Text Mining and Sentiment Analysis

Prof. Annamaria Bianchi
A.Y. 2024/2025

Lecture 10

18 March 2025

UNIVERSITÀ DEGLI STUDI DI BERGAMO | Dipartimento di Scienze Economiche

Outline

SA in the tidy data format

Visualization of SA results

Packages: **tidytext, tidyverse, wordcloud**, **reshape2**

Functions: `dplyr::inner_join()`,`tidyr::pivot_wider()`, `wordcloud::comparison.cloud()`, `reshape2::acast()`

Sentiment Analysis using the tidy data principle

With reference to the Airbnb Boston apartments data, during Lecture 9 we computed a polarity score according to the Bing lexicon.

```
> library(tidyverse)
> library(tidytext)
> View(bos.pol)
```

	ID	positive	negative	polarity
1	1	4	0	4
2	2	3	0	3
3	3	3	0	3
4	4	6	0	6
5	5	2	0	2

Sentiment Analysis

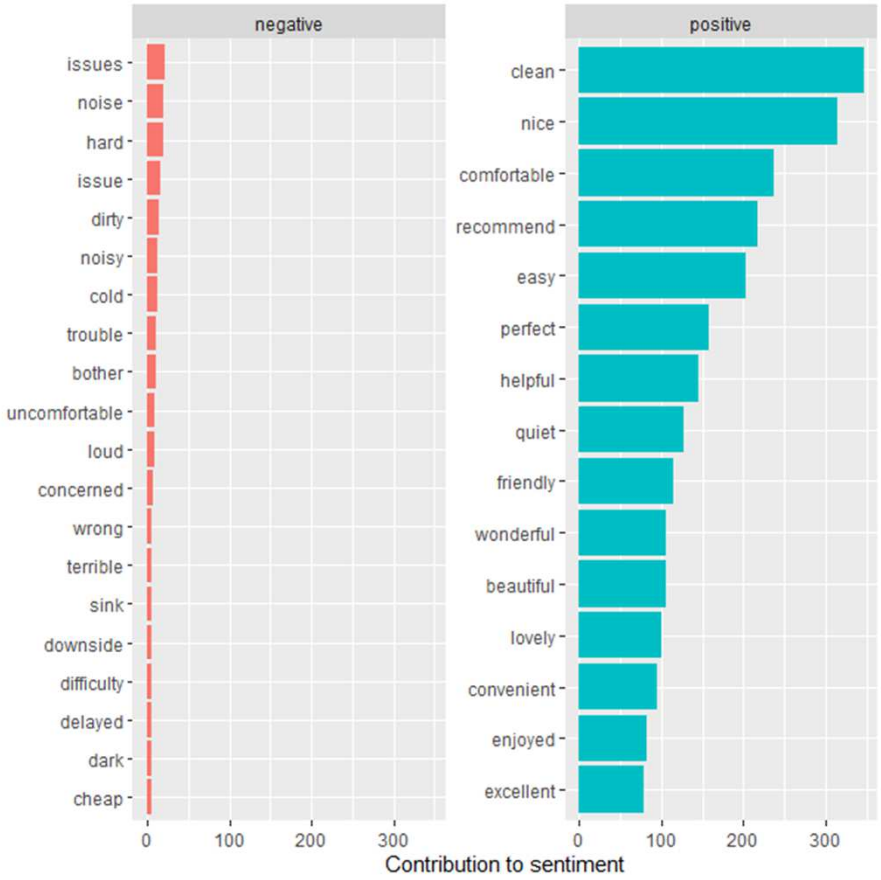
We shall investigate visually word counts that contribute to each sentiment. By implementing `count()` with arguments *word* and *sentiment*, we find how much each word contributes to each sentiment

```
> bing = get_sentiments("bing")
> bos.word.count = tidy.bos.airbnb |>
+ inner_join(bing) |>
+ count(word, sentiment, sort = T)
Joining with `by = join_by(word)`
> bos.word.count
# A tibble: 624 × 3
  word          sentiment      n
  <chr>         <chr>      <int>
1 clean         positive     346
2 nice          positive     314
3 comfortable   positive     237
4 recommend     positive     218
5 easy          positive     202
6 perfect       positive     158
7 helpful       positive     145
```

