SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 7
Data 26.05.2023
Temat: Problemy NLP w uczeniu maszynowym
Wariant 1

Rafał Klinowski
Informatyka II stopień,
Stacjonarne,
1 semestr,
Gr. a

1. Polecenie: Wariant 1

Zadanie dotyczy analizy tekstu, w tym listę częstotliwości słów, budowanie chmury słów, kojarzeń, sentiment analysis, emotion analysis, bigramów, grafów powiązań. Warianty zadania są określone tekstem w języku angielskim umieszczonym na portalu `en.wikipedia.org` (główna część artykułu bez literatury)

1. `https://en.wikipedia.org/wiki/Machine_learning`

2. Wprowadzane dane:

Dane tekstowe zostały pobrane „ręcznie” ze strony https://en.wikipedia.org/wiki/Machine_learning i zapisane do pliku tekstowego „Machine learning.txt”.

3. Wykorzystane komendy:

Poniżej można znaleźć wszystkie wykorzystane komendy:

```
# Autor: Rafal Klinowski, wariant: 1.

setwd('C:\\Users\\klino\\Pulpit\\Studia magisterskie\\APU\\Lab7')

# I
# Instalacja i zaimportowanie niezbednych pakietow
# install.packages(pkgs=c("tm", "SnowballC", "wordcloud", "RColorBrewer", "syuzhet", "ggplot2"))
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")

# Zaladowanie pliku - tekstu z wikipedii
# Strona: https://www.wikiwand.com/en/Machine_learning
text <- readLines("Machine_learning.txt")
TextDoc <- Corpus(VectorSource(text))

# Wyczyszczenie tekstu
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
TextDoc <- tm_map(TextDoc, toSpace, "^a“")
TextDoc <- tm_map(TextDoc, toSpace, ":")
TextDoc <- tm_map(TextDoc, toSpace, ";")
TextDoc <- tm_map(TextDoc, toSpace, ",")
TextDoc <- tm_map(TextDoc, content_transformer(tolower))
TextDoc <- tm_map(TextDoc, removeNumbers)
```

