Sentiment analysis

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

Sentiment analysis

- Assess subjective information from text
- Types of sentiment analysis:
  - positive vs negative
  - words eliciting emotions
- Each word is given a meaning and sometimes a score
  - abandon -> fear
  - accomplish -> joy

Tidytext sentiments

```
library(tidytext)
sentiments
```

```
# A tibble: 27,314 x 4
   word        sentiment lexicon score
   <chr>       <chr>     <chr>   <int>
 1 abacus      trust     nrc        NA
 2 abandon     fear      nrc        NA
 3 abandon     negative  nrc        NA
 4 abandon     sadness   nrc        NA
 5 abandoned   anger     nrc        NA
```

3 lexicons

- `AFINN` : scores words from -5 (extremely negative) to 5 (extremely positive)
- `bing` : positive/negative label for all words
- `nrc` : labels words as fear, joy, anger, etc.

```
library(tidytext)
get_sentiments("afinn")
```

```
# A tibble: 2,476 x 2
 1 abandon        -2
 2 abandoned      -2
 3 abandons       -2
 ...
```

Prepare your data.

```
# Read the data
animal_farm <- read.csv("animal_farm.csv", stringsAsFactors = FALSE)
animal_farm <- as_tibble(animal_farm)

# Tokenize and remove stop words
animal_farm_tokens <- animal_farm %>%
  unnest_tokens(output = "word", token = "words", input = text_column) %>%
  anti_join(stop_words)
```

The afinn lexicon

```
animal_farm_tokens %>%
  inner_join(get_sentiments("afinn"))
```

```
# A tibble: 1,175 x 3
   chapter   word        score
   <chr>     <chr>       <int>
 1 Chapter 1 drunk          -2
 2 Chapter 1 strange        -1
 3 Chapter 1 dream           1
 4 Chapter 1 agreed          1
 5 Chapter 1 safely          1
```

afinn continued

```
animal_farm_tokens %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(chapter) %>%
  summarise(sentiment = sum(score)) %>%
  arrange(sentiment)
```

```
# A tibble: 10 x 2
   chapter    sentiment
   <chr>          <int>
 1 Chapter 7       -166
 2 Chapter 8       -158
 3 Chapter 4        -84
```

The bing lexicon

```
word_totals <- animal_farm_tokens %>%
  group_by(chapter) %>%
  count()
```

```
animal_farm_tokens %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(chapter) %>%
  count(sentiment) %>%
  filter(sentiment == 'negative') %>%
  transform(p = n / word_totals$n) %>%
  arrange(desc(p))
```

```
      chapter sentiment   n          p
1   Chapter 7  negative 154 0.11711027
2   Chapter 6  negative 106 0.10750507
3   Chapter 4  negative  68 0.10559006
4  Chapter 10  negative 117 0.10372340
5   Chapter 8  negative 155 0.10006456
6   Chapter 9  negative 121 0.09152799
7   Chapter 3  negative  65 0.08843537
8   Chapter 1  negative  77 0.08603352
9   Chapter 5  negative  93 0.08462238
10  Chapter 2  negative  67 0.07395143
```

The nrc lexicon

```
as.data.frame(table(get_sentiments("nrc")$sentiment)) %>%
  arrange(desc(Freq))
```

```
          Var1 Freq
1     negative 3324
2     positive 2312
3         fear 1476
4        anger 1247
5        trust 1231
6      sadness 1191
...
```

nrc continued

```
fear <- get_sentiments("nrc") %>%
  filter(sentiment == "fear")
animal_farm_tokens %>%
  inner_join(fear) %>%
  count(word, sort = TRUE)
```

```
# A tibble: 220 x 2
   word           n
   <chr>      <int>
 1 rebellion     29
 2 death         19
 3 gun           19
 4 terrible      15
 5 bad           14
 6 enemy         12
 7 broke         11
 ...
```

Sentiment time.

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Word embeddings

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

The flaw in word counts

Two statements:

- Bob is the smartest person I know.
- Bob is the most brilliant person I know.

