

# COMP8210 — Big Data Technologies

## Week 8 Lecture 1: Analysing Unstructured Data

Diego Mollá

Department of Computer Science  
Macquarie University

COMP8210 2020H2

# Programme

- 1 Analysing Unstructured Data
- 2 Analysing Text Data

## Reading

- Lecture notes.
- Text Analytics — Microsoft Azure Machine Learning Studio.

# Programme

1 Analysing Unstructured Data

2 Analysing Text Data

# Why Analyse Unstructured Data

## It's about Variety

- Probably the biggest impact of Big Data in companies is the possibility to analyse unstructured data.
- Unstructured data contains information that can potentially be very useful.
- It opens up the possibility to access yet untapped information from multiple sources.

## Sources of Unstructured Data

**Video:** surveillance cameras, videos in social media.

**Images:** Web images, images in social media, satellite images.

**Sound:** Call centre recordings.

**Text:** Documents, reports, webpages, social media posts.

## Motivating Example: United Healthcare

- Have recorded voice files from customer calls to call centres.
- The voice data was converted to text using speech-to-text conversion tools.
- The text was then analysed using natural language processing software.
- Their analysis focused on identifying customers who use terms suggesting strong dissatisfaction.
- A United representative can then make some sort of intervention.

# Use Cases of Image Analytics

<https://www.zencos.com/blog/5-amazing-use-cases-of-image-analytics/>,  
13/6/2018

- 1 Identify bags at airports.
- 2 Analyse social media images for missing persons.
- 3 Real-time vehicle damage assessment.
- 4 Detect pneumonia from chest x-rays.

# Programme

## 1 Analysing Unstructured Data

## 2 Analysing Text Data

- Characteristics of Text
- Common Building Blocks for Text Analytics

# Why Analysing Text?

## Information Overload

- A lot of information is available as free text.
- The most natural form to write information is through free text.
- A great deal of digital information is available as free text.
- People can read and understand free text easily.
- But it's very hard for machines!





# Examples of Using Text for Big Data

## Analysis of social media posts

- What do people think about us?
- What do people think about our product?
- What do people think about our competitors?
- What are the most common topics mentioned in media?
- What are the common trends?

## Analysis of text documents

- Is this patent claim related to other patents?
- Evidence based medicine: What treatment has best clinical evidence?
- Is this message spam?
- Who is the best person to forward this user request?

# Using Text Analytics to Help Combat COVID-19

<http://kdcovid.nl/>

The screenshot shows the KDCovid website interface. At the top, the URL bar displays `kdcovid.nl/search.html?search_term=Treatments+for+SARS`. The website header includes the 'KDCovid' logo and an 'About' link. The main heading is 'Knowledge Discovery', followed by the subtitle 'Search the full text of COVID19 research papers on COVID 19'. A search bar contains the text 'Treatments for SARS', with a 'Search' button to its right. Below the search bar are two checkboxes: 'Sort By Date' and 'COVID-19 only'. The search results are displayed in two columns. The left column shows a search result titled 'Literature-related discovery: Potential treatments and preventatives for SARS' by Kostoff, Ronald N., dated 2011-09-30. The text snippet highlights 'Literature-related discovery: Potential treatments and preventatives for SARS' and 'HIGHLIGHT Literature-related discovery (LRD) is the linking of two or more previously disjoint concepts in order to produce novel, interesting, plausible, and intelligible connections (i.e., potential discovery). LRD has been used to identify potential treatments or preventative actions for challenging medical problems, among myriad other applications. Severe acute respiratory syndrome of ( SARS ) was the first pandemic of the 21st century. SARS was eventually controlled through increased hygienic measures (e.g., face mask protection, frequent hand washing, living quarter disinfection), travel restrictions, and quarantine. According to recent reviews of SARS, none of the drugs that were used during the pandemic'. The right column is titled 'Gene-Disease Association' and includes the instruction 'Click on the gene/disease for more information'. It features a diagram with orange boxes for genes (CCL2, CXCL10, IL6, IL18, IL2) and yellow ovals for diseases (Influenza, Parkinson Disease, Neoplasms, leukemia). Lines connect the genes to the diseases: CCL2 and CXCL10 to Influenza; IL6 to Parkinson Disease; IL18 and IL2 to Neoplasms; and IL2 to leukemia.

