Sentiment Analysis in R

Jelena Jovanović

Get the tweets

The data set consists of 844 tweets with #apple hashtag. All tweets are labeles as either positive ('POS') or negative ('NEG'), sentiment wise.

```
# load the tweets data
tweets.data <- read.csv(file = "data/tweets_sentiment_labelled.csv",
                        colClasses = c("character", "factor"))
str(tweets.data)
```

```
## 'data.frame':    844 obs. of  2 variables:
##  $ Tweet: chr  "I have to say, Apple has by far the best customer care service I have ever received!
##  $ Lbl  : Factor w/ 2 levels "NEG","POS": 2 2 2 2 2 2 2 2 2 2 ...
```

Examine a few positive and a few negative tweets.

```
tweets.data$Tweet[tweets.data$Lbl=="POS"][1:10]
```

```
##  [1] "I have to say, Apple has by far the best customer care service I have ever received! @Apple @A
##  [2] "iOS 7 is so fricking smooth & beautiful!! #ThanxApple @Apple"
##  [3] "LOVE U @APPLE"
##  [4] "Thank you @apple, loving my new iPhone 5S!!!!!  #apple #iphone5S pic.twitter.com/XmHJCU4pcb"
##  [5] ".@apple has the best customer service. In and out with a new phone in under 10min!"
##  [6] "@apple ear pods are AMAZING! Best sound from in-ear headphones I've ever had!"
##  [7] "Omg the iPhone 5S is so cool it can read your finger print to unlock your iPhone 5S and to mak
##  [8] "the iPhone 5c is so beautiful <3 @Apple"
##  [9] "#AttributeOwnership is exactly why @apple will always be #one! #apple #marketing #marketer #bu
## [10] "Just checked out the specs on the new iOS 7...wow is all I have to say! I can't wait to get th
```

```
tweets.data$Tweet[tweets.data$Lbl=="NEG"][1:10]
```

```
##  [1] "It's important that Apple not become the developer for the world. We need people to invent the
##  [2] "@APPLE @GOOGLE #CHROME - #TEST #TEST #TEST - #iPad (Retina) #Chrome REALLY #UGLY PAGE RENDERIN
##  [3] "@fulltimebro @Apple #Hip 1/10"
##  [4] "@Microsoft getting #desperate willing to pay you $200  to trade your @apple ipads for tablets.
##  [5] "@Google 's laughing at @Apple and I am laughing at what was left of my love for @Chrome #stink
##  [6] "@Twohat007 @Apple  ppl will buy it BC lodes of ppl like it (I don't) but most of the ppl in th
##  [7] "@battalalgoos @apple"
##  [8] "A harder look at @Apple's #iPhone 5S security scanner, which allows the user's fingerprint to
##  [9] "We need emergency battery @Apple"
## [10] "I need to somehow keep headphones attached to my phone at all times. @apple"
```

Examine the distribution of class labels.

```
table(tweets.data$Lbl)
```

```
##
## NEG POS
## 541 303
```

Preprocessing of tweets' text

We use the 'tm' package that allows performing text mining related tasks.

```
## Loading required package: NLP
```

We will also load out custom R script with auxiliary functions.

```
source("text_mining_utils.R")
```

In TM terminology, *corpora* are collections of documents containing (natural language) text. In order to work with textual datasets in 'tm', we need to create a 'Corpus' instance

```
# build a corpus
tw.corpus <- Corpus(VectorSource(tweets.data$Tweet))
tw.corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 844
```

tm_map() function (from the 'tm' package) allows for performing different transformations on the corpus. List of frequently used transformations can be obtained with the 'getTransformations()' function.

```
getTransformations()
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

The purpose of all the transformations is to reduce the diversity among the words and remove words that are of low importance.

The first transformation will be to convert text to lower case.

```
tw.corpus <- tm_map(tw.corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, tolower): transformation drops
## documents
```

```
print.tweets(tw.corpus, 1, 10)
```

```
## [[1]] i have to say, apple has by far the best customer care service i have ever
## received! @apple @appstore
##
## [[2]] ios 7 is so fricking smooth & beautiful!! #thanxapple @apple
##
## [[3]] love u @apple
##
## [[4]] thank you @apple, loving my new iphone 5s!!!!!  #apple #iphone5s
## pic.twitter.com/xmhjcu4pcb
##
## [[5]] .@apple has the best customer service. in and out with a new phone in under
## 10min!
##
## [[6]] @apple ear pods are amazing! best sound from in-ear headphones i've ever had!
##
## [[7]] omg the iphone 5s is so cool it can read your finger print to unlock your
## iphone 5s and to make purchases without a passcode #apple @apple
##
## [[8]] the iphone 5c is so beautiful <3 @apple
```

```
##
## [[9]] #attributeownership is exactly why @apple will always be #one! #apple
## #marketing #marketer #business #innovation #fb
##
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo @apple
```

When processing tweets, we often remove user references completely. However, this corpus is specific - it has many (meaningful) user references; those are mostly references to Twitter accounts of various tech companies. So, we will remove only '@' sign that marks usernames (@username). This will be done using regular expressions. An excellent introduction to regular expression is The Bastards Book of Regular Expressions.