Without stop words:

- Bob smartest person
- Bob brilliant person

Word meanings

Additional data:

- The smartest people ...
- He was the smartest ...
- Brilliant people ...
- His was so brilliant ...

word2vec

- represents words as a large vector space
- captures multiple similarities between words
- words of similarly meaning are closer within the space

[1] https://www.adityathakker.com/introduction [2] to [3] word2vec [4] how [5] it [6] works/

Preparing data

```
library(h2o)
h2o.init()
```

```
h2o_object = as.h2o(animal_farm)
```

Tokenize using h2o:

```
words <- h2o.tokenize(h2o_object$text_column, "\\\\W+")
words <- h2o.tolower(words)
words = words[is.na(words) || (!words %in% stop_words$word),]
```

word2vec modeling

```
word2vec_model <-
    h2o.word2vec(words, min_word_freq = 5, epochs = 5)
```

- `min_word_freq` : removes words used fewer than 5 times
- `epochs` : number of training iterations to run

Word synonyms

```
h2o.findSynonyms(w2v.model, "animal")
```

```
    synonym     score
1     drink 0.8209088
2       age 0.7952490
3   alcohol 0.7867004
4       act 0.7710537
5      hero 0.7658424
```

```
h2o.findSynonyms(w2v.model, "jones")
```

```
       synonym     score
1       battle 0.7996588
2  discovered 0.7944554
3      cowshed 0.7823287
4      enemies 0.7766532
5        yards 0.7679787
```

Additional uses

- classification modeling
- sentiment analysis
- topic modeling

Apply word2vec

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Additional NLP analysis

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

BERT, and ERNIE.

What is it:

- BERT: Bidirectional Encoder Representations from Transformers
- A model used in transfer learning for NLP tasks
- is pre-trained on unlabeled data to create a language representation
- requires only small amounts of labeled data to train for specific task

What is it used for:

- supervised tasks
- to create features for NLP models

ERNIE: Enhanced Representation through kNowledge IntEgration

Named Entity Recognition

What is it:

- classifies named entities within text
- Examples: names, locations, organizations, values

What is it used for:

- extracting entities from tweets
- aiding recommendation engines
- search algorithms

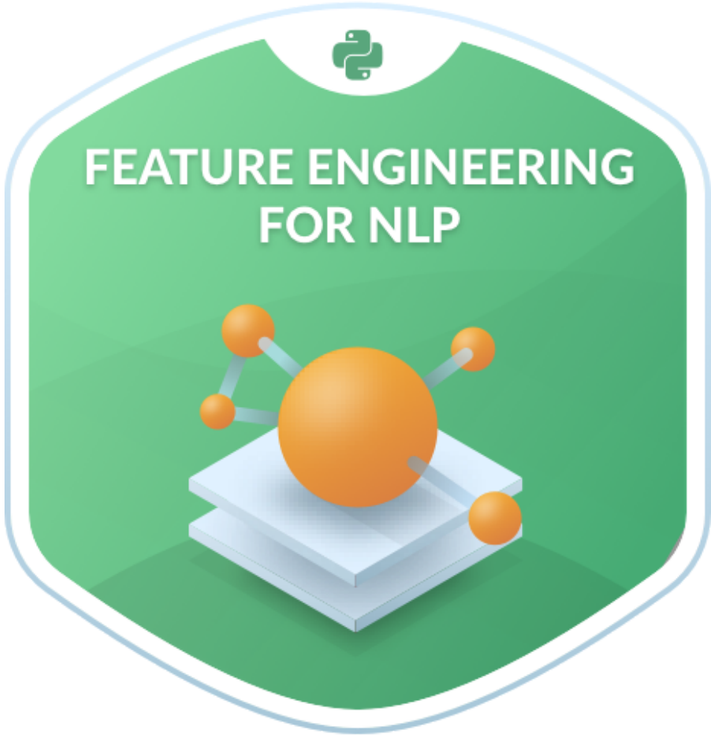
Part-of-speech tagging

What is it:

- tagging words with their part-of-speech
  - nouns, verbs, adjectives, etc.

How is it used:

- aids in sentiment analysis
- creates features for NLP models
- enhances what a model knows about each word in text



Let's recap.

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Conclusion

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R

Kasey Jones
Research Data Scientist

Course recap

- The pre-processing:
  - tokenization
  - stop-word removal
  - data formats (tibbles, VCorpus, h2o frame)
- The classics:
  - sentiment analysis
  - text classification
  - topic modeling

Recap continued

- The advanced techniques
  - word embeddings
  - BERT/ERNIE
- The Next Steps
  - practice
  - master the basics

Course complete!

INTRODUCTION TO NATURAL LANGUAGE PROCESSING IN R