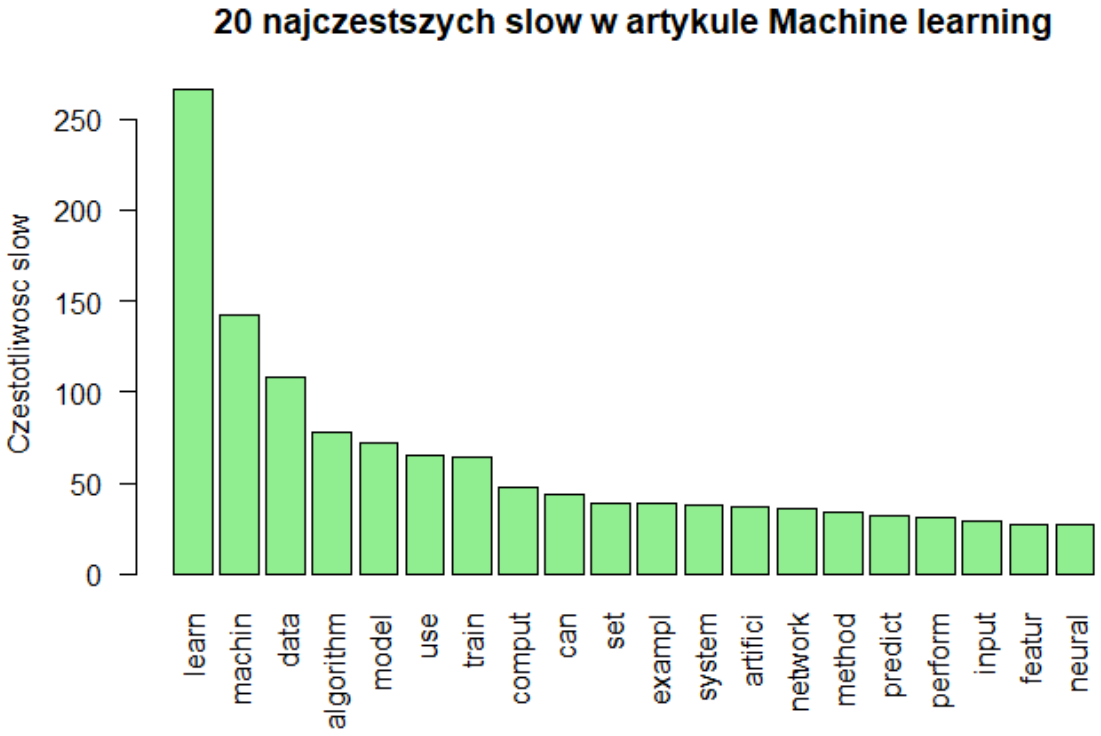
```
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
TextDoc <- tm_map(TextDoc, removeWords, c("s", "company", "team"))
TextDoc <- tm_map(TextDoc, removePunctuation)
TextDoc <- tm_map(TextDoc, stripWhitespace)
TextDoc <- tm_map(TextDoc, stemDocument)
TextDoc <- tm_map(TextDoc, content_transformer(
  function(x) gsub(x, pattern = "mathemat", replacement = "math")))
TextDoc <- tm_map(TextDoc, content_transformer(
  function(x) gsub(x, pattern = " r ", replacement = " Rlanguage ")))

# Budowanie macierzy dokumentu
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)

# Pokazanie 5 najczestszych slow
head(dtm_d, 5)
# learn, machin, data, algorithm, model

barplot(dtm_d[1:20,]$freq, las = 2, names.arg = dtm_d[1:20,]$word,
        col ="lightgreen",
        main ="20 najczestszych slow w artykule Machine learning",
        ylab = "Czestotliwosc slow")

# Chmura slow
set.seed(1234)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, scale=c(5,0.5),
          min.freq = 1,
          max.words=100, random.order=FALSE,
          rot.per=0.40,
          colors=brewer.pal(8, "Dark2"))

# Kojarzenie slow
findAssocs(TextDoc_dtm, terms = findFreqTerms(TextDoc_dtm, lowfreq = 30),
           corlimit = 0.5)

# Analiza sentymentu
# syuzhet
syuzhet_vector <- get_sentiment(text, method="syuzhet")
head(syuzhet_vector)
summary(syuzhet_vector)

# bing
bing_vector <- get_sentiment(text, method="bing")
head(bing_vector)
summary(bing_vector)

# affin
afinn_vector <- get_sentiment(text, method="afinn")
head(afinn_vector)
summary(afinn_vector)

rbind(
  sign(head(syuzhet_vector)),
```

```
  sign(head(bing_vector)),
  sign(head(afinn_vector))
)

# W naszym przypadku tylko metoda syuzhet dala jakiekolwiek efekty

# Analiza emocji
d<-get_nrc_sentiment(as.vector(dtm_d$word)) # Analiza trwa bardzo dlugo
head (d,10)

td<-data.frame(t(d))
td_new <- data.frame(rowSums(td[1:56]))
names(td_new)[1] <- "count"
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2<-td_new[1:8,]
quickplot(sentiment, data=td_new2, weight=count, geom="bar", fill=sentiment,
        ylab="count")+ggtitle("Survey sentiments")

barplot(
  sort(colSums(prop.table(d[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Text", xlab="Percentage"
)

# II
# Grafy powiazan
# install.packages(pkgs=c("tidytext", "igraph", "ggraph"))
library("tidytext")
library("igraph")
library("ggraph")

fileName <- "Machine_learning.txt"
text <- readChar(fileName, file.info(fileName)$size)
library(dplyr)
text_df <- data_frame(line = 1, text = text)
text_df

library(tidytext)
tidy_text <- text_df %>%
  unnest_tokens(word, text)
data(stop_words)
stop_words <- rbind(stop_words,de)

tidy_text <- tidy_text %>%
  anti_join(stop_words)

tidy_text %>%
  count(word, sort = TRUE)

# Tworzenie bigramow

text_bigrams <- text_df %>%
```

