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

CASE STUDY: @BOTUS AND TRUMP&DUMP

TEXT MINING AND SENTIMENT ANALYSIS

Sentiment Analysis of Tweets

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

@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 []

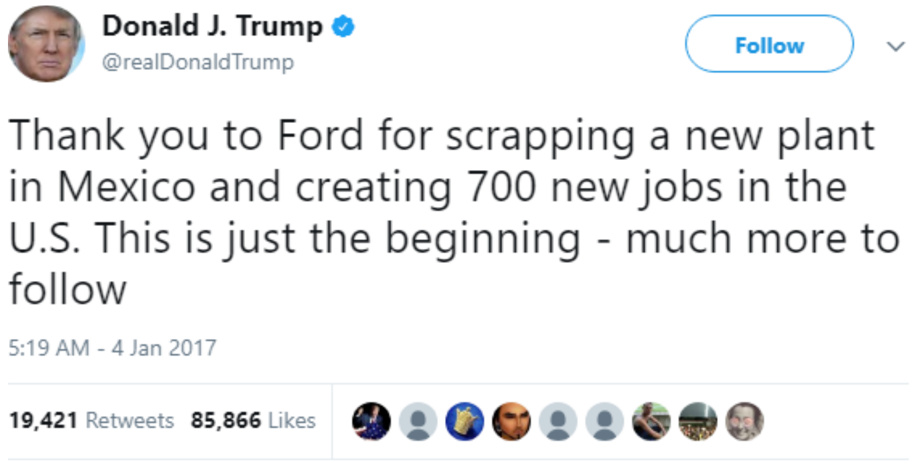
@BOTUS AND T&D

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.

Trump and Dump is not a financial product. You can't use it so don't ask. No legal claims are being made.
This is all meant for fun and to save puppies. Profit and number of puppies saved are all based on projections and simulations.
Do not pass go. Do not collect $100. And no puppies were harmed in the making of this bot.

“I can’t believe YOU’re the President!!!” vs. “I can’t believe you’re the PRESIDENT!!!”

@BOTUS AND T&D



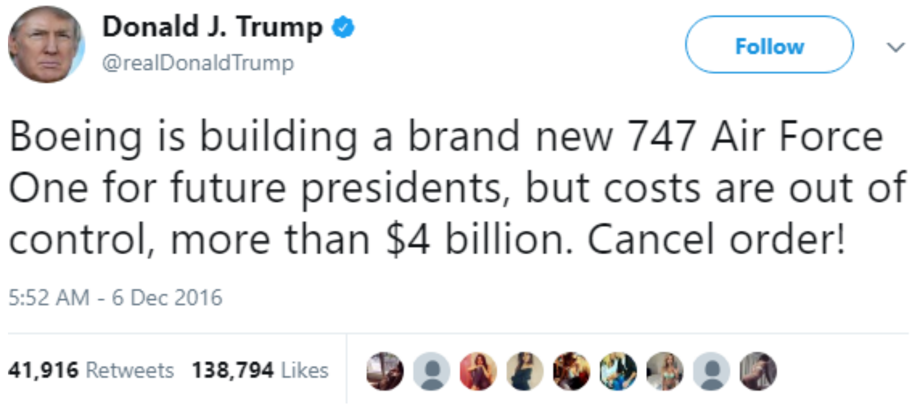
Donald J. Trump
@realDonaldTrump

Follow

Thank you to Ford for scrapping a new plant in Mexico and creating 700 new jobs in the U.S. This is just the beginning - much more to follow

