Text As Data

UC3M/IC3JM Graduate Workshop, Fall 2024

Patrick Kraft

2024-12-10

Outline for Today

1. Working with Text Data

2. Sentiment Analysis

3. Topic Modeling

1. Working with Text Data

Text Data in the Social Sciences

- Long tradition in the social sciences
  - Content analysis (communication studies, political science, sociology…)
  - Open-ended survey questions
- With the rise of the internet, tons of new data sources
  - Social media data
  - Automatic transcripts of speeches, videos, news, …
- Requires new analytical techniques
  - Text mining, Computational text analysis (CTA), quantitative text analysis, distant reading, etc.
- Related disciplines: Natural language processing (e.g., translation, chat bots), Information retrieval (e.g., search)

Text Data in the Social Science: New challenges

- Social scientists used to work with structured data (e.g., survey data)
- Text often comes as unstructured data (characters, words, sentences, paragraphs, …)
- Text and language often have many nuances, ambiguous meaning, sarcasm, …
- Computational text analysis often…
  - requires a lot of (simplifying) assumptions
    - e.g. assume standard English (social media?!)
  - is more qualitative/subjective than the methods suggest

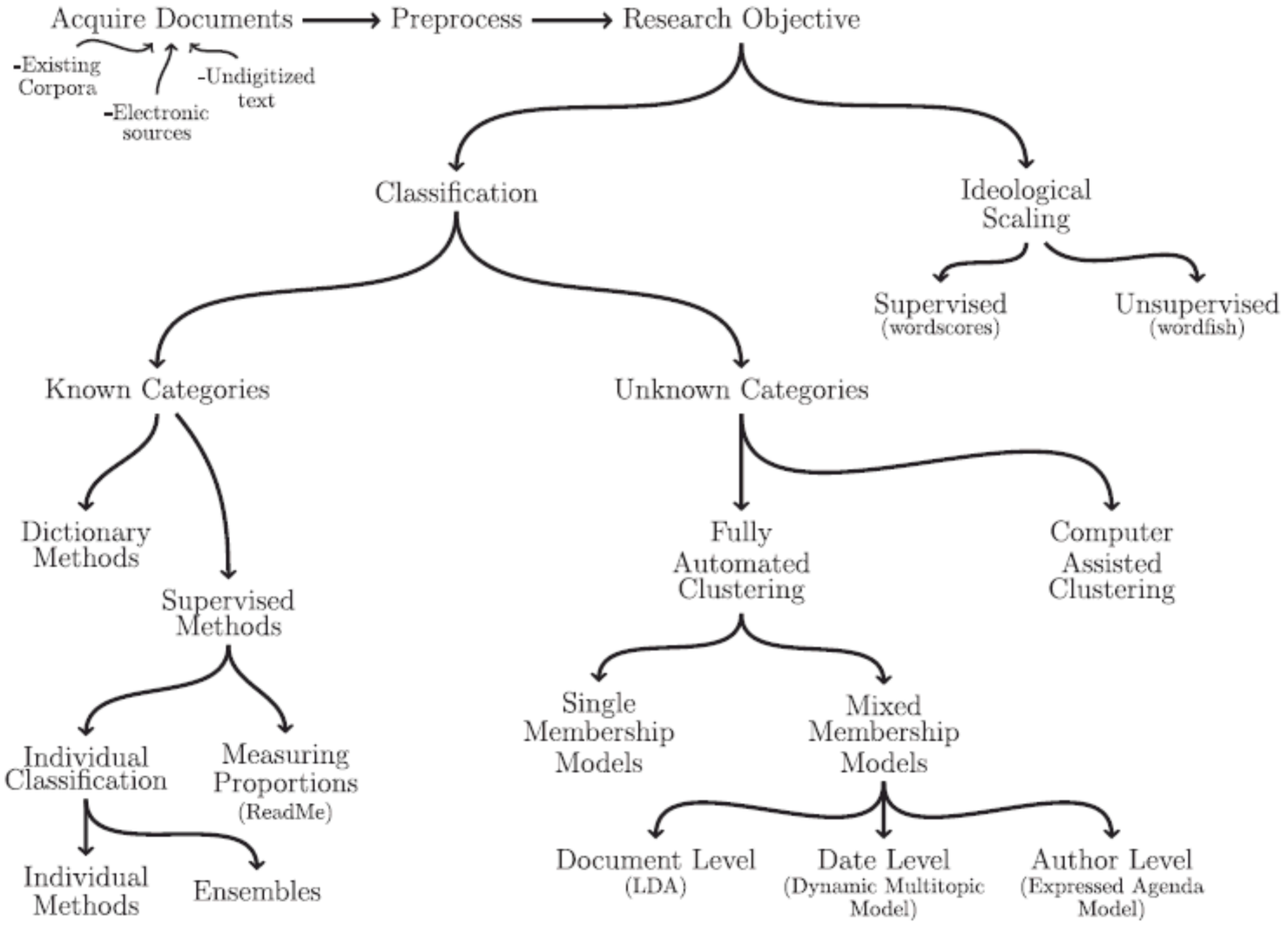
Text as data: Examples (1)