# The COVID-19 Open Research Dataset Challenge (CORD-19)

<https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge>

The screenshot shows the Kaggle dataset page for the COVID-19 Open Research Dataset Challenge (CORD-19). The header is dark blue with the title 'COVID-19 Open Research Dataset Challenge (CORD-19)' and a subtitle 'An AI challenge with AI2, CZI, MSR, Georgetown, NIH & The White House'. It also shows the AI2 logo and that it was updated 3 hours ago (Version 47). On the right, there is a circular diagram with red arrows and a yellow circle, and a box indicating 8147 files. Below the header, there are tabs for Data, Tasks (17), Notebooks (1,615), Discussion (368), Activity, and Metadata. A 'Download (18 GB)' button and a 'New Notebook' button are also present. The 'Usability' is 8.8, and the 'License' is 'Other (specified in description)'. The 'Tags' include 'business, earth and nature, computer science, health, biology and 3 more'. The 'Description' section is titled 'Dataset Description' and contains the following text: 'In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 200,000 scholarly articles, including over 100,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new insights in support of the ongoing fight against this infectious disease. There is a growing urgency for these approaches because of the rapid acceleration in new coronavirus literature, making it difficult for the medical research community to keep up.'

Dataset

## COVID-19 Open Research Dataset Challenge (CORD-19)

An AI challenge with AI2, CZI, MSR, Georgetown, NIH & The White House

AI2 Allen Institute For AI and 8 collaborators • updated 3 hours ago (Version 47)

8147

Data Tasks (17) Notebooks (1,615) Discussion (368) Activity Metadata

Download (18 GB) New Notebook

Usability 8.8 License Other (specified in description) Tags business, earth and nature, computer science, health, biology and 3 more

### Description

#### Dataset Description

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 200,000 scholarly articles, including over 100,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new insights in support of the ongoing fight against this infectious disease. There is a growing urgency for these approaches because of the rapid acceleration in new coronavirus literature, making it difficult for the medical research community to keep up.

# Programme

## 1 Analysing Unstructured Data

## 2 Analysing Text Data

- Characteristics of Text
- Common Building Blocks for Text Analytics

# Text as Arbitrary Symbols

- Words are encoded as arbitrary symbols.
- Different languages use different representations to represent the same word.
- Even within one language there is no clear correspondence between a word symbol and its meaning.



<https://www.linguisticsociety.org/content/how-many-languages-are-there-world>

# Ambiguity everywhere I

Language features ambiguity at multiple levels.

## Lexical

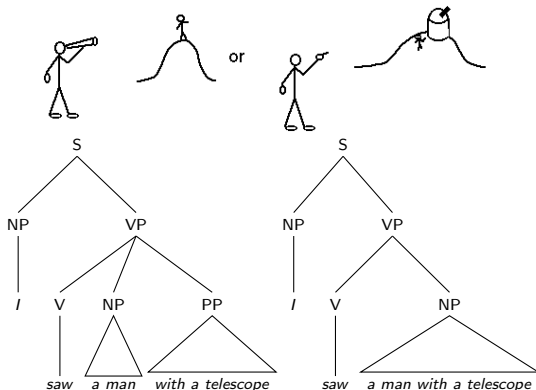
Example from Google's dictionary:

- bank (n): the land alongside or sloping down a river or lake.
- bank (n): financial establishment that uses money deposited by customers for investment, . . .
- bank (v): form in to a mass or mound.
- bank (v): build (a road, railway, or sports track) higher at the outer edge of a bend to facilitate fast cornering.
- . . .

# Ambiguity everywhere II

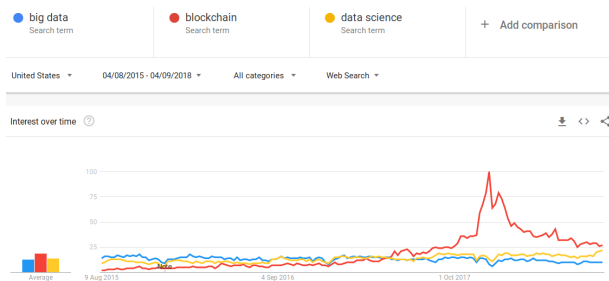
## Syntactic

- “I saw a man with a telescope” ... who has the telescope?



