouds
wordcloud(names(mentionsfreq), mentionsfreq, min.freq = 100, scale=c(5, .1), colors=brewer.pal(6, "Dark
```

```
#Single Term Correlation Analysis
findAssocs(mtdm, c("Twitter", "Amazon", "chromecast"), corlimit = .2)
```

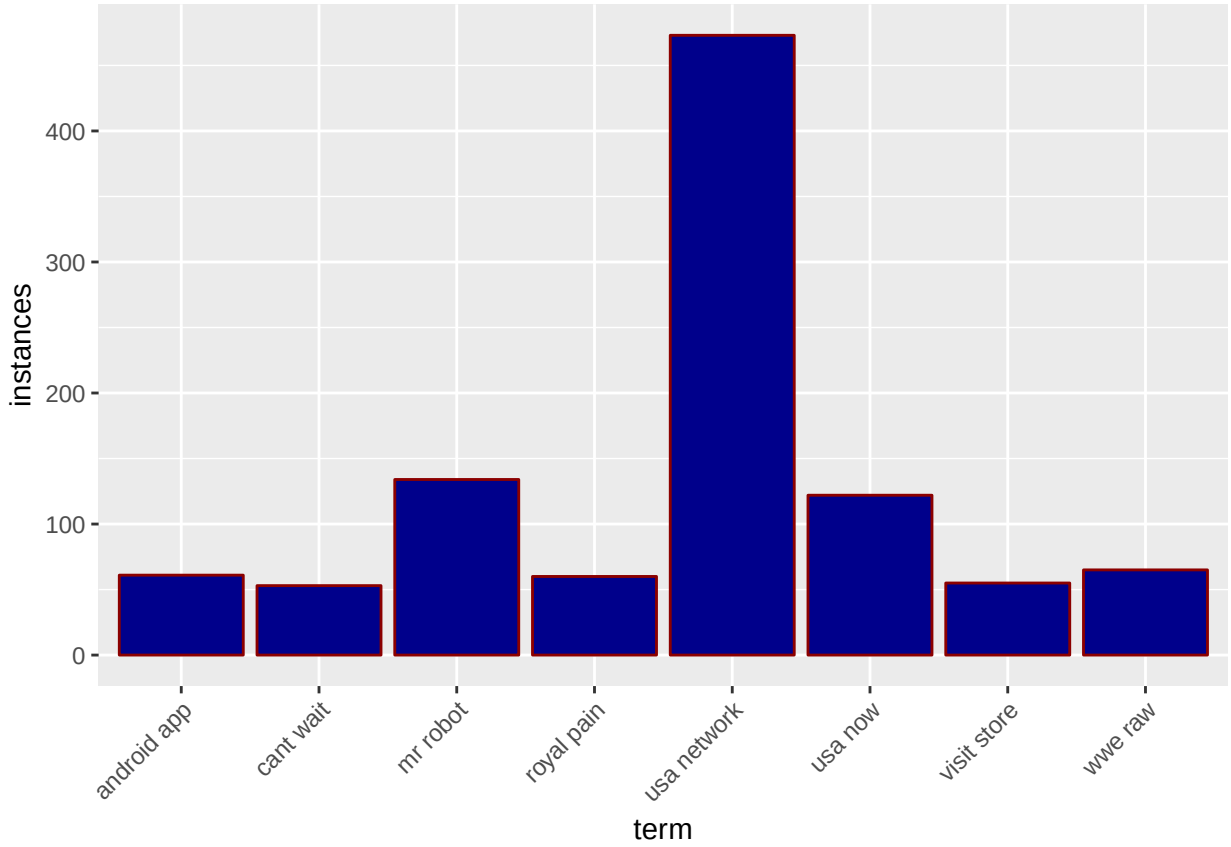
```
## $Twitter
## numeric(0)
##
## $Amazon
## numeric(0)
##
## $chromecast
## numeric(0)
```

