Text analytics with Amazon reviews data

Amazon Reviews

Data format: product/productId: B001E4KFG0 review/userId: A3SGXH7AUHU8GW review/profileName: delmartian review/helpfulness: 1/1 review/score: 5.0 review/time: 1303862400 review/summary: Good Quality Dog Food review/text: I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.

URL: http://snap.stanford.edu/data/web-FineFoods.html

Citation: J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.

```r
#package
library(readr)
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(knitr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(GGally)
```

```
## Loading required package: ggplot2
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library("psych")
```

```
## Warning: package 'psych' was built under R version 4.0.2
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(ggplot2)
library(stringr)
library("ggExtra")
```

```
## Warning: package 'ggExtra' was built under R version 4.0.2
```

```r
library(psych)
library(dplyr)
library(tidyr)
library(purrr)
library(readr)
#install.packages("topicmodels")
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.0.2
```

```r
library(widyr)
```

```
## Warning: package 'widyr' was built under R version 4.0.2
```

```r
library(ggraph)
```

```
## Warning: package 'ggraph' was built under R version 4.0.2
```

```r
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
## The following object is masked from 'package:tidyr':
##
##     crossing

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.0.2

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 4.0.2
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.0.2

## Loading required package: RColorBrewer
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library("ldatuning")
```

```
## Warning: package 'ldatuning' was built under R version 4.0.2
```

```r
#Stemming
#https://github.com/juliasilge/tidytext/issues/17
library(SnowballC)
```

Stemming

```r
wordStem(c('taste','tasted','tasteful','tastefully','tastes','tasting'), language = "english")
```

```
## [1] "tast" "tast" "tast" "tast" "tast" "tast"
```

load data

```r
amazon_reviews_full <- read_tsv("foods.txt",
                                col_names = FALSE
                                #delim = "",
                                #n_max = 24
                                )
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character()
## )
```

```r
View(head(amazon_reviews_full, 10))

amazon_reviews <- amazon_reviews_full %>%
                  #head(1000) %>%
                  separate(col = X1,
                           into = c("head", "value"),
                           sep = ": ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 28437 rows [48, 576,
## 904, 944, 1176, 1272, 1496, 1576, 1776, 1856, 1928, 2112, 2120, 2368, 2544,
## 3160, 3320, 3391, 3439, 3528, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 7 rows [753580,
## 1416685, 1521590, 2270671, 2809464, 3018833, 4306898].
```

```r
                  #mutate(seq_num = row_number())

head(amazon_reviews)
```

```
## # A tibble: 6 x 2
##   head                value
##   <chr>               <chr>
## 1 product/productId  B001E4KFG0
## 2 review/userId      A3SGXH7AUHU8GW
## 3 review/profileName delmartian
## 4 review/helpfulness 1/1
## 5 review/score       5.0
## 6 review/time        1303862400
```

```
review <- data.frame(rev_id = 1:nrow(filter(amazon_reviews, head == "product/productId")),
                     productId = filter(amazon_reviews, head == "product/productId")$value,
                     userId    = filter(amazon_reviews, head == "review/userId")$value,
                     rating     = as.numeric(filter(amazon_reviews, head == "review/score")$value),
                     text      = filter(amazon_reviews, head == "review/text")$value,
                     time      = as.numeric(filter(amazon_reviews, head == "review/time")$value),
                     stringsAsFactors = FALSE)

View(head(review,10))
```

Tidy text

Clean up text so that we can get it ready for analysis

```
#Remove stop words

tidy_amzn <- review %>%
              unnest_tokens(word, text) %>%
              anti_join(stop_words) %>%
              filter(word != "br") %>% #HTML tag <br /><br /> results in the word "br"
              mutate(word = wordStem(word))
```

```
## Joining, by = "word"
```