```
replaceUserRefs <- function(x) gsub("@(\\w+)", "\\1", x)
tw.corpus <- tm_map(tw.corpus, replaceUserRefs)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, replaceUserRefs): transformation
## drops documents
```

```
print.tweets( tw.corpus, from = 10, to = 20 )
```

```
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo apple
##
## [[11]] i love the new ios so much!!!!! thnx apple phillydvibing
##
## [[12]] can't wait to get my #iphone5s!!! apple
##
## [[13]] v2vista fingerprint scanner: the killer feature of iphone 5s. this is so bloody
## brilliant. apple timesnow http://toi.in/w0o-3z
##
## [[14]] interesting how so many people seem to be almost willing the demise of apple.
## what's going on? still by far my favorite #brand. fantastic
##
## [[15]] i love bnbuzz nookstudy nookbn and apple made my life so much easier this
## morning !
##
## [[16]] just watched the keynote of apple latest iphones. i just love the #iphone5s and
## #iphone5c ??? i guess, i have a christmas gift already????
##
## [[17]] my iphone wasn't calling correctly so i went to an apple store (first time)
## told them and they gave me a brand new phone. #wow
##
## [[18]] great job apple on providing best users experience #thinkauto
##
## [[19]] swapped my #galaxys2 for an #iphone4s. after one day i'd say i'm an apple
## convert!
##
## [[20]] can't wait for my #orange phone upgrade in november :-) #apple iphone 5s here i
## come ;-) orangehelpers apple
```

Remove hash (#) sign from hastags.

```
removeHash <- function(x) gsub("#([[:alnum:]]+)", "\\1", x)
tw.corpus <- tm_map(tw.corpus, removeHash)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, removeHash): transformation drops
## documents
```

```
print.tweets( tw.corpus, from = 10, to = 20 )
```

```
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo apple
##
## [[11]] i love the new ios so much!!!!! thnx apple phillydvibing
##
## [[12]] can't wait to get my iphone5s!!! apple
##
## [[13]] v2vista fingerprint scanner: the killer feature of iphone 5s. this is so bloody
## brilliant. apple timesnow http://toi.in/w0o-3z
##
## [[14]] interesting how so many people seem to be almost willing the demise of apple.
## what's going on? still by far my favorite brand. fantastic
##
## [[15]] i love bnbuzz nookstudy nookbn and apple made my life so much easier this
## morning !
##
## [[16]] just watched the keynote of apple latest iphones. i just love the iphone5s and
## iphone5c ??? i guess, i have a christmas gift already????
##
## [[17]] my iphone wasn't calling correctly so i went to an apple store (first time)
## told them and they gave me a brand new phone. wow
##
## [[18]] great job apple on providing best users experience thinkauto
##
## [[19]] swapped my galaxys2 for an iphone4s. after one day i'd say i'm an apple
## convert!
##
## [[20]] can't wait for my orange phone upgrade in november :-) apple iphone 5s here i
## come ;-) orangehelpers apple
```

Replace URLs with the "URL" term.

```
replaceURL <- function(x) gsub("(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", "URL", x)
tw.corpus <- tm_map(tw.corpus, replaceURL)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, replaceURL): transformation drops
## documents
```

```
print.tweets( tw.corpus, from = 10, to = 20 )
```

```
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo apple
##
## [[11]] i love the new ios so much!!!!! thnx apple phillydvibing
##
## [[12]] can't wait to get my iphone5s!!! apple
##
## [[13]] v2vista fingerprint scanner: the killer feature of iphone 5s. this is so bloody
## brilliant. apple timesnow URL
##
## [[14]] interesting how so many people seem to be almost willing the demise of apple.
## what's going on? still by far my favorite brand. fantastic
##
## [[15]] i love bnbuzz nookstudy nookbn and apple made my life so much easier this
```

```
## morning !
##
## [[16]] just watched the keynote of apple latest iphones. i just love the iphone5s and
## iphone5c ??? i guess, i have a christmas gift already????
##
## [[17]] my iphone wasn't calling correctly so i went to an apple store (first time)
## told them and they gave me a brand new phone. wow
##
## [[18]] great job apple on providing best users experience thinkauto
##
## [[19]] swapped my galaxys2 for an iphone4s. after one day i'd say i'm an apple
## convert!
##
## [[20]] can't wait for my orange phone upgrade in november :-) apple iphone 5s here i
## come ;-) orangehelpers apple
```

Replace links to pictures (e.g. pic.twitter.com/lbu9diufrf) with 'TW_PIC'.

