TEXT MINING AND SENTIMENT ANALYSIS

OUTLINE

1. Case Study: @BOTUS
2. Text Mining and NLP
3. Text Analysis Basics
4. Sentiment Analysis
5. Example: Movie Reviews

[Greenstone, S. [2017]. Mettler, K. [2017]]

CASE STUDY: @BOTUS AND T&D

Some evidence pointed to the 45th POTUS' tweets affecting the stock market.

Can sentiment analysis and A.I. be used to take real-time (fast) advantage of his unpredictable tweeting nature?

Enter NPR's *Planet Money's* **@BOTUS** and T3's **Trump&Dump**.

https://www.youtube.com/watch?v=ZqI89XDNqg8

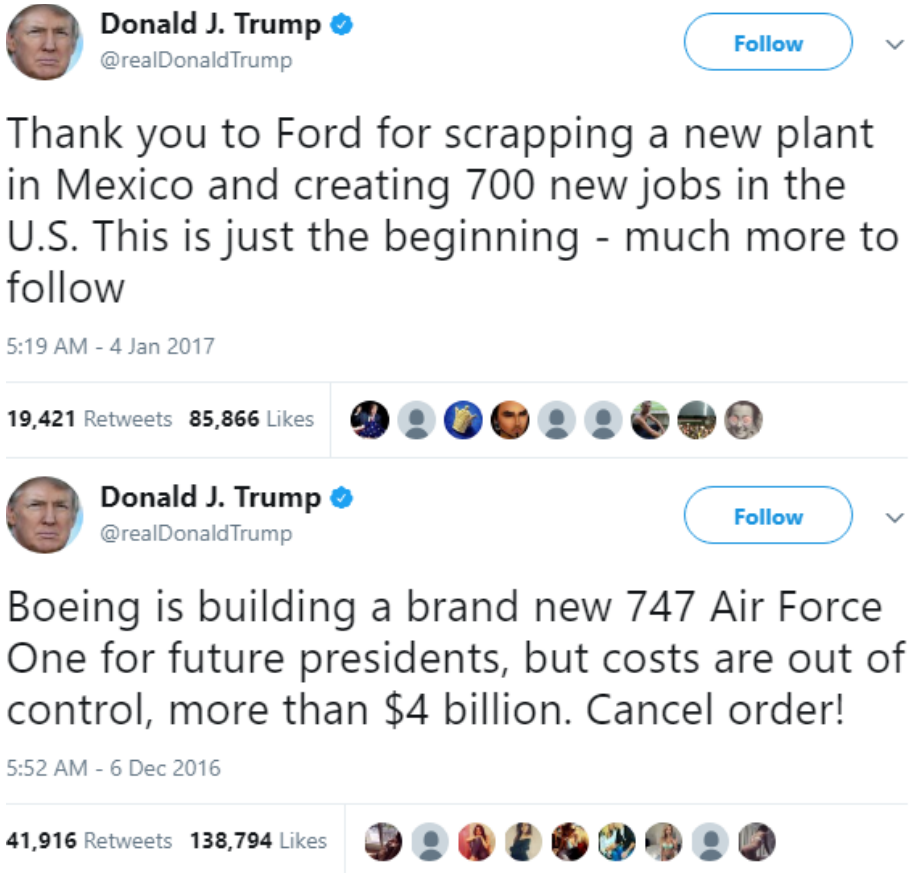
[]