# So many words!

- Any language features a large number of distinct words.
- New words are coined.
- Words change their use in time.
- There are also names, numbers, dates... the possibilities are infinite.



<https://trends.google.com>



# Programme

## 1 Analysing Unstructured Data

## 2 Analysing Text Data

- Characteristics of Text
- Common Building Blocks for Text Analytics

# Tokenisation

- **Tokenisation**: Break down the input into words and other kinds of tokens.
- **Sentence Segmentation**: Break down the input into sentences.
- Tokenisation needs to be done as a first step in other applications.
- Same process as identifying separate units in programming languages, but harder.
- Tokenisation in space-delimited languages is fairly easy but some languages have no clear-cut way to separate words, or even sentences.

## Keyword Extraction and Word Clouds

- **Keyword extraction:** Extract the most important words in a document or collection of documents.
- **Word cloud:** a graphical interface that displays words according to their importance.

## How to Select and Score words?

- Remove stop words.
- Select words by frequency.
- Use tf.idf
- ...



# Removing Stop Words

- Many packages offer lists of stop words.
- These lists include words that usually are not important.
- There is no universal list of stop words.

## Stop words in the Python NLTK package

```
>>> from nltk.corpus import stopwords
>>> stopwords.words('english')
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're",
"you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he',
'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's",
'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which',
'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are',
'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do',
'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because',
'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against',
'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below',
'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again',
'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all',
'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no',
'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't',
'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll',
'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven',
'haven't', 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't",
'needn', "needn't", 'shan', "shan't", "shouldn", "shouldn't", 'wasn', "wasn't"]
```



# Selecting Words by Frequency

- If you want to find words that **discriminate** between different documents ...
  - Very frequent words are not useful (because they are in most documents).
  - Very rare words are not useful (because they are in too few documents).
  - The right solution is somewhere in the middle.
- A practical solution is to apply this sequence:
  - 1 Remove stop words.
  - 2 Select the most frequent remaining words.

# Selecting Words by $tf.idf$

- $tf.idf$  strikes a balance between words that are frequent but are not too frequent.
- **tf**: Term frequency. Words that are very frequent are more important.

$tf(w, d)$  = number of times word  $w$  occurs in document  $d$

- **idf**: Inverse document frequency. Words that occur in many documents are less important.

$$idf(w) = 1 + \log\left(\frac{\text{number of documents}}{\text{number of documents containing word } w}\right)$$

- $tf.idf(w, d) = tf(w, d) \times idf(w)$
- We select words from document  $d$  with high  $tf.idf$ , possibly after removing stop words.

# Stemming and Lemmatisation

- Words in many languages (e.g. English) have inflections.
  - Singular, plural, verb-ing, etc.
- Stemming and lemmatisation allow to group words that are different only because of their inflections.
- **Stemming**: Remove the part of a word that has the inflection to produce the **stem**.
- **Lemmatisation**: Convert an inflected word into a word without inflections to produce the **lemma** or **base form**.
- Stemming is easier and requires less knowledge of the language. Often stemming is all you need.
- Lemmatisation is useful when you want to produce real words.
  - E.g. if you want to display keywords.

# Part of Speech Tagging

- Words with the same part of speech have similar grammatical properties.
- In general, one can replace a word with another of the same part of speech and the sentence is still grammatical.
- Most words belong to **open class types**: nouns, verbs, adjectives, adverbs.
  - These words usually determine the topic of the sentence.
  - For example, keywords would normally be words in open class types.
- Words in the **closed class types** are useful to connect other words: prepositions, determiners, pronouns, conjunctions, . . . .
  - These words are usually removed by some text applications.
  - For example, stop words are normally words from closed class types.



# Parts of Speech in the Penn Treebank

CC	Coordinating conjunction	PRP\$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP\$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

# Named Entity Recognition

- **Named entities** are (often multi-word) expressions that refer to proper names of specific types.
  - Persons, organisations, locations, artifacts, dates, ...
- Named entity recognition is one of the most common tasks in text analytics.

When **Sebastian Thrun** PERSON started working on self-driving cars at **Google** ORG in **2007** DATE, few people outside of the company took him seriously. "I can tell you very senior CEOs of major **American** NORP car companies would shake my hand and turn away because I wasn't worth talking to," said **Thrun** PERSON, now the co-founder and CEO of online higher education startup Udacity, in an interview with **Recode** ORG **earlier this week** DATE.