```
replaceTWPic <- function(x) gsub("pic\\.twitter\\.com/[[:alnum:]]+",
                                 "TW_PIC", x)
tw.corpus <- tm_map(tw.corpus, replaceTWPic)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, replaceTWPic): transformation
## drops documents
```

```
print.tweets( tw.corpus, from = 10, to = 20 )
```

```
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo apple
##
## [[11]] i love the new ios so much!!!!! thnx apple phillydvibing
##
## [[12]] can't wait to get my iphone5s!!! apple
##
## [[13]] v2vista fingerprint scanner: the killer feature of iphone 5s. this is so bloody
## brilliant. apple timesnow URL
##
## [[14]] interesting how so many people seem to be almost willing the demise of apple.
## what's going on? still by far my favorite brand. fantastic
##
## [[15]] i love bnbuzz nookstudy nookbn and apple made my life so much easier this
## morning !
##
## [[16]] just watched the keynote of apple latest iphones. i just love the iphone5s and
## iphone5c ??? i guess, i have a christmas gift already????
##
## [[17]] my iphone wasn't calling correctly so i went to an apple store (first time)
## told them and they gave me a brand new phone. wow
##
## [[18]] great job apple on providing best users experience thinkauto
##
## [[19]] swapped my galaxys2 for an iphone4s. after one day i'd say i'm an apple
## convert!
##
## [[20]] can't wait for my orange phone upgrade in november :-) apple iphone 5s here i
## come ;-) orangehelpers apple
```

Replace :-), :), ;-), :D, :o with the "POS_SMILEY" term.

```
replaceHappySmiley <- function(x) gsub("[:|;](-?)[\\)|o|O|D]",
                                       "POS_SMILEY", x)
tw.corpus <- tm_map(tw.corpus, replaceHappySmiley)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, replaceHappySmiley):
## transformation drops documents
```

```
print.tweets( tw.corpus, from = 10, to = 20 )
```

```
## [[10]] just checked out the specs on the new ios 7...wow is all i have to say! i can't
## wait to get the new update ?? bravo apple
##
## [[11]] i love the new ios so much!!!!! thnx apple phillydvibing
##
## [[12]] can't wait to get my iphone5s!!! apple
##
## [[13]] v2vista fingerprint scanner: the killer feature of iphone 5s. this is so bloody
## brilliant. apple timesnow URL
##
## [[14]] interesting how so many people seem to be almost willing the demise of apple.
## what's going on? still by far my favorite brand. fantastic
##
## [[15]] i love bnbuzz nookstudy nookbn and apple made my life so much easier this
## morning !
##
## [[16]] just watched the keynote of apple latest iphones. i just love the iphone5s and
## iphone5c ??? i guess, i have a christmas gift already????
##
## [[17]] my iphone wasn't calling correctly so i went to an apple store (first time)
## told them and they gave me a brand new phone. wow
##
## [[18]] great job apple on providing best users experience thinkauto
##
## [[19]] swapped my galaxys2 for an iphone4s. after one day i'd say i'm an apple
## convert!
##
## [[20]] can't wait for my orange phone upgrade in november POS_SMILEY apple iphone 5s
## here i come POS_SMILEY orangehelpers apple
```

Replace :(, :-(, :/, >:(, >:O with the "NEG_SMILEY" term.

```
replaceSadSmiley <- function(x) gsub("((>?):(-?)[\\(|/|O|o])",
                                     "NEG_SMILEY", x)
tw.corpus <- tm_map(tw.corpus, replaceSadSmiley)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, replaceSadSmiley): transformation
## drops documents
```

Remove stopwords from the corpus. First, examine the tm's set of stopwords.

```
stopwords('english')[100:120]
```

```
##  [1] "that's"  "who's"   "what's"  "here's"  "there's" "when's"  "where's"
##  [8] "why's"   "how's"   "a"       "an"      "the"     "and"     "but"
## [15] "if"      "or"      "because" "as"      "until"   "while"   "of"
```

Add a few extra ('corpus-specific') stop words (e.g. "apple", "rt") to the 'general' stopwords for the English language.