Method	Goal	Examples
Search	Finding relevant content	Literature reviews
Topic detection and Clustering	Understanding text topics	Assigning social media content to topics
Classification	Categorizing text	Detecting trolls and bots in social networks; Identifying fake news
Sentiment analysis	Understanding sentiment of text	What do social media users think of politicians
Word clustering/Synonyms	Finding words with similar meanings	Understanding social media content, e.g., what issues are republicans concerned with?

Text as data: Examples (2)

Method	Goal	Examples
Named entity recognition	Recognition and tagging of named entities	Automated analysis of laws, court rulings, places etc.
General extraction	Recognition and tagging of specific word classes	Understanding user activities from social media
Visualization	Visualizing text data	Networks of politicians
Summarization	Automated summarization of long texts	Laws, news media content, diaries
Translation	Translating between languages	Understanding social media and news content

Text as data: Methods



Grimmer and Stewart (2013)

Text as data: Framework

- Computers can calculate numbers (estimates) from large documents
- Humans must evaluate if the estimates are useful

Document		Numeric		Outcome
	$\longrightarrow$	C_1	$\longrightarrow$	$\hat{V}_1, V_1$
	$\longrightarrow$	C_2	$\longrightarrow$	$\hat{V}_2, V_2$
$\vdots$		$\vdots$		$\vdots$
	$\longrightarrow$	C_n	$\longrightarrow$	$\hat{V}_n, V_n$

Text as data: Terminology

- **Token**
  - A meaningful unit of text
- **Document**
  - A purposefully organized text written in a given genre and style and serving a specific audience
- **Corpus**
  - A collection of documents that are typically similar in purpose, genre and style and share similar statistical properties

Text as data: Typical steps

- **Step 1**: Initial Processing: Get raw text, remove unnecessary content. Split up sentences in words, remove unnecessary words (*Q: examples?*)
- **Step 2**: Converting text to a (sparse) matrix: Define rows, columns and cell content
- **Step 3**: Analysis
- Sometimes **linguistic features** are added after Step 1
  - Part-of-speech tags – identify grammatical structure

Step 1: Cleaning and pre-processing text (1)

- Tokenization possibilities
  - Text: "`X and Y are 2 Kremlin trolls! Trolling day and night for a few rubles.`"
- Sentences: [`"X and Y are 2 Kremlin trolls !", "Trolling day and night for a few rubles."`]
- Words / Unigrams [`"X", "and", "Y", "are", "2", "Kremlin", "trolls", "!", "Trolling", ..., "rubles", "."`]
- Letters
- N-grams (Unigrams, Bigrams, Trigrams, …) (*Q: examples?*)

Step 1: Cleaning and pre-processing text (2)

- Remove punctuation
- Stopwords
  - Remove words that transport little semantic meaning: prepositions, articles, common nouns, etc
  - `"and", "are", "also", ....`
  - ![`"and","a","for","few"`]
  - However, sometimes we do need them for our analysis!
  - *Q: Do you think it's a good idea to remove stopwords?*