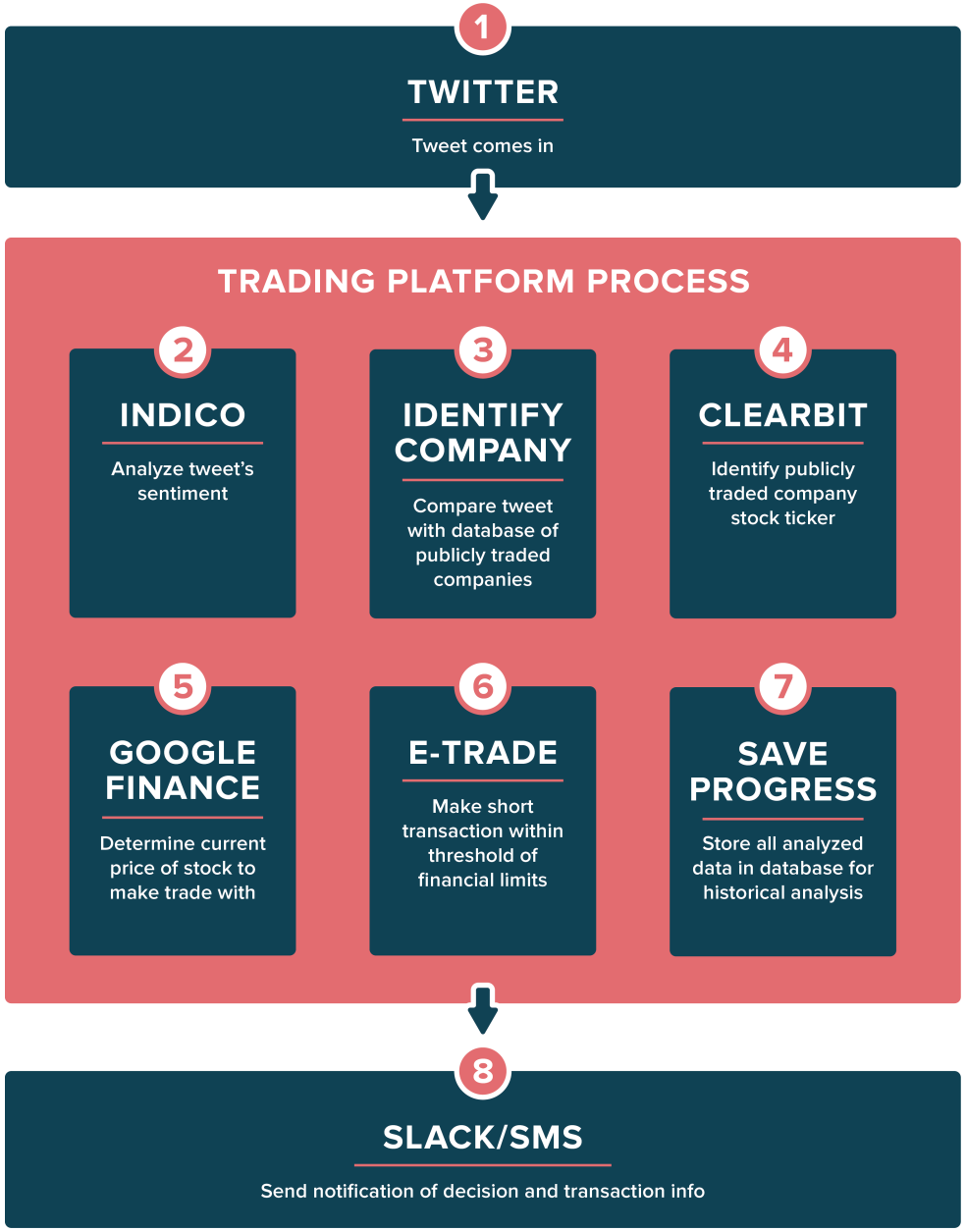
CASE STYUDY: @BOTUS AND T&D

Sentiment analysis (or opinion mining) is the collection of algorithms used to identify the text writer's attitude (positive, negative, neutral, etc.) towards a specific topic/product.

"I can't believe YOU're the President!!!" vs. "I can't believe you're the PRESIDENT!!!"

CASE STUDY: @BOTUS AND T&D





CASE STUDY: @BOTUS AND T&D

T3's president claimed T&D was profitable, but no details were provided and the website was recently taken down.

@BOTUS did not make a single trade in its first 4 months of operation (for various reasons)

Trading strategy was relaxed… leading to a loss on 1st trade.

Bot of the U.S.
@BOTUS
Follow

I see a company name. ✔ I know the stock ticker (AMZN) ✔ I can analyze the sentiment. ✔ (It's pretty negative). But market wasn't open. 🚫

> **Donald J. Trump** @realDonaldTrump
> The #AmazonWashingtonPost, sometimes referred to as the guardian of Amazon not paying internet taxes (which they should) is FAKE NEWS!

