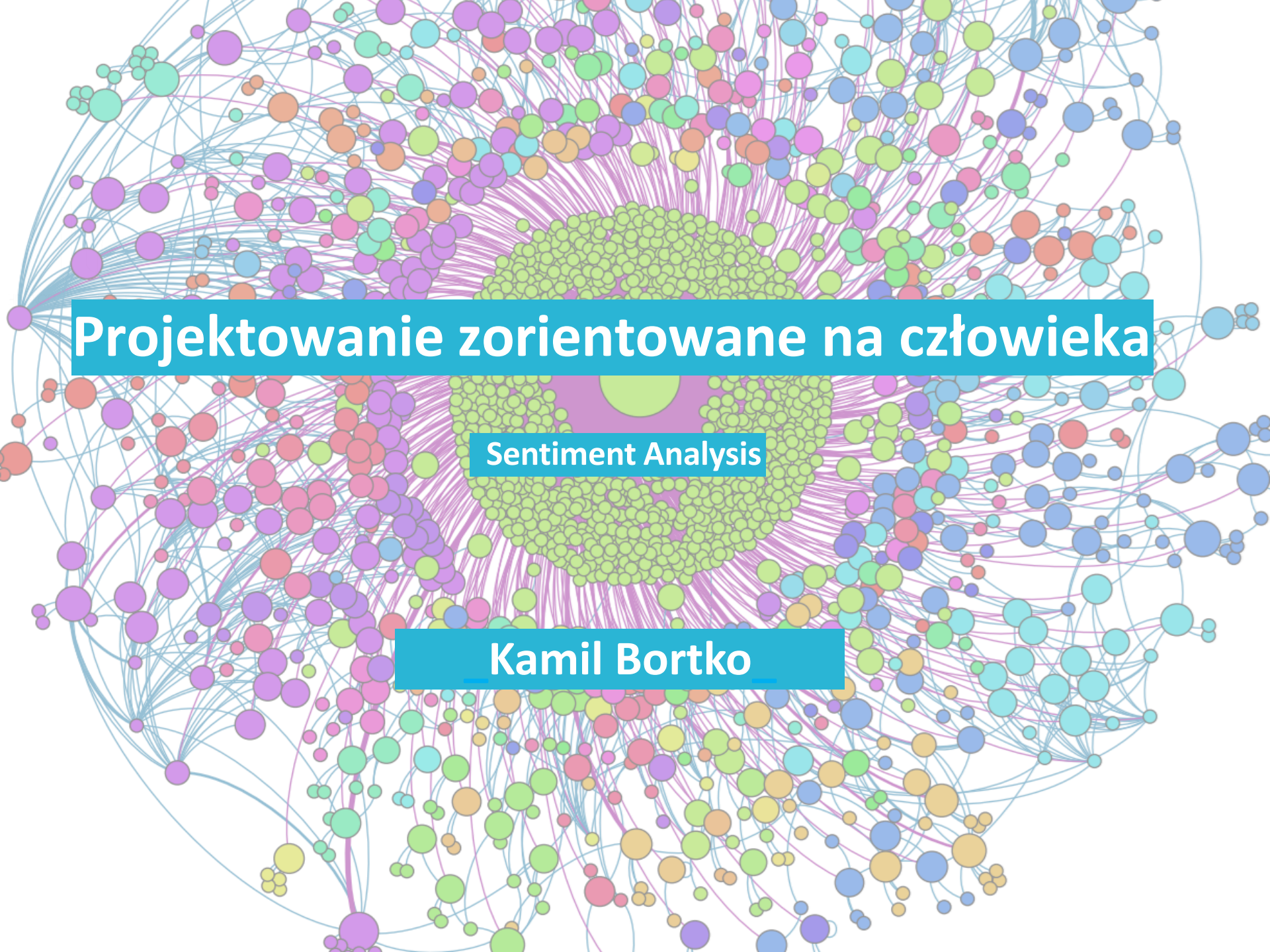


Projektowanie zorientowane na człowieka

Sentiment Analysis

Kamil Bortko

Zadanie 1

Zapoznanie się z podstawowymi funkcjami biblioteki **sentimentr** – slajdy 3-12

Zadanie 2

Zgodnie z opisem slajd 13

Biblioteki do analizy sentymentu

R

Sentimentr https://cran.r-project.org/web/packages/sentimentr/

RSentiment https://cran.r-project.org/web/packages/RSentiment/

SentimentAnalysis https://github.com/sfeuerriegel/SentimentAnalysis

Python

Natural Language Toolkit http://www.nltk.org/ https://sourceforge.net/projects/nltk/