5:19 AM - 4 Jan 2017

19,421 Retweets 85,866 Likes



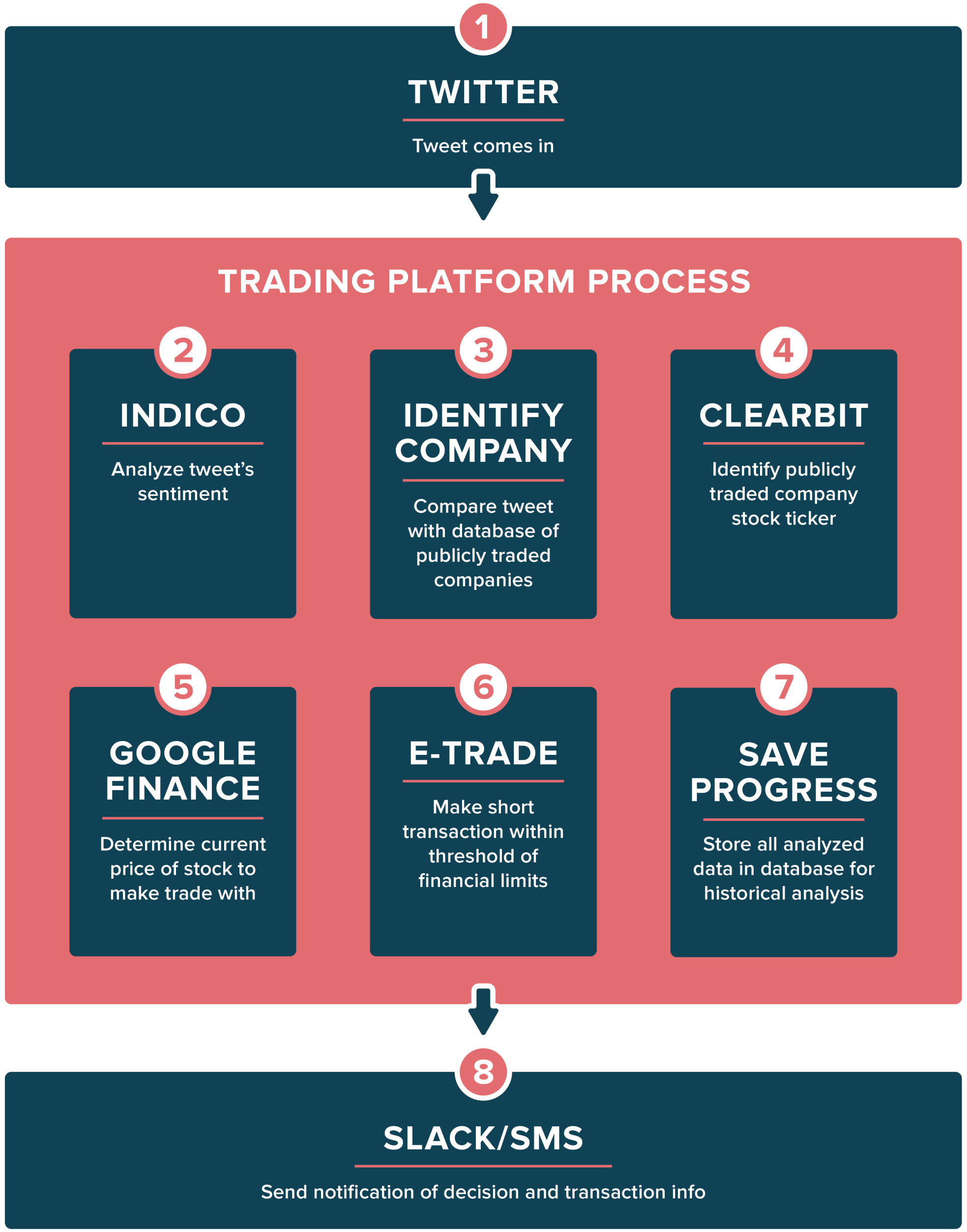
Donald J. Trump
@realDonaldTrump

Follow

Boeing is building a brand new 747 Air Force One for future presidents, but costs are out of control, more than $4 billion. Cancel order!

5:52 AM - 6 Dec 2016

41,916 Retweets 138,794 Likes



1 TWITTER

Tweet comes in

TRADING PLATFORM PROCESS

2 INDICO

Analyze tweet's sentiment

3 IDENTIFY COMPANY

Compare tweet with database of publicly traded companies

4 CLEARBIT

Identify publicly traded company stock ticker

5 GOOGLE FINANCE

Determine current price of stock to make trade with

6 E-TRADE

Make short transaction within threshold of financial limits

7 SAVE PROGRESS

Store all analyzed data in database for historical analysis

8 SLACK/SMS

Send notification of decision and transaction info

@BOTUS AND T&D

Natural languages are rich, flexible, and can allow for syntax variations (+ for humans, – for bots).

A word’s meaning can be highly **context-dependent**.

Sarcasm, idioms, figures of speech… humans don’t even always recognize them.

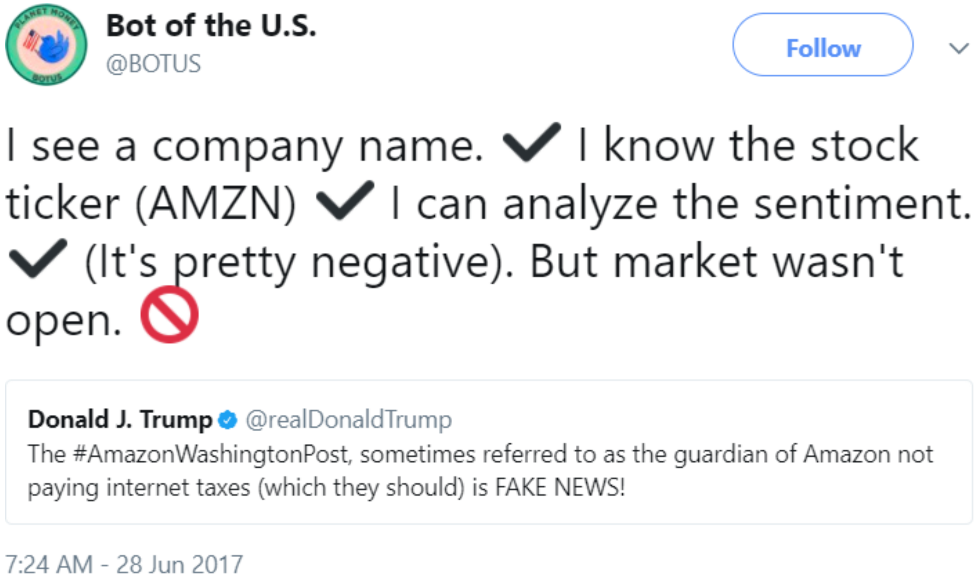
Named-entity recognition: Apple (company) vs. apple (food).

@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.

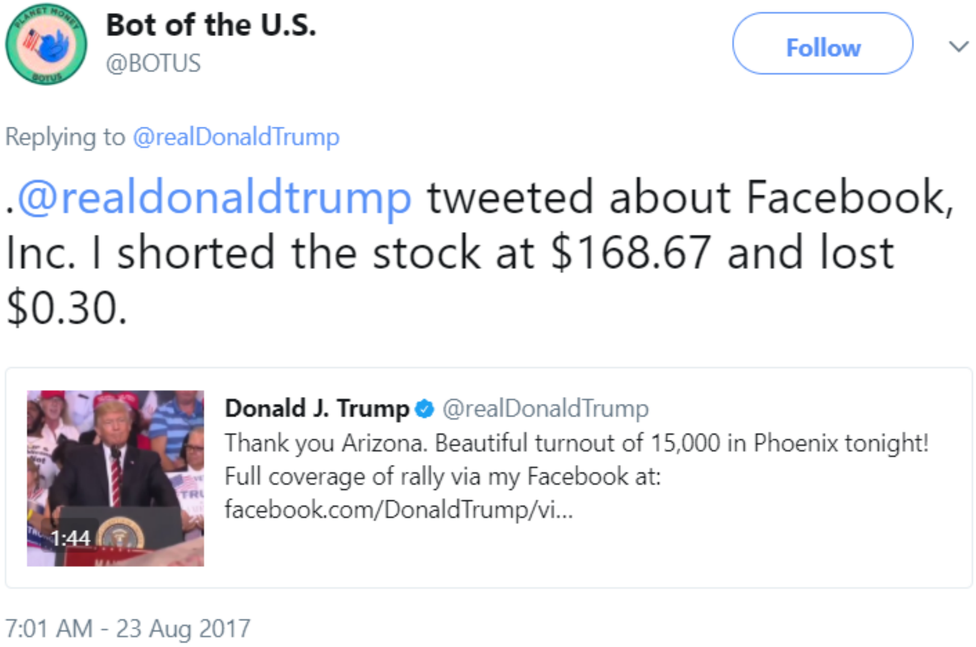


Bot of the U.S.
@BOTUS

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

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

@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).

