

subtitle_text_mining

Adam Kolipiński, Ludwik Przyrowski

27 maja 2017

biblioteki

```
# calculations
library(tm)
library(dplyr)
library(SnowballC)

#visualization
library(wordcloud2)
```

wstęp

Celem projektu jest znalezienie najbardziej pozytywnych filmów ze zbioru. Źródło dialogów pochodzi z serwisu <https://nlds.soe.ucsc.edu/fc2>. Pliki podzielone są na kategorie:

```
path = file.path(getwd(), "dialogs/")
category_list = dir(path)
category_list

## [1] "Action"      "Adventure"    "Animation"   "Biography"   "Comedy"
## [6] "Crime"       "Drama"        "Family"      "Fantasy"     "Film-Noir"
## [11] "History"     "Horror"       "Music"       "Musical"     "Mystery"
## [16] "Romance"     "Sci-Fi"       "Short"       "Sport"       "Thriller"
## [21] "War"         "Western"
```

W celu przyśpieszenia procesu ograniczono się tylko do części kategorii.

```
path = file.path(getwd(), "dialogs_selected/")
category_list = dir(path)
category_list

## [1] "Action"      "Adventure"    "Biography"   "Comedy"     "Crime"
## [6] "Drama"       "Family"      "Fantasy"     "Horror"     "Sci-Fi"
## [11] "Short"
```

wszystkie filmy w podkategoriach zostały zainportowane do “korpusu”

```
corpus <- Corpus(DirSource(path, recursive=T))
```

W uzyskanym źródle imiona bohaterów oraz opisy sA pisane samymi dużymi literami. Dlatego napisana została niestandardowa funkcja dla preprocessingu, usuwająca takie wystąpienia Usunięte będzie w ten sposób również kilku okrzyków ale ich wpływ uznany jest za nieznaczny.

```
remAllCap <- function (x){gsub("\b[A-Z]+\b", "", x)}
corpus <- tm_map(corpus, remAllCap)
```

Zastosowano serię narzędzi do odpowiedniej obróbki wstępnej dialogów oraz zamaskowano bardzo popularnego angielskiego przekleństwa:

```

corpus <- tm_map(corpus, tolower)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, c(stopwords("english")))
remSwear <- function(x){gsub("fuck", "f**k", x)}
corpus <- tm_map(corpus, remSwear)
corpus_org <- corpus
corpus_org <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, stemDocument)
corpus <- tm_map(corpus, stripWhitespace)

```

Utworzenie Macierzy wyrażenie-dokument

```
tdm <- TermDocumentMatrix(corpus)
```

analiza pozytywnego lub negatywnego znaczenia dialogów na podstawie występowania słów pozytywnych lub negatywnych.

Użyta została lekko zmodyfikowana funkcja przerabiana na zajęciach. Przerobiony został wynik funkcji jako różnica udziału procentowego pozytywnych i negatywnych słów do wszystkich negatywnych i pozytywnych słów. Za słowniki negatywnych i pozytywnych słów zostały urzyta baza prezentowana w instrukcji do zajęć.

```

hu.liu.pos = scan(file.path(getwd(), "opinion-lexicon-English","positive-words.txt"),
                  what='character', comment.char=';')
hu.liu.neg = scan(file.path(getwd(), "opinion-lexicon-English","negative-words.txt"),
                  what='character', comment.char=';')

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  require(plyr)
  require(stringr)
  scores = laply(sentences, function(sentence, pos.words, neg.words) {
    word.list = str_split(sentence, '\\s+')
    words = unlist(word.list)
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)
    score = c(sum(pos.matches)/length(words), sum(neg.matches)/length(words))
    return(score)
  }, pos.words, neg.words, .progress=.progress )
}

```

Funkcja została użyta do wcześniej zaczytanego ciała:

```

max = length(list.files(path=path, recursive = T))

i <- 1
names = c()
pos = c()
neg=c()

while(i<=max){
  sample.text = corpus[[i]]$content
  result = score.sentiment(sample.text, hu.liu.pos , hu.liu.neg)
  names = c(names, gsub("_dialog.txt","",corpus[[i]]$meta$id))
}

```

```

pos = c(pos, result[1])
neg = c(neg, result[2])
i=i+1
}

```

Wyniki ograniczono tylko do tych filmów w których pozytywne i negatywne wyrażenia stanowiły co najmniej 5% wszystkich wyrażeń.

```

df = data.frame(names, pos, neg )
df <- df %>% filter(pos >0.5 & neg >0.5) %>% distinct() %>% mutate(per_pos = pos/(pos+neg))
df.pos <- df %>% arrange(desc(per_pos))
df.neg <- df %>% arrange(per_pos)

```

W ten sposób udało się wyodrębnić najbardziej pozytywne filmy w zbiorze:

```
top_n(df.pos, 15)
```

```

## Selecting by per_pos

## [1] names    pos      neg      per_pos
## <0 rows> (or 0-length row.names)

