

SENTIMENT ANALYSIS PENGGUNA TWITTER TENTANG #JOKOWISILAHKANMUNDUR

Revanto Alif Nawasta (123170060) - Taufiqul Aptiyan Bagaskara (123170066)

5/31/2020

SCRAPING DATA DARI TWITTER DAN SCORING DATA TWEET

1. Pengumpulan Data Tweet Dari Twitter Memanggil library yang digunakan untuk pengambilan data tweet dari Twitter.

```
library(twitteR)
library(ROAuth)
```

2. Mengambil Key Dari Twitter dan Test Key tersebut Masuk ke dalam akun twitter developer dan ambil key API dari Twitter dan kemudian test key tersebut apakah bisa dijalankan atau tidak.

```
access_secret<-"MivFN08rsUlqnown3D5yYBgHqlSZ0hssVHIhZL7rFHoDT"
access_token<-"1471672230-7TKNN3cuuFG1G8JlyR00bwy3schnnxu50z4GuCB"
consumer_key<-"q4Yccg2KIno1Pff0d0sKbA2RB"
consumer_secret<-"OPJgaCurPorQXkScLIQcFmcE0bBNprGzwqo4W4e6EuIzQf4CV7"
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

```
## [1] "Using direct authentication"
```

3. Cari Tweet Yang Akan Dicari dan Kemudian Disimpan Hasil Pencarian Tersebut Data tweet dicari dengan memasukkan keyword dan juga batas tweet yang ingin diambil.

```
jokowi_tweets = searchTwitter("jokowisilahkanmundur", n=1000, lang="id")
```

```
## Warning in doRppAPICall("search/tweets", n, params = params, retryOnRateLimit =
## retryOnRateLimit, : 1000 tweets were requested but the API can only return 409
```

4. Pembersihan data tweet Data tweet dibersihkan dengan menghilangkan titik,koma, dll. dan juga tweet diubah ke non kapital.

```
jokowiTweets <- sapply(jokowi_tweets, function(x) x$getText())

catch.error = function(x)
{
  y = NA
}
```

```

catch_error = tryCatch(tolower(x), error=function(e) e)

if (!inherits(catch_error, "error"))
  y = tolower(x)

return(y)
}

cleanTweets <- function(tweet) {
  #remove html links:
  tweet = gsub("(f|ht)(tp)(s?)(:|/|)(.*)[.|/|](.*)", " ", tweet)
  #remove retweet entities:
  tweet = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", tweet)
  #remove #hashtags:
  tweet = gsub("#\\w+", " ", tweet)
  #remove all "@people":
  tweet = gsub("@\\w+", " ", tweet)
  #remove all punctuations:
  tweet = gsub("[[:punct:]]", " ", tweet)
  #remove numbers, kita hanya butuh teks untuk analytics
  tweet = gsub("[[:digit:]]", " ", tweet)
  #remove unnecessary spaces (white spaces, tabs, etc)
  tweet = gsub("[ \\t]{2,}", " ", tweet)
  tweet = gsub("^\\s+|\\s+$", "", tweet)
  #remove amp
  tweet = gsub("&", " ", tweet)
  #remove crazy character
  tweet = gsub("[^a-zA-Z0-9]", " ", tweet)
  #remove alphanumeric
  tweet = gsub("[^:alnum:]", " ", tweet)

  #ubah semua kata menjadi lowercase:
  tweet = catch_error(tolower(tweet))
  tweet
}

cleanTweetsAndRemoveNAs <- function(Tweets) {
  TweetsCleaned = sapply(Tweets, cleanTweets)
  #remove "NA" tweets:
  TweetsCleaned = TweetsCleaned[!is.na(TweetsCleaned)]
  names(TweetsCleaned) = NULL
  #remove repetitive tweets:
  TweetsCleaned = unique(TweetsCleaned)
  TweetsCleaned
}

tjokowiCleaned = cleanTweetsAndRemoveNAs(jokowiTweets)

```

5. Menyimpan Data Tweet Ke File .csv Data tweet yang sudah dibersihkan kemudian disimpan kedalam file csv.

```
write.csv(tjokowiCleaned, file = "tjokowiCleaned.csv")
```

6. Memisahkan Kata Positif dan Negatif Dari Data Tweet Data tweet yang sudah bersih dipisahkan

dengan menggunakan fungsi lexicon yang mengambil kata positif dari file “s-pos.txt” dan kata negatif dari file “s-neg.txt”.

```
opinion.lexicon.pos = scan("s-pos.txt", what = "character", comment.char = ";")
opinion.lexicon.neg = scan("s-neg.txt", what = "character", comment.char = ";")

pos.words = c(opinion.lexicon.pos)
neg.words = c(opinion.lexicon.neg)
```

7. Membuat Scoring Untuk Data Tweet Data tweet akan discoring dari hasil pemilahan kata positif dan negatif.

