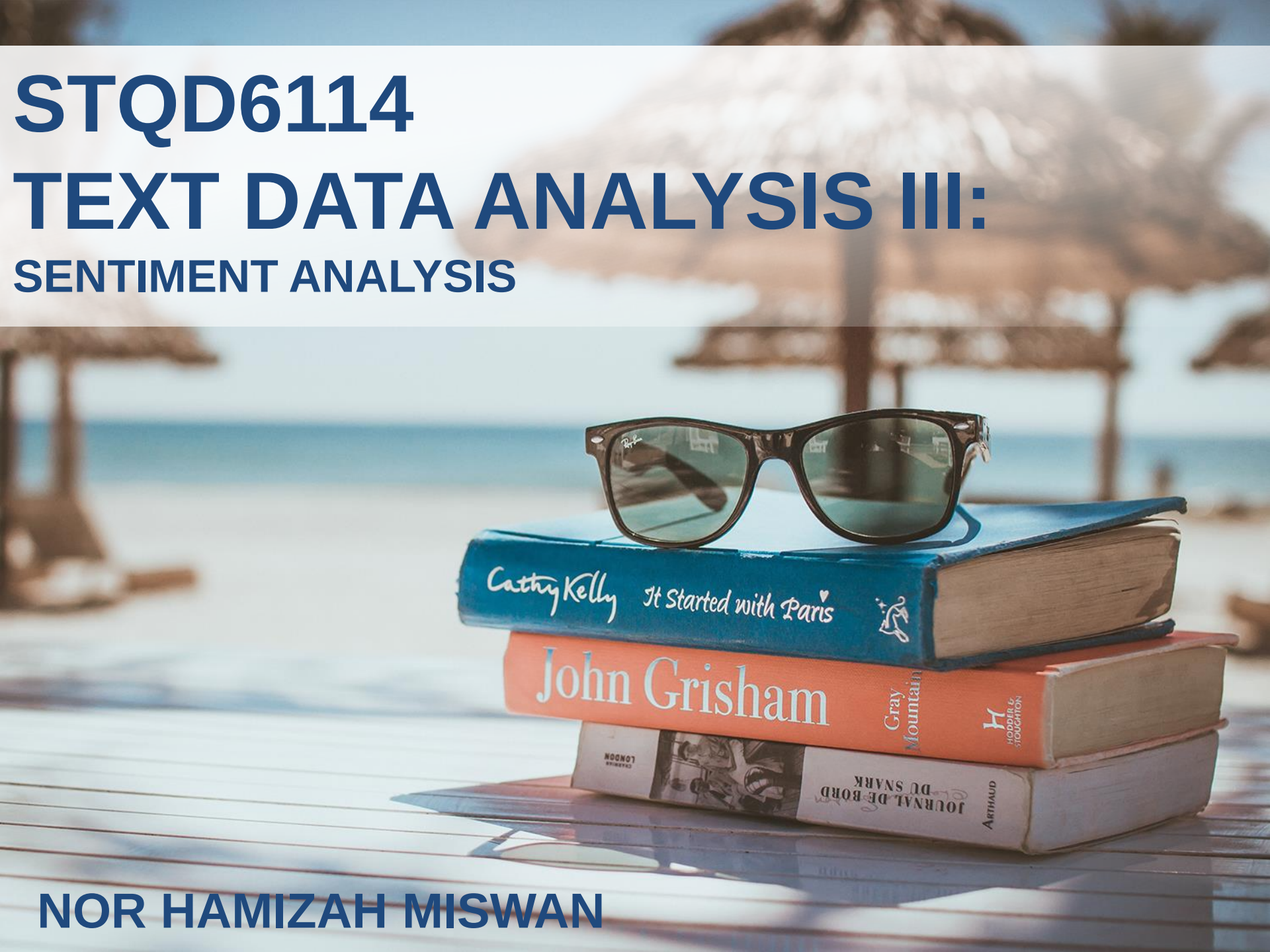
STQD6114
TEXT DATA ANALYSIS III:
SENTIMENT ANALYSIS



NOR HAMIZAH MISWAN

Introduction

- Sentiment: a view, an opinion
- Sentiment analysis: a process of computationally identifying and categorizing sentiments typically expressed in a text
- Determining emotional tone behind a series of word

Why?