DISCUSSION

How important are visual cues in communications and business negotiations? How important is context?

On a related note, how easy is it to learn from someone whose context is different from yours (culturally AND professionally)?

TEXT MINING AND NLP

TEXT MINING AND SENTIMENT ANALYSIS

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

DREAM OF THE RED CHAMBER (红楼梦)

宝玉道：“一言难尽.” 说者便把梦中之事细说与袭人听了. 然后说至警幻所授云雨之情，羞得袭人掩面伏身而笑. (*original by Cao Xueqin*)

“It’s a long story,” answered Pao-yu, then told his dream in full, concluding with his initiation by Disenchantment into the “sport of cloud and rain”. His-jen, hearing this, covered her face and doubled up in a fit of giggles. (*translated by Yang Xianyi*)

After much hesitation he proceeded to give her a detailed account of his dream. But when he came to the part of it in which he made love to Two-in-one, Aroma threw herself forward with a shriek of laughter and buried her face in her hands. (*translated by D. Hawkes*)

Bao Yudao: "A word is hard to do." The speaker described the dream things in detail and hit people. Then he talked about the cloud and rain sent by the police illusion, and was ashamed to hide behind and laugh. (*machine translation*)

[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)

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

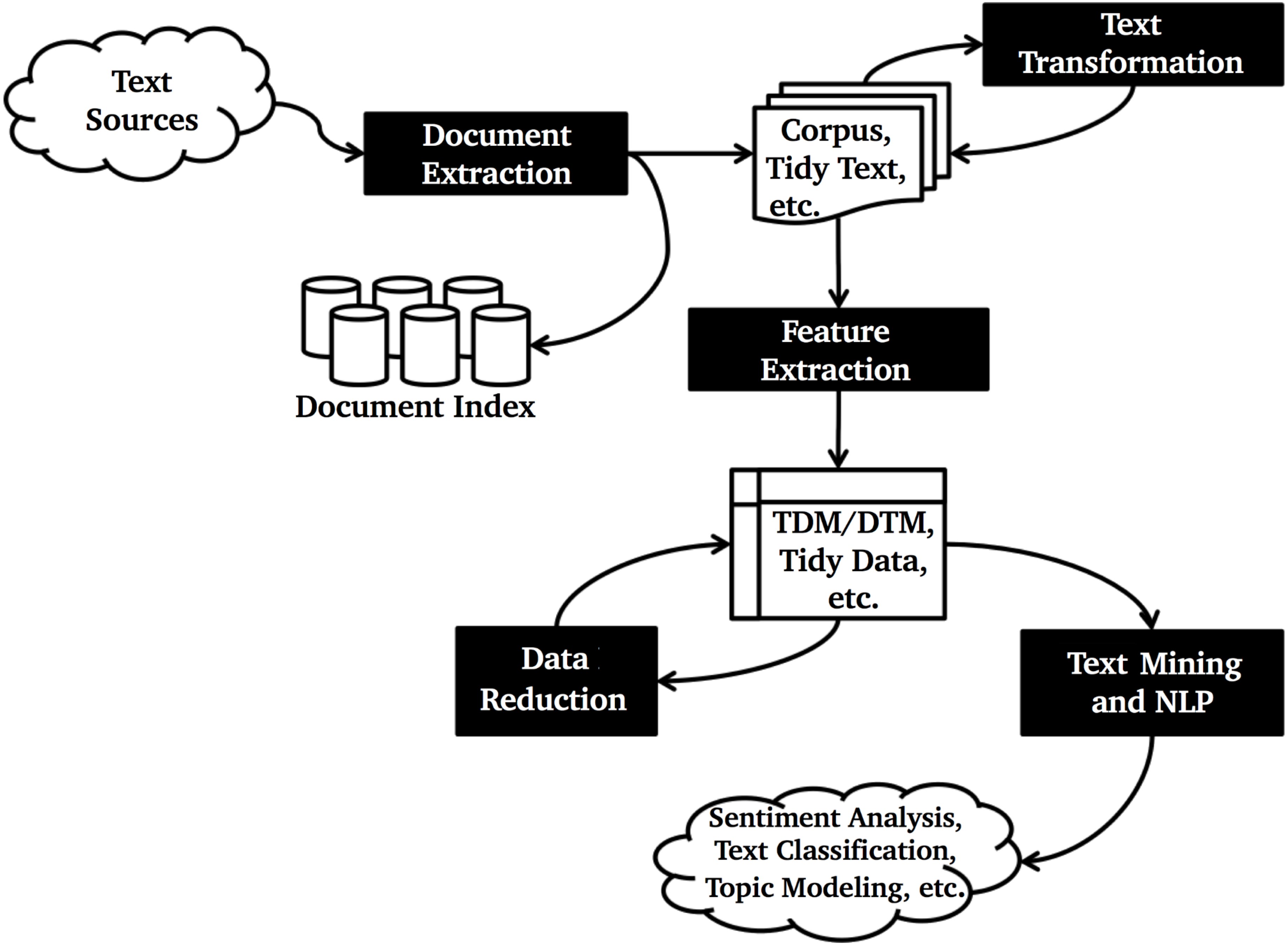
MACHINE TRANSLATION

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

DISCUSSION

In numerical data analysis, it can be difficult (even for experts) to tell when the results are nonsensical. That’s not the case for text data analysis, as most of us can tell at a glance when something is off.

What steps can you take to help catch nonsensical results from being deployed/released too soon?