7:24 AM - 28 Jun 2017

Bot of the U.S.
@BOTUS
Follow

Replying to @realDonaldTrump

.@realdonaldtrump tweeted about Facebook, Inc. I shorted the stock at $168.67 and lost $0.30.

> **Donald J. Trump** @realDonaldTrump
> Thank you Arizona. Beautiful turnout of 15,000 in Phoenix tonight! Full coverage of rally via my Facebook at: facebook.com/DonaldTrump/vi...

7:01 AM - 23 Aug 2017

CASE STUDY: @BOTUS AND T&D

Successes:

- presented well-executed sentiment analyses
- simulated a process that finds an optimal trading strategy

But not so good as a **predictive** tool (unrelated to TM & NLP).

Descriptive data analysis can explain what has happened.

Modeling assumptions are not always applicable to the real world (in the predictive domain).

TEXT MINING VS. NATURAL LANG. PROC.

Text Analysis is the collection of quantitative processes by which we attempt to extract **useful** (actionable) insights from text.

In many ways, text mining is about transitions from **unorganized** states to **organized** states (unstructured data to structured data). Natural language processing (NLP) is about getting machines to react "**appropriately**" when interacting with human languages.

In this course:

- **Text Mining** refers to applications of data science tasks to text data
- **NLP** is reserved for tasks that seek an "understanding" of languages

TEXT MINING APPLICATIONS

Classification

- authorship questions, distinguishing true/false statements, etc.

Value Estimation

- sentiment analysis, bias detection, etc.

Clustering

- topic modeling, information retrieval and recommendations, etc.

Others

- text description, text visualization, etc.

UNDERSTANDING LANGUAGE

Syntax

- lemmatization, part-of-speech tagging, sentence boundary disambiguation, etc.

Semantics

- machine translation, language generation, named entity recognition, topic segmentation, questions and answers, etc.

Discourse

- discourse analysis, summarization, etc.

Speech

- recognition, segmentation, text-to-speech, etc.

TM IS EASY, NLP IS AI-HARD

[https://www.youtube.com/watch?time_continue=13&v=EdsIMJg1m2E]