```
tw.stopwords <- c(stopwords('english'), "apple", "rt")
tw.corpus <- tm_map(tw.corpus, removeWords, tw.stopwords)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, removeWords, tw.stopwords):
## transformation drops documents
```

```
print.tweets( tw.corpus, from = 20, to = 30 )
```

```
## [[20]] wait orange phone upgrade november POS_SMILEY iphone 5s come POS_SMILEY
## orangehelpers
##
## [[21]] colored iphone!  new 5c iphone comes colors.  love !!  wait till can get one!
## iphone5c
##
## [[22]] whether fan , iphone 5c video worth watching, lots reasons URL
##
## [[23]] impressive features iphone 5s - fingerprint recognition, now thats
## impressive!!! POS_SMILEY
##
## [[24]] kuqogroup jimmykimmel blackberry unbelievably awesome!!!!! clickitandlickit
##
## [[25]] earpods amazing, thanks
##
## [[26]] luv iphone 5s luv champagne colour fingerprint reader
##
## [[27]] used tv explainevrythng demonstrate student understanding factors!  awesome!
##
## [[28]] brody_knibbs haha mint arent ! top company
##
## [[29]] back yaw, thanks !!!!!
##
## [[30]] found new way work today courtesy iphone. way go, maps!  first time anyone's
## said ?
```

Remove punctuation.

```
tw.corpus <- tm_map(tw.corpus, removePunctuation,
                    preserve_intra_word_contractions = TRUE,
                    preserve_intra_word_dashes = TRUE)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, removePunctuation,
## preserve_intra_word_contractions = TRUE, : transformation drops documents
```

```
print.tweets( tw.corpus, from = 20, to = 30 )
```

```
## [[20]] wait orange phone upgrade november POSSMILEY iphone 5s come POSSMILEY
## orangehelpers
##
## [[21]] colored iphone new 5c iphone comes colors love wait till can get one iphone5c
##
## [[22]] whether fan iphone 5c video worth watching lots reasons URL
##
## [[23]] impressive features iphone 5s fingerprint recognition now thats impressive
## POSSMILEY
```

```
##
## [[24]] kuqogroup jimmykimmel blackberry unbelievably awesome clickitandlickit
##
## [[25]] earpods amazing thanks
##
## [[26]] luv iphone 5s luv champagne colour fingerprint reader
##
## [[27]] used tv explainevrythng demonstrate student understanding factors awesome
##
## [[28]] brodyknibbs haha mint arent top company
##
## [[29]] back yaw thanks
##
## [[30]] found new way work today courtesy iphone way go maps first time anyone's said
```

Remove stand-alone numbers (but not numbers in e.g. iphone7 or g3)

```
removeStandAloneNumbers <- function(x) gsub(" \\d+ ", "", x)
tw.corpus <- tm_map(tw.corpus, removeStandAloneNumbers)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, removeStandAloneNumbers):
## transformation drops documents
```

```
print.tweets( tw.corpus, from = 20, to = 30 )
```

```
## [[20]] wait orange phone upgrade november POSSMILEY iphone 5s come POSSMILEY
## orangehelpers
##
## [[21]] colored iphone new 5c iphone comes colors love wait till can get one iphone5c
##
## [[22]] whether fan iphone 5c video worth watching lots reasons URL
##
## [[23]] impressive features iphone 5s fingerprint recognition now thats impressive
## POSSMILEY
##
## [[24]] kuqogroup jimmykimmel blackberry unbelievably awesome clickitandlickit
##
## [[25]] earpods amazing thanks
##
## [[26]] luv iphone 5s luv champagne colour fingerprint reader
##
## [[27]] used tv explainevrythng demonstrate student understanding factors awesome
##
## [[28]] brodyknibbs haha mint arent top company
##
## [[29]] back yaw thanks
##
## [[30]] found new way work today courtesy iphone way go maps first time anyone's said
```

Strip whitespace.

```
tw.corpus <- tm_map(tw.corpus, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, stripWhitespace): transformation
## drops documents
```

Do word stemming using the Snowball stemmer. To use the Snowball stemmer in R, we need the 'SnowballC' package.

```
## Warning: package 'SnowballC' was built under R version 3.5.2
```

Since we might later want to have words in their 'regular' form, we will keep a copy of the corpus before stemming it.

```
tw.corpus.backup <- tw.corpus
```

Now, do the stemming.