http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/
http://www.nltk.org/howto/sentiment.html

TextBlob https://textblob.readthedocs.io/en/dev/

Pomocnicze

TwitteR https://www.rdocumentation.org/packages/twitteR/versions/1.1.9

Bibilioteka sentimentr

Metodę słownika rozszerzonego sentimentr, która może dawać lepsze wyniki niż proste podejście do słownika odnośników, które nie uwzględnia zmian walencyjnych z udziałem modyfikatorów. Algorytm wykorzystuje słownik sentymentów (np. Jockers, (2017)) do oznaczania spolaryzowanych słów. Każdy akapit (pi = {s1, s2,..., sn}) złożony ze zdań jest podzielony na odrębne zdania. Każde zdanie (sj) jest podzielone na uporządkowany zbiór słów. Interpunkcja jest usuwana, z wyjątkiem interpunkcji pauzy (przecinki, dwukropki, średniki), które są uważane za słowo w zdaniu. Słowa w każdym zdaniu (wi, j, k) są wyszukiwane i porównywane ze słownikiem spolaryzowanych słów (np. Połączona i rozszerzona wersja Jockera (2017) Wyrazy dodatnie (wi, j, k +) i ujemne (wi, j, k-) są odpowiednio oznaczone +1 i -1. Spolaryzowany klaster kontekstu domyślnie przyjmuje 4 słowa przed i dwa słowa po analizowanym słowie, aby uznać je za zmienniki wartościowości.

valence shifters

Zmiany wartościowości wpływają na spolaryzowane słowa. W przypadku negatorów i spójników cały sentyment klauzuli może zostać odwrócony lub unieważniony. Jeśli zmiany wartościowości występują dość często, proste wyszukiwanie w słowniku może nie modelować odpowiednio sentymentu.

Zmienne wartościowości współwystępują ze spolaryzowanymi słowami, potencjalnie zmieniając, a nawet odwracając i unieważniając sentyment klauzuli. Poniższa tabela pokazuje współczynnik współwystępowania na poziomie zdania zmienników wartościowości ze słowami spolaryzowanymi w kilku rodzajach tekstów

Text	Negator	Amplifier	Deamplifier	Adversative
Cannon reviews	21%	23%	8%	12%
2012 presidential debate	23%	18%	1%	11%
Trump speeches	12%	14%	3%	10%
Trump tweets	19%	18%	4%	4%
Dylan songs	4%	10%	0%	4%
Austen books	21%	18%	6%	11%
Hamlet	26%	17%	2%	16%

Negator zamienia znak spolaryzowanego słowa (e.g., "**I do not like it**.")
lexicon::hash_valence_shifters[y==1]

Wzmacniacz zwiększa wpływ spolaryzowanego słowa (e.g., "**I really like it**.").
lexicon::hash_valence_shifters[y==2]

Osłabiacz zmniejsza wpływ spolaryzowanego słowa (e.g., "**I hardly like it**.").
lexicon::hash_valence_shifters[y==3]

Konwersja przestawna zastępuje poprzednią klauzulę zawierającą spolaryzowane słowo(e.g., "**I like it but it's not worth it**."). lexicon::hash_valence_shifters[y==4]