Step 1: Cleaning and pre-processing text (3)

- Remove capitalization
- Stemming
  - Reducing inflected words to their word stem
  - Cutting off common suffixes
    - `trolling` → `troll`
    - `trolls` → `troll`
    - `systems` / `systematic` / `systemic` → `system`

Step 2: Turning Text into a Matrix (1)

- **Rows**: Documents
- **Columns**: Pre-processing results
- **Cells**
  - Binary 0-1 columns: Token occurs/ does not occur in a document
  - Term frequency (TF): Counts of how many times a token occurs in a document
    - $\text{TF}(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$
- **Weighting** often included as many words will occur very often with little added information
  - How often does a word occur in one document compared to the overall collection of docs?

Step 2: Turning Text into a Matrix (2)

- **Term frequency - inverse document frequency** (TF-IDF)
  - **statistical measure** used to evaluate importance of a word (or term) within document relative to collection of documents (= corpus)
  - Value increases proportionally to number of times **word/term** appears in a document offset by number of documents in corpus that contain **word/term**

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) * \text{IDF}(t)$$

- where $\text{TF}(t, d)$ is the frequency of term t in document d

- $\text{IDF}(t)$ measures how unique or rare a term t is across a collection/corpus of documents
  - $\text{IDF}(t) = log\left(\frac{N}{df(t)}\right)$ where N is the total number of documents in the corpus and $df(t)$ is the number of documents in which the term t appears
  - log usually taken to reduce effect of extremely common words

Step 2: Turning Text into a Matrix (3): Document example

- Document: ***Time flies like an arrow. Fruit flies like a banana.***

- Same document after cleaning and processing:

	arrow	**banana**	**fli**	**fruit**	**like**	**time**
$C_i = 1$	1	1	2	1	2	1

- *Q: What is the problem with this representation?*

- Pre-processing aims to simplify the document without losing important information, but..
  - …meaning of words is ignored (e.g. “like”)
  - …word order is ignored (so-called “bag-of-words” representation)
  - → **Optimal approach depends on research question and planned analysis**

Step 2: Turning Text into a Matrix (4): Document-Feature Matrix

- Preprocessing converts a *corpus* (= a set of documents) into a *Document-Feature Matrix*

$$C = \begin{pmatrix} C_1 \\ \vdots \\ C_i \\ \vdots \\ C_n \end{pmatrix} =$$

arrow	banana	fli	fruit	like	time	…
0	0	0	0	0	0	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	2	1	2	1	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	1	1	…

- Matrix is **sparse** (= many zeros) and **high-dimensional** (= several thousand columns)

→ **Statistical learning useful**

Step 3: Analysis

- Sentiment Analysis
- Topic Modeling
- Text Classification
- Word Embeddings

Additional pre-processing steps (Step 2): Linguistic Analysis

- Meaning of text/terms can be enhanced in various ways

- **Part-of-speech (POS) tagging** (cf. Chiche and Yitagesu 2022)
  - process of identifying and labeling the grammatical categories (or "parts of speech") of words in a sentence
  - Allows better understanding of text: verb? noun? prep? adj? adv?
  - Position matters
    - e.g., "Plants/N need light and water."
    - e.g., "Everyone plants/V a tree."

- **Dependency parsing** (cf. Bunt, Merlo, and Nivre 2010)
  - type of syntactic analysis in natural language processing (NLP) that focuses on identifying the grammatical relationships between words in a sentence

Additional pre-processing steps (Step 2): Information Extraction

- **Named-entity recognition** (cf. Goyal, Gupta, and Kumar 2018)
  - Tag organizations, persons, or places within text
  - Different providers can identify different entities
  - see Spacy.io, Google NLP, GATE, …
- **Relation extraction** (cf. Cui et al. 2017)
  - detect and classify predefined relationships between entities identified in text
  - “Barack and Michelle Obama are married.”
  - “Madrid is the capital of Spain.”
- **Event extraction** (cf. Xiang and Wang 2019)
  - e.g., create a dataset of people killed by the police
  - verb phrases may provide a reasonable start to identify events