```

Oraz najbardziej negatywne filmy w zbiorze:

```
top_n(df.neg, -15)
```

```

## Selecting by per_pos

## [1] names    pos      neg      per_pos
## <0 rows> (or 0-length row.names)

```

Ostatnim etapem jest wizualizacja częstości występowania słów poprzez chmurę wyrazów. Całość przygotowania została zamknięta w postaci funkcji:

```

for.cloud = function(name, names, corpus_org){
  id = match(name, names)
  print(corpus_org[[id]]$meta$id)
  print(id)
  ##POS tagging
  library(NLP)
  library(openNLP)
  library(tm)
  sent_token_annotator <- Maxent_Sent_Token_Annotator()
  word_token_annotator <- Maxent_Word_Token_Annotator()
  sample.text = corpus_org[[id]]$content
  a1 = annotate(sample.text,list(sent_token_annotator,word_token_annotator))
  pos_tag_annotator <- Maxent_POS_Tag_Annotator()
  a3 = annotate(sample.text, pos_tag_annotator, a1)
  a3w = subset(a3, type=='word')
  max = length(a3w)
  k = 1
  words = c()
  while(k<=max){
    p = unlist(a3w[k]$features)
    if(p=="NN" || p=="VB"){
      word <- substr(sample.text,a3w[k]$start, a3w[k]$end)
    }
    words = c(words, word)
  }
}

```

```

    k = k + 1
}
words= words[words!="m" & words!="ll" & words!="ve" & words!="dont"]
tb <- as.data.frame(table(words))
colnames(tb) <- c('word','freq')
tb <- tb %>% arrange(desc(freq))
return(tb)
}

```

Poniżej przykład jednego z pozytywnych filmów: Amadeus

```
words.pos <- for.cloud('amadeus', names, corpus_org)
```

```
## [1] "amadeus_dialog.txt"
## [1] 924
#path.png = file.path(getwd(), "sample pictures/play.png")
#wordcloud2(data = words.neg, figPath = path.png, size = 1.5)
```

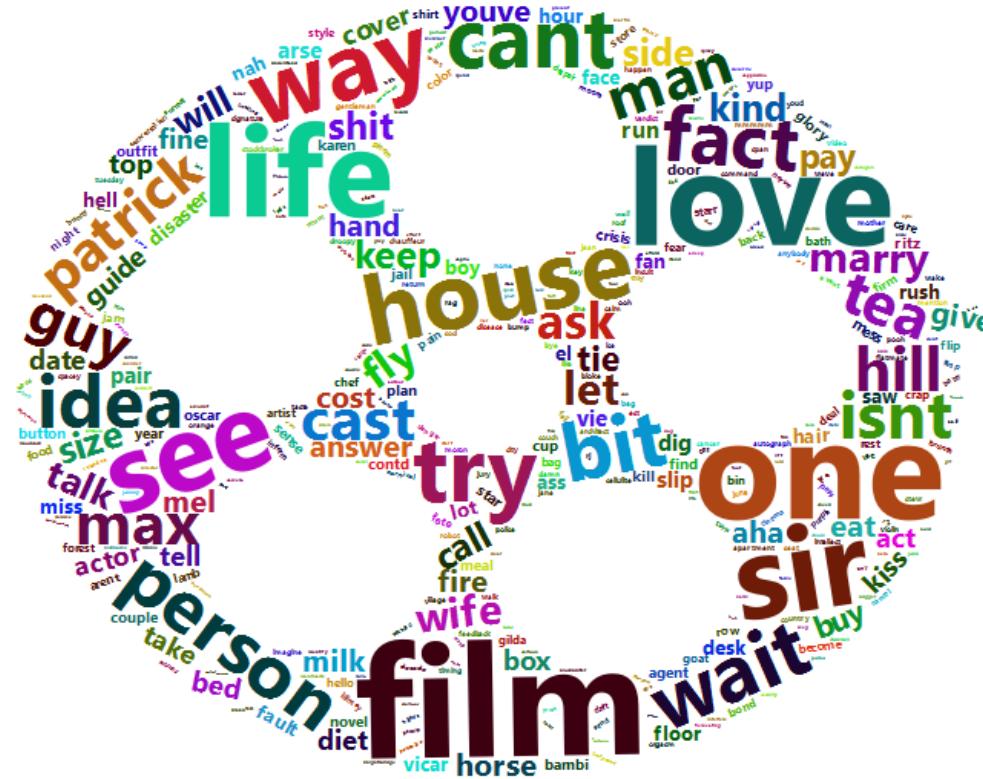


oraz negatywny “Pine Apple Express”

```
#words.neg <- for.cloud('pineappleexpress', names, corpus_org)
path.png = file.path(getwd(), "sample pictures/movie.png")
#wordcloud2(data = words.pos, figPath = path.png, size = 1)
```



Figure 1: Caption for the picture.



Oraz bonusowy pozytywn "Notting Hill"