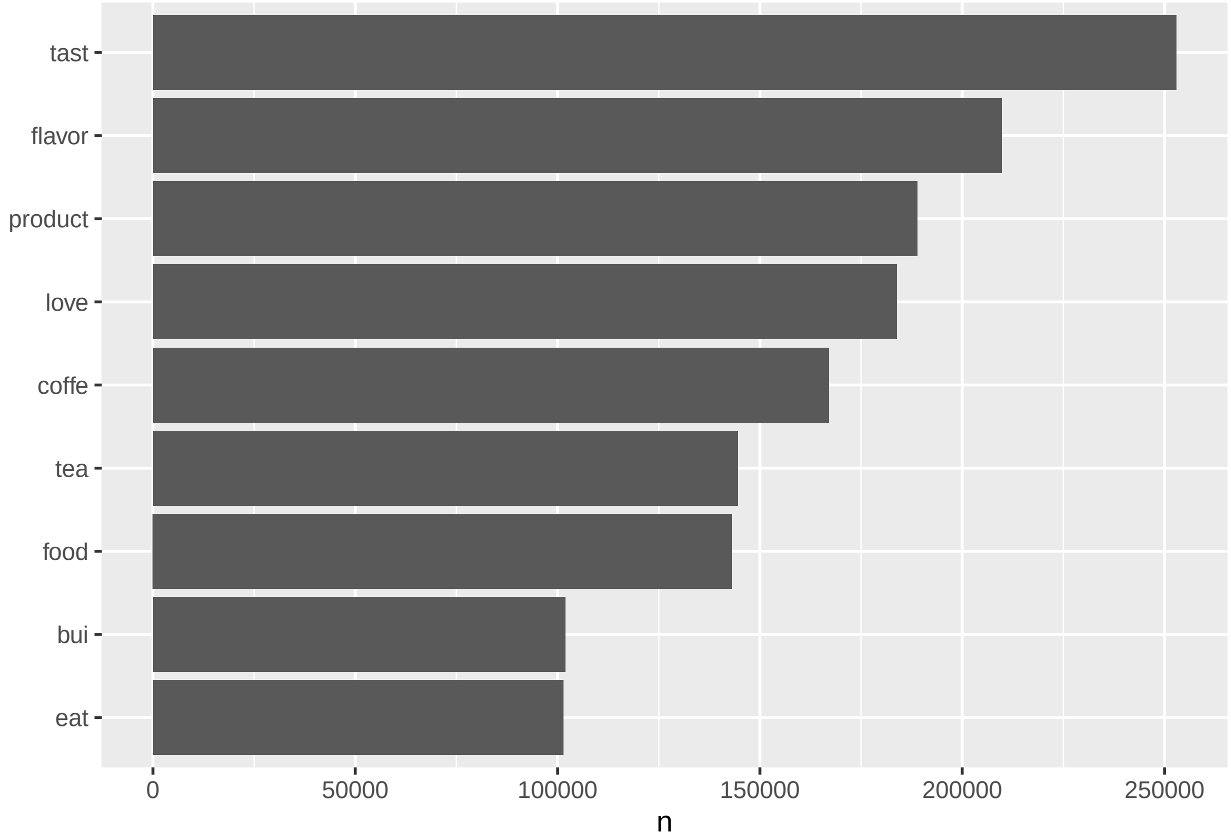
```
View(head(tidy_amzn,10))
```

Word count analysis

```
tidy_amzn %>%
count(word, sort = TRUE) %>%
  slice(1:5)
```

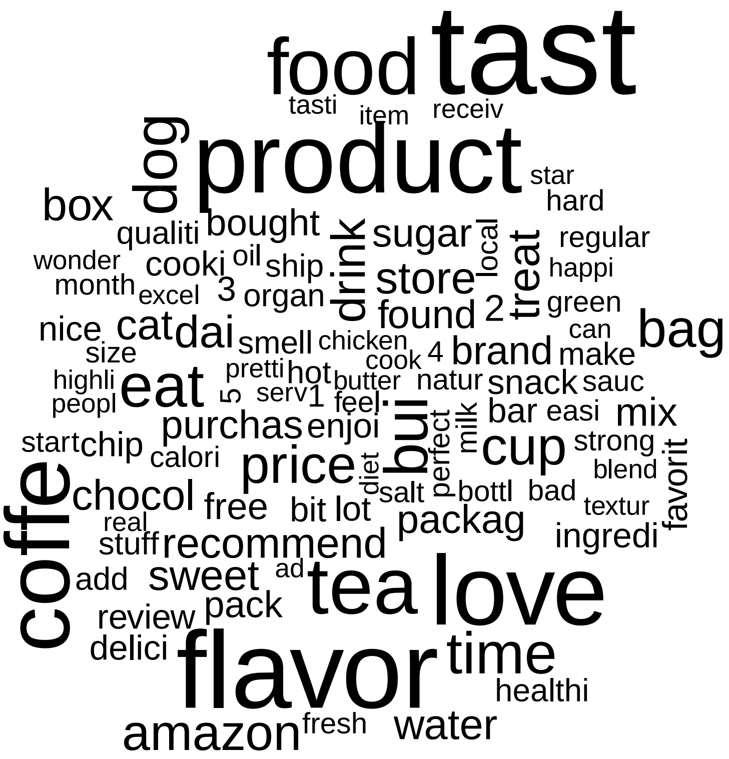
```
##      word      n
## 1    tast 252881
## 2  flavor 209758
## 3 product 188905
## 4    love 183847
## 5   coffe 166978
```

```
tidy_amzn %>%
  count(word, sort = TRUE) %>%
  filter(n > 100000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```



## Word cloud