Resources (1): Resources for R (Welbers, Van Atteveldt, and Benoit 2017)

	R packages	
Operation	example	alternatives
Data preparation		
importing text	*readtext*	*jsonlite, XML, antiword, readxl, pdftools*
string operations	*stringi*	*stringr*
preprocessing	*quanteda*	*stringi, tokenizers, snowballC, tm, etc.*
document-term matrix (DTM)	*quanteda*	*tm, tidytext, Matrix*
filtering and weighting	*quanteda*	*tm, tidytext, Matrix*
Analysis		
dictionary	*quanteda*	*tm, tidytext, koRpus, corpustools*
supervised machine learning	*quanteda*	*RTextTools, kerasR, austin*
unsupervised machine learning	*topicmodels*	*quanteda, stm, austin, text2vec*
text statistics	*quanteda*	*koRpus, corpustools, textreuse*
Advanced topics		
advanced NLP	*spacyr*	*coreNLP, cleanNLP, koRpus*
word positions and syntax	*corpustools*	*quanteda, tidytext, koRpus*

Source: Welbers et al. (2017)

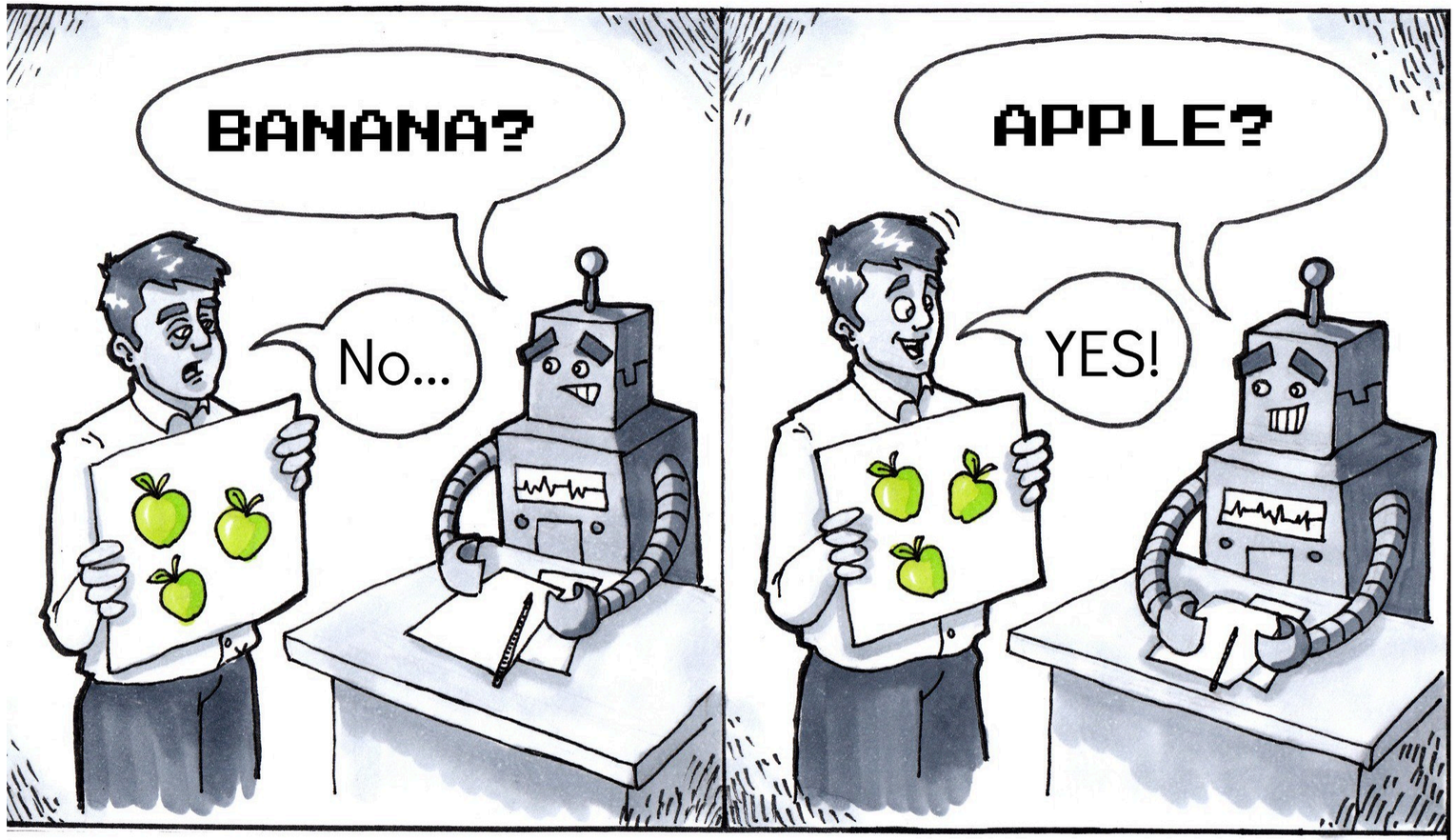
See also: CRAN Natural Language Processing Task View & recipes R package (step functions)