TEXT MINING BASICS

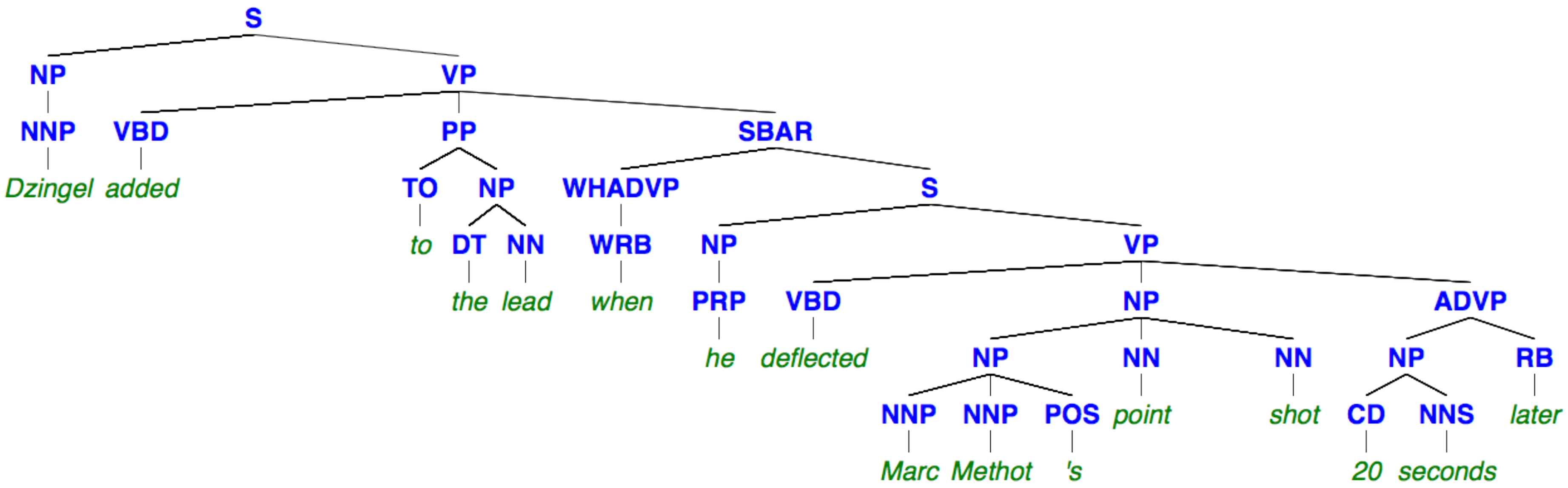
TEXT MINING AND SENTIMENT ANALYSIS

“Dzingel added to the lead when he deflected Marc Methot’s point shot 20 seconds later.”
(Associated Press game recap, Ottawa Senators vs. Toronto Maple Leafs, February 18, 2017)

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**.

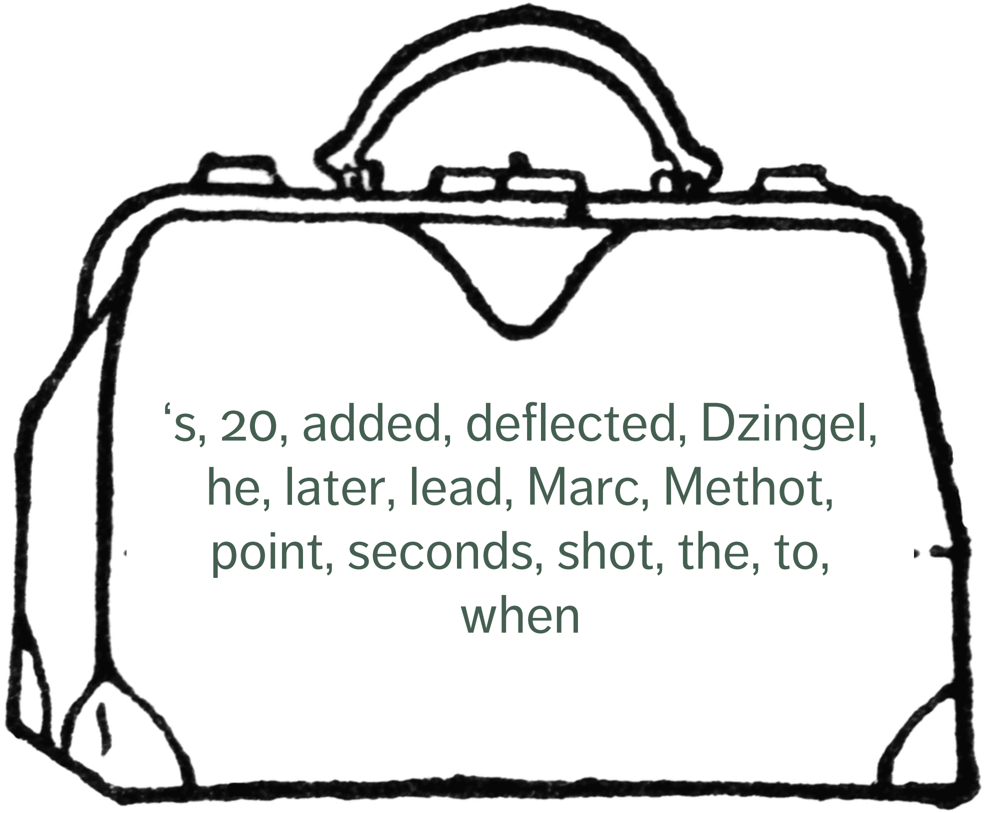


BAG OF "WORDS" (BOW)

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.



's, 20, added, deflected, Dzingel, he, later, lead, Marc, Methot, point, seconds, shot, the, to, when

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

"Dzingel added lead deflected
Marc Methot point shot twenty seconds later"

added, deflected, Dzingel, later, lead, Marc, Methot, point, seconds, shot, twenty

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; #$&#!

EXERCISE

How would you process the following bit of text?

"<i>He<i> went to bed at   2 A.M. It\'s way too late! He was only 20% asleep at first, but sleep eventually came."

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.

DTM/TDM REPRESENTATION

	Document 1	Document 2	Document 3	…	Document *N*
Token 1	0	0	1	62	3
Token 2	0	1	0	61	2
Token 3	1	0	3	101	0
…	112	24	38	84	0
Token *M*	2	2	0	12	3

Sum
66
64
105
258
19

Sum	115	27	42	320	8

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

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

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

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

$tf^*_{t,d}$		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	1/7	0	0	1/7	0	1/7	1/7	0	0	1/7	0	0	1/7	1/7
	2	0	0	1/3	0	0	0	0	0	0	0	0	1/3	0	1/3
	3	0	1/7	0	2/7	1/7	0	0	1/7	1/7	0	1/7	0	0	0

TEXT STATISTICS

The **relative document frequency** of t is

$$df_t^* = \frac{\text{\# of documents in which } t \text{ occurs}}{N} = \frac{\sum_d \mathrm{sign}(tf_{t,d}^*)}{N}$$

df_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
	1/3	1/3	1/3	2/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	2/3

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 $\text{df}_t^* = 1$ 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.