```
tw.corpus <- tm_map(tw.corpus, stemDocument, language = "english")
```

```
## Warning in tm_map.SimpleCorpus(tw.corpus, stemDocument, language =
## "english"): transformation drops documents
```

```
print.tweets( tw.corpus, from = 20, to = 30)
```

```
## [[20]] wait orang phone upgrad novemb POSSMILEi iphon 5s come POSSMILEi orangehelp
##
## [[21]] color iphon new 5c iphon come color love wait till can get one iphone5c
##
## [[22]] whether fan iphon 5c video worth watch lot reason URL
##
## [[23]] impress featur iphon 5s fingerprint recognit now that impress POSSMILEi
##
## [[24]] kuqogroup jimmykimmel blackberri unbeliev awesom clickitandlickit
##
## [[25]] earpod amaz thank
##
## [[26]] luv iphon 5s luv champagn colour fingerprint reader
##
## [[27]] use tv explainevrythng demonstr student understand factor awesom
##
## [[28]] brodyknibb haha mint arent top compani
##
## [[29]] back yaw thank
##
## [[30]] found new way work today courtesi iphon way go map first time anyon said
```

Building a Document-Term matrix

Document Term Matrix (DTM) represents the relationship between terms and documents, where each row stands for a document and each column for a term, and entry is the weight of the term in the corresponding document.

```
min.freq <- round(0.005*length(tw.corpus))
max.freq <- round(0.95*length(tw.corpus))
dtm <- DocumentTermMatrix(tw.corpus,
                          control = list(bounds = list(global = c(min.freq,max.freq)),
                                         wordLengths = c(2,16), # the restriction on the word length
                                         weighting = weightTf)) # term freq. weighting scheme
```

Note: the 'global' parameter is altered to require a word to appear in at least ~0.5% and at most in 95% of tweets to be included in the matrix. Check the documentation of the *TermDocumentMatrix* function for other useful control parameters.

Let's examine the built DTM matrix.

```
inspect(dtm)
```

```
## <<DocumentTermMatrix (documents: 844, terms: 391)>>
## Non-/sparse entries: 4178/325826
## Sparsity           : 99%
## Maximal term length: 15
## Weighting          : term frequency (tf)
## Sample             :
##      Terms
## Docs  5c 5s freak get iphon make new phone twpic url
##   21   1  0     0   1     2    0   1     0     0   0
##   320  1  0     0   0     1    0   0     1     0   0
##   335  0  0     0   0     0    0   0     0     1   0
##   394  0  0     0   0     0    0   0     0     1   0
##   407  0  0     0   0     0    0   0     0     1   0
##   430  0  0     0   0     0    0   0     0     0   0
##   563  0  0     0   0     0    0   0     0     0   0
##   606  0  0     0   0     0    0   0     0     0   0
##   81   0  0     0   0     1    0   1     0     0   0
##   83   0  0     0   0     1    0   1     0     0   0
```

We have very sparse DTM matrix; so, we should better reduce the sparsity by removing overly sparse terms.

```
dtm.trimmed <- removeSparseTerms(dtm, sparse = 0.9875)
inspect(dtm.trimmed)
```

```
## <<DocumentTermMatrix (documents: 844, terms: 108)>>
## Non-/sparse entries: 2542/88610
## Sparsity           : 97%
## Maximal term length: 15
## Weighting          : term frequency (tf)
## Sample             :
##      Terms
## Docs  5c 5s freak get iphon make new phone twpic url
##   143  1  1     0   0     1    0   1     0     0   1
##   21   1  0     0   1     2    0   1     0     0   0
##   255  0  0     0   0     0    0   0     0     0   0
##   299  0  0     0   0     0    0   0     0     0   0
##   302  0  0     0   0     0    0   0     0     0   0
##   320  1  0     0   0     1    0   0     1     0   0
##   335  0  0     0   0     0    0   0     0     1   0
##   430  0  0     0   0     0    0   0     0     0   0
##   468  0  0     0   1     0    0   1     1     0   0
##   532  0  0     0   0     1    1   1     0     0   0
```

Examine the resulting DTM matrix. First, check the terms that appear at least 20 times in the whole corpus.

```
findFreqTerms(dtm.trimmed, lowfreq = 20)
```

```
##  [1] "io"          "love"        "5s"          "iphon"       "iphone5"
##  [6] "new"         "thank"       "twpic"       "phone"       "can"
## [11] "make"        "5c"          "one"         "will"        "get"
## [16] "just"        "updat"       "fingerprint" "url"         "go"
## [21] "iphone5c"    "store"       "time"        "come"        "now"
## [26] "use"         "back"        "app"         "think"       "ipad"
## [31] "freak"       "like"        "want"        "ios7"        "itun"
```

```
## [36] "look"        "need"        "ipod"        "realli"
```

We can also inspect the frequency of accurance of all the terms.

```
head(colSums(as.matrix(dtm)))
```

```
## appstor    best    care  custom    ever     say
##       6      12      11      12       9      19
```

It is better if they are sorted.

```
head(sort(colSums(as.matrix(dtm)), decreasing = T), n = 10)
```

```
## iphon   url   new phone   get    5s twpic freak  make    5c
##   215   115    91    84    65    63    59    55    50    45
```

Classifying tweets using Naive Bayes method

Since we want to use DTM for classification purposes, we need to transform it into a data frame that can be passed to a function for building a classifier.