Resources (2): Text mining with tidy data via `tidytext`

- Consistent with tidy data principles
- Conversion of text to and from tidy formats
- Connects text mining and tidyverse packages

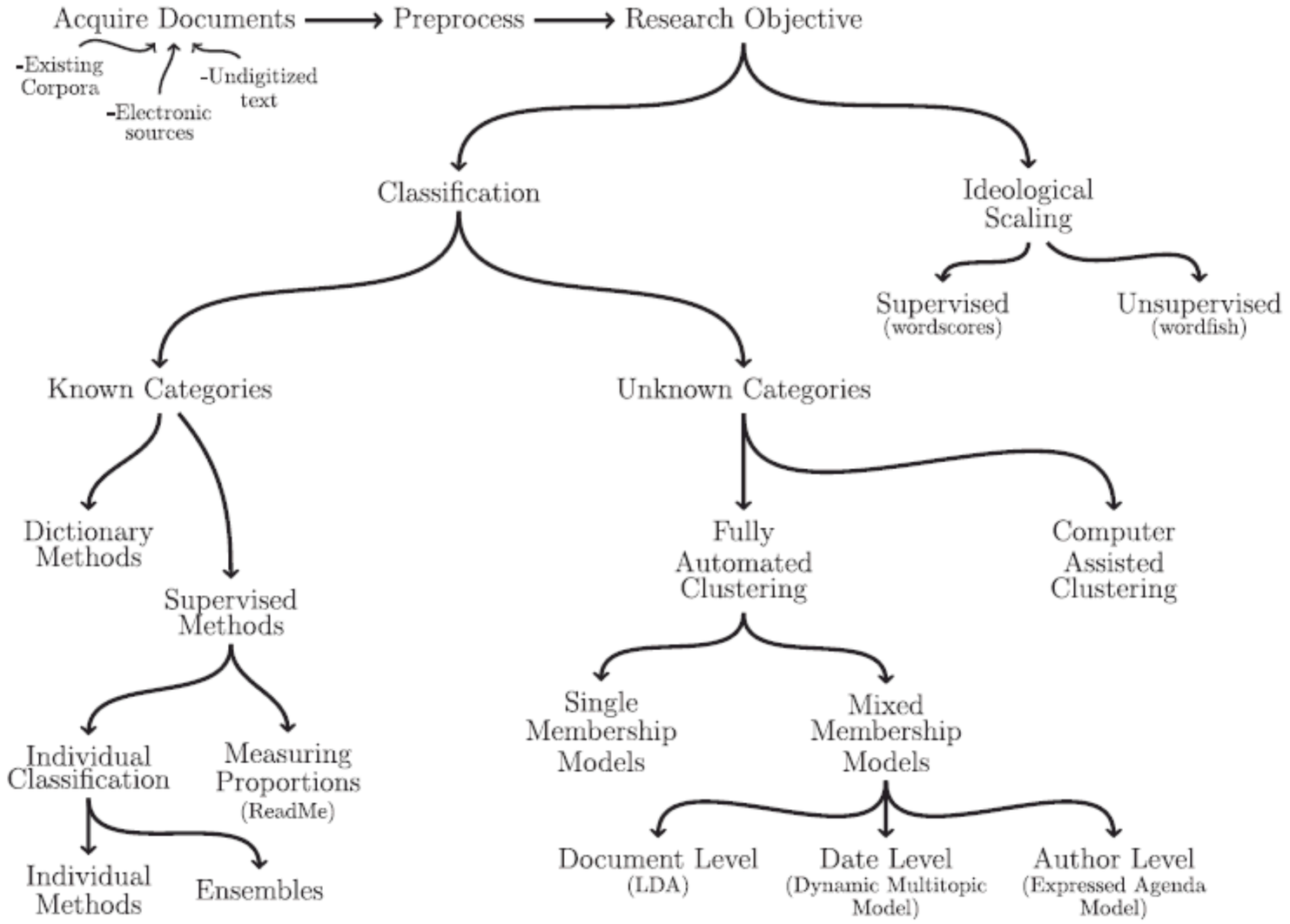
https://github.com/juliasilge/tidytext

R Session



2. Sentiment Analysis

Text as data



Grimmer and Stewart (2013)

Sentiment Analysis (1)

- Humans use understanding of emotional intent of words to infer whether a section of text is positive or negative
  - `"The laptop really sucks and broke after 4 months"`
- Sentiment analysis allows us to automate this task

Sentiment Analysis (2)

- **Untargeted sentiment**
  - Assign sentiment at sentence or document level
  - Lexicon only:
    - Human labeled word lists with sentiment strength, polarity, emotion…
  - Lexicon + heuristics:
    - Add pre-processing to understand “context”; negations, symbols etc.
  - Lexicon + model:
    - Add model to optimize term weights in aggregating word scores to overall sentiment
  - Classification (of sentiment)
    - Train model (e.g., random forest) using your labelled data (or finetune an LLM.)
- **Targeted sentiment**
  - Identify sentiment toward opinion target/entity
  - (Deep) learning models:
    - Identify entity and classify target-dependent sentiment