```
getSentimentScore = function(sentences, pos.words, neg.words, .progress = "none")
{
  require(plyr)
  require(stringr)
  scores = laply(sentences, function(sentence, pos.words, neg.words) {
    #remove digit, punctuation, dan special/control character:
    sentence = gsub("[[:cntrl:]]", "", gsub("[[:punct:]]", "", gsub("\\d+", "", sentence)))
    #convert semua teks menjadi lowercase:
    sentence = tolower(sentence)
    #pisahkan setiap kalimat menggunakan spasi (space delimiter):
    words = unlist(str_split(sentence, "\\s+"))
    #lakukan boolean match dari setiap kata-kata menggunakan pos &amp;amp; neg opinion-lexicon:
    pos.matches = !is.na(match(words, pos.words))
    neg.matches = !is.na(match(words, neg.words))
    #score sentimen = total positive sentiment - total negative:
    score = sum(pos.matches) - sum(neg.matches)
    return(score)
  }, pos.words, neg.words, .progress=.progress)
  #return data frame berisi kalimat beserta sentimennya:
  return(data.frame(text = sentences, score = scores))
}

#Terapkan ke data tweet yang telah kita bersihkan:
jokowiResult = getSentimentScore(tjokowiCleaned, pos.words, neg.words)
```

```
## Loading required package: plyr
```

```
##
```

```
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:twitter':
```

```
##
```

```
## id
```

```
## Loading required package: stringr
```

8. Simpan Hasil Scoring Ke Dalam File Csv Data tweet yang sudah diberi score akan disimpan ke file csv untuk diolah ke tahap selanjutnya.

```
write.csv(jokowiResult, file = "jokowiResult.csv")
```

VISUALISASI DATA TWEET

Library yang digunakan :

```
library(sentiment)
```

```
## Loading required package: tm
```

```
## Loading required package: NLP
```

```
## Loading required package: Rstem
```

```
library(ggplot2)
```

```
##
```

```
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
```

```
##
```

```
##      annotate
```

```
library(tm)
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

1. Klasifikasi Data Tweet dan Simpan File Csv Data tweet diklasifikasikan jenis emosinya yaitu anger, disgust, fear, joy, sadness, surprise, bestfit dengan menggunakan algoritma Naive Bayes kemudian di simpan ke file csv, kemudian data tweet juga diklasifikasi polaritasnya menggunakan algoritma Naive Bayes.

```
#Classify_emotion function returns an object of class data frame with 7 columns  
#(anger, disgust, fear, joy, sadness, surprise, best_fit) and one row for each document:  
tjokowiclassemo = classify_emotion(tjokowiCleaned, algorithm="bayes", prior=1.0)
```

```
#Ganti NA dengan unknown
```

```
#pilih kolom ke-7 atau best_fit:
```

```
jokowiEmotion = tjokowiclassemo[,7]
```

```
jokowiEmotion[is.na(jokowiEmotion)] = "unknown"
```

```
#Simpan data ke csv
```

```
write.csv(jokowiEmotion, file = "jokowiEmotion.csv")
```

```
#Measure polarity (pos, neg, neutral):
```

```
tjokowiClassPol = classify_polarity(tjokowiCleaned, algorithm = "bayes")
```

```
#Fetch polarity category best_fit for our analysis purposes:
```

```
jokowiPol = tjokowiClassPol[,4]
```

```
#lihat hasilnya:  
head(jokowiPol, 5)
```

```
## [1] "positive" "positive" "positive" "positive" "positive"
```

```
#buat data frame dari hasil di atas:
```

```
jokowiSentimenDF = data.frame(text=tjokowiCleaned, emotion=jokowiEmotion, polarity=jokowiPol, stringAsFactors=TRUE)
```

```
#lihat hasilnya:
```

```
head(jokowiSentimenDF, 5)
```

```
##
```

```
## 1 giliran yg begini selalu selesai dgn kata gangguan jiwa emang tuh antek pki sdh ada uji klinis smp
```

```
## 2 giliran yg begini selalu selesai dgn kata gangguan jiwa emang tuh antek pki sdh ada uji klinis smp
```

```
## 3 kalau sudah kosong kas a
```

```
## 4 paya deh pemerintah sdh tidak waras lagi dana haji mau dijadikan pemerku
```

```
## 5 geger warga mengenakan kaos pendukung jokowi ditemukan
```

```
## emotion polarity stringAsFactors
```

```
## 1 unknown positive FALSE
```

```
## 2 unknown positive FALSE
```

```
## 3 unknown positive FALSE
```

```
## 4 unknown positive FALSE
```

```
## 5 unknown positive FALSE
```

2. Membuat Grafik dan Wordcloud

```
#function for plotting
```

```
plotSentiments1 <- function(sentiment_dataframe, title)
```

```
{
```

```
  ggplot(sentiment_dataframe, aes(x=emotion)) +
```

```
    geom_bar(aes(y=..count.., fill=emotion)) +
```

```
    scale_fill_brewer(palette="Dark2") +
```

```
    ggtitle(title) +
```

```
    theme(legend.position="right") +
```

```
    ylab("Number of Tweets") +
```