DISCUSSION

At the analysis stage, it is easy to forget where the data comes from and what it really applies to.

Text comes unstructured and unorganized. After processing, text is clean, but still unstructured. Bag of Words provides a framework for a structured numerical representation of text.

How does this affect the choice of text statistic in the DTM/TDM?

tf-idf may not always be the ideal choice.... An approach based on weighted log odds could be preferable at times.

SENTIMENT ANALYSIS

TEXT MINING AND SENTIMENT ANALYSIS

"[...] classifying social media posts by hand isn't practical at scale (though some firms do manually classify samples to make their algorithms better). But a simple thumbs up or down as "sentiment" is worse than meaningless–it's simply not true."

(S. Kessler, *The Problem With 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

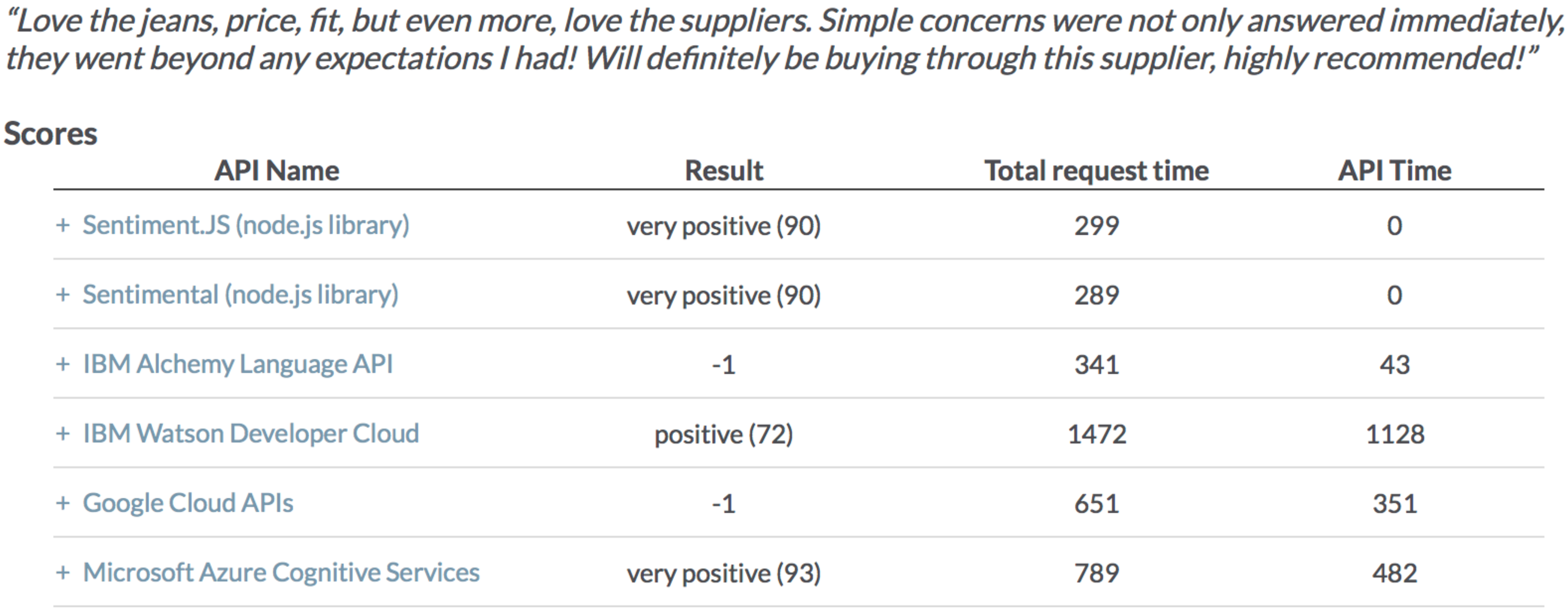
Element of **subjectivity**

EXERCISE

Three reviews (1-star, 3-star, 5-star) were found on Amazon.ca. Can you identify them? Can you identify the product?

- “Love the jeans, price, fit, but even more, love the suppliers. Simple concerns were not only answered immediately, they went beyond any expectations I had! Will definitely be buying through this supplier, highly recommended!”
- “DON'T BUY. Great series aside, this special addition is pathetic. They're basically mass-market paperbacks: small and uncomfortable to hold. The regular paperback versions are far superior for basically the same price.”
- “Beginning the second use, the bowl keeps falling out 30 seconds after the mixing starts. A bit disappointed.”

[http://aiplaybook.a16z.com/docs/guides/nlp#user-content-apiexamples]