- Social media monitoring – gain overview on public opinions for a certain topic
- Able to quickly understand consumer needs and react to it
- Example: Expedia Canada commercial case

Loves the German bakeries in Sydney. Together with my imported honey it feels like home	Positive
@VivaLaLauren Mine is broken too! I miss my sidekick	Negative
Finished fixing my twitter...I had to unfollow and follow everyone again	Negative
@DinahLady I too, liked the movie! I want to buy the DVD when it comes out	Positive
@frugaldougal So sad to hear about @OscarTheCat	Negative
@Mofette briliant! May the fourth be with you #starwarsday #starwars	Positive
Good morning thespians a bright and sunny day in UK, Spring at last	Positive
@DowneyisDOWNEY Me neither! My laptop's new, has dvd burning/ripping software but I just can't copy the files somehow!	Negative

Jadual 4.6 Perbandingan Kutipan Filem dengan Markah Akhir Kajian Filem Antarabangsa Tahun 2015

Filem (Antarabangsa)	Kutipan (USD)*	Bilangan Positif	Bilangan Negatif	Markah Akhir
Avengers: Age of Ultron	1,405,413,868	542	94	85.22
Furious 7	1,516,045,911	529	101	83.97
Jurassic World	1,670,400,637	764	248	75.49
Minions	1,159,398,397	448	406	52.46
Star Wars: The Force Awakens	2,066,960,090	303	132	69.66

Hotel	Skor sentimen keseluruhan Agoda	Skor penarafan Agoda	Skor sentimen keseluruhan Booking.com	Skor penarafan Booking.com
One World	6.85	8.5	6.59	8.5
Pullman Putrajaya	6.6	8	6.66	8.2
Vibrant Studio	6.87	8.6	8.23	8.8
Golden Triangle	6.37	7.1	6.47	7.1
The Gardens	6.83	8.3	6.79	8.4
Ascott Kuala Lumpur	6.57	8.5	6.64	8.7
Summer Suite	7.27	8.4	7.1	8.7
Sarang Vacation	7.1	8.6	6.8	9.1

ntimen keseluruhan dengan skor bintang

Skor sentimen keseluruhan Booking.com	Skor Bintang Booking.com
6.59	10
6.66	10
8.23	0
6.47	0
6.79	10
6.64	10
7.1	0
6.8	0

Vibrant Studio	6.87	7	8.23
Golden Triangle	6.37	7	6.47
The Gardens	6.83	10	6.79
Ascott Kuala Lumpur	6.57	9	6.64
Summer Suite	7.27	8	7.1
Sarang Vacation	7.1	6	6.8

- Teaching machine to identify context and sentiment of human language is very difficult
- Human language itself is already complex , and add on the lack of intuitively in a machine: how can we do it?
- Example: Wow, Astro doesn't broadcast when its rain! Verrryyyyyy gooooddddd!!
- A human know that the above sentence need to be read in a sarcastic way; hence it is a negative tone, however a machine sees the word "good" & might categorize as a positive tone statement
- Hence, the algorithm is evolving (as we talk!) to include comprehensively phrases/statements to increase the ability of a machine in conducting the sentiment analysis.