```
features.final <- as.data.frame(as.matrix(dtm.trimmed))
str(features.final, list.len = 50)
```

```
## 'data.frame':    844 obs. of  108 variables:
##  $ best           : num  1 0 0 0 1 1 0 0 0 0 ...
##  $ care           : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ custom         : num  1 0 0 0 1 0 0 0 0 0 ...
##  $ say            : num  1 0 0 0 0 0 0 0 0 1 ...
##  $ servic         : num  1 0 0 0 1 0 0 0 0 0 ...
##  $ io             : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ love           : num  0 0 1 1 0 0 0 0 0 0 ...
##  $ 5s             : num  0 0 0 1 0 0 2 0 0 0 ...
##  $ iphon          : num  0 0 0 1 0 0 2 1 0 0 ...
##  $ iphone5        : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ new            : num  0 0 0 1 1 0 0 0 0 2 ...
##  $ thank          : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ twpic          : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ phone          : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ can            : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ make           : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ print          : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ 5c             : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ innov          : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ market         : num  0 0 0 0 0 0 0 0 2 0 ...
##  $ one            : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ will           : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ get            : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ just           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ updat          : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ wait           : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ fingerprint    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ url            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ go             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ peopl          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ still          : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ iphone5c       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ watch          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ store          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ time           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ come           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ possmilei      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ color          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ now            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ use            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ compani        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ back           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ way            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ work           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ dear           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ good           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ app            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ product        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ android        : num  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
```

Add the class label.

```
features.final$CLASS_LBL <- tweets.data$Lbl
colnames(features.final)
```

```
##   [1] "best"           "care"           "custom"
##   [4] "say"            "servic"         "io"
##   [7] "love"           "5s"             "iphon"
##  [10] "iphone5"        "new"            "thank"
##  [13] "twpic"          "phone"          "can"
##  [16] "make"           "print"          "5c"
##  [19] "innov"          "market"         "one"
##  [22] "will"           "get"            "just"
##  [25] "updat"          "wait"           "fingerprint"
##  [28] "url"            "go"             "peopl"
##  [31] "still"          "iphone5c"       "watch"
##  [34] "store"          "time"           "day"
##  [37] "come"           "possmilei"      "color"
##  [40] "now"            "use"            "compani"
##  [43] "back"           "way"            "work"
##  [46] "dear"           "good"           "app"
##  [49] "product"        "android"        "think"
##  [52] "help"           "ipad"           "nokia"
##  [55] "free"           "give"           "macbook"
##  [58] "well"           "gold"           "screen"
##  [61] "freak"          "next"           "got"
##  [64] "let"            "stop"           "better"
##  [67] "like"           "news"           "pleas"
##  [70] "samsung"        "us"             "want"
##  [73] "batteri"        "thing"          "ios7"
##  [76] "know"           "mobil"          "everi"
##  [79] "microsoft"      "announc"        "itun"
##  [82] "buy"            "yet"            "sure"
##  [85] "releas"         "take"           "look"
```

```
##  [88] "devic"           "right"           "need"
##  [91] "googl"           "hey"             "tri"
##  [94] "price"           "twitter"         "even"
##  [97] "fix"             "ipod"            "ipodplayerpromo"
## [100] "lol"             "realli"          "stuff"
## [103] "charger"         "hate"            "facebook"
## [106] "cheap"           "wtf"             "amazon"
## [109] "CLASS_LBL"
```

Split the data into training and test sets.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
set.seed(1212)
train.indices <- createDataPartition(y = features.final$CLASS_LBL,
                                     p = 0.85,
                                     list = FALSE)
train.data <- features.final[train.indices,]
test.data <- features.final[-train.indices,]
```

Build NB classifier using all the features.

Since each feature (word) has numerous zero values, when fitting the model, we include the Laplace smoothing to avoid zero values of conditional probabilities.

Let's make the predictions.

```
nb1.pred <- predict(nb1, newdata = test.data, type = "class")
```

Create confusion matrix.

```
cm1 <- table(true = test.data$CLASS_LBL, predicted = nb1.pred)
cm1
```

```
##      predicted
## true  NEG POS
##   NEG  26  55
##   POS   5  40
```

Evaluate the model.

```
eval1 <- compute.eval.measures(cm1)
eval1
```

```
##  accuracy precision    recall        F1
##    0.5238    0.8387    0.3210    0.4643
```

Try to improve the performance by using a different probability threshold (instead of the default one of 0.5). To that end, we'll make use of ROC curves.

```
## Warning: package 'pROC' was built under R version 3.5.2
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

Get predictions as probabilities.