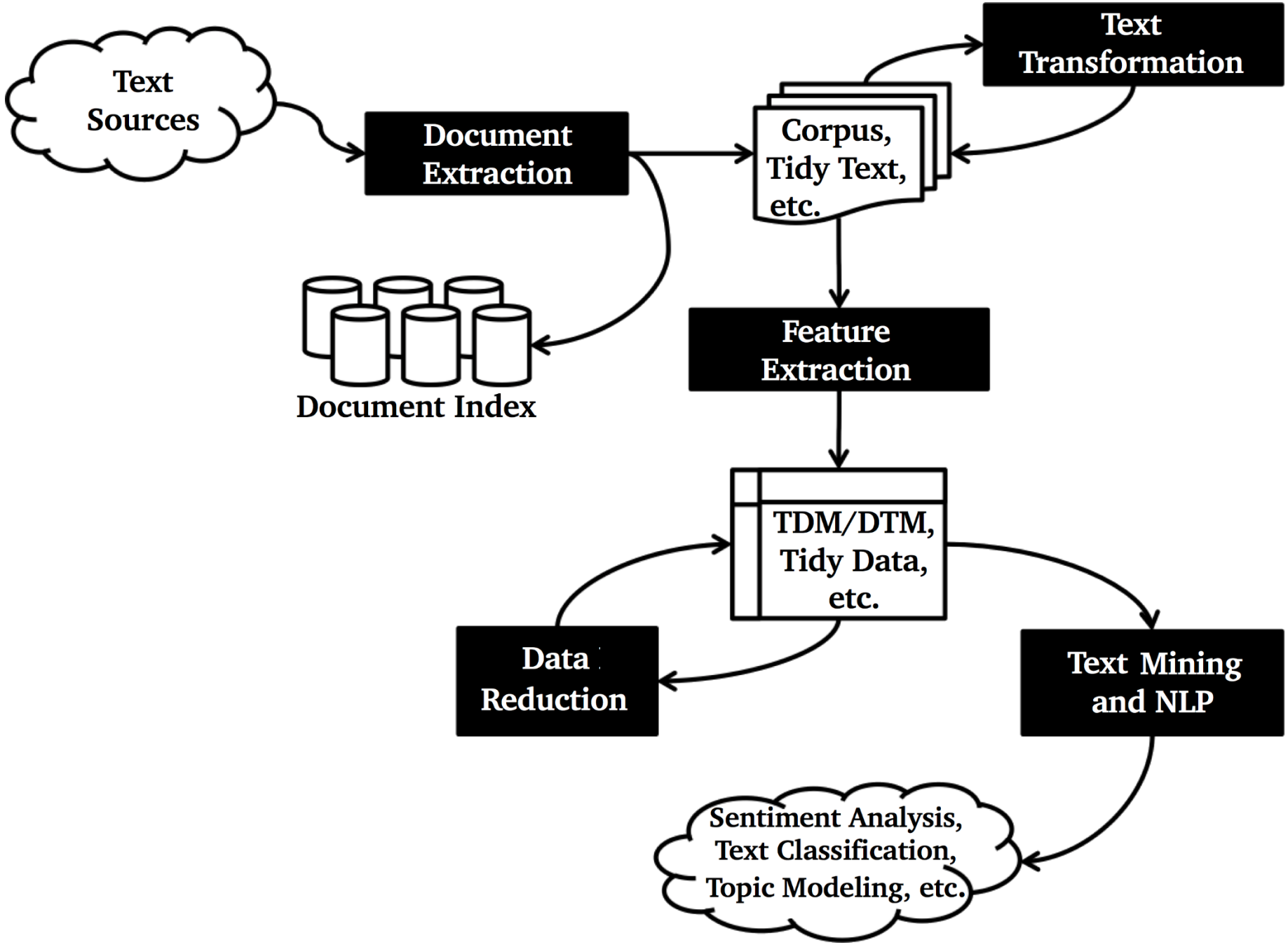
MACHINE TRANSLATION

J'ai été au sud du sud au soleil
Bleu blanc rouge les palmiers
Et les cocotiers glacés
Dans les pôles aux esquimaux bronzés
Qui tricottent des ceintures flèchés
Farcies
Et toujours la Sophie
Qui venait de partir

(*Lindberg*, R. Charlebois)

I was south of south in the sun
Blue white red palm trees
And frozen coconut palms
In the poles to the tanned Eskimos
Who knit arrow belts
Stuffed
And always Sophie
Who had just left

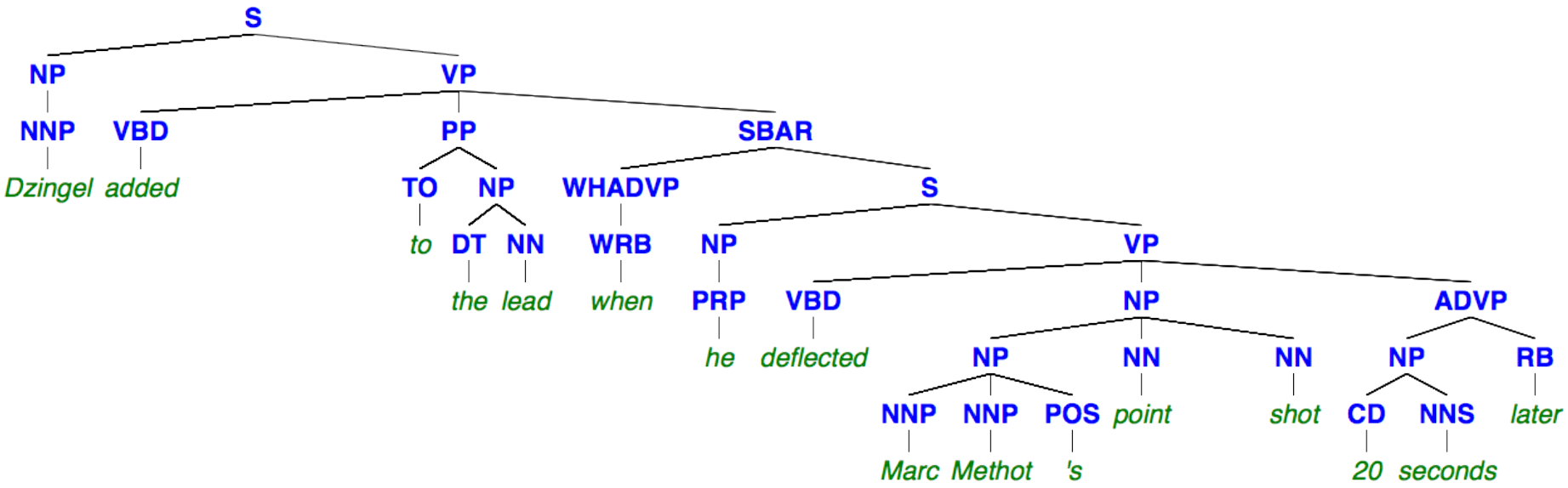
???