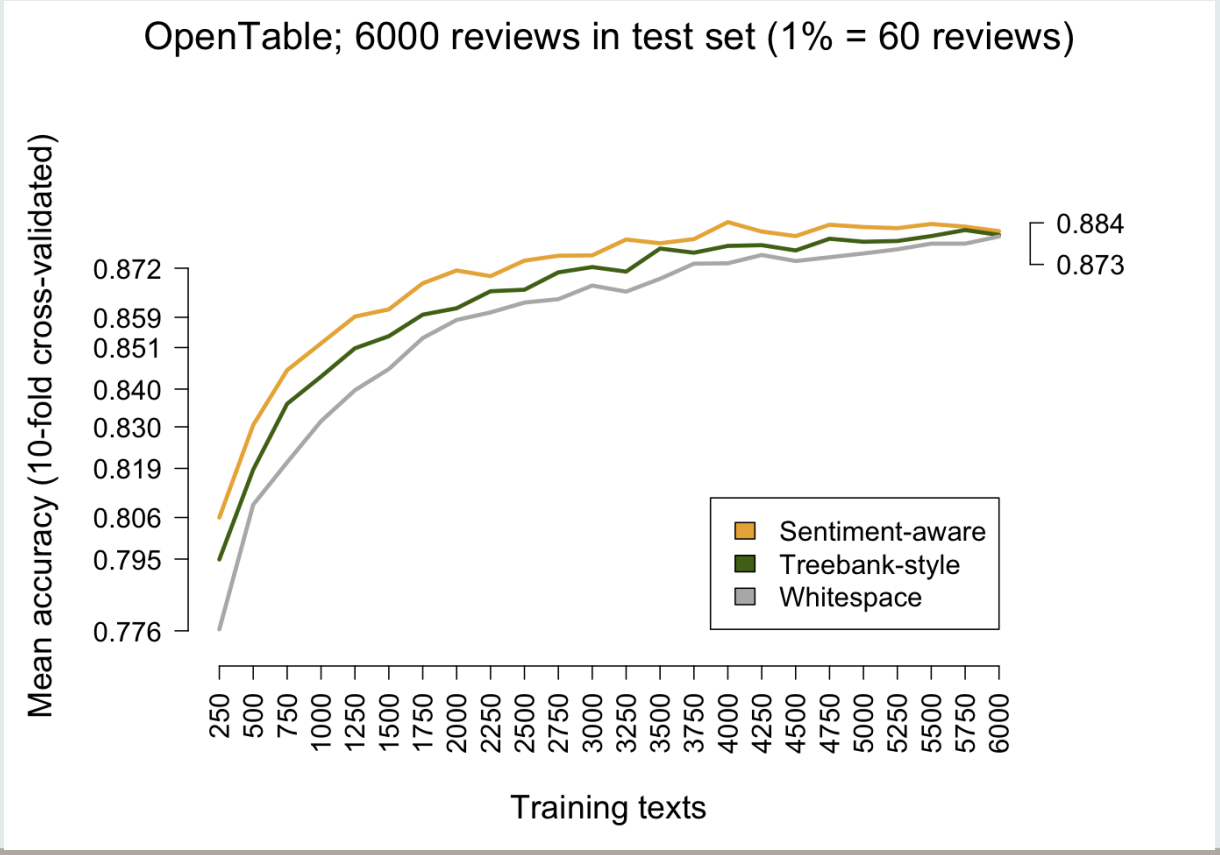
- Hence, the sentiment analysis results needs to be taken ‘with a pinch of salt’ (in precaution; with warning)
- It is not 100% accurate (yet!) but it do provides the overview / general idea especially on public sentiment


Text Input
Tokenization
Stop Word Filtering
Negation Handling
Stemming
Classification
Sentiment Class
Sentiment Analysis


It Started with Paris
John Grisham

Data Preprocessing (Cleaning & Scrubbing)

- Text input
- Tokenization: splitting a string into its desired constitutes parts.



- Stop word filtering
- Negation handling: not good, not not good; not pretty, not n ot pretty, pretty ugly?
- Stemming

Data Sentiment Analysis

- Classification: classifying positive, negative words
- Sentiment class: determine the polarity of the topic or the data

How to determine the sentiment score

- Capitalization
  - Words that are capitalized often signify a stronger expression
- Emotion
  - Usage of emotions alone
  - Usage of emotions along with the words
- The length of phrase
  - Longer phrases
  - Repetition on synonymous words
- Examples:
  - I am beautiful
  - I am very beautiful
  - I am BEAUTIFUL
  - I am superrrr beautiful
  - Beautifullllllllll

Sentiment Analysis in R (Simple codes)

Data prepping:

```
library(tm)
library(SnowballC)
library(wordcloud)
text=readLines(file.choose())
docs=Corpus(VectorSource(text))
inspect(docs)
toSpace=content_transformer(function(x,pattern)gsub(pattern," ",x))
docs=tm_map(docs, toSpace, "/")
docs=tm_map(docs, toSpace, "@")
docs=tm_map(docs, toSpace, "\\|")
docs=tm_map(docs,content_transformer(tolower))
docs=tm_map(docs,removeNumbers)
docs=tm_map(docs,removeWords, stopwords("english"))
docs=tm_map(docs,removeWords, c("dan","dengan","atau","sebagai","yang","itu", "ini","
asm","dari","daripada"))
docs=tm_map(docs,removePunctuation)
docs=tm_map(docs,stripWhitespace)
docs=tm_map(docs,stemDocument)
dtm=TermDocumentMatrix(docs)
m=as.matrix(dtm)
v=sort(rowSums(m),decreasing=TRUE)
d=data.frame(word=names(v),freq=v)
m=d$word
```