Podstawowe funkcje

sentiment	Sentiment at the sentence level
sentiment_by	Aggregated sentiment by group(s)
profanity	Profanity at the sentence level
profanity_by	Aggregated profanity by group(s)
emotion	Emotion at the sentence level
emotion_by	Aggregated emotion by group(s)
uncombine	Extract sentence level sentiment from sentiment_by
get_sentences	Regex based string to sentence parser (or get sentences from sentiment/sentiment_by)
replace_emoji	repalcement
replace_emoticon	Replace emoticons with word equivalent
replace_grade	Replace grades (e.g., “A+”) with word equivalent
replace_internet_slang	replacment
replace_rating	Replace ratings (e.g., “10 out of 10”, “3 stars”) with word equivalent
as_key	Coerce a data.frame lexicon to a polarity hash key
is_key	Check if an object is a hash key
update_key	Add/remove terms to/from a hash key
highlight	Highlight positive/negative sentences as an HTML document
general_rescale	Generalized rescaling function to rescale sentiment scoring
sentiment_attribute	Extract the sentiment based attributes from a text
validate_sentiment	Validate sentiment score sign against known results

sentiment(text.var, polarity_dt = lexicon::hash_sentiment_jockers_rinker, valence_shifters_dt = lexicon::hash_valence_shifters, hyphen = "", amplifier.weight = 0.8, n.before = 5, n.after = 2, question.weight = 1, adversative.weight = 0.25, neutral.nonverb.like = FALSE, missing_value = 0, ...)

text.var The text variable. Can be a get_sentences object or a raw character vector though get_sentences is preferred as it avoids the repeated cost of doing sentence boundary disambiguation every time sentiment is run.

polarity_dt A data.table of positive/negative words and weights with x and y as column names. The lexicon package has several dictionaries that can be used, including:

- lexicon::hash_sentiment_jockers_rinker
- lexicon::hash_sentiment_jockers
- lexicon::emojis_sentiment
- lexicon::hash_sentiment_emojis
- lexicon::hash_sentiment_huliu
- lexicon::hash_sentiment_loughran_mcdonald
- lexicon::hash_sentiment_nrc
- lexicon::hash_sentiment_senticnet
- lexicon::hash_sentiment_sentiword
- lexicon::hash_sentiment_slangsd
- lexicon::hash_sentiment_socal_google

valence_shifters_dt - A data.table of valence shifters that can alter a polarized word's meaning and an integer key for negators (1), amplifiers [intensifiers] (2), de-amplifiers [downtoners] (3) and adversative conjunctions (4) with x and y as column names.

hyphen - The character string to replace hyphens with. Default replaces with nothing so 'sugar-free' becomes 'sugarfree'. Setting hyphen = " " would result in a space between words (e.g., 'sugar free'). Typically use either " " or default "".

amplifier.weight The weight to apply to amplifiers/de-amplifiers [intensifiers/downtoners] (values from 0 to 1). This value will multiply the polarized terms by 1 + this value.

n.before The number of words to consider as valence shifters before the polarized word. To consider the entire beginning portion of a sentence use n.before = Inf.

n.after The number of words to consider as valence shifters after the polarized word. To consider the entire ending portion of a sentence use n.after = Inf.

Analiza sentymentu

```
mytext <-c(
   'do you like it? But I hate really bad dogs',
   'I am the best friend.',
   'Do you really like it? I\'m not a  fan')

mytext <-get_sentences(mytext)
```

sentiment(mytext)

element_id	sentence_id	word_count	sentiment
1:	1	1	4 0.2500000
2:	1	2	6 -1.8677359
3:	2	1	5 0.5813777
4:	3	1	5 0.4024922
5:	3	2	4 0.0000000

sentiment_by(mytext)