```
nb1.pred.prob <- predict(nb1, newdata = test.data, type = "raw")
nb1.pred.prob[1:10,]
```

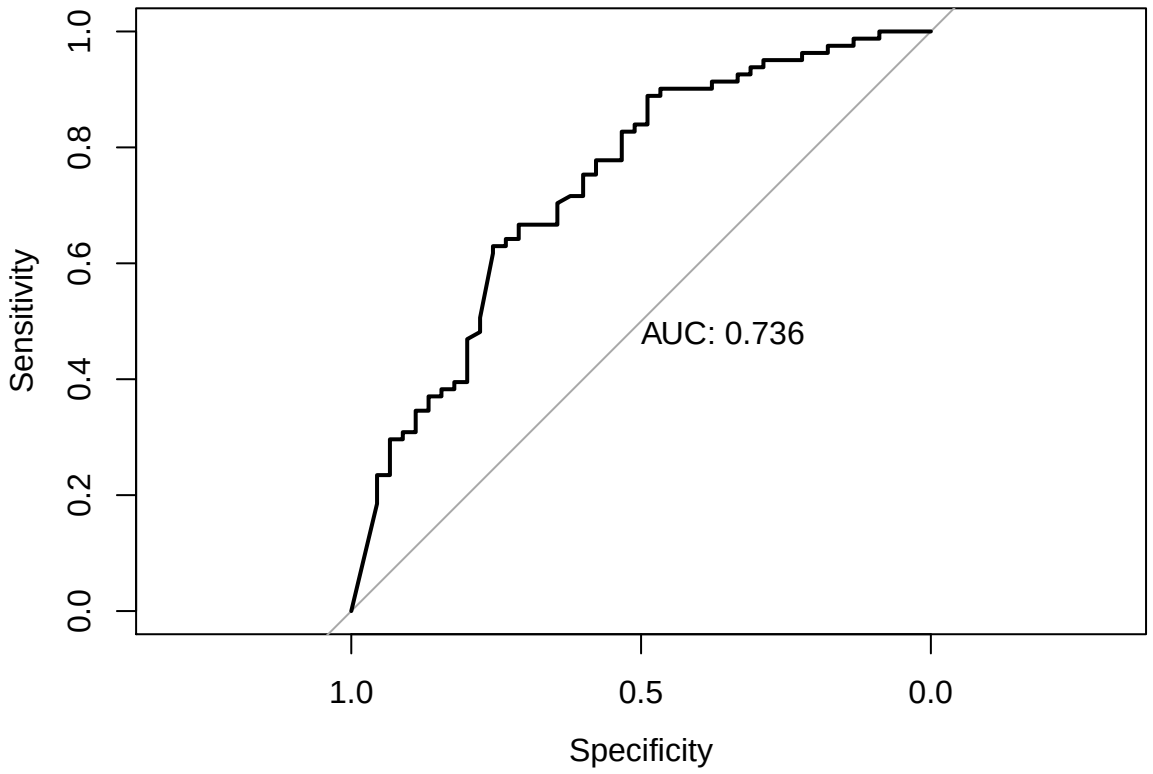
```
##                  NEG          POS
##  [1,]  2.877585e-25 1.000000e+00
##  [2,]  6.245556e-35 1.000000e+00
##  [3,]  1.491943e-23 1.000000e+00
##  [4,]  1.000000e+00 7.116476e-28
##  [5,]  8.693614e-36 1.000000e+00
##  [6,] 2.550802e-197 1.000000e+00
##  [7,]  9.886615e-12 1.000000e+00
##  [8,]  1.226974e-10 1.000000e+00
##  [9,]  3.617684e-25 1.000000e+00
## [10,]  2.499523e-21 1.000000e+00
```

Compute the stats for the ROC curve.

```
nb.roc <- roc(response = as.numeric(test.data$CLASS_LBL),
              predictor = nb1.pred.prob[,1], # probabilities of the 'positive' class
              levels = c(2,1)) # define the order of levels corresponding to the negative (controls)
                               # and positive (cases) class
```

Plot the curve:

```
plot.roc(x = nb.roc,
         print.auc = TRUE) # print AUC measure
```



Get the evaluation measures and the threshold for the local maxima of the ROC curve.

