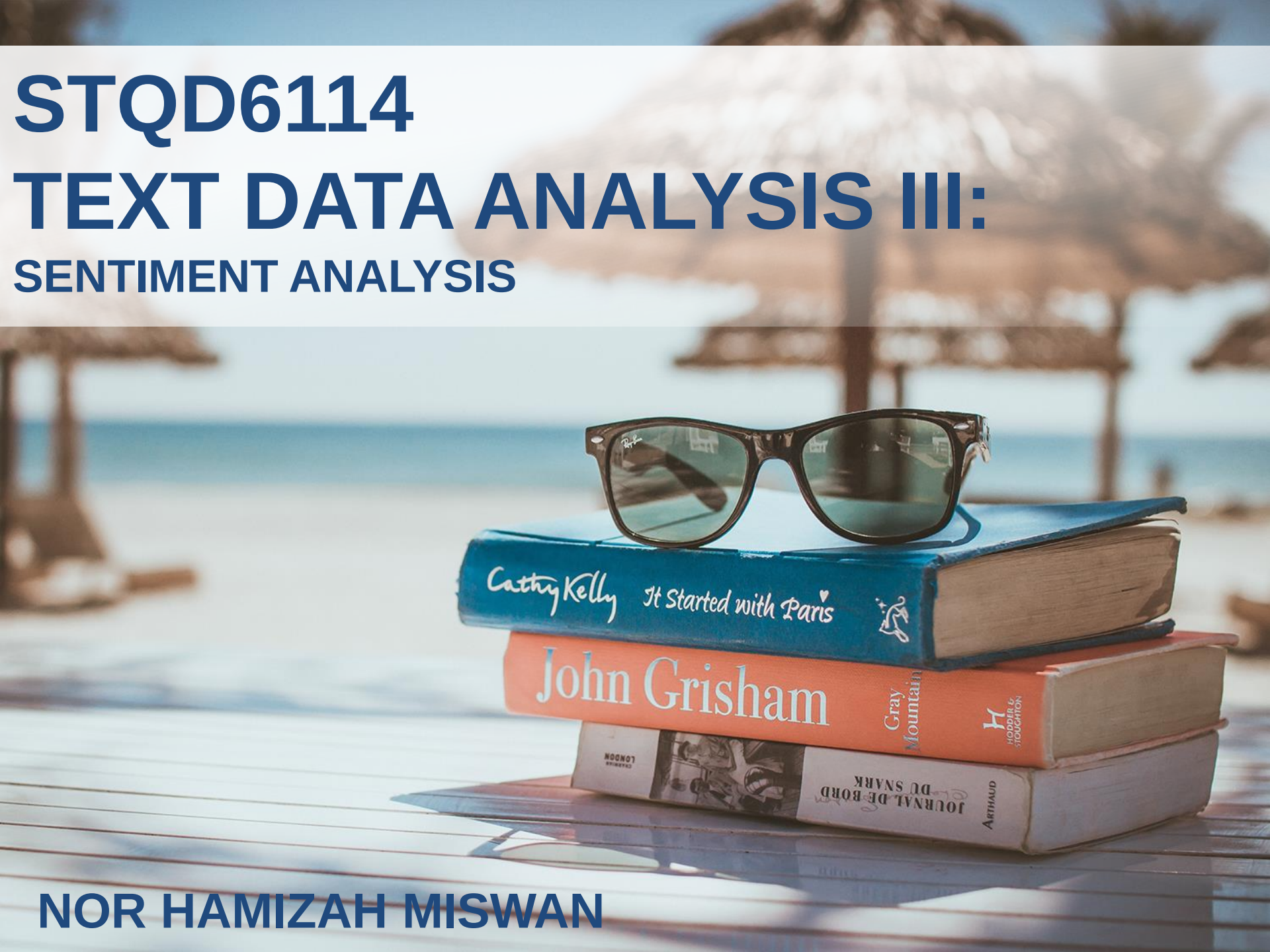
STQD6114
TEXT DATA ANALYSIS III:
SENTIMENT ANALYSIS



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.