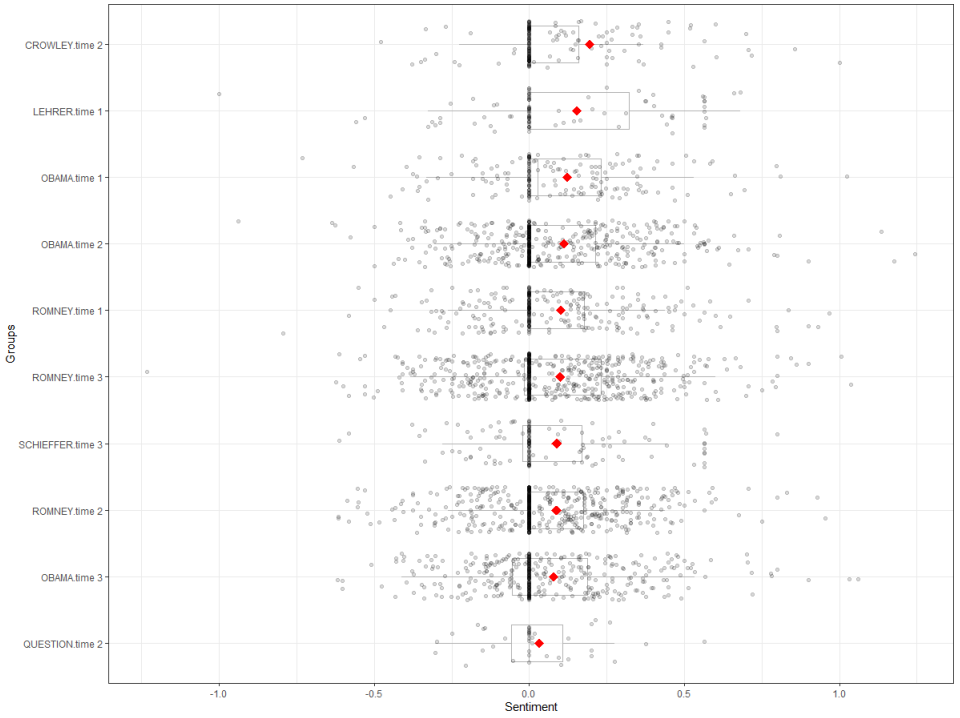
element_id	word_count	sd	ave_sentiment
1:	1	10 1.497465	-0.8088680
2:	2	5 NA	0.5813777
3:	3	9 0.284605	0.2196345

Zbiór wypowiedzi i emocje

```
p <- presidential_debates_2012
View(p)

(out <- with(
   presidential_debates_2012,
   sentiment_by(
      get_sentences(dialogue),
      list(person, time)
   )
))
```

person	time	word_count	sd	ave_sentiment
1: OBAMA	time 1	3599	0.2535006	0.12256892
2: OBAMA	time 2	7477	0.2509177	0.11217673
3: OBAMA	time 3	7243	0.2441394	0.07975688
4: ROMNEY	time 1	4085	0.2525596	0.10151917
5: ROMNEY	time 2	7536	0.2205169	0.08791018
6: ROMNEY	time 3	8303	0.2623534	0.09968544
7: CROWLEY	time 2	1672	0.2181662	0.19455290
8: LEHRER	time 1	765	0.2973360	0.15473364
9: QUESTION	time 2	583	0.1756778	0.03197751
10: SCHIEFFER	time 3	1445	0.2345187	0.08843478

```
plot(out)
```



Emocje

```
em <- emotion(p$dialogue)
View(em)

em <- emotion_by(p$dialogue,by=list( p$person,p$time))
```