Sentiment Analysis in R

Sentiment analysis:

```
mysentiment<-function(m)
{
mydictpos=c("baik","cantik","bijak","kuat”)
mydictneg=c("jahat","buruk","bodoh","lemah")
pos_score=sum(!is.na(match(m,mydictpos)))
neg_score=(-1)*sum(!is.na(match(m,mydictneg)))
sentiment_score=pos_score+neg_score
sentiment_score
}
```

Sentiment Analysis by lexicon

Lexicon means dictionary

Example: affin, bing

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # … with 2,467 more rows
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # … with 6,776 more rows
```

Sentiment Analysis by lexicon in R

We will refer to:

https://www.tidytextmining.com/sentiment.html

Another example:

https://rstudio-pubs-static.s3.amazonaws.com/302066_fe1dd2a635fa41198b18c87a64f5620c.html

Appendix

```
library(tm) # for text mining
library(SnowballC) # for text stemming
library(wordcloud) # word-cloud generator
library(RColorBrewer) # color palettes
library(syuzhet) # for sentiment analysis
library(ggplot2) # for plotting graphs

# Read the text file from local machine , choose file interactively
text <- readLines(file.choose())

# Load the data as a corpus
docs <- Corpus(VectorSource(text))

#Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "₩₩|")

docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, removeWords, c("s", "company","team"))
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, stemDocument)
```

Appendix

```
# Build a term-document matrix
dtm <- TermDocumentMatrix(docs)
dtm_m <- as.matrix(dtm)
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE) # Sort by descending value of frequency
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
head(dtm_d, 5) # Display the top 5 most frequent words

# Plot the most frequent words
barplot(dtm_d[1:5,]$freq, las = 2, names.arg = dtm_d[1:5,]$word,
        col ="lightgreen", main ="Top 5 most frequent words",
        ylab = "Word frequencies")

#generate word cloud
set.seed(1234)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 5,
          max.words=100, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(8, "Dark2"))

# Word Association
findAssocs(dtm, terms = c("good","work","health"), corlimit = 0.25) # Find associations
findAssocs(dtm, terms = findFreqTerms(dtm, lowfreq = 50), corlimit = 0.25) # Find associations for words that occur at l
east 50 times
```

Appendix

```
## Sentiment scores
# regular sentiment score using get_sentiment() function and method of your choice
# please note that different methods have different scales
syuzhet_vector <- get_sentiment(text, method="syuzhet")
head(syuzhet_vector)
head(syuzhet_vector,10) # see the first 10 elements of the vector
summary(syuzhet_vector)

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

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

#nrc
nrc_vector <- get_sentiment(text, method="nrc")
head(nrc_vector)
summary(nrc_vector)
```

```
#compare the first row of each vector using sign function
rbind(
  sign(head(syuzhet_vector)),
  sign(head(bing_vector)),
  sign(head(afinn_vector))
)
```

Appendix

```
## Emotion classification
# run nrc sentiment analysis to return data frame with each row classified as one of the following
# emotions, rather than a score :
# anger, anticipation, disgust, fear, joy, sadness, surprise, trust
# and if the sentiment is positive or negative
d<-get_nrc_sentiment(text)
head (d,10) # head(d,10) - just to see top 10 lines

#Visualization
td<-data.frame(t(d)) #transpose
td_new <- data.frame(rowSums(td)) #The function rowSums computes column sums across rows for each level of a groupin
g variable.
names(td_new)[1] <- "count" #Transformation and cleaning
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2<-td_new[1:8,]

#Plot 1 - count of words associated with each sentiment
quickplot(sentiment, data=td_new2, weight=count, geom="bar",fill=sentiment,ylab="count")+ggtitle("Survey sentiments")

        #Plot 2 - count of words associated with each sentiment, expressed as a percentage
        barplot(
          sort(colSums(prop.table(d[, 1:8]))),
          horiz = TRUE,
          cex.names = 0.7,
          las = 1,
          main = "Emotions in Text", xlab="Percentage"
        )
```