Text Sources
Document Extraction
Text Transformation
Corpus, Tidy Text, etc.
Document Index
Feature Extraction
TDM/DTM, Tidy Data, etc.
Data Reduction
Text Mining and NLP
Sentiment Analysis, Text Classification, Topic Modeling, etc.

SEMANTIC PARSING

The process of converting a sentence in a natural language to a **formal meaning representation**.

Word **order** and word **type**/role provide the word's **attributes**.

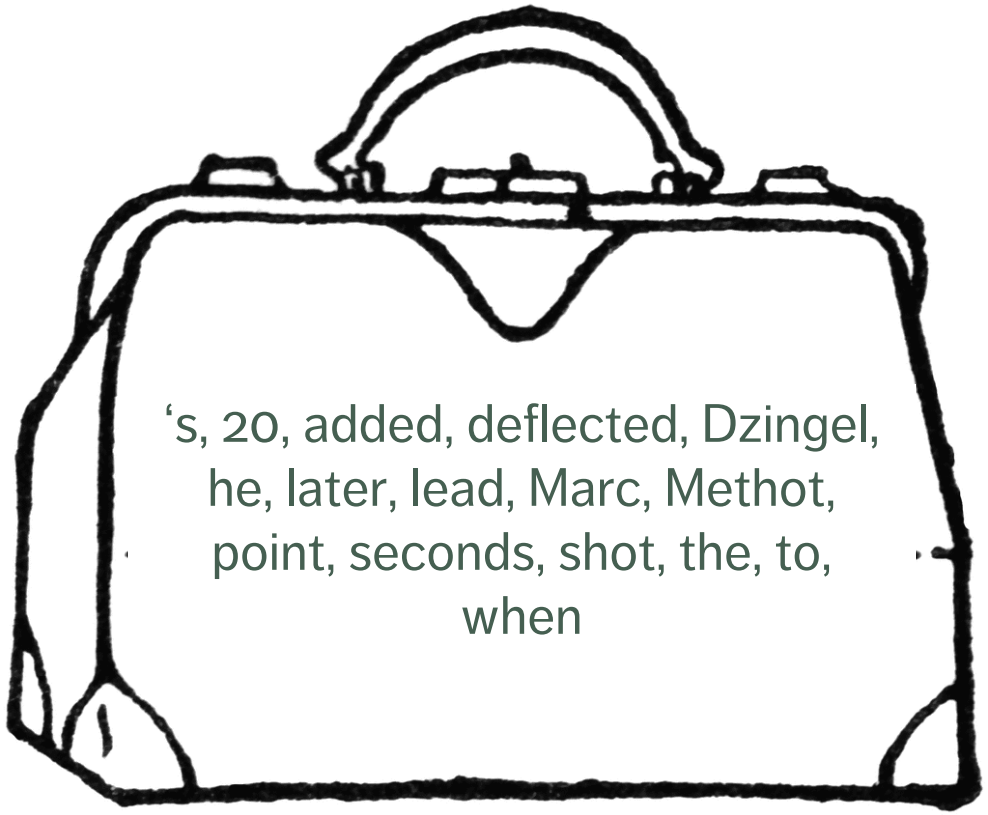


BAG OF "WORDS" (BOW)

Only the **presence** (or **absence**) of "words" (stems, n-grams, sentences, etc.) is important.