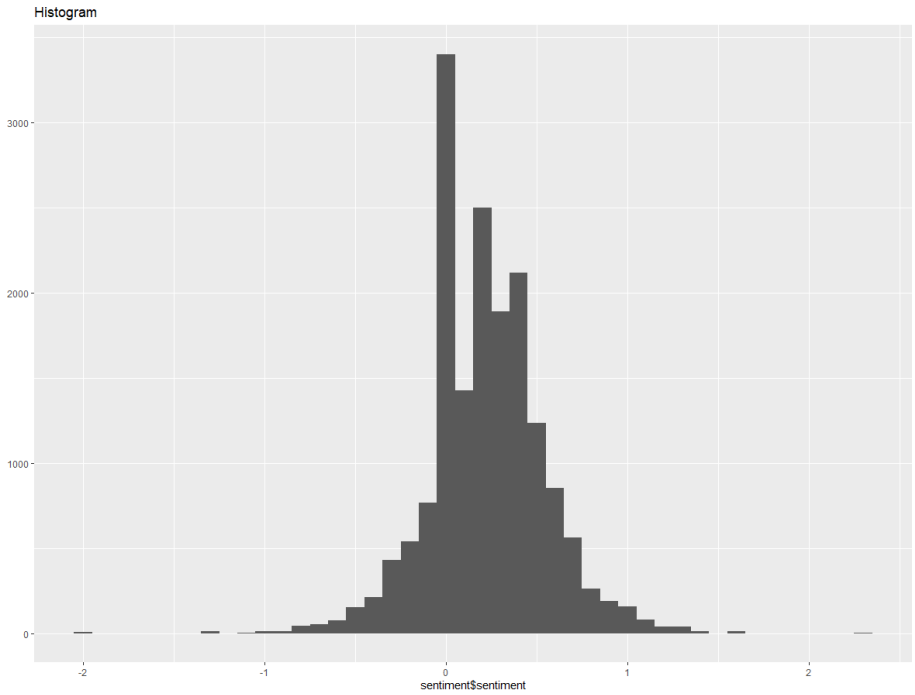
Opinie i komentarze

```
install.packages("sentimentr")
library(sentimentr)
```

```
install.packages("ndjson")
library(ndjson)
```

```
setwd("c:/R_data");
```

```
df = stream_in("opinions.json")
head(df)
```




```
sentiment=sentiment(df$reViewText, emotion_dt = lexicon::hash_sentiment_sentiword)

View(sentiment)

library(ggplot2)

qplot(sentiment$sentiment,    geom="histogram",binwidth=0.1,main="Histogram")
```

Opinie i komentarze z grupowaniem

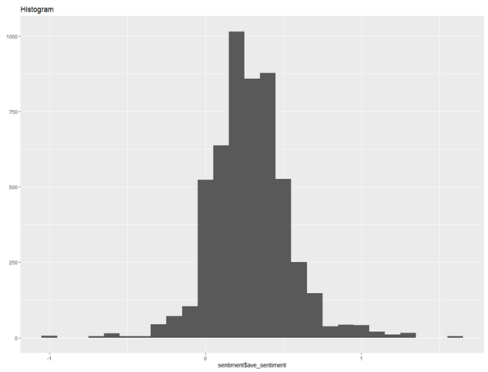
```
install.packages("sentimentr")
library(sentimentr)

install.packages("ndjson")
library(ndjson)

setwd("c:/R_data");

df = stream_in("opinions.json")
head(df)

sentiment=sentiment_by(df$reViewText)

View(sentiment)
```

element_id — The id number of the reView
word_count — The word count of the reView
sd — odchylenie standardowe sentymetu dla zdań
ave_sentiment — średni sentyment dla zdań w wypowiedzi

```
summary(sentiment$ave_sentiment)

df$ave_sentiment=sentiment$ave_sentiment
df$sd_sentiment=sentiment$sd

library(ggplot2)
qplot(sentiment$ave_sentiment,    geom="histogram",binwidth=0.1,main="Histogram")
```

Filter

	element_id	word_count	sd	ave_sentiment
1	1	4	0.500000000	0.3858585042
2	2	44	0.049484020	0.0919777849
3	3	29	0.246244375	-0.2007321540
4	4	262	0.158548937	0.1356085444
5	5	45	0.217477291	0.3561249613
6	6	20	0.028093109	0.1966415231
7	7	32	0.377022094	0.0169661915
8	8	33	0.298797395	0.0719261063
9	9	60	0.377336270	-0.3319925269
10	10	26	0.153960072	0.1301989833
11	11	4	0.041217521	0.7791451884
12	12	18	0.319801075	-0.2467959285
13	13	17	0.506131757	0.1039910178
14	14	44	0.396479196	0.1036032903
15	15	8	*NA*	-0.7247844507
16	16	33	0.230285141	-0.2836654287
17	17	15	*NA*	-0.0129099445
18	18	63	0.453962302	-0.1889216947
19	19	2	*NA*	0.5303300859
20	20	44	0.177355497	-0.0276925761
21	21	22	0.199617138	0.3532826659