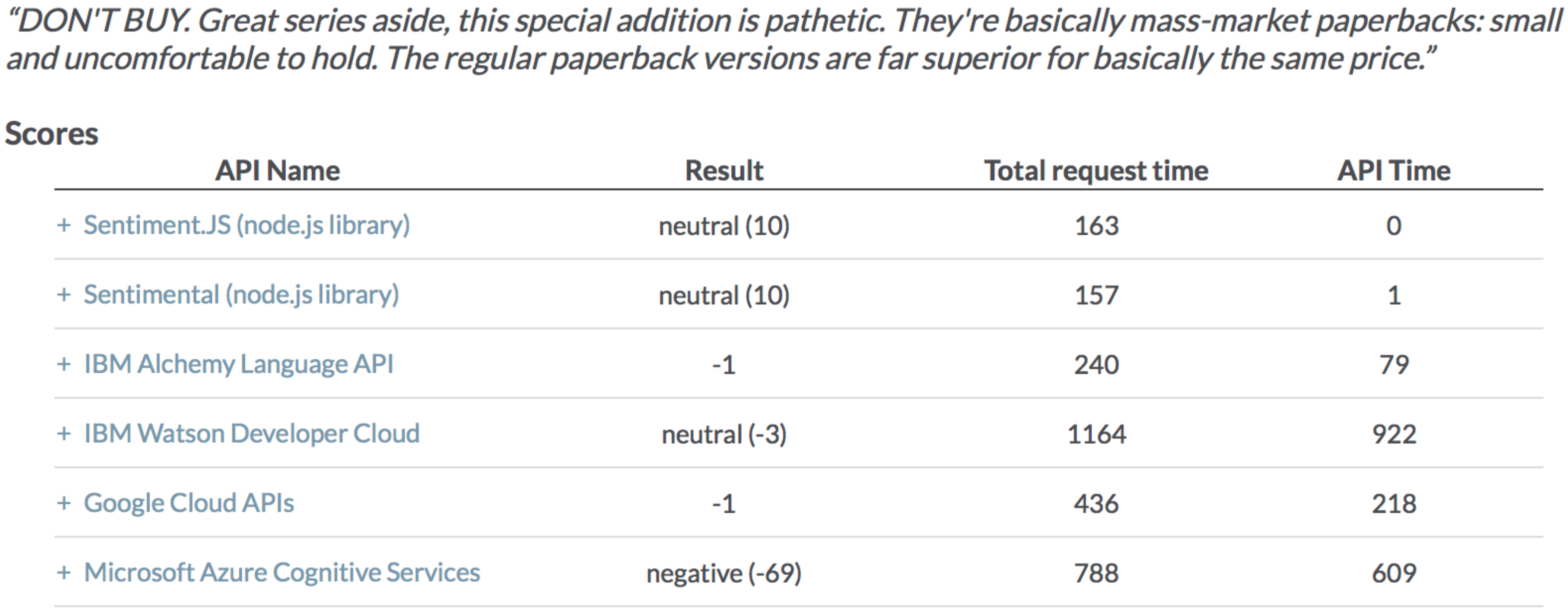


"Love the jeans, price, fit, but even more, love the suppliers. Simple concerns were not only answered immediately, they went beyond any expectations I had! Will definitely be buying through this supplier, highly recommended!"

Scores

API Name	Result	Total request time	API Time
+ Sentiment.JS (node.js library)	very positive (90)	299	0
+ Sentimental (node.js library)	very positive (90)	289	0
+ IBM Alchemy Language API	-1	341	43
+ IBM Watson Developer Cloud	positive (72)	1472	1128
+ Google Cloud APIs	-1	651	351
+ Microsoft Azure Cognitive Services	very positive (93)	789	482

[http://aiplaybook.a16z.com/docs/guides/nlp#user-content-apiexamples]

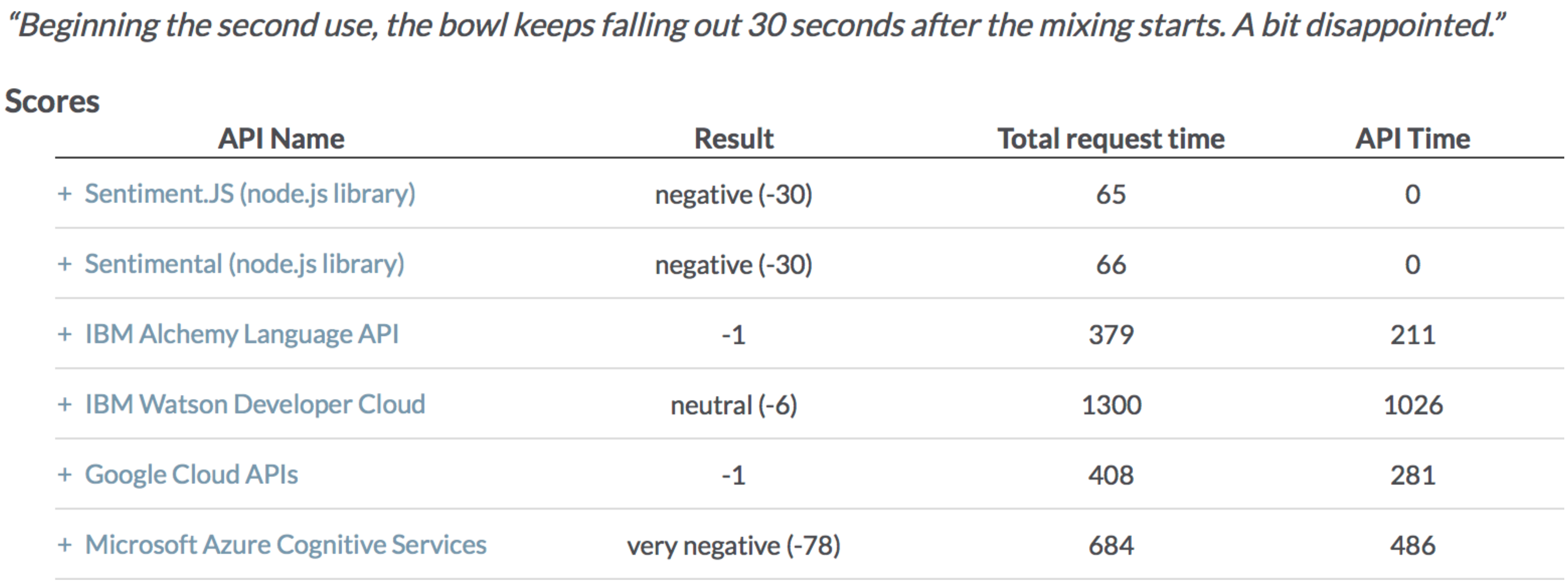


"DON'T BUY. Great series aside, this special addition is pathetic. They're basically mass-market paperbacks: small and uncomfortable to hold. The regular paperback versions are far superior for basically the same price."

Scores

API Name	Result	Total request time	API Time
+ Sentiment.JS (node.js library)	neutral (10)	163	0
+ Sentimental (node.js library)	neutral (10)	157	1
+ IBM Alchemy Language API	-1	240	79
+ IBM Watson Developer Cloud	neutral (-3)	1164	922
+ Google Cloud APIs	-1	436	218
+ Microsoft Azure Cognitive Services	negative (-69)	788	609

[http://aiplaybook.a16z.com/docs/guides/nlp#user-content-apiexamples]



"Beginning the second use, the bowl keeps falling out 30 seconds after the mixing starts. A bit disappointed."

Scores

API Name	Result	Total request time	API Time
+ Sentiment.JS (node.js library)	negative (-30)	65	0
+ Sentimental (node.js library)	negative (-30)	66	0
+ IBM Alchemy Language API	-1	379	211
+ IBM Watson Developer Cloud	neutral (-6)	1300	1026
+ Google Cloud APIs	-1	408	281
+ Microsoft Azure Cognitive Services	very negative (-78)	684	486

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.