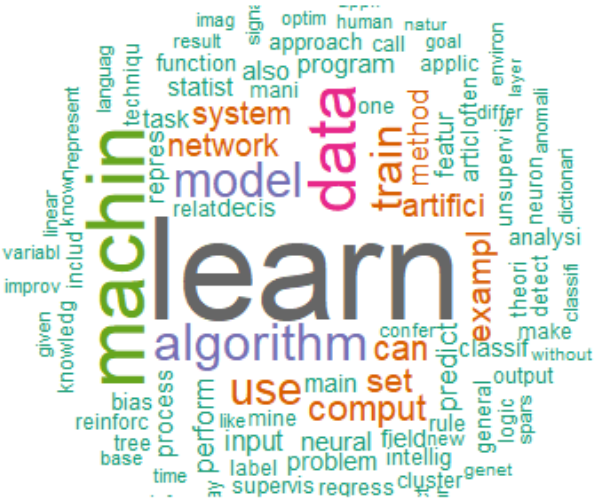
```
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
text_bigrams
text_bigrams %>%
  count(bigram, sort = TRUE)

library(tidyr)
bigrams_separated <- text_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
bigram_counts

bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")
bigrams_united
```

4. Wynik działania:

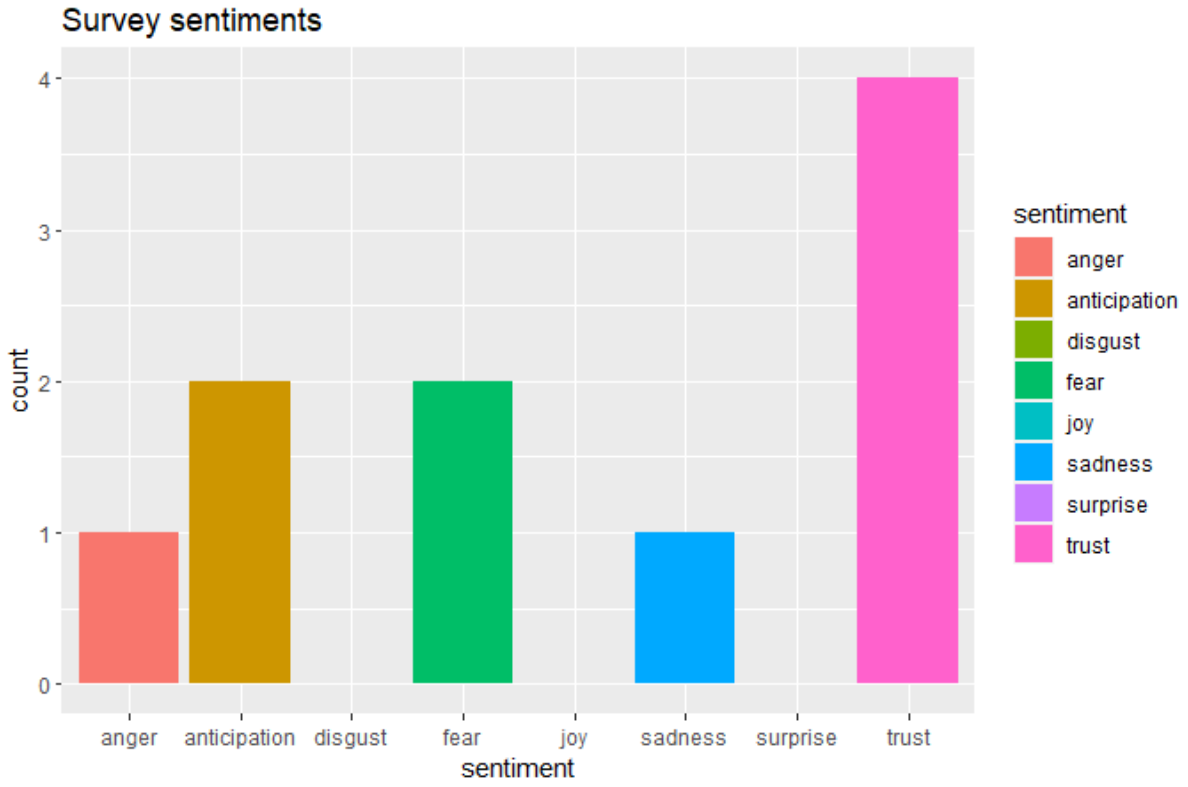
Wyniki poleceń w konsoli można znaleźć w pliku „wyniki z konsoli.txt”, link do repozytorium poniżej.



