**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

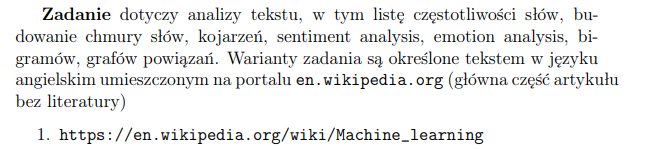
Informatyka II stopień,

Stacjonarne,

1 semestr,

Gr. a

1. **Polecenie: Wariant 1**



1. **Wprowadzane dane:**

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

1. **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, "ˆa“")

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

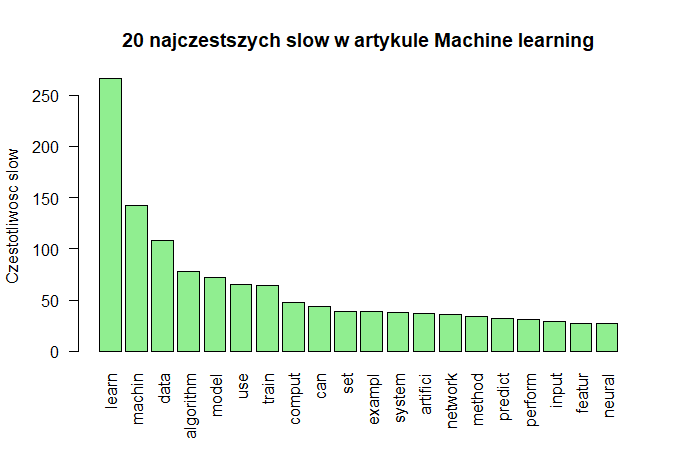
bigrams\_united <- bigrams\_filtered %>%

unite(bigram, word1, word2, sep = " ")

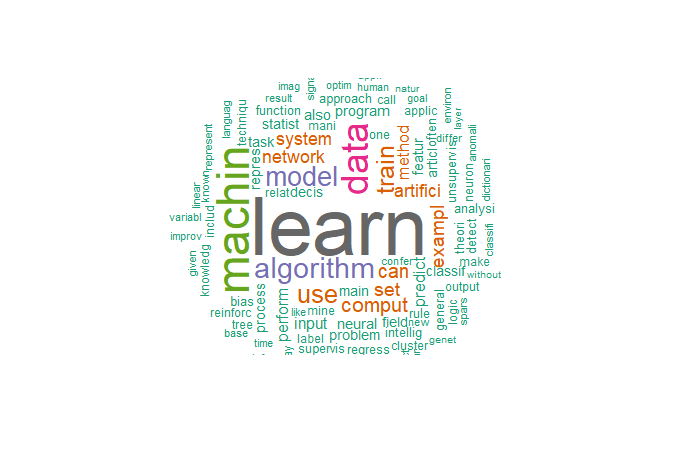
bigrams\_united

**4. Wynik działania:**

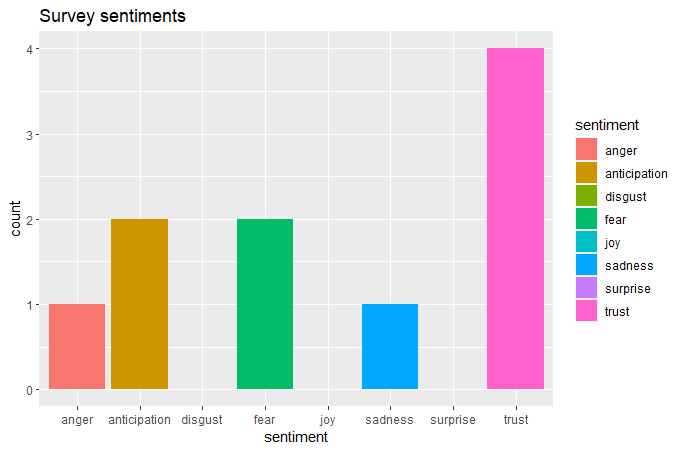
Wyniki poleceń w konsoli można znaleźć w pliku „wyniki z konsoli.txt”, link do repozytorium poniżej.



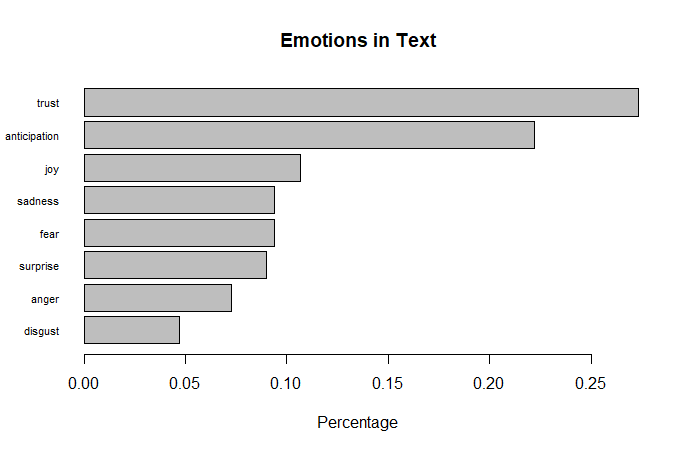
Rysunek . 20 najczęściej występujących w tekście słów.



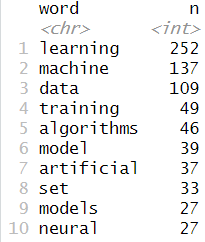
Rysunek . Chmura słów z tekstu o Machine learning.



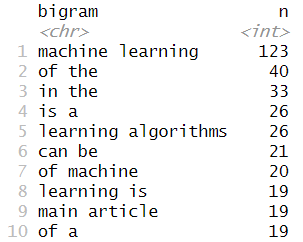
Rysunek . Analiza emocji w tekście – wykres 1.



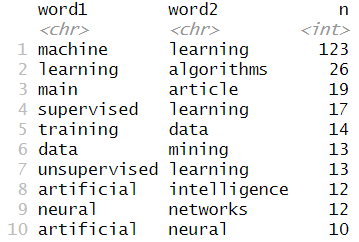
Rysunek . Analiza emocji w tekście – wykres 2.



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



Rysunek . Kilka najczęściej występujących bigramów.



Rysunek . Najczęściej występujące bigramy po filtrowaniu.

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

1. **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.