Sentiment Analysis

The information can be shown visually as follows

```
> bos.word.count |>
+ group_by(sentiment) |>
+ slice_max(order_by = n, n = 15) |>
+ mutate(word = reorder(word,n)) |>
+ ggplot(aes(word, n, fill=sentiment))+
+ geom_col(show.legend = F) +
+ facet_wrap(~ sentiment, scales = "free_y")+
+ xlab(NULL)+
+ ylab("Contribution to sentiment")+
+ coord_flip()
```



Sentiment Analysis – dictionary-based approach

We shall create a sentiment-based word cloud. We do this using the function **wordcloud**::`comparison.cloud()`, that plots a cloud comparing the frequencies of words across groups.

```
comparison.cloud(term.matrix,
                 colors=brewer.pal(max(3,ncol(term.matrix)),"Dark2"),
                 max.words=300,...)
```

`term.matrix`	A term frequency matrix whose rows represent words and whose columns represent groups.
`max.words`	Maximum number of words to be plotted. Least frequent terms dropped
`colors`	Color words in the order of columns in term.matrix

Sentiment-based wordcloud

In order to apply the `comparison.cloud()` function, we need first to turn the data frame into a matrix.

```
> tidy.bos.airbnb |> inner_join(bing) |>
+ count(word, sentiment, sort=T) |>
+ pivot_wider(names_from = sentiment, values_from =n,
values_fill = 0) |>
+ column_to_rownames(var="word") |>
+ comparison.cloud(colors=c("gray20", "gray80"),
max.words = 100)
```

The size of a word's text in the plot is proportional
to its frequency within its sentiment.
We shall use this visualization to see the most important positive and negative words, but the sizes of the words are not comparable across sentiments.

AFINN and NRC sentiment datasets

Beyond Bing lexicon, other two sentiment lexicons are available in the **tidytext** package:

- AFINN lexicon
- NRC lexicon.

All of them are based on unigrams

AFINN and NRC sentiment datasets

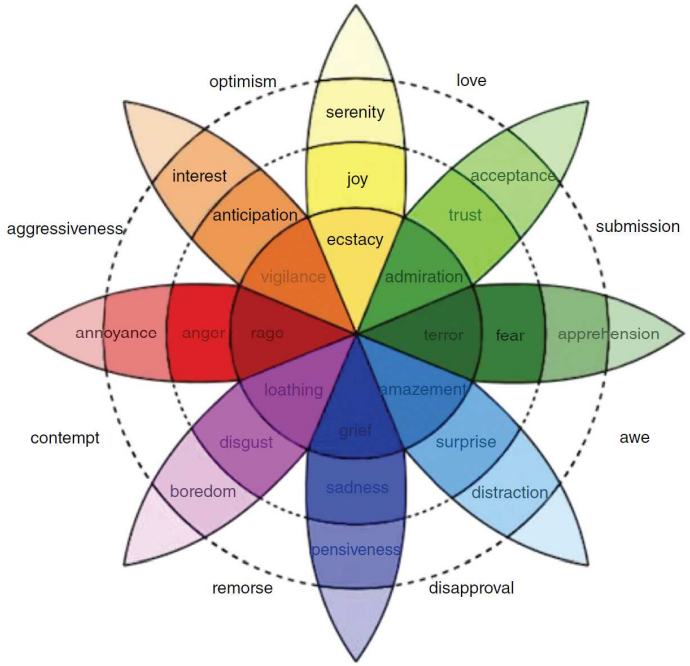
AFINN lexicon has been developed by Finn Arup Nielsen (a Danish researcher at a technical university) and contains 2476 English words. This lexicon assigns words with a score ranging from -5 and +5, with negative scores indicating negative sentiment and positive scores indicating positive sentiments.

```
> install.packages("textdata")
> afinn = get_sentiments("afinn")
> afinn
# A tibble: 2,477 × 2
   word       value
  <chr>      <dbl>
1 abandon       -2
2 abandoned     -2
3 abandons      -2
4 abducted      -2
5 abduction     -2
6 abductions    -2
7 abhor         -3
...
```

AFINN and NRC sentiment datasets

NRC, from Saif Mohammad and Peter Turney, contains 13 901 words associated to emotions. Categories are positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

This lexicon refers to a popular academic sentiment framework called Plutchik's wheel of emotion



AFINN and NRC sentiment datasets

```
> nrc = get_sentiments("nrc")
> nrc
# A tibble: 13,872 × 2
   word      sentiment
   <chr>     <chr>
 1 abacus    trust
 2 abandon   fear
 3 abandon   negative
 4 abandon   sadness
 5 abandoned anger
 6 abandoned fear
 7 abandoned negative
….
```

AFINN and NRC sentiment datasets

Exercise. Explore AFINN and NRC lexicons.

