Text Processing with tm

Karen Mazidi

Amazon review data from Kaggle: https://www.kaggle.com/bittlingmayer/amazonreviews

Load data and package tm

```
library(tm)
```

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

```
reviews <- read.csv("data/reviews.csv", header=TRUE, stringsAsFactors=F)
```

counts for ratings

Rating = 1 means 1 or 2 stars; Rate=2 mean 4 or 5 stars; 3 stars were ignored

About 46% were low and 54% high.

```
low_ratings <- nrow(reviews[reviews$Rating == 1,])
high_ratings <- nrow(reviews[reviews$Rating == 2,])
```

Make a simple corpus

```
am_corpus <- Corpus(VectorSource(reviews$Review))
inspect(am_corpus[1])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
## [1] Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your min
```

Preprocess

```
am_clean <- tm_map(am_corpus, content_transformer(tolower))
am_clean <- tm_map(am_clean, removeNumbers)
am_clean <- tm_map(am_clean, removePunctuation)
am_clean <- tm_map(am_clean, removeWords, stopwords())
am_clean <- tm_map(am_clean, stripWhitespace)
```

Make Document Term Matrix

```
am_dtm <- DocumentTermMatrix(am_clean)
am_dtm
```

```
## <<DocumentTermMatrix (documents: 4139, terms: 21669)>>
## Non-/sparse entries: 143646/89544345
## Sparsity           : 100%
## Maximal term length: 106
## Weighting          : term frequency (tf)
```

Divide into test and train

```
set.seed(1234)
i <- sample(nrow(reviews), 0.75*nrow(reviews), replace=FALSE)
# labels
train_labels <- reviews[i, 1]
test_labels <- reviews[-i, 1]
# data
train <- am_clean[i]
test <- am_clean[-i]
```

Ignore rare words

```
freq_words <- findFreqTerms(am_dtm, 5)
train <- DocumentTermMatrix(train, control=list(dictionary=freq_words))
test <- DocumentTermMatrix(test, control=list(dictionary=freq_words))
inspect(train[50:55, 200:209])
```

```
## <<DocumentTermMatrix (documents: 6, terms: 10)>>
## Non-/sparse entries: 3/57
## Sparsity           : 95%
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs bought send someone spent term thing unless want worst written
##   50      0    0       0     0    0     0      0    0     0       0
##   51      0    1       0     0    0     0      0    0     0       0
##   52      0    2       0     0    0     0      0    0     0       0
##   53      0    1       0     0    0     0      0    0     0       0
##   54      0    0       0     0    0     0      0    0     0       0
##   55      0    0       0     0    0     0      0    0     0       0
```

Create binary matrix

```
convert_count <- function(x) {
  y <- ifelse(x>0, 1, 0)
  y <- factor(y)
  y
}

train <- apply(train, 2, convert_count)
test <- apply(test, 2, convert_count)
```

Naive Bayes

```
library(e1071)
nb1 <- naiveBayes(train, factor(train_labels))
```

Evaluate on test data

We only got 54% accuracy. Further improvements might be made by removing proper nouns and stemming, but we leave that for future work.

```
pred <- predict(nb1, newdata=test)
table(pred, test_labels)
```

```
##     test_labels
## pred   1   2
##    1 303 233
##    2 239 260
```

```
mean(pred == test_labels)
```

```
## [1] 0.5439614
```