```
nb2.coords <- coords(roc = nb.roc,
                     x = "local maximas",
                     ret = c("accuracy", "sensitivity", "specificity", "thr"))
nb2.coords
```

```
##             local maximas local maximas local maximas local maximas
## accuracy     6.746032e-01  6.825397e-01  6.904762e-01  6.984127e-01
## sensitivity  1.000000e+00  9.876543e-01  9.753086e-01  9.629630e-01
## specificity  8.888889e-02  1.333333e-01  1.777778e-01  2.222222e-01
## threshold    3.322707e-63  7.792052e-58  3.834678e-47  2.612230e-39
##             local maximas local maximas local maximas local maximas
## accuracy     7.142857e-01  7.142857e-01  7.142857e-01  7.222222e-01
## sensitivity  9.506173e-01  9.382716e-01  9.259259e-01  9.135802e-01
## specificity  2.888889e-01  3.111111e-01  3.333333e-01  3.777778e-01
## threshold    1.353712e-32  2.267413e-31  2.075719e-29  1.273441e-27
##             local maximas local maximas local maximas local maximas
## accuracy     7.460317e-01  7.460317e-01  7.222222e-01  7.222222e-01
## sensitivity  9.012346e-01  8.888889e-01  8.395062e-01  8.271605e-01
## specificity  4.666667e-01  4.888889e-01  5.111111e-01  5.333333e-01
## threshold    1.587395e-24  2.016691e-23  6.310090e-22  5.403557e-21
##             local maximas local maximas local maximas local maximas
## accuracy     7.063492e-01  6.984127e-01  6.825397e-01  6.825397e-01
## sensitivity  7.777778e-01  7.530864e-01  7.160494e-01  7.037037e-01
## specificity  5.777778e-01  6.000000e-01  6.222222e-01  6.444444e-01
## threshold    1.474845e-17  3.818925e-17  2.379819e-16  3.000622e-16
##             local maximas local maximas local maximas local maximas
## accuracy     6.825397e-01  6.746032e-01  6.746032e-01  6.031746e-01
## sensitivity  6.666667e-01  6.419753e-01  6.296296e-01  5.061728e-01
```

```
## specificity  7.111111e-01  7.333333e-01  7.555556e-01  7.777778e-01
## threshold    2.590452e-14  4.533540e-14  9.470877e-14  1.720116e-13
##             local maximas local maximas local maximas local maximas
## accuracy     5.873016e-01  5.476190e-01  5.476190e-01  5.476190e-01
## sensitivity  4.691358e-01  3.950617e-01  3.827160e-01  3.703704e-01
## specificity  8.000000e-01  8.222222e-01  8.444444e-01  8.666667e-01
## threshold    6.763594e-13  1.132039e-11  1.809326e-08  4.680649e-06
##             local maximas local maximas local maximas local maximas
## accuracy     0.5396825397     0.5238095     0.5238095     0.4920635
## sensitivity  0.3456790123     0.3086420     0.2962963     0.2345679
## specificity  0.8888888889     0.9111111     0.9333333     0.9555556
## threshold    0.0002830784     0.9999914     0.9999993     1.0000000
##             local maximas
## accuracy        0.3571429
## sensitivity     0.0000000
## specificity     1.0000000
## threshold             Inf
```

As we want to assure that the company (Apple) will not miss tweets with negative sentiment, and since we set the negative sentiment as our positive class (i.e. class in our focus), we should look for a probability threshold that will maximize sensitivity (i.e., true positive rate). Still, we should keep the other measures (accuracy, specificity) at a decent level.

The local maximum that corresponds to the 9th column looks like a good candidate. Let's examine it more closely:

```
nb2.coords[,9]
```

```
##     accuracy  sensitivity  specificity    threshold
## 7.460317e-01 9.012346e-01 4.666667e-01 1.587395e-24
```

Select the threshold that corresponds to the 9th local maximum:

```
opt.threshold <- nb2.coords[4,9]
```

Assign class labels based on the chosen threshold:

```
nb1.pred.opt <- ifelse(test = nb1.pred.prob[,1] > opt.threshold,
                       yes = "NEG", no = "POS")
nb1.pred.opt <- as.factor(nb1.pred.opt)
```

Create a confusion matrix based on the newly assigned class labels:

```
cm.opt <- table(actual = test.data$CLASS_LBL, predicted = nb1.pred.opt)
cm.opt
```

```
##       predicted
## actual NEG POS
##    NEG  73   8
##    POS  24  21
```

Examine evaluation measures:

```
eval2 <- compute.eval.measures(cm.opt)
eval2
```

```
##  accuracy precision    recall        F1
##    0.7460    0.7526    0.9012    0.8202
```

Compare evaluation measures:

```r
data.frame(rbind(eval1, eval2), row.names = c("default_threshold", "ROC_based_theshold"))
```

```
##                     accuracy precision recall     F1
## default_threshold     0.5238    0.8387 0.3210 0.4643
## ROC_based_theshold    0.7460    0.7526 0.9012 0.8202
```

Acknowledgements

This example is partially based on *Chapter 10* of the R and Data Mining