1. Take a look at the score distribution in AFINN. How many words have score +5? And -5?
2. Take a look at the sentiment distribution in NRC. How many words are associated with *sadness*? And *surprise*?
3. List the words in NRC associated with *joy* and *fear*.

AFINN and NRC sentiment datasets

Let us use the NRC lexicon to answer the question:

What are the most common trust words used in the comments?

```
> nrctrust = nrc |>
+ filter(sentiment == "trust")
> tidy.bos.airbnb |>
+ inner_join(nrctrust) |>
+ count(word, sort = T)
Joining with `by = join_by(word)`
```

```
# A tibble: 235 x 2
   word           n
   <chr>      <int>
 1 clean        346
 2 recommend    218
 3 perfect      158
 4 helpful      145
 5 friendly     114
 6 wonderful    105
 7 lovely       100
 8 excellent     78
 9 found         56
10 safe          54
# ... with 225 more rows
```

Comparing the three sentiment dictionaries

With several options for sentiment lexicons, you might want some more information on which one is most appropriate for your purposes. Let us use all three sentiment lexicons and then compare the results.

Let us first add a column to the polarity score computed according to the Bing lexicon:

```
> sentiment.B = bos.pol |>
+ select(ID, sentiment) |>
+ mutate(method = "bing")
```

	ID	sentiment	method
1	1	4	Bing
2	2	3	Bing
3	3	3	Bing
4	4	6	Bing
5	5	2	Bing

Comparing the three sentiment dictionaries

Let us now compute the polarity score according to the *NRC* lexicon:

```
> sentiment.N = tidy.bos.airbnb |>
+ inner_join(nrc) |>
+ filter(sentiment %in% c("positive", "negative")) |>
+ count(ID, sentiment) |>
+ pivot_wider(names_from = sentiment, values_from = n, values_fill =0) |>
+ mutate(sentiment = positive - negative, method = "nrc") |>
+ select(ID, sentiment, method)
> View(sentiment.N)
```

	ID	sentiment	method
1	1	5	nrc
2	2	3	nrc
3	3	2	nrc
4	4	1	nrc
5	5	0	nrc
6	6	0	nrc

Comparing the three sentiment dictionaries

Polarity score according to the AFINN lexicon

```
> sentiment.A = tidy.bos.airbnb |>
+ inner_join(afinn) |>
+ group_by(ID) |>
+ summarise(sentiment=sum(value)) |>
+ mutate(method = "AFINN")
Joining with `by = join_by(word)`
> View(sentiment.A)
```

	ID	sentiment	method
1	1	11	AFINN
2	2	6	AFINN
3	3	10	AFINN
4	4	13	AFINN

Exercise for you

Exercise 1

Suppose you have gained subject matter expertise in Airbnb reviews so that you understand that some authors wrote comments in Italian. You decide you want to include their reviews in the sentiment computation. Of course, Italian terms are not included in Bing lexicon. You will need to add some selected terms to the list of positive words.

1. Add the words «carina», «confortevole», «gentili», and «positivo» to the list of positive words in Bing lexicon.
2. Apply the `inner_join()` function using the customised lexicon and verify that the customized words are properly classified.
3. Perform the polarity scoring using the customised lexicon.
4. Compare the polarity scores computed using the two lexicons.
5. Compare the overall summary statistics for the polarity scores based on the two lexicons.

Exercise for you

Exercise 2

1. Count the number of positive and negative words by lexicon. What do you notice?
2. How many words are in common among lexicons?

Exercise 3

1. Reshape the nrc lexicon so to have rows corresponding to words and columns corresponding to sentiments/emotions. What do you notice?
2. How many words are classified into more than one category?
3. Are all positive/negative words also associated to an emotion?