```
tidy_amzn %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



Sentiment analysis

The AFINN lexicon assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment.

```
tidy_amzn_sentiment <- tidy_amzn %>%
                          inner_join(get_sentiments("afinn"), by = "word")
View(head(tidy_amzn_sentiment,10))
```

get average sentiment score for each productId to plot rating vs. avg_score

```
tidy_amzn_sentiment_prod <- tidy_amzn_sentiment %>%
                            group_by(productId) %>%
                              summarise(avg_score=mean(value),
                                        sum_score=sum(value),
                                        avg_rating = mean(rating))
```

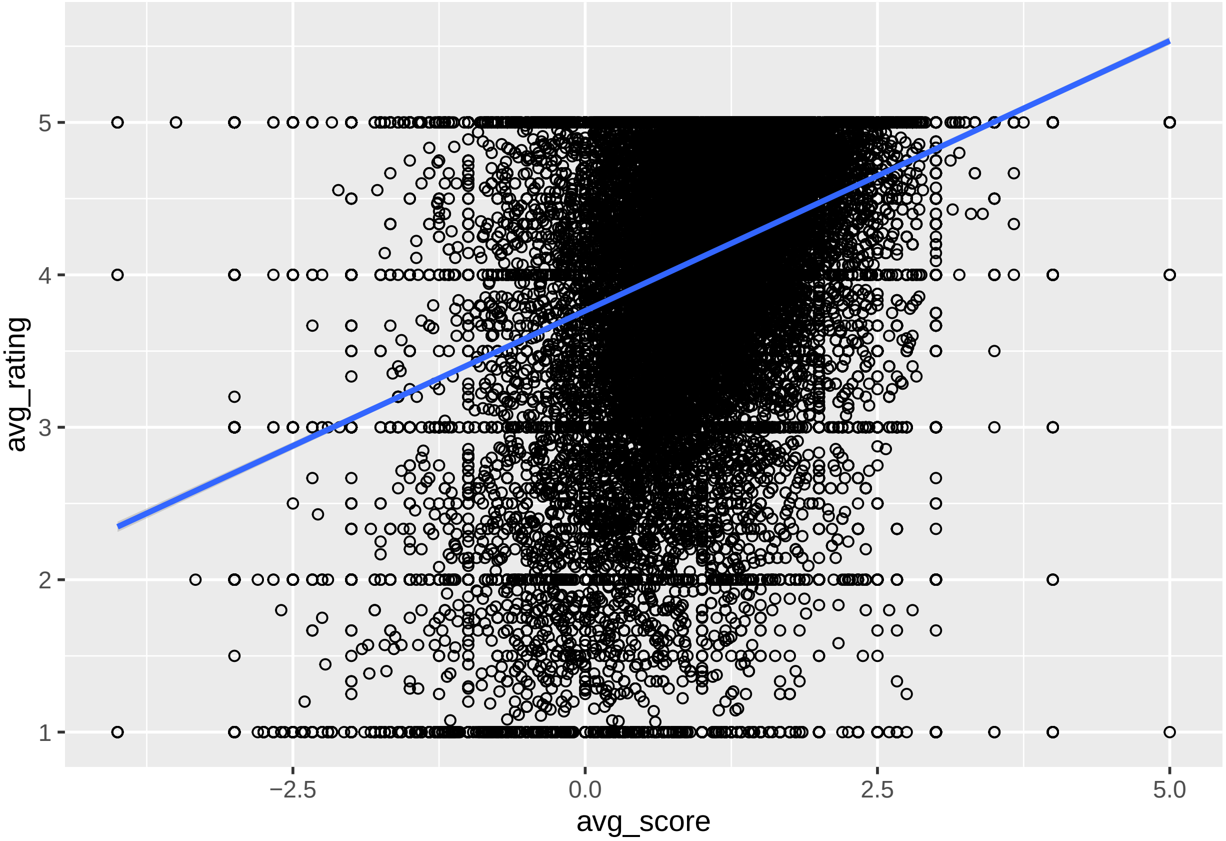
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
View(head(tidy_amzn_sentiment_prod,10))
```

plot

```
ggplot(tidy_amzn_sentiment_prod, aes(x=avg_score, y=avg_rating)) +
    geom_point(shape=1) +     # Use hollow circles
    geom_smooth(method=lm,    # Add linear regression line
                se=TRUE)     # Don't add shaded confidence region
```

```
## `geom_smooth()` using formula 'y ~ x'
```



## Topic Modelling

Latent Dirichlet allocation (LDA) is one of the most common algorithms for topic modeling.