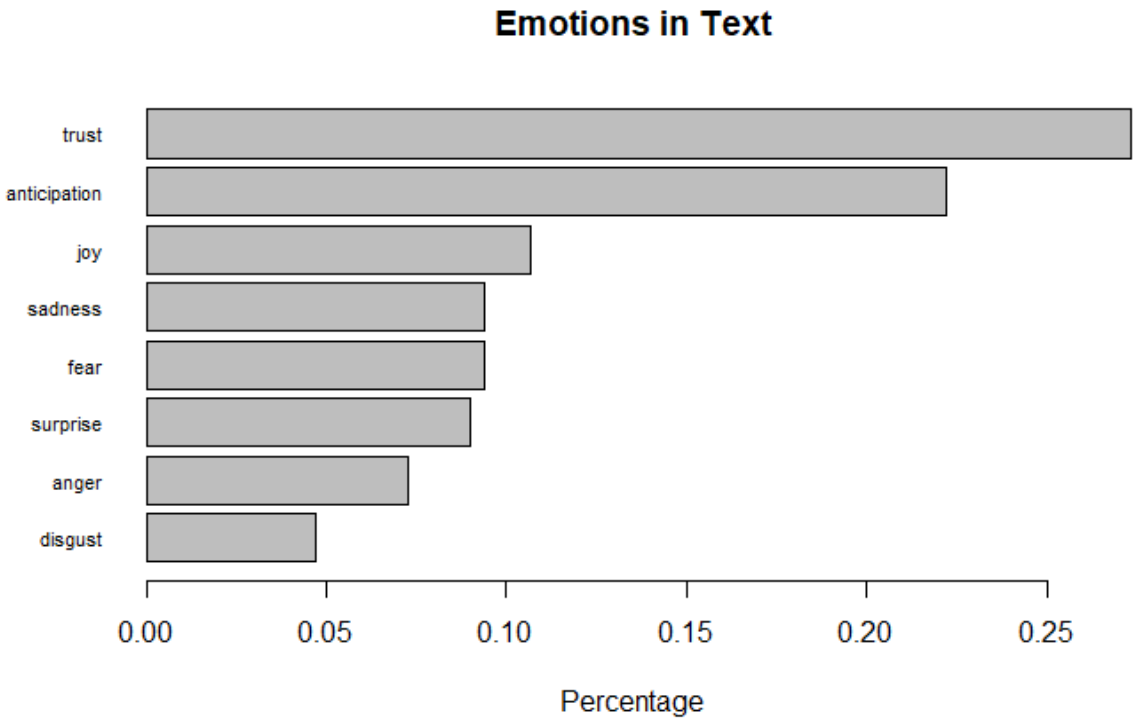
Rysunek 1. 20 najczęściej występujących w tekście słów.



Rysunek 2. Chmura słów z tekstu o Machine learning.



Rysunek 3. Analiza emocji w tekście – wykres 1.



Rysunek 4. Analiza emocji w tekście – wykres 2.

	word	n
	<chr>	<int>
1	learning	252
2	machine	137
3	data	109
4	training	49
5	algorithms	46
6	model	39
7	artificial	37
8	set	33
9	models	27
10	neural	27

Rysunek 5. Kilka najczęściej występujących słów – analiza przy tworzeniu bigramów.

	bigram	n
	<chr>	<int>
1	machine learning	123
2	of the	40
3	in the	33
4	is a	26
5	learning algorithms	26
6	can be	21
7	of machine	20
8	learning is	19
9	main article	19
10	of a	19

Rysunek 6. Kilka najczęściej występujących bigramów.

	word1	word2	n
	<chr>	<chr>	<int>
1	machine	learning	123
2	learning	algorithms	26
3	main	article	19
4	supervised	learning	17
5	training	data	14
6	data	mining	13
7	unsupervised	learning	13
8	artificial	intelligence	12
9	neural	networks	12
10	artificial	neural	10

Rysunek 7. Najczęściej występujące bigramy po filtrowaniu.

Link do repozytorium: https://github.com/Stukeley/APU_Lab7

5. Wnioski:

Realizacja laboratorium związanego z przetwarzaniem tekstu była wyjątkowo prosta i przyjemna w środowisku R – wykorzystane biblioteki pozwalały na łatwe przetwarzanie tekstu (w tym jego oczyszczenie) oraz niemal automatyczne wyświetlanie istotnych informacji na ekranie w formie wykresów lub list.

Jedynym problemem napotkanym podczas realizacji ćwiczenia była konieczność ręcznego pobrania zawartości stron internetowych – z dokumentacji wynikało, że funkcje operują wyłącznie na plikach, niemożliwe było więc podanie adresu URL, z którego należy pobrać zawartość strony.