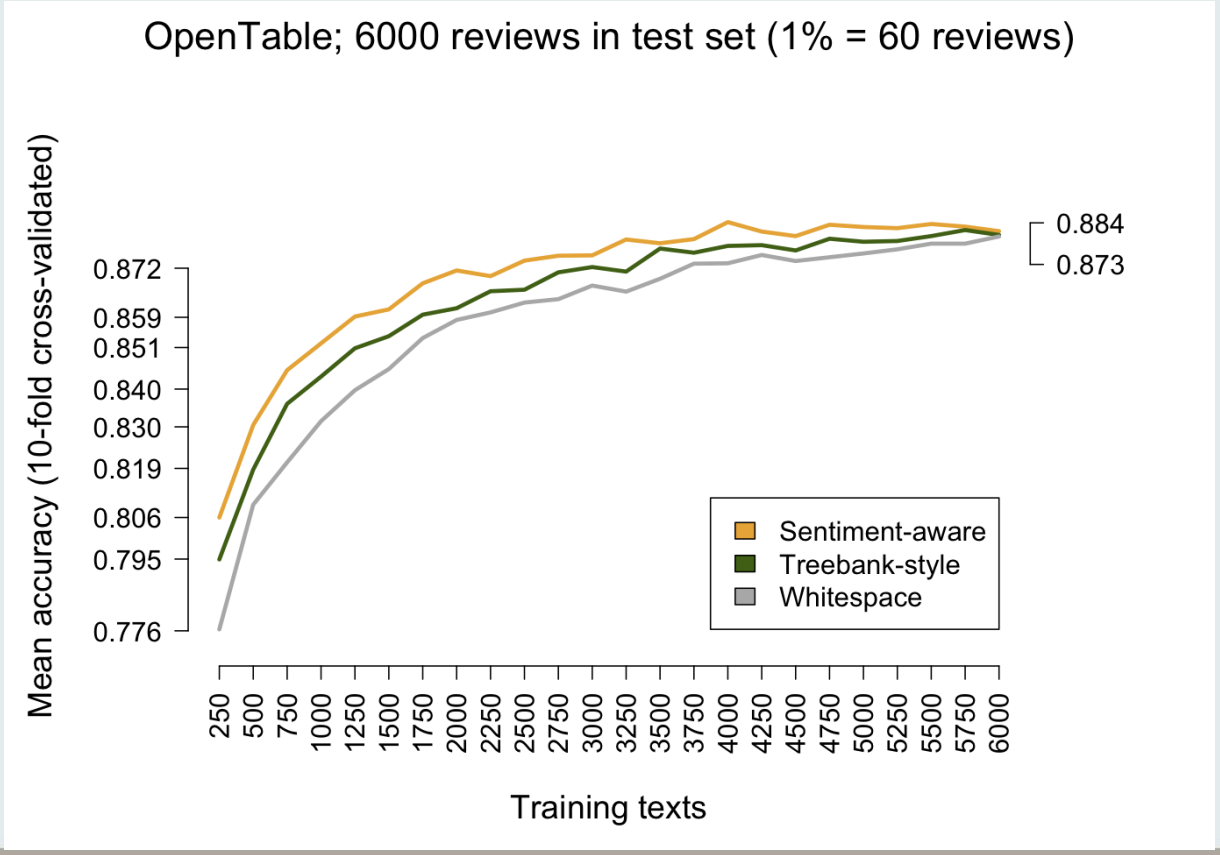
- Hence, the sentiment analysis results needs to be taken ‘with a pinch of salt’ (in precaution; with warning)
- It is not 100% accurate (yet!) but it do provides the overview / general idea especially on public sentiment


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 not 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”) #dictionary for positive words
mydictneg=c("jahat","buruk","bodoh","lemah") #dictionary for negative words
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, nrc

```
get_sentiments("afinn")
```

```
#> # A tibble: 2,477 × 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
```

```
get_sentiments("bing")
```

```
#> # A tibble: 6,786 × 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
```

```
get_sentiments("nrc")
```

```
#> # A tibble: 13,901 × 2
#>    word        sentiment
#>    <chr>       <chr>
#>  1 abacus      trust
#>  2 abandon     fear
#>  3 abandon     negative
#>  4 abandon     sadness
#>  5 abandoned   anger
#>  6 abandoned   fear
#>  7 abandoned   negative
#>  8 abandoned   sadness
#>  9 abandonment anger
#> 10 abandonment fear
#> # … with 13,891 more rows
```

Sentiment Analysis by lexicon in R

Please refer to:

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

Another example:

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

Sentiment Analysis by machine learning

As what you have learned in machine learning, training data is needed to for the algorithm to learn.

Then, the algorithm is applied to the testing data.

Sentiment Analysis by machine learning in R

```
library(RTextTools)
library(e1071)

pos_tweets =  rbind(
  c('I love this car', 'positive'),
  c('This view is amazing', 'positive'),
  c('I feel great this morning', 'positive'),
  c('I am so excited about the concert', 'positive'),
  c('He is my best friend', 'positive')
)

neg_tweets = rbind(
  c('I do not like this car', 'negative'),
  c('This view is horrible', 'negative'),
  c('I feel tired this morning', 'negative'),
  c('I am not looking forward to the concert', 'negative'),
  c('He is my enemy', 'negative')
)

test_tweets = rbind(
  c('feel happy this morning', 'positive'),
  c('larry friend', 'positive'),
  c('not like that man', 'negative'),
  c('house not great', 'negative'),
  c('your song annoying', 'negative')
)

tweets = rbind(pos_tweets, neg_tweets, test_tweets)
```

```
# train the model
mat = as.matrix(matrix)
classifier = naiveBayes(mat[1:10,], as.factor(tweets[1:10,2]) )
```

```
# test the validity
predicted = predict(classifier, mat[11:15,]); predicted
table(tweets[11:15, 2], predicted)
recall_accuracy(tweets[11:15, 2], predicted)
```