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

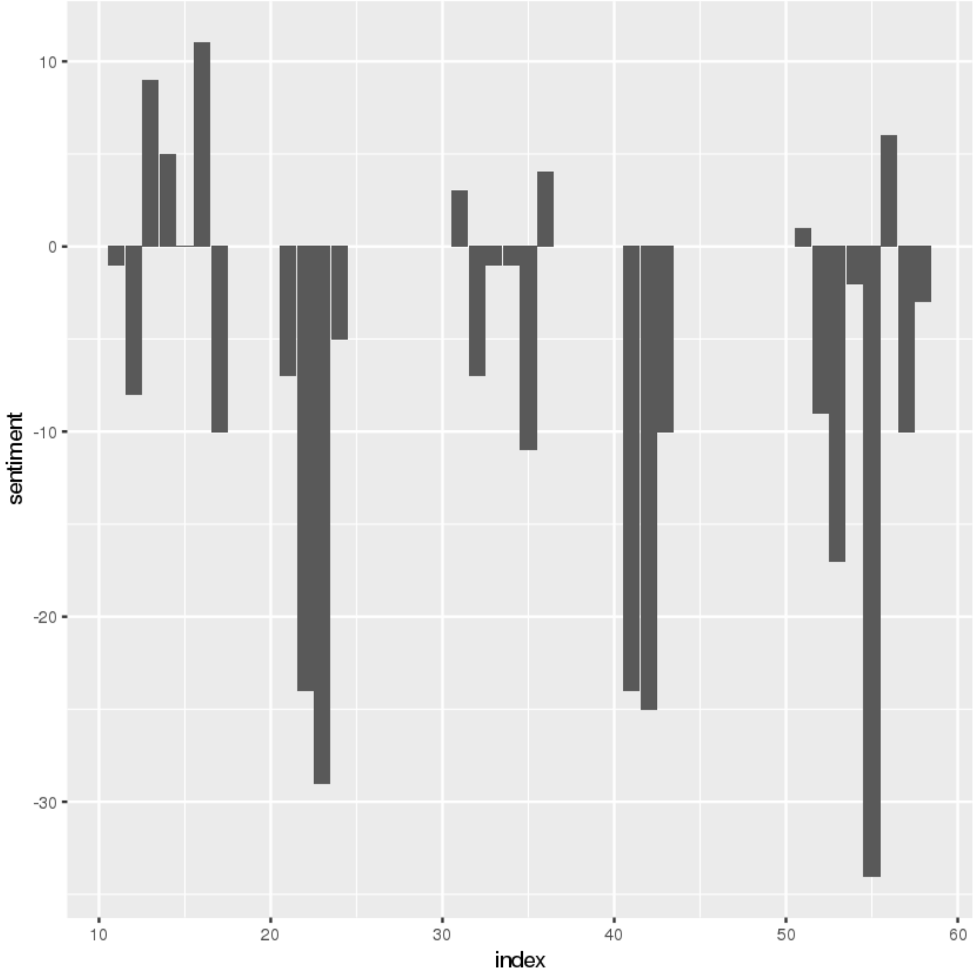
“egregious”

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)

DISCUSSION

Most words in the English language are neutral. Why are most lexicon words negative, then? Is this also the case for written Chinese? For French?

Are lexicons interchangeable (time, culture, context)?

EXAMPLE: MOVIE REVIEWS

TEXT MINING AND SENTIMENT ANALYSIS

> “No good movie is too long and no bad movie is short enough”
>
> (Roger Ebert)

STATEMENT OF EXERCISE

The emphasis of this exercise is not on programming; rather it is to showcase a complete analytical workflow using Python's Natural Language Toolkit (NLTK).

The goal is to develop a sentiment analysis model for movie reviews. The dataset contains 50000 movie reviews labeled as either **positive** or **negative**.

With an accurate sentiment model we'll have the ability to automatically classify new reviews in order to aggregate review data, say.

EXERCISE

1. **Dataset Information**
   - How many positive reviews are there in the training set? How many negative reviews?
   - How many positive reviews are there in the testing set? How many negative reviews?
   - What is the range of scores for positive and negative reviews in the training and testing sets?
   - What effect can the absence of neutral reviews in the training and testing set have?
2. **Data Preparation**
   - Select 10 words which you think explain why the review of *Haunted Boat* (3446_1.txt) is a 1-star review.
   - Select 10 words which you think explain why the review of *Night Listener* (10015_8.txt) is an 8-star review.

EXERCISE

3. **Bag-of-Words Processing**
   - All the processed tokens in all the training set reviews are used to create a document-term matrix (DTM). Describe the process to go from the full review text to the review tokens.
   - How many tokens are retained in the DTM?
   - How is this number dependent on the nature of the tokenizer? (notice the mistake in the notebook: the shape provided by the output is not the shape described in the explanation).

4. **Multinomial Naïve Bayes**
   - Does the multinomial naïve Bayes classifier built on the training DTM suggests that review 9999_1.txt is positive or negative? Are any of the words you identified in Question 2 found in this review.
   - Same question, but for review 9999_10.txt.

EXERCISE

5. **Performance Evaluation**

   - Describe the sentiment analyzer's performance provided by the classification report.
   - Why do you think that the performance of VADER (NLTK's pre-trained sentiment analyzer) poorer than the model that was trained on the review data?

REFERENCES

TEXT MINING AND SENTIMENT ANALYSIS

NOTEBOOKS

Text Processing, Text Visualization, Text Clustering, Sentiment Analysis Notebooks (in HTML format)
https://www.data-action-lab.com/wp-content/uploads/2019/03/TMNotebooks.zip

REFERENCES

Basu, T. [2017], NPR's Fascinating Plan to Use A.I. on Trump's Tweets, retrieved from inverse.com on September 12, 2017.

Goldmark, A. [2017], Episode 763: BOTUS, Planet Money podcast, retrieved from NPR.org's Planet Money on September 12, 2017.