```
#Single Term Cluster Analysis, requires Document-Text Matrix

dendro <- dist(t(mentionsdtm2), method="euclidean")
cluster <- hclust(d=dendro, method="ward.D")
plot(cluster, hang=-1)
rect.hclust(cluster, k=5, border="orange")
```



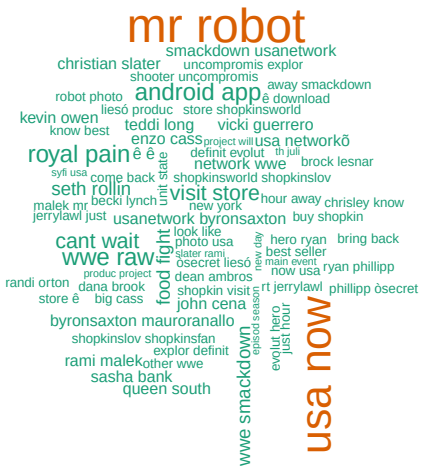

```
#Bigram Corpus, see source for "Bigram Text-Document Matrices" in endnotes
BigramTokenizer <-
     function(x)
        unlist(lapply(ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)
mtdm2 <- TermDocumentMatrix(mentiondesc, control = list(tokenize = BigramTokenizer))
mdtm2 <- DocumentTermMatrix(mentiondesc, control = list(tokenize = BigramTokenizer))
mentionsfreq2 <- rowSums(as.matrix(mtdm2))


#Bigram Frequency Chart
```

```
tf <- data.frame(term = names(mentionsfreq2), instances=mentionsfreq2)
subset(tf, mentionsfreq2>50) %>%
  ggplot(aes(term,instances)) +
  geom_bar(stat="identity", fill="darkblue", colour="darkred") +
  theme(axis.text.x=element_text(angle = 45, hjust = 1))
```

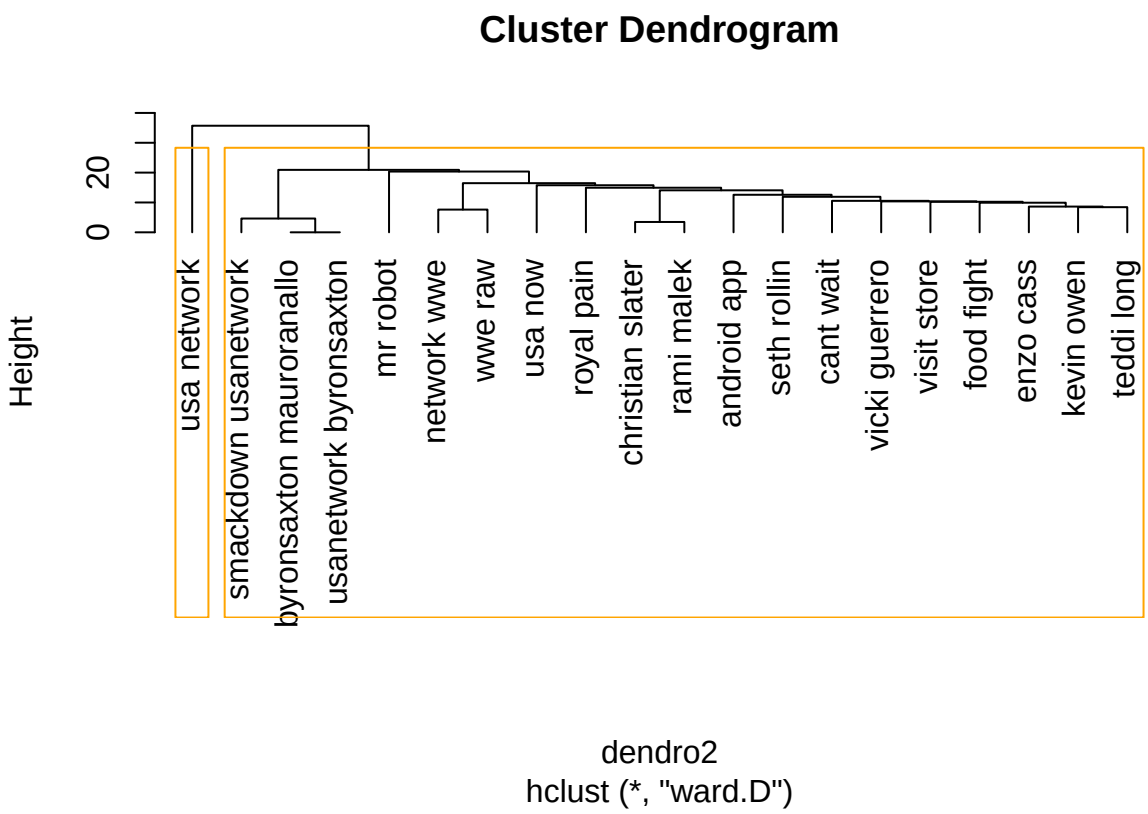



