An Introduction to Sentiment Analysis in R

CCBS Virtual Event – R for Non-Econometrics

James Brookes

Advanced Analytics Division, Bank of England

24/11/2022

Agenda

1. Introduction to Sentiment Analysis
2. Sentiment Analysis & Modeling in R

Section 1

Introduction to Sentiment Analysis

Subsection 1

What is Sentiment Analysis?

What is Sentiment Analysis?

- Task/research field in Natural Language Processing (NLP) since ca. 2000
- Is this piece of language positive or negative in sentiment? Thumbs up or thumbs down?
- "Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in [language]. The entities can be products, services, organizations, individuals, events, issues, or topics.'' (Liu 2020)
- Usually written text, but also work being done using different types of input data: images, videos, and audio
- Other names (but subtly different tasks): opinion mining, opinion analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining
- Task setup:
  - Most typically a binary classification task: positive vs. negative
  - Occasionally ordinal (e.g. star ratings; negative/neutral/positive; very negative—very positive; etc.)

Subsection 2

Stuff that Falls Under Sentiment Analysis

Stuff that Falls Under Sentiment Analysis

- subjectivity detection (Pang and Lee 2004)
- hawkishness/dovishness (Tobback et al. 2017)
- toxic language detection (Zhou et al. 2021)
- fake news detection (Oshikawa et al. 2020)
- stance detection (Anand et al. 2011)
- rumor detection (Ma et al. 2018)
- polarization (Demszky et al. 2019)

Subsection 3

Applications

General

- businesses:
  - what are customers saying about their products and services?
  - tweet sentiment used to predict movie revenues
- consumers:
  - holidaymakers want to know whether they should this or that hotel
  - product reviews used to rank products and merchants (McGlohon et al. 2010)
- governments and decision-making institutions:
  - what are public opinions about existing or proposed policies?
- political elections (e.g., Bermingham and Smeaton 2011)
  - what are people's opinions about electoral candidates?
  - positive/negative tweets used to train a linear regression model to predict election results (Bermingham and Smeaton 2011)

CB & EconFin

- stock market prediction (Das and Chen 2007)
- regulatory communications to failing firms (Bholat et al. 2017)
- sentiment gap between market participants' views and those of the MPC (with Carlos Canon)
- measuring news sentiment (Turrell et al.; Shapiro et al.) forcasting

Subsection 4

Levels of Sentiment Analysis

Document Level

- classify whether a whole document expresses a positive or negative sentiment
- e.g. given a product review, the sentiment analysis system determines whether the whole review expresses an overall positive or negative opinion about the product
- problem:
  - assumes each document is about a single entity
  - so difficult to extract reasons

Sentence Level

- classify whether a sentence expresses a positive, neutral, or negative opinion
- similar problems as with document-level analysis:
  - e.g. *The food arrived cold, but the service was fast*

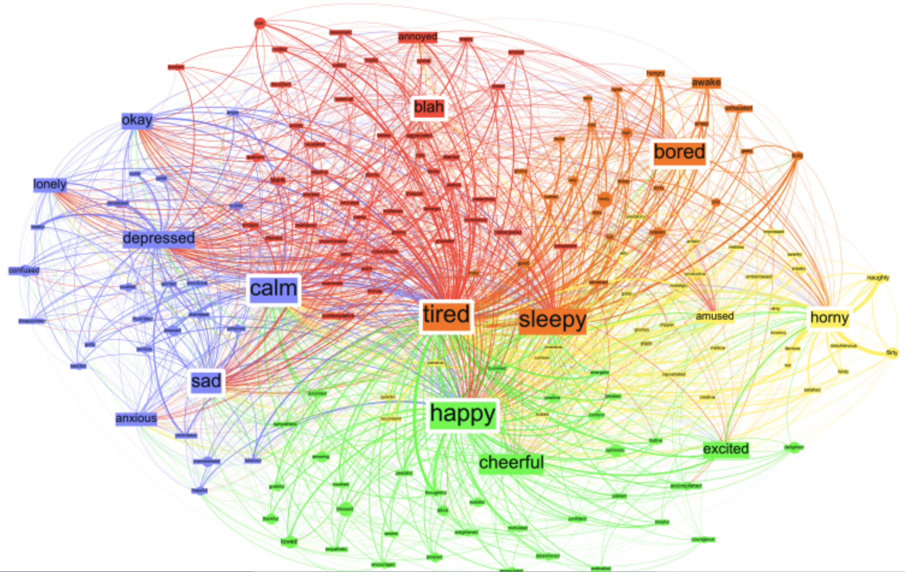
Aspect (Target/Feature/Entity) Level

- often want to know what each sentiment/opinion is about
- takes a document and extracts all (aspect, sentiment) pairs
  - e.g. *Apple is doing well in this poor economy*
    - (Apple, positive), (economy in general, negative)
- or we could go even more fine-grained
  - E.g. "When I arrived, I was so upset with the manager of the hotel' '
    - entity - hotel
    - stimulus - manager of the hotel
    - emotion - anger
    - experiencer - the speaker/writer
    - time - upon arrival
- problem:
  - a bit more involved, need two systems - one to extract aspects, another to classify sentiment

Subsection 5

Issues in Sentiment Analysis

Affective Computing



Language Use

1. There was an earthquake in California / on Mars
2. This phone sucks
3. (a) Can you recommend a good camera?
   (b) Do you know a place where I can get this rubbish camera fixed?
4. This camera cost an arm and a leg
5. Many said this movie would be... It was terrible.
6. Coke tastes better than Pepsi
7. The team failed to complete the challenge. (We win/lose!)
8. I'm so upset that X's share price has gone up
9. I did not love that movie