Greenstone, S. [2017], When Trump Tweets, This Bot Makes Money, retrieved from NPR.org on September 12, 2017

Mettler, K. [2017], 'Trump and Dump': When POTUS tweets and stocks fall, this animal charity benefits, retrieved from the *Washington Post* on September 19, 2017

Jockers, M.L. [2014], *Text Analysis with R for Students of Literature*, Springer.

Anastasia, D.C., Tagarelli, A., Karypis, G. [2014], *Document Clustering: The Next Frontier,* in *Data Clustering: Algorithms and Applications* (Aggarwal, C.C., Reddy, C.K., eds.), CRC Press.

REFERENCES

Aggarwal, C.C., Zhai, C.X. [2015], *Text Classification*, in *Data Classification: Algorithms and Applications* (Aggarwal, C.C., ed.), CRC Press.

Srivastava, A.N., Sahami, M. (eds.) [2009], *Text Mining: Classification, Clustering, and Applications*, CRC Press.

Silge, J., Robinson, D. [2017], *Text Mining with R: a Tidy Approach*, O'Reilly.

Jurafsky, D., Martin, J.H. [2009], *Speech and Language Processing* (2nd ed), Pearson.

Aggarwal, C.C., Zhai, C.X. (eds.) [2012], *Mining Text Data*, Springer.

Bird, S., Klein, E., Loper, E. [2009], *Natural Language Processing with Python*, O'Reilly.

http://aiplaybook.a16z.com/docs/guides/nlp#user-content-apiexamples

https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/

SUPPLEMENTAL MATERIAL – MULTINOMIAL NAÏVE BAYES CLASSIFICATION

TEXT MINING AND SENTIMENT ANALYSIS

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

Multinomial naïve Bayes is an algorithm where the feature vectors in each class are assumed to have a multinomial distribution (best known application: **spam filters**)

Dataset M with a number of email messages (the **records**)

Each record has n features (the **frequencies** of n selected terms in the email message body)

Each record is represented by a **feature vector** denoted by

$$\boldsymbol{x} = (x_1, \dots, x_n)$$

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

Assume that there are K categories in which a record could be **classified**

- labels: spam, quarantined, personal, business, etc.

Let $\{C_k : k = 1, \ldots, K\}$ denote the categories

The classification problem is to determine

$$\boxed{P(\boldsymbol{x} \in C_k \,|\, x_1, \ldots, x_n) \text{ for each } k}$$

Prediction is given by the class for which this value is **highest**

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

Fix k. From **Bayes' Theorem**, we have

$$P(\boldsymbol{x} \in C_k | x_1, \ldots , x_n) \propto P(C_k) \times P(x_1, \ldots , x_n | \boldsymbol{x} \in C_k)$$

The **naïve** assumption is

$$P(x_1, \ldots , x_n | \boldsymbol{x} \in C_k) = P(x_1 | \boldsymbol{x} \in C_k) \times \cdots \times P(x_n | \boldsymbol{x} \in C_k)$$

so that

$$P(\boldsymbol{x} \in C_k | x_1, \ldots , x_n) \propto P(C_k) \times \prod_{i=1}^{n} P(x_i | \boldsymbol{x} \in C_k)$$

The **multinomial** assumption is

$$P(x_i | \boldsymbol{x} \in C_k) \propto p_{k,i}^{x_i}, \text{ where } p_{k,i} \in [0,1] \text{ for each word } i$$

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

Combining these assumptions, the posterior "probabilities" is

$$P(\boldsymbol{x} \in C_k | x_1, \dots, x_n) \propto P(C_k) \times \prod_{i=1}^{n} p_{k,i}^{x_i}$$

The model can be linearized by taking logarithms

$$\boxed{\log P(\boldsymbol{x} \in C_k | x_1, \dots, x_n) \propto b_k + \sum_{i=1}^{n} x_i \cdot \log p_{k,i}}$$

The classifier is **trained** by estimating the parameters $p_{k,i}$ on a subset of all records and by specifying the "priors" b_k

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

If a message has a token that has never been seen before, it is **impossible** to predict its most likely class membership using (non-existent) past behaviour

To avoid divisions by 0, one could use the corrected estimate

$$\hat{p}_{k,i}=\frac{\sum_{x\in C_k}x_i+1}{\sum_{x\in C_k}\sum_{j=1}^{n}x_j+|v|}=\frac{\#w_i\in C_k+1}{W_k+|v|}$$

$|v|$: size of vocabulary, $\#w_i\in C_k$: count of w_i in C_k, W_k: count of all words in C_k

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

Training set

cl	ID	text
+	i1	I love this phone
+	i2	amazing sound quality
+	i3	Love this great phone
-	i4	i hate it
-	i5	bad quality
-	i6	so bad Hate it

Testing set

?? i7 hate hate HATE the phone quality

Processed training set

cl	ID	text
+	i1	love phone
+	i2	amazing sound quality
+	i3	love great phone
-	i4	hate
-	i5	bad quality
-	i6	bad hate

Processed testing set

?? i7 hate hate hate phone quality

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

$$P(+) = \frac{3}{6} = 0.5 \quad \text{and} \quad P(-) = \frac{3}{6} = 0.5 \quad \text{so} \quad b_{+} = b_{-} = \ln 0.5$$

$$|v| = 8, W_{+} = 8, W_{-} = 5$$

$$\hat{p}_{+,\text{amazing}} = \frac{(\#\text{amazing} \in +) + 1}{W_{+} + 8} = \frac{1+1}{8+8} = \frac{1}{8}$$

$$\hat{p}_{-,\text{amazing}} = \frac{(\#\text{amazing} \in -) + 1}{W_{-} + 8} = \frac{0+1}{5+8} = \frac{1}{13}$$

$$\ldots$$

MULTINOMIAL NAÏVE BAYES CLASSIFICATION

$\hat{p}$	amazing	bad	great	hate	love	phone	quality	sound
+	0.1250	0.6025	0.1250	0.0625	0.1875	0.1875	0.1250	0.1250
−	0.0769	0.2308	0.0769	0.2308	0.0769	0.0769	0.1538	0.0769

Testing set

	amazing	bad	great	hate	love	phone	quality	sound
i7	0	0	0	3	0	1	1	0

$$P(+\mid \boldsymbol{x}) \propto 2.9 \times 10^{-6}$$

$$\boxed{P(-\mid \boldsymbol{x}) \propto 9.7 \times 10^{-6}}$$