    - Identify sentiment toward unnamed target (stance)
  - Re-train for new target or domain?

Sentiment Analysis (3)

Method	Supervised/ Unsupervised	Type	Output	Reference	Implementation
VADER	unsupervised	UTS	compound score between [-1, 1]	Hutto and Gilbert	https://github.com/cjhutto/vaderSentiment
MPQA	unsupervised	UTS	Ratio of positive and negative score	Hu and Liu	https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
LabMT	unsupervised	UTS	Ratio of positive and negative score	Dodds et al.	https://hedonometer.org/words/labMT-en-v1/
Sentistrength (STS)	supervised	UTS	[-5,5]	Thelwall	http://sentistrength.wlv.ac.uk/
SentiTreeBank (STB)	supervised	UTS	[very positive positive, neutral, negative, very negative]	Socher et al.	https://nlp.stanford.edu/sentiment/treebank.html
TD-LSTM	supervised	TS	[negative, none, positive]	Tang et al.	https://github.com/jimmyyfeng/TD-LSTM
SVM-SD	supervised	ST	[favor, none, against]	Mohammad et al.	
DSSD	supervised	ST	[favor, none, against]	Augenstein et al.	https://github.com/sheffieldnlp/stance-conditional
Custom (LR)	supervised	ST	[favor, none, against]		
Custom (SVM)	supervised	ST	[favor, none, against]		
Custom (MNB)	supervised	ST	[favor, none, against]		
Custom (BERT)	supervised	ST	[favor, none, against]		