Subsection 6

Approaches

Lexicon-Based Approaches

- Use an existing sentiment lexicon or induce one from the data
  - Bing Liu's lexicon
  - Loughran and McDonald (EconFin)
  - Harvard General Inquirer
  - Warriner et al.'s affective ratings
- Many available in R via (e.g.) `tidytext::get_sentiments(lexicon = c("bing", "loughran"))`
- Use some kind of scoring function, e.g:

$$sentimentScore = \frac{count(pos, doc) - count(neg, doc)}{count(words, doc)}$$

$$sentimentClass = \begin{cases} +, , & sentimentScore > 0 \\ -, & sentimentScore \leq 0 \end{cases}$$

Traditional Supervised Sentiment Analysis

- Take some sparse, high-dimensional bag-of-words/handcrafted feature representation of the text
- Feed in (*representation*, *label*) pairs into some traditional ML algorithm:
    - naive Bayes
    - logistic regression (usually needs to be regularized because of vast feature spaces)
    - random forest
    - SVM
    - ...
- Same as any other ML problem (hyperparameter tuning, feature ablation experiments, ...)

Features in Traditional Supervised Sentiment Analysis

- bag-of-words:

```
library(tidyverse)
library(textrecipes)
tibble(label = c("neg", "pos"),
       text = c("This movie was boring", "This book was really interesting")) %>%
         recipe(label ~ text) %>%
         step_tokenize(text) %>%
         step_tf(text) %>%
         prep() %>%
         bake(new_data = NULL)
```

```
## # A tibble: 2 x 8
##   label tf_text_book tf_text_boring tf_text_in~1 tf_te~2 tf_te~3 tf_te~4 tf_te~5
##   <fct>        <dbl>          <dbl>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 neg              0              1            0       1       0       1       1
## 2 pos              1              0            1       0       1       1       1
## # ... with abbreviated variable names 1: tf_text_interesting, 2: tf_text_movie,
## #   3: tf_text_really, 4: tf_text_this, 5: tf_text_was
```

Features in Traditional Supervised Sentiment Analysis

- handcrafted features (depends on the application/domain):
  - repeated exclamation marks `This movie was awful!!!!!!`
  - all caps, `This movie was AWFUL`
  - emojis, `This book was great :)`
  - word lengthening `It was so booooorrrring`
  - POS-tags
  - syntactic dependency parses
  - ....

Neural Supervised Sentiment Analysis

- Take some dense, low-dimensional representation of the text
- Feed in (*representation*, *label*) pairs into a neural network of some kind:
  - feed-forward neural networks
  - convolutional neural networks
  - recurrent neural networks (RNNs, LSTMs)
  - transformer-based models (e.g. BERT)

Features in Neural Supervised Sentiment Analysis

```r
glove_embeddings <- read_delim("data/glove6b100d.txt", delim = "\t")
```

```
## Rows: 134638 Columns: 101
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr   (1): word
## dbl (100): w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11, w12, w13, w14, w15,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(glove_embeddings)
```

```
## # A tibble: 6 x 101
##   word       w1      w2     w3      w4      w5     w6      w7      w8      w9
##   <chr>   <dbl>   <dbl>  <dbl>   <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>
## 1 the   -0.0382 -0.245  0.728  -0.400   0.0832 0.0440 -0.391   0.334  -0.575
## 2 ,     -0.108   0.111  0.598  -0.544   0.674  0.107   0.0389  0.355   0.0635
## 3 .     -0.340   0.209  0.463  -0.648  -0.384  0.0380  0.171   0.160   0.466
## 4 of    -0.153  -0.243  0.898   0.170   0.535  0.488  -0.588  -0.180  -1.36
## 5 to    -0.190   0.0500 0.191  -0.0492 -0.0897 0.210  -0.550   0.0984 -0.201
## 6 and   -0.0720  0.231  0.0237 -0.506   0.339  0.196  -0.329   0.184  -0.181
```

Features in Neural Supervised Sentiment Analysis

```
tibble(label = c("neg", "pos"),
       text = c("This movie was boring", "This book was really interesting")) %>%
         recipe(label ~ text) %>%
         step_tokenize(text) %>%
         step_word_embeddings(text, embeddings = glove_embeddings ) %>%
         prep() %>%
         bake(new_data = NULL)
```

```
## # A tibble: 2 x 101
##   label wordem~1 worde~2 worde~3 worde~4 worde~5 worde~6 worde~7 worde~8 worde~9
##   <fct>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 neg     -0.655   0.695    1.94   -1.89  -0.865   0.256   0.293   1.27    0.622
## 2 pos     -1.17    1.62     1.87   -1.51   0.205   0.682   1.01    0.360   0.575
## # ... with 91 more variables: wordembed_text_w10 <dbl>,
## #   wordembed_text_w11 <dbl>, wordembed_text_w12 <dbl>,
## #   wordembed_text_w13 <dbl>, wordembed_text_w14 <dbl>,
## #   wordembed_text_w15 <dbl>, wordembed_text_w16 <dbl>,
## #   wordembed_text_w17 <dbl>, wordembed_text_w18 <dbl>,
## #   wordembed_text_w19 <dbl>, wordembed_text_w20 <dbl>,
## #   wordembed_text_w21 <dbl>, wordembed_text_w22 <dbl>, ...
```