Relative **frequencies** provide infor-mation (intent, theme, feeling, etc.) about the corpus.

The words **themselves** are attributes of the document.



TEXT PROCESSING

Text data requires extensive cleaning and processing.

There are a number of challenges due to the nature of the data:

- what is an anomaly in the text?
- what is an outlier?
- are these concepts even definable?
- how do we deal with encoding errors?

Spelling mistakes and typographical errors are difficult to catch in large documents, even with spell-checkers.

TEXT PROCESSING

The process can be simplified to some extent with the help of **regular expressions** and **text pre-processing functions**.

Specific pre-processing steps vary depending on the problem:

- *tweetish* uses a different vocabulary than *legalese*
- ditto for a child who's learning to speak and a Ph.D. candidate

As is almost everything else related to text mining, the cleaning process is **strongly context-dependent**.

Note that the order of pre-processing tasks can affect results.

TEXT PROCESSING – OPTIONS

Convert all letters to **lower case** (avoid when seeking names)

Remove all **punctuation** marks (avoid if seeking emojis)

Remove all **numerals** (avoid when mining for quantities)

Remove all extraneous **white space**

Remove characters within **brackets** (avoid if seeking tags)

Replace all **numerals with words**

TEXT PROCESSING – OPTIONS

Replace **abbreviations**

Replace **contractions** (avoid if seeking non-formal speech)

Replace all **symbols with words**

Remove **stop words** and **uninformative words** (language-, era- and context-dependent)

Stem words and **complete stems** to remove empty variation

- "sleepiness", "sleeping", "sleeps", "slept" convey the meaning of "sleep"
- in "operations research", "operating systems" and "operative dentistry", the stem "operati" needs to stand it for **different meanings**

TEXT PROCESSING

Phonetic accent representation
ya new cah's wicked pissa!

Neologisms and portmanteaus
I'm planning prevenge?

Poor translations/foreign words

Puns and play-on-words

Mark-up, tags, and uninformative text
<b>; \includegraphics; ISBN blurb

Specialized vocabulary
clopen; poset; retro encabulator

Fictional names and places
Qo'noS; Kilgore Trout

Slang and curses
skengfire; #$&#!

TEXT REPRESENTATION

Text must be stored to data structures with right properties:

- a **string** or vector of characters, with language-specific encoding
- a **corpus** (collection) of text documents (with meta information)
- a **document-term matrix** (DTM) where the rows are documents, the columns are terms, and the entries are an appropriate text statistic (or the transposed **term-document matrix** (TDM)
- a **tidy text dataset** with one **token** (single word, *n*-gram, sentence, paragraph) per row

No magic recipe: best format depends on the problem at hand. But this step is **crucial**, both for semantic analysis and BoW.

TEXT STATISTICS

Consider a corpus $\mathcal{C} = \{d_1, \dots, d_N\}$ consisting of N **documents** and M BoW **terms** $\mathcal{C} = \{t_1, \dots, t_M\}$.

For instance, if

$$\mathcal{C} = \left\{ \begin{array}{c} \text{“the dogs who have been let out”}, \\ \text{“who did that”}, \\ \text{“my dogs breath smells like dogs food”} \end{array} \right\},$$

then

$$N = 3, d_1 = \text{“the dogs who have been let out”},$$

$$d_2 = \text{“who did that”}, d_3 = \text{“my dogs breath smells like dogs food”}$$

TEXT STATISTICS

The **relative term frequency** of t in d is

$$tf^*_{t,d} = \frac{\#\text{ of times } t \text{ occurs in } d}{M_d}$$

The **relative document frequency** of t is

$$df^*_t = \frac{\#\text{ of documents in which } t \text{ occurs}}{N} = \frac{\sum_d \text{sign}(tf^*_{t,d})}{N}$$

