

40.016: The Analytics Edge

Week 10 Lecture 1

TEXT ANALYTICS (PART 1)

Term 5, 2022



SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Outline

- 1 Text Analytics
- 2 Sentiment Analysis
- 3 Sentiment Analysis with Twitter data
- 4 Modelling process

Outline

- 1 Text Analytics
- 2 Sentiment Analysis
- 3 Sentiment Analysis with Twitter data
- 4 Modelling process

Text Analytics

- Process of (automatically) deriving high-quality information from text
 - translating large volumes of unstructured text into quantitative data to uncover insights, trends, and patterns.
- Common tasks are
 - (1) text categorization and summarization (will be discussed this Wednesday)
 - (2) sentiment analysis (this lecture)
- Text analytics build on several steps, e.g.,
 - processing the text
 - finding patterns
 - learning a classification model

- Twitter is a social networking and communication website established in March 2006.
- The service enables users to send and read short messages called “tweets”, which were originally restricted to 140 characters (this limit was recently doubled for most languages).
- Twitter is one of the biggest social networks worldwide: as of April 2018, the company has
 - more than 300 million users
 - a total revenue of about \$2.5 billion
 - an evaluation of over \$20 billion

Twitter (cont'd)

- A study by Pear Analytics in 2009 estimated that 40% of Twitter messages are just “babble”, this means that 60% are not.
 - We can take advantage of that!
- When a lot of people share a lot of messages on a daily basis, we will get a large amount of data.
- We can use smart computer algorithms to analyze this data and create information from it.

Twitter (cont'd)

- Twitter is not only used by celebrities to reach out to their followers, but also by companies to communicate with their customers, hear their thoughts, and understand trends.
 - Product reviews (for retailers)
- “Twitter mood” has been shown to have a predictive power on stock market prices.
 - Johan Bollen et al, “Twitter mood predicts the stock market.” *Journal of Computational Science*, 2011. (<https://www.sciencedirect.com/science/article/pii/S187775031100007X>)
- Twitter has been used to predict box office performance after movies have been released based upon how many times films are mentioned in tweets.

Outline

- 1 Text Analytics
- 2 Sentiment Analysis**
- 3 Sentiment Analysis with Twitter data
- 4 Modelling process

Sentiment analysis

Sentiment Analysis refers to the use of

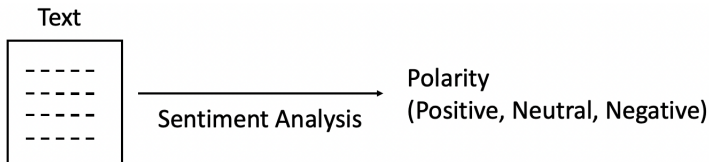
- text analytics
- natural language processing
- computational linguistics

to identify and extract subjective information in source materials.

Tasks of sentiment analysis

It can be seen as a classification problem (**binary** or **multi-class**), where one wants to determine the

- Polarity of a given text (i.e., whether the opinion expressed in a document is positive, neutral, or negative)

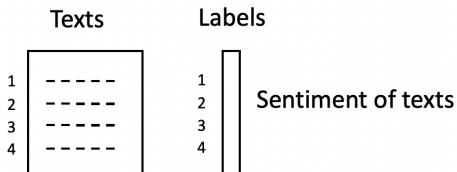


- Or emotional states (e.g., angry, sad) – More advanced tasks

Typical applications

- Political sentiment
- Opinion polling
- Recommendation systems (Week 11)

Which data do we need?



Outline

- 1 Text Analytics
- 2 Sentiment Analysis
- 3 Sentiment Analysis with Twitter data
- 4 Modelling process

Working with data from Twitter

Where can we get data? Some options:

- Twitter's API (Application Programming Interface)
- Specialized websites, such as sentiment140 (www.sentiment140.com)
- R package `TwitterR` – can be used to import tweets directly from Twitter

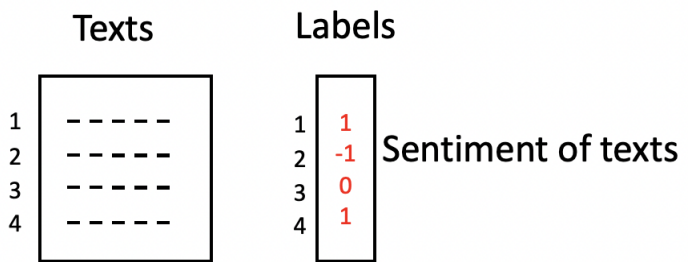
How to get labels?