A little **less than a decade later** DATE, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

<https://explosion.ai/demos/displacy-ent>

# Entities in the Message Understanding Conference

- Named Entities
  - Organization
  - Person
  - Location
- Temporal Expressions
  - Date
  - Time
- Number Expressions
  - Money
  - Percent

## MUC

- Initiated and financed by DARPA (Defense Advanced Research Projects Agency).
- From 1987 to 1997.
- The goal was to advance methods for information extraction from text.
- MUC-6 (1995) introduced the task of named entity recognition.
- The MUC named entities have been used by many systems since then.

# Text Classification

Many different tasks can be seen as text classification.

- E-mail filtering, spam detection, sentiment analysis . . .

To classify text it needs to be converted into a vector of features.

## Feature Selection

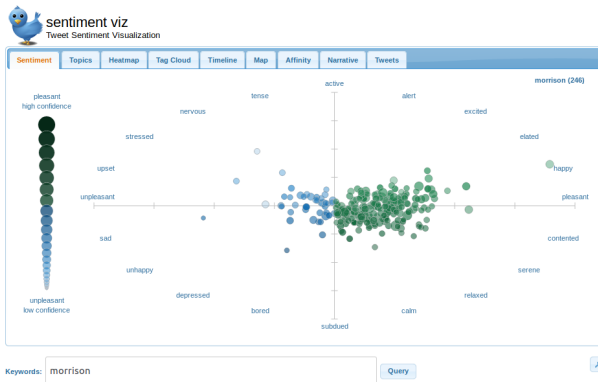
- Extract key words and use them to build document vectors for classification.
- For example, remove **stop words** and/or select words with high tf.idf.

## Feature Extraction

- Generate document vectors based on mathematical and statistical combinations of the entire information of the text.
- **Latent Semantic Analysis (LSA)**, **Singular Value Decomposition (SVD)** and **Principal Component Analysis (PCA)** are traditionally used for feature extraction.
- More recent approaches use **neural networks** and **word embeddings**.

# Sentiment Analysis

- Sentiment analysis is a popular example of text classification.
- Often needed for market analysis.
- A well known approach to analyse social media posts.



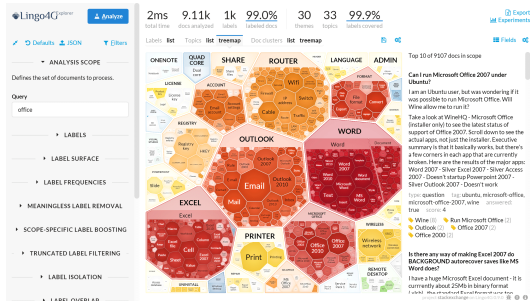
[https://www.csc2.ncsu.edu/faculty/healey/tweet\\_viz/tweet\\_app/](https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/)

# Text Retrieval / Filtering

- Often needed to find specific information in large volumes of text.
- Search engines are the first popular applications of text retrieval.
- A common step before doing other processing tasks such as sentiment analysis.

# Text Clustering

- Nothing to do with computer clusters ...
- Useful when we have large volumes of text but no labels.
- Can help characterise types of customers, common views of opinion, etc.



<https://get.carrotsearch.com/lingo4g/1.4.0/doc/>

# Topic Modelling

- Topic modelling is a more complex form of unsupervised text processing.
- The task is to find the common topics in a collection of texts (e.g. tweets).
- Implementations such as **Latent Dirichlet Allocation (LDA)** return keywords that are most characteristic of each topic.

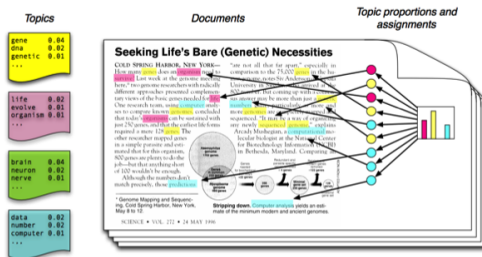


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4), 77-84.



# Take-home Messages

- Sources of unstructured data.
- Impact of unstructured data.
- Characteristics of text.
- Common building blocks for text analytics.

# What's Next

## Week 9

- Text Analytics (II).