```
#Bigram Word Cloud
wordcloud(names(mentionsfreq2), mentionsfreq2, min.freq = 25, scale=c(5, .1), colors=brewer.pal(6, "Dark
```

```
## Warning in wordcloud(names(mentionsfreq2), mentionsfreq2, min.freq = 25, :
## usa network could not be fit on page. It will not be plotted.
```

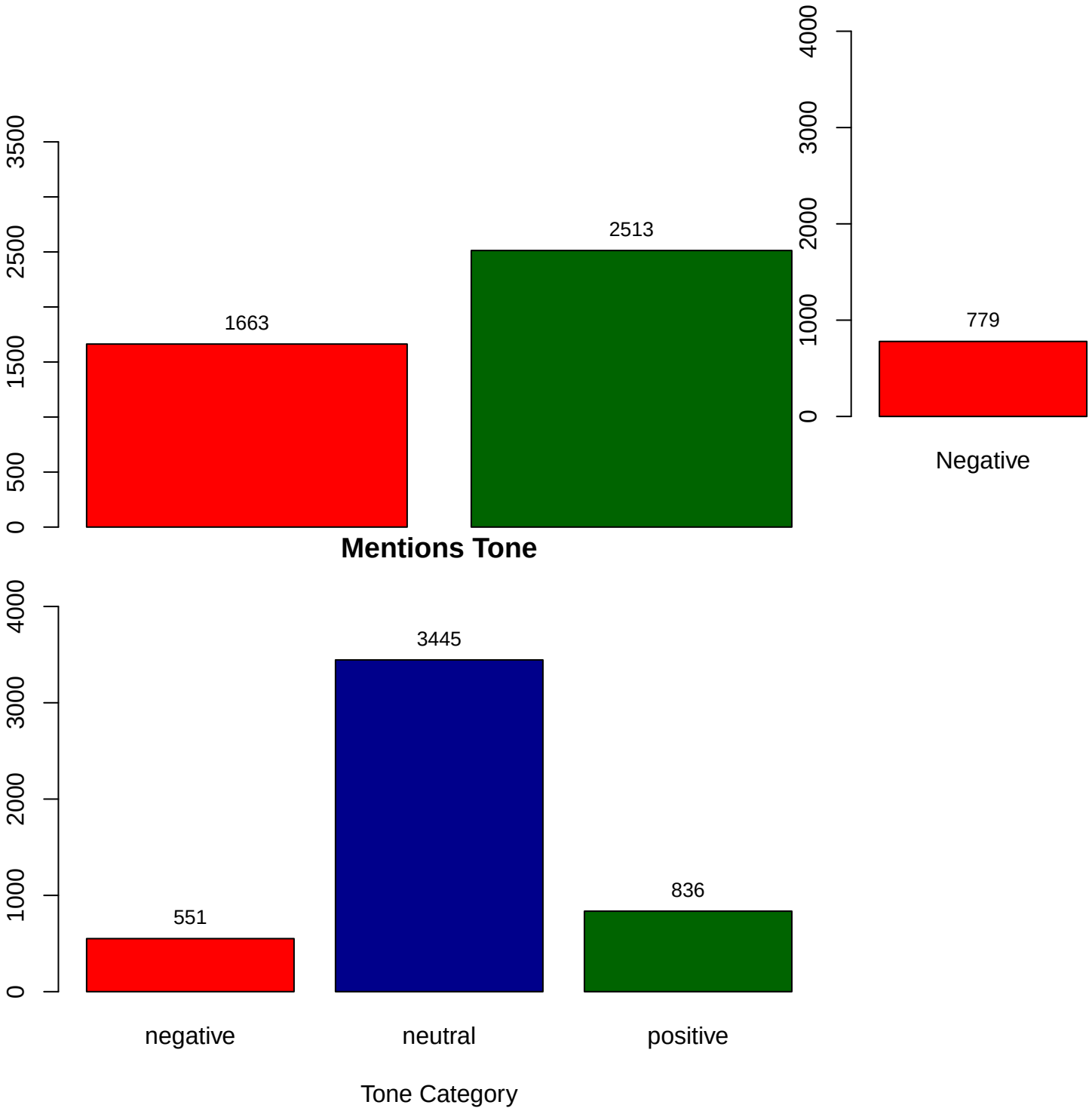


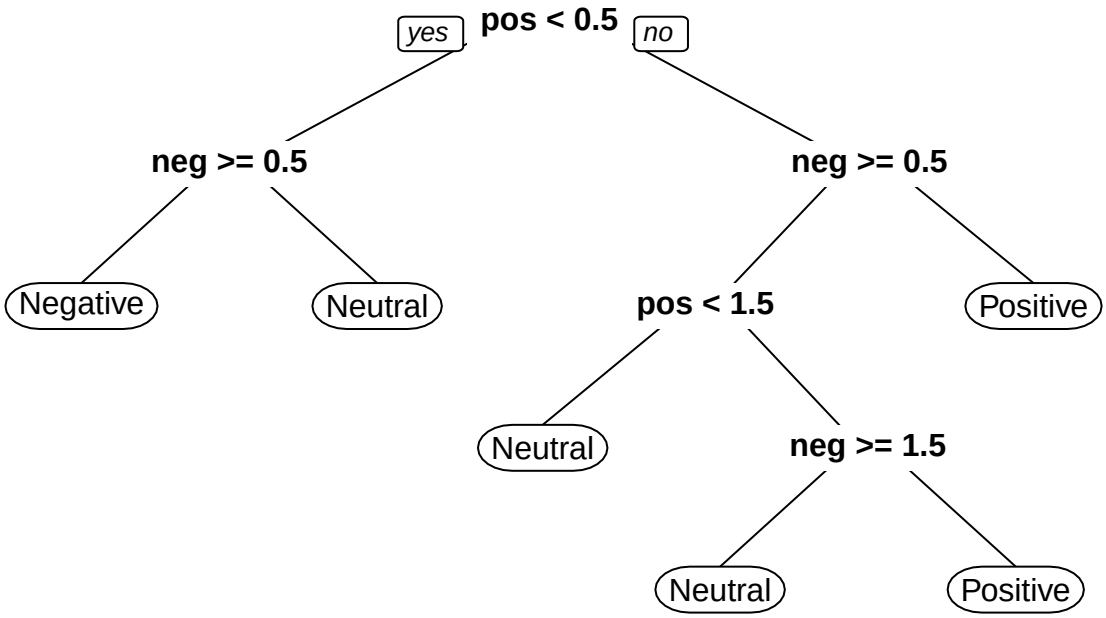
```
#Bigram Dendrogram
mdtm2a <- removeSparseTerms(mdtm2, .993)
dendro2 <- dist(t(mdtm2a), method="euclidean")
cluster <- hclust(d=dendro2, method="ward.D")
plot(cluster, hang=-1)
rect.hclust(cluster, k=2, border="orange")
```




Mentions Classification & Analysis

```
#pos and neg sentiment sourced from Hu and Liu
#header notes removed from source file, adjust file path
pos_words = read.table("/Users/digitalmarketer1977/Desktop/positive-words.txt", header = F, stringsAsFa
neg_words = read.table("/Users/digitalmarketer1977/Desktop/negative-words.txt", header = F, stringsAsFa
```


0
500
1500
2500
3500
1663
2513
0
1000
2000
3000
4000
779
Negative
Mentions Tone
0
1000
2000
3000
4000
551
3445
836
negative
neutral
positive
Tone Category




```
##           Pred
## Obs        Negative Neutral Positive
##   Negative      150       5        0
##   Neutral         0     560        0
##   Positive        0       1      251
```

Code Source: Basic Text Mining in R

Code Source: Bigram Text-Document Matrices

Reference: Automated Data Collection with R, Wiley (2015)

Code Source: Predictive Modeling

Data Source: Minqing Hu and Bing Liu. “Mining and Summarizing Customer Reviews.”