To predict the sentiment of tweets, a **fundamental piece of information** we need are the labels (sentiments) associated to each tweet.

Where do we get them? Some options:

- Manual labelling
 - carefully go through every tweet and provide a sentiment polarity for the tweets;
- Centralized work places, such as Amazon Mechanical Turk
 - small tasks are assigned to individuals who work remotely and do the sentiment categorization for a few tweets at a small price;
- Leverage the information contained in emoticons

How to get labels? (cont'd)



Key Question

Is it possible to correctly predict (classify) the sentiment of a tweet based on the information contained in previous tweets?

Sentiment Analysis with Twitter data

Challenges:

- Tweets are textual data, typically with **poor spelling** (short forms) and use of **non-traditional grammar**
 - Example: “U say that iphone 5S didnt bring anything new 2?”
 - U = You
 - didnt = didn't
 - 2 = too
- There is an additional source of complexity, namely **the ambiguity in the English language** that sometimes even humans cannot decipher.
 - Examples?

Examples of “the ambiguity in the English language”

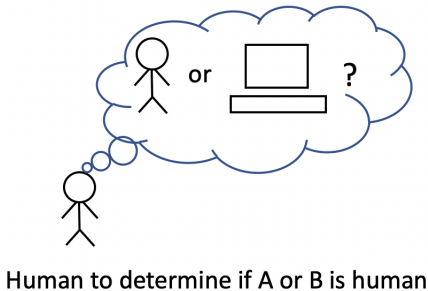
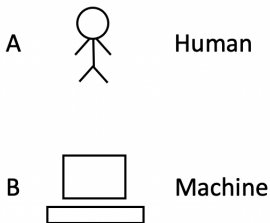
- “John saw the man on the mountain with a telescope”.
Who has the telescope? John, the man on the mountain, the man?
- “John and Mary took two trips around France. They were both wonderful.”
They refers to John and Mary or to the two trips?
- “Medicine helps dog bite victims.”
Does the medicine help the dog to bite victims or does it help the victims who are bitten by the dog?

Relation with Turing test

- The problem of classifying tweets with a computer program is part of the broader problem of understanding and analyzing human language as it is spoken.
- In this regard, Alan Turing introduced the Turing test, which is described in his 1950 paper “Computing machinery and Intelligence”.
- This is a test of a machine’s ability to exhibit intelligent behavior that is undistinguishable from a human.

Relation with Turing test (cont'd)

- Turing proposed that the human evaluator would judge between natural language conversations with a human and a machine.
 - If the evaluator cannot reliably tell the machine from the human using a text only channel, then the machine passes the test.



Summary

- **Data:** large unstructured datasets containing tweets and a corresponding value determining the class/polarity of each tweet.
- **Model:** A classification model (e.g., logistic regression, CART, Random Forest) that predicts the sentiment of a tweet based on key words contained in the tweet itself.
- **Value and Decision:** the model replaces the option of polling users on their opinions and allows exploiting the information contained in tweets.

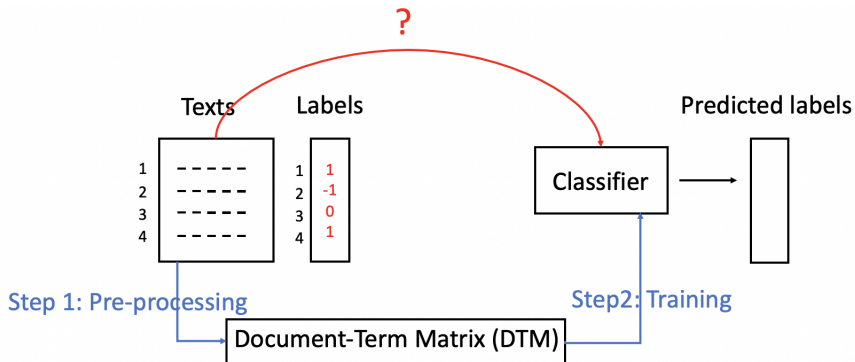
Outline

- 1 Text Analytics
- 2 Sentiment Analysis
- 3 Sentiment Analysis with Twitter data
- 4 **Modelling process**