```
amzn_dtm <- tidy_amzn %>%
                count(productId, word, sort = TRUE) %>%
                ungroup() %>%
                cast_dtm(productId, word, n)
```

```
amzn_dtm[1:100,]
```

```
## <<DocumentTermMatrix (documents: 100, terms: 101537)>>
## Non-/sparse entries: 197561/9956139
## Sparsity           : 98%
## Maximal term length: 124
## Weighting          : term frequency (tf)
```

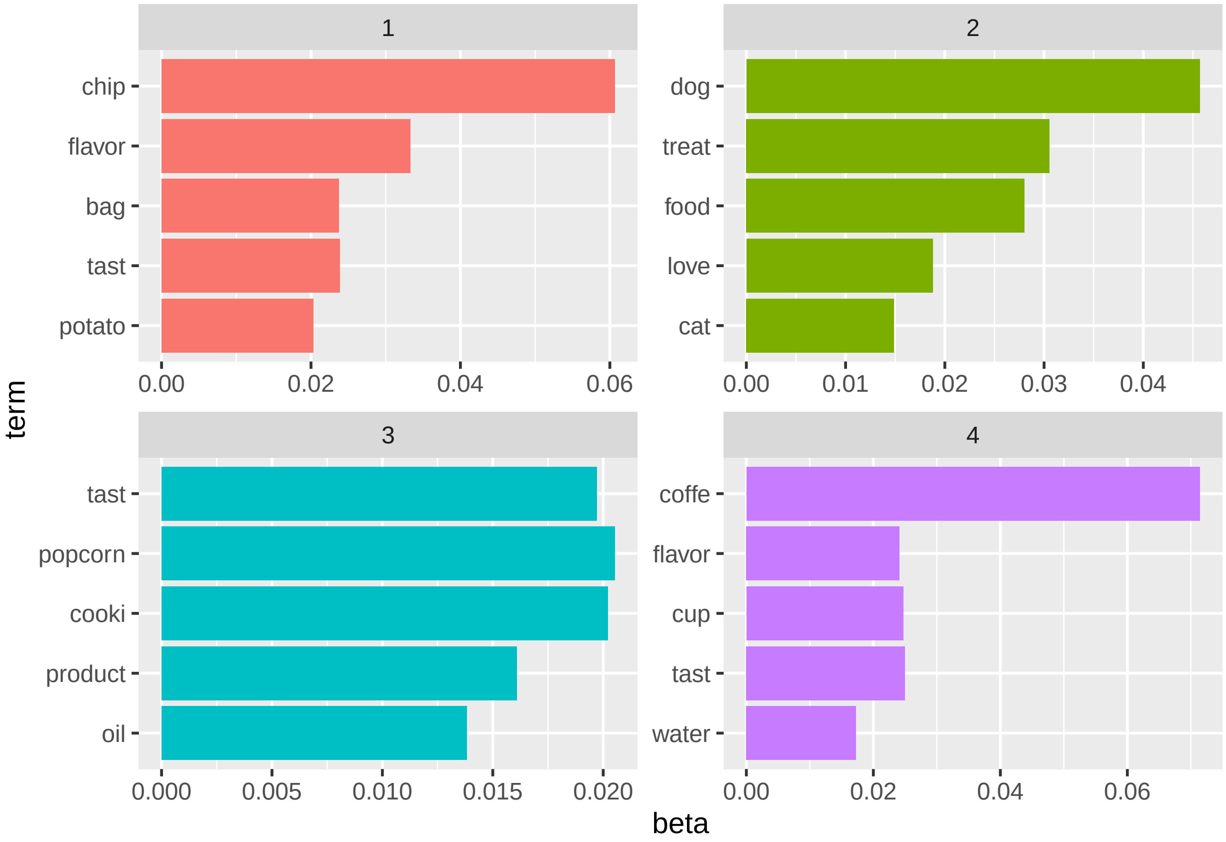
4 topics

```
product_lda <- LDA(amzn_dtm[1:100,], k = 4, control = list(seed = 1))

product_topics <- tidy(product_lda, matrix = "beta")

top_terms <- product_topics %>%
                group_by(topic) %>%
                top_n(5, beta) %>%
                ungroup() %>%
                arrange(topic, -beta)
```

```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



```
top_terms
```

```
## # A tibble: 20 x 3
##    topic term      beta
##    <int> <chr>    <dbl>
##  1     1 chip    0.0607
##  2     1 flavor  0.0333
##  3     1 tast    0.0239
##  4     1 bag     0.0237
##  5     1 potato  0.0203
##  6     2 dog     0.0457
##  7     2 treat   0.0306
##  8     2 food    0.0280
##  9     2 love    0.0188
## 10     2 cat     0.0149
## 11     3 popcorn 0.0205
## 12     3 cooki   0.0202
## 13     3 tast    0.0197
## 14     3 product 0.0161
## 15     3 oil     0.0138
## 16     4 coffe   0.0714
## 17     4 tast    0.0250
## 18     4 cup     0.0247
## 19     4 flavor  0.0241
## 20     4 water   0.0173
```