SPRAWOZDANIE

Zajęcia: Analiza procesów uczenia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 7 Data 26.05.2023 Temat: Problemy NLP w uczeniu maszynowym Wariant 4	Maksymilian Grygiel Informatyka II stopień, stacjonarne, Semestr I, gr.1a

Link do repozytorium: https://github.com/Maksiolo20/APU

Zadania:

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**): https://en.wikipedia.org/wiki/History_of_poetry

Wykonanie zadania:

Instalacja pakietów:

```
> install.packages("tm")
> library(tm)
> install.packages("SnowballC")
> library(SnowballC)
> install.packages("wordcloud")
> library(wordcloud)
> install.packages("RColorBrewer")
> library(RColorBrewer)
> install.packages("syuzhet")
> library(syuzhet)
> install.packages("ggplot2")
> library(ggplot2)
```

Odczytanie tekstu:

```
> text <- readLines("poetryHistory.txt", warn=FALSE)
```

Konwersja tekstu do obiektu:

```
> TextDoc <- Corpus(VectorSource(text))
```

Czyszczenie tekstu ze zbędnych znaków:

Uswanie specjalnych znaków:

```
> toSpace <- content_transformer(function(x,pattern) gsub(pattern,"",x))
Warning message:
In mget(objectNames, envir = ns, inherits = TRUE) :
  strings not representable in native encoding will be translated to UTF-8
Warning message:
In mget(objectNames, envir = ns, inherits = TRUE) :
  strings not representable in native encoding will be translated to UTF-8
> TextDoc<- tm_map(TextDoc,toSpace,"/")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, "/") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,"@")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, "@") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,"\\|")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, "\\|") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,":")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, ":") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,";")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, ";") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,",")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, ",") :
  transformation drops documents
> TextDoc<- tm_map(TextDoc,toSpace,"/")
Warning message:
In tm_map.SimpleCorpus(TextDoc, toSpace, "/") :
  transformation drops documents
>
```

Usuwanie liczb:

```
> TextDoc <- tm_map(TextDoc,removeNumbers)
```

Usuwanie znaków stop:

```
> TextDoc <- tm_map(TextDoc,removeWords, stopwords("english"))
```

Usuwanie tajnych znaków:

```
> TextDoc <- tm_map(TextDoc,removeWords,c("\\[","\\]"))
```

Usuwanie znaków interpunkcyjnych:

```
> TextDoc <- tm_map(TextDoc,removePunctuation)
```

Usuwanie białych przestrzeniL

```
> TextDoc <- tm_map(TextDoc,stripWhitespace)
```

Zmiana do formy bazowej:

```
> TextDoc <- tm_map(TextDoc,stemDocument)
```

Budowanie macierzy tekstowej:

```
> TextDoc_dtm<-TermDocumentMatrix(TextDoc)
> dtm_m <- as.matrix(TextDoc_dtm)
```

Sortowanie malejąco bazując na tym, jak częst słowo się pojawia:

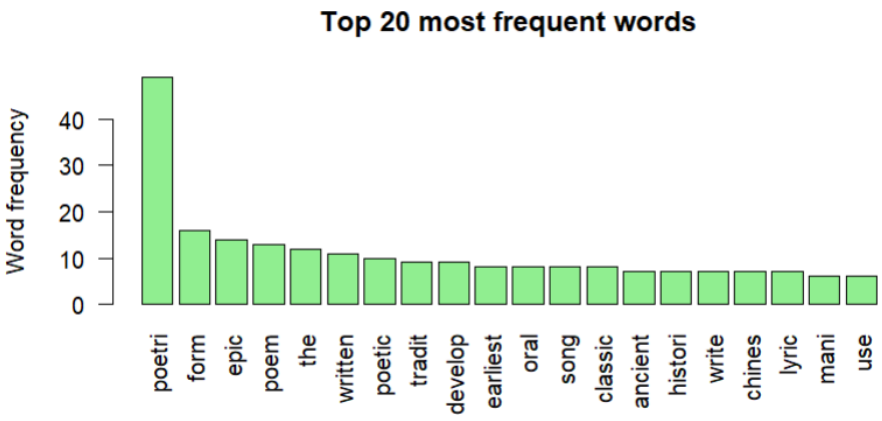
```
> dtm_v <-sort(rowSums(dtm_m), decreasing = TRUE)
> dtm_D <- data.frame(word=names(dtm_v),freq=dtm_v)
```

Pokaż 5 najczęstszych słów:

```
> head(dtm_D,5)
         word freq
poetri poetri   49
form     form   16
epic     epic   14
poem     poem   13
the       the   12
```

Najczęstsze słowa - wykres:

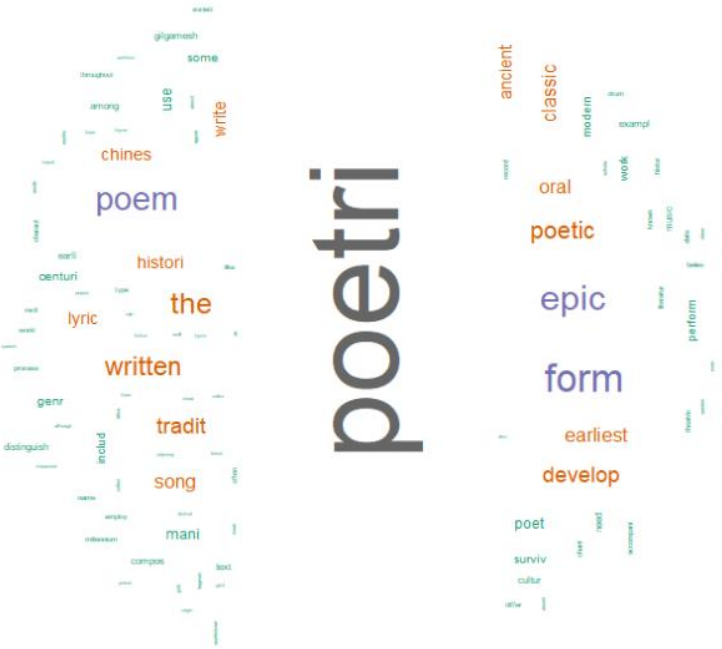
```
> barplot(
+     dtm_D[1:20,]$freq,
+     las=2,
+     names.arg = dtm_D[1:20,]$word,
+     col="lightgreen",
+     main="Top 20 most frequent words",
+     ylab = "Word frequency"
+ )
```



Wygenerowanie chmury słów:

```
> set.seed(1234)
> wordcloud(
+     words=dtm_D$word,
+     freq = dtm_D$freq,
+     scale=c(5,0,0),
+     min.freq = 1,
+     max.words = 100,
+     random.order = FALSE,
+     rot.per = 0.40,
+     colors = brewer.pal(8,"Dark2")
+ )
```


poetr
ancient
classic
modern
oral
poetic
epic
form
earliest
develop
poet
perform
some
use
write
chines
poem
histori
the
lyric
written
genr
tradit
song
mani


poetri
some
use
write
chines
poem
histori
the
lyric
written
genr
tradit
song
mani
ancient
classic
modern
oral
poetic
epic
form
earliest
develop
poet
perform

Kojarzenia słów:

```
> findAssocs(
+     TextDoc_dtm,
+     term=c("learn","machine","algorithm","train"),
+     corlimit = 0.5
+ )
$learn
numeric(0)

$machine
numeric(0)

$algorithm
numeric(0)

$train
numeric(0)
```

Znalezienie kojarzeń słów które pojawiają się co najmniej 20 razy:

```
> findAssocs(
+     TextDoc_dtm,
+     terms=findFreqTerms(TextDoc_dtm,lowfreq = 20),
+     corlimit = 0.5
+ )
$poetri
throughout      often       type      addit   earliest       form       work        way       rule
      0.74       0.70       0.65       0.65       0.61       0.60       0.60       0.59       0.59
    period     poetic       text       time    popular       this    histori      world     differ
      0.59       0.58       0.54       0.54       0.54       0.54       0.53       0.53       0.53
    employ       mani
      0.50       0.50

>
```

Analiza sentymentu:

```
> syuzhet_vector <- get_sentiment(text,method="syuzhet")
> bing_vecor <- get_sentiment(text, method="bing")
> nrc_vector <- get_sentiment(text,method = "nrc")
>
```

Porównanie analizy:

```
> rbind(
+     sign(head(syuzhet_vector)),
+     sign(head(bing_vecor)),
+     sign(head(nrc_vector))
+ )
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    1    0    1    0
[2,]    1    0    1    0   -1    0
[3,]    1    0    1    0    1    0
>
```

Klasyfikacja emocji w tekście:

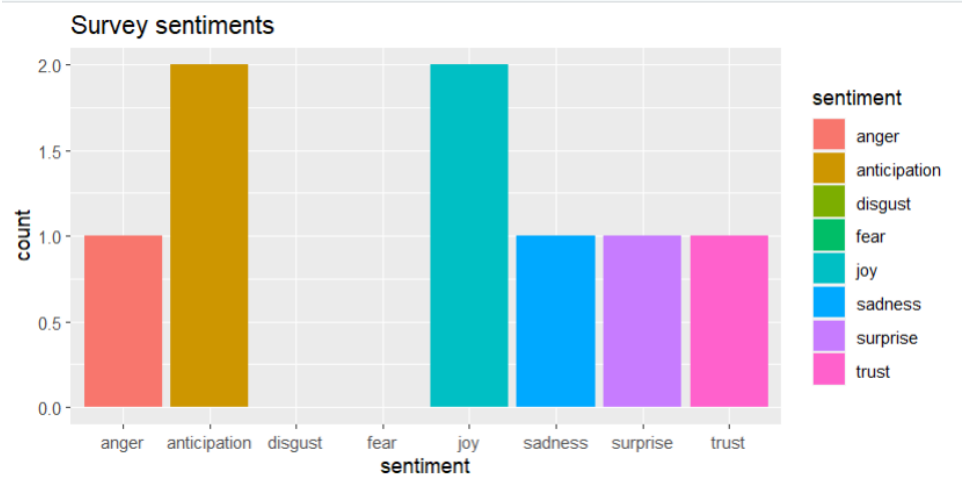
```
> d <- get_nrc_sentiment(as.vector(dtm_D$word))
Warning message:
`spread_()` was deprecated in tidyr 1.2.0.
i Please use `spread()` instead.
i The deprecated feature was likely used in the syuzhet package.
  Please report the issue to the authors.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> head(d,10)
   anger anticipation disgust fear joy sadness surprise trust negative positive
1      0            0       0    0   0       0        0     0        0        0
2      0            0       0    0   0       0        0     0        0        0
3      0            0       0    0   0       0        0     0        0        1
4      0            0       0    0   0       0        0     0        0        0
5      0            0       0    0   0       0        0     0        0        0
6      0            0       0    0   0       0        0     0        0        0
7      0            0       0    0   0       0        0     0        0        0
8      0            0       0    0   0       0        0     0        0        0
9      0            1       0    0   0       0        0     0        0        1
10     0            0       0    0   0       0        0     0        0        0
```

Transpozycja, czyszczenie wyników, sumowanie częstotliwości emocji w pierwszych 56 słowach:

```
> 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]
Error in `[.data.frame`(td_new, 1:8) : nie wybrano kolumn
> td_new2<-td_new[1:8,]
```

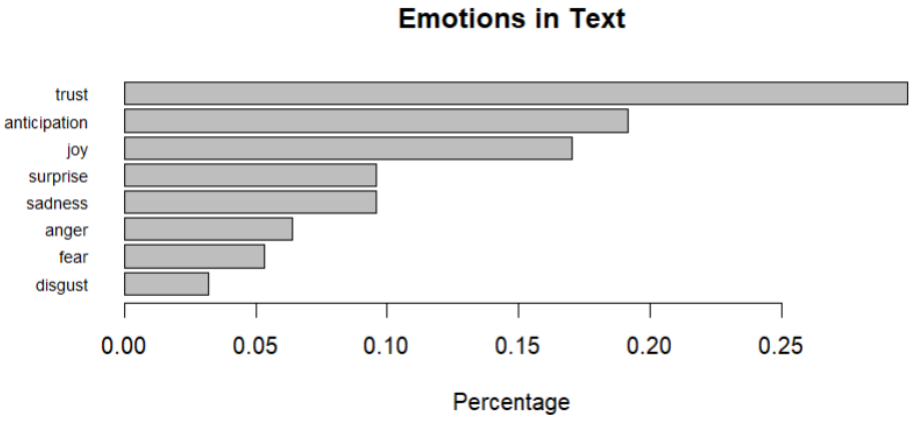
Drugi wykres - słowa przekazujące emocje:

```
> quickplot(
+      sentiment,
+      data=td_new2,
+      weight=count,
+      geom="bar",
+      fill=sentiment,
+      ylab="count"
+ )+ggtitle("Survey sentiments")
Warning message:
```



Trzeci wykres – procent emocji:

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



Wnioski:

Sprawozdanie z R Studio na temat problemów związanych z przetwarzaniem języka naturalnego (NLP) w uczeniu maszynowym wykazało, że NLP jest obszarem o unikalnych wyzwaniach, takich jak zrozumienie i generowanie tekstu, analiza sentymentu oraz tłumaczenie maszynowe. Wnioskiem z tego sprawozdania jest, że skuteczne stosowanie uczenia maszynowego w obszarze NLP wymaga zastosowania specjalistycznych technik i narzędzi, takich jak tokenizacja, wektoryzacja tekstu czy modele językowe. Ponadto, wykorzystanie gotowych zbiorów danych oraz dostęp do mocnych obliczeniowo środowisk, takich jak R Studio, jest kluczowe dla skutecznego modelowania NLP.