Modelling process

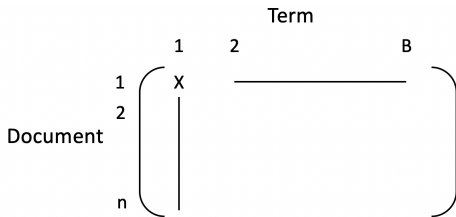
Challenges:

- This is a **classification problem**, where we want to predict the sentiment associated to a tweet a



Bag of words model

- Bag of words is a simple approach to represent text in a computer program.
- In particular, the text is represented as a set (bag) of words, disregarding the grammar and word order – but keeping multiplicity.
- Document-term matrix (DTM) – N documents and B terms



- Each element in the document-term matrix represents a measure of the frequency of occurrence of the terms (words) in the document.

Example

- “John likes to watch movies. Mary likes movies too.”
- “John also likes to watch football.”

In this example, we have two documents and nine words, which we can organize in the following matrix:

$$\begin{pmatrix} & \text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{also} & \text{Mary} & \text{football} & \text{too} \\ \text{Doc1} & 1 & 2 & 1 & 1 & 2 & 0 & 1 & 0 & 1 \\ \text{Doc2} & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

- (1) number of terms p can be large ($p \gg n$, where n is the number of documents)
- (2) sparsity

Pre-processing

Data are often unstructured, and hence pre-processing needs to be done.
For example:

- **Stopwords** are words that are filtered out in the processed text.
 - These typically refer to the most common words in the language, such as “the”.
 - Example: “Dharma and Greg rock”.
 - In a bag of words model, the stopword “and” would be removed.
 - However, “Dharma and Greg rock” might refer to the show “Dharma and Greg”, where it is part of the same name.
- In these cases, it is possible to use an n-gram, namely a contiguous sequence of n items (see Google Ngram Viewer). We will not use these features in this work, though.

Pre-processing (cont'd)

- **Removing punctuations, converting upper case to lower case.**

These are other types of preprocessing commonly used.

- **Stemming**

- Martin Porter in 1980 invented the Porter stemmer, one of the most common algorithms for stemming in English.
- The Porter stemming algorithm is used to remove inflected words to their word stem, base or root form.
- Example
 - “cats” should be identified with the root “cat”.
 - “revive” and “revival” would be stemmed to “reviv”.

Modelling workflow

- Pre-processing
 - Convert text to lower case
 - Remove stopwords
 - Remove punctuation
 - Stemming
 - Create DTM
 - Removing sparse terms
- Preparing the DTM for model learning
- Train and test a classifier

Back to R!

- Pre-processing

In R, the pre-processing steps can be carried out with the `tm` (text mining) package.

```
twitter <- read.csv("twitter.csv", stringsAsFactors=FALSE)
```

– Load data.

```
corpus <- Corpus(VectorSource(twitter$tweet))
```

– Create a corpus, which represents a collection of documents

- Convert text to lower case

```
corpus <- tm_map(corpus, function(x) iconv(enc2utf8(x),  
sub = "byte"))
```

```
corpus <- tm_map(corpus, content_transformer(function(x)  
iconv(enc2utf8(x), sub = "bytes")))
```

```
corpus <- tm_map(corpus, content_transformer(tolower))
```

Back to R! (cont'd)

- Pre-processing

- Convert text to lower case

- Remove stopwords

```
corpus <- tm_map(corpus,removeWords,stopwords("english"))
```

- Remove punctuation

```
corpus <- tm_map(corpus,removePunctuation)
```

- Stemming

```
corpus <- tm_map(corpus,stemDocument)
```

– using package SnowballC

- Create DTM

```
dtm <- DocumentTermMatrix(corpus)
```

- Removing sparse terms

```
dtm <- removeSparseTerms(dtm,0.995)
```

Back to R! (cont'd)

- Pre-processing
 - Convert text to lower case
 - Remove stopwords
 - Remove punctuation
 - Stemming
 - Create DTM
 - Removing sparse terms
- Preparing the DTM for model learning

```
twittersparse <- as.data.frame(as.matrix(dtm))  
colnames(twittersparse)<-make.names(colnames(twittersparse))
```

Basic visualization with package `wordcloud`

References

- Teaching notes.