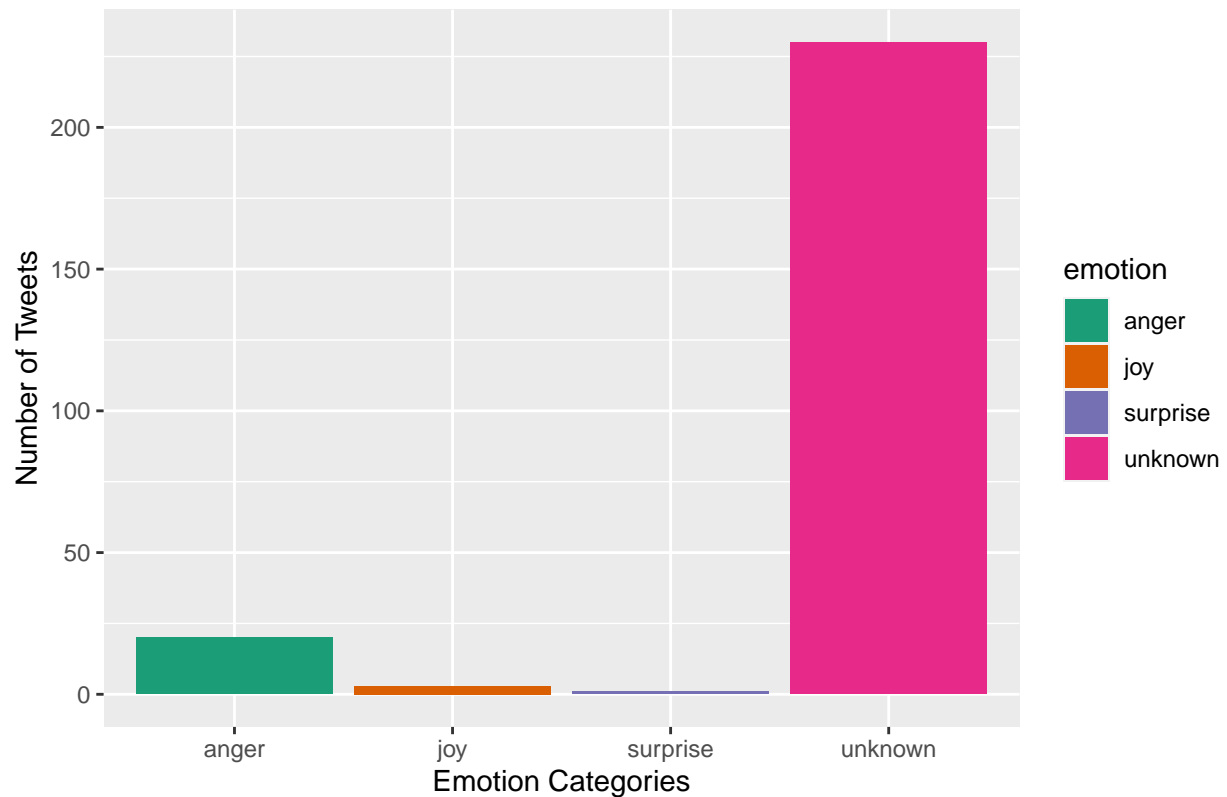
```
    xlab("Emotion Categories")
```

```
}
```

```
#plotting tweets emotions
```

```
plotSentiments1(jokowiSentimenDF, "Analisis Sentimen Dari Tweet Tentang #JokowiSilahkanMundur")
```

Analisis Sentimen Dari Tweet Tentang #JokowiSilahkanMundur



```
#tinjau polaritas sentimen (positif vs negatif), untuk melihat tingkat dukungan:
#plot distribusi polaritas tweet:
plotSentiments2 <- function(sentiment_dataframe, title)
{
  ggplot(sentiment_dataframe, aes(x=polarity)) +
    geom_bar(aes(y=..count.., fill=polarity)) +
    scale_fill_brewer(palette="RdGy") +
    ggtitle(title) +
    theme(legend.position="right") +
    ylab("Number of Tweets") +
    xlab("Polarity Categories")
}

#plotting tweets polarity
plotSentiments2(jokowiSentimenDF, "Analisis Polaritas Dari Tweet Tentang #JokowiSilahkanMundur")
```

Polarity Category	Number of Tweets
negative	15
neutral	2
positive	240

7

```
emo.docs[i] = paste(TweetsCleaned[Emotion == emos[i]], collapse=" ")
}
corpus <- Corpus(VectorSource(emo.docs))
tdm = TermDocumentMatrix(corpus)
tdm = as.matrix(tdm)
colnames(tdm) = emos
#require(wordcloud)
suppressWarnings(comparison.cloud(tdm, colors = brewer.pal(n_emos, "Dark2"), scale = c(3,.5), random.
})
getWordCloud(jokowiSentimenDF, tjokowiCleaned, jokowiEmotion)
```



```
#export ke csv
write.csv(jokowiSentimenDF, file = "jokowisentimen.csv")
```