TEXT STATISTICS

The **term frequency – inverse document frequency** of t in d is

$$tf\text{–}idf^*_{t,d} = -tf^*_{t,d} \times \ln(df^*_t)$$

$tf\text{–}idf^*_t$		t													
		1 been	2 breath	3 did	4 dogs	5 food	6 have	7 let	8 like	9 my	10 out	11 smells	12 that	13 the	14 who
d	1	0.16	0	0	0.06	0	0.16	0.16	0	0	0.16	0	0	0.16	0.06
	2	0	0	0.37	0	0	0	0	0	0	0	0	0.37	0	0.14
	3	0	0.16	0	0.12	0.16	0	0	0.16	0.16	0	0.16	0	0	0

TEXT STATISTICS

If **all the documents** contain the term t, then $\mathrm{df}_t^* = 1$ and

$$tf\text{–}idf_{t,d}^* = -tf_{t,d}^* \times \ln(1) = 0$$

(that terms does not provide information)

If a term t **rarely occurs** in a document d, then $tf_{t,d}^* \approx 0$ and

$$tf\text{–}idf_{t,d}^* \approx -0 \times \ln(df_t^*) \approx 0.$$

Terms that appear relatively often only in a small subset of documents are crucial to understanding those documents **in the general context** of the corpus.

SENTIMENT ANALYSIS BASICS

Most of us have a good native understanding of the emotional intent of words, which leads us to infer **surprise**, **disgust**, **joy**, **pain** (and so forth) from a text segment

The process, when applied by machines to a block of text, is called **sentiment analysis** (opinion mining).

Typical SA questions:

- "Is this movie review positive or negative?"
- "Is this customer email a complaint?"
- "Have newspapers' attitudes about the PM changed since the election?"

CHALLENGES

Most humans would **typically** be able to answer these questions when presented with the appropriate text documentation. For machines, that is not as obvious a problem to solve.

Challenges:

- we don't always agree on the emotional content of a text
- words may have different meaning/emotional value depending on the context (anti-antonyms)
- qualifiers can drastically change a term's emotional value
- topic changes
- rhetorical devices

RELATED TASKS

Sentiment analysis is a **supervised learning** problem, requiring dictionaries of emotional content to have been compiled ahead of time (internally or externally)

Related Tasks:

- discarding subjective information (information extraction)
- recognizing opinion-oriented questions (question answering)
- accounting for multiple viewpoints (summarization)
- identifying suitability of videos for kids, bias in news sources, and appropriate content for ad placement

Element of **subjectivity**

TYPES OF SENTIMENT ANALYSIS

In this course, we differentiate 2 types of sentiment analyses:

- **term-by-term** (tbt) looks at the emotional content of tokens and tries to deduce a score for passages containing them
- **document-by-document** (dbd) looks at scored passages and tries to find tokens which carry the emotional load or predict how a new passage would score on some emotional spectrum

TBT is not a complicated technical task: it only requires the ability to match a lexicon score to a term, and to add the scores.

DBD is basically a classification problem – it requires labeled text data, but the principle is exactly the same: predict "**positive/negative**" labels (see exercise).

SENTIMENT LEXICONS

TBT sentiment analysis relies heavily on **lexicons** – list of terms which have been ranked on some emotional scale

- AFINN: words on a scale from -5 (negative) to 5 (positive)
- BING: binary negative/positive
- NRC: words are assigned category(ies) of sentiments
- LOUGHRAN: categorical bins

Each of these lexicons contains a majority of **negative** terms.

The best choice of lexicon is dictated by **context**.

SENTIMENT LEXICONS

“abandon”

AFINN: -2
BING: NA
NRC: fear, negative, sadness
LOUGHRAN: negative

“not”

AFINN: NA
BING: NA
NRC: NA
LOUGHRAN: NA

“bad”

AFINN: -3
BING: negative
NRC: anger, disgust, fear, etc.
LOUGHRAN: negative

“egregious”

AFINN: ?
BING: ?
NRC: ?
LOUGHRAN: ?

SENTIMENT LEXICONS

Once a lexicon has been selected, TBT is simply a matter of **chunking the text** and computing sentiment scores on each block (every 100 words, every 100 lines, every chapter, etc.)

Is there any reason to expect the various lexicons to give the same scores?

(Shakespeare's *Macbeth*, AFINN scene scores)