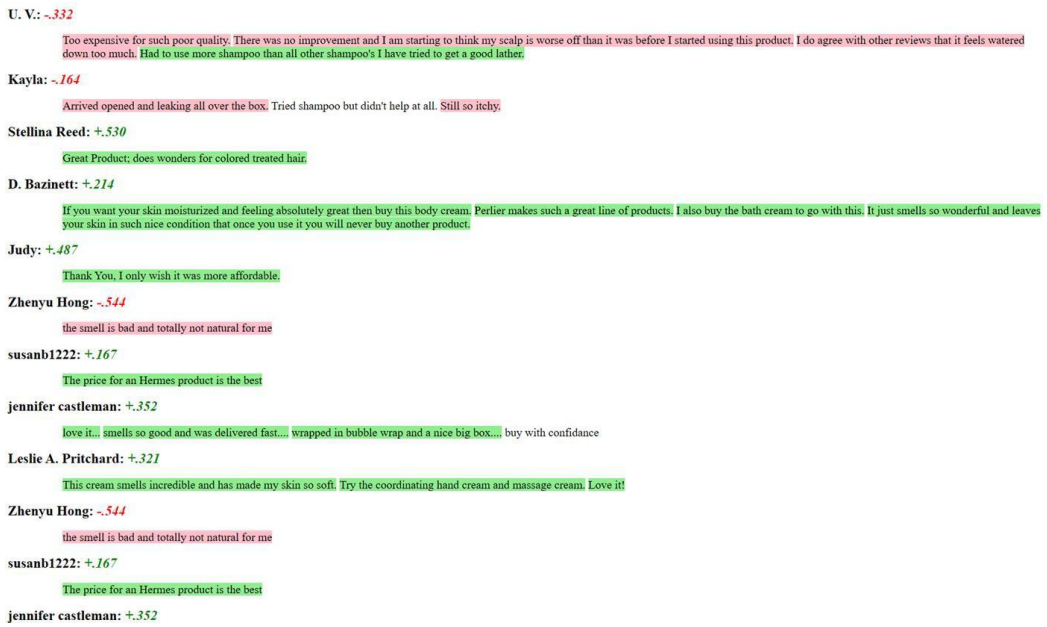


Budowanie własnego słownika

```
key <- data.frame(
   words = c("cat", "dog", "bunny"),
   polarity = c(1,-1,0),
   stringsAsFactors = FALSE
)

mykey <- as_key(key)

is_key(mykey)

sentiment_by("cat", polarity_dt = mykey)

sentiment_by("cat, dog, bunny", polarity_dt = mykey)

mykey_new <- update_key(mykey, drop = c("cat"))

sentiment_by("cat, dog, bunny", polarity_dt = mykey_new)
```

Wizualizacja

```
library(magrittr)
library(dplyr)
set.seed(2)

setwd("c:/R_data");

df = stream_in("opinions.json")
head(df)
```




```
df %>%
filter(reViewerName %in% sample(unique(reViewerName), 10)) %>%
mutate(reView = get_sentences(reViewText)) %$%
sentiment_by(reView, reViewerName) %>%
highlight()
```

Zadanie 2

- Pobrać zbiór danych dla dowolnej kategorii produktowej ze strony https://nijianmo.github.io/amazon/index.html

- Wyznaczyć średnie wartości sentymentu dla wartości overal reView 1,2,3,4,5 dla różnych słowników (3) i parametrów valence_shifters (3) i porównać różnice.

    1 avg(sent1)
    2 avg(sent2)
    3 avg(sent3)
    4 avg(sent4)
    5 avg(sent5)

- Porównać wartości sentymentu, histogramy dla dwóch wybranych, konkurencyjnych produktów

- Wyznaczyć udział poszczególnych emocji w zbiorze komentarzy dla dwóch wybranych produktów

- Wybrać produkt z dużą liczbą ocen, dokonać podziału na miesiące, lub dni w zależności od liczby ocen i przeanalizować zmiany sentymentu w czasie **negative/positive**