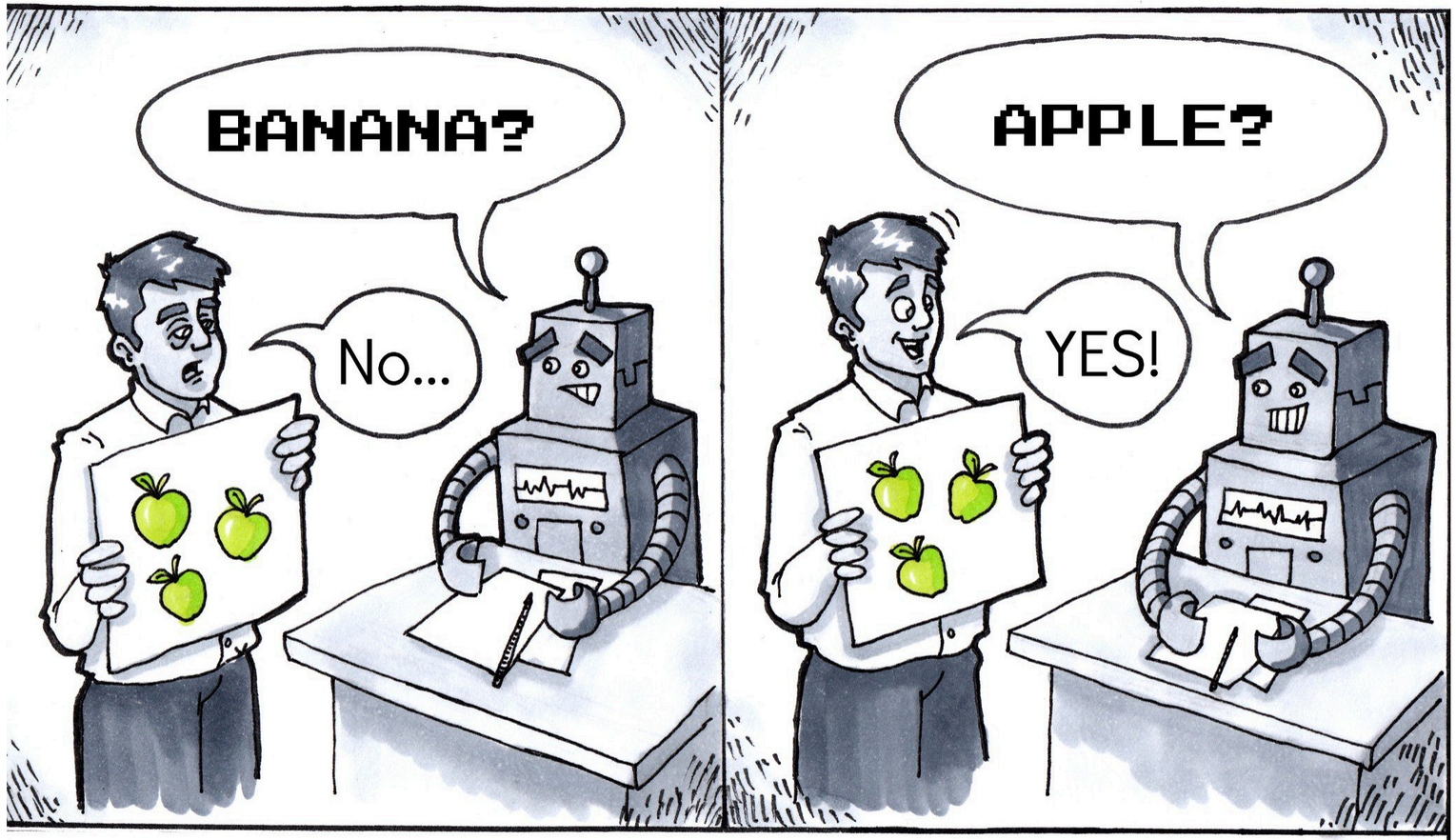
Comparison of methods used to measure approval (Sen, Flöck, and Wagner 2020)

- But LLMs also revolutionize sentiment analysis…
  - Q: What speaks against using LLMs?

Sentiment Analysis (4): in R

- **tidytext** – comes with AFINN, Bing and NRC lexicons/EmoLex. See https://tidytextmining.com for application
- **syuzhet** – comes with AFINN, Bing, NRC lexicons (aka EmoLex): https://cran.r-project.org/web/packages/syuzhet/index.html
- **VADER** – especially helpful for social media texts: https://cran.r-project.org/web/packages/vader/index.html
- **sentimentR** – takes into account valence shifters (i.e., negators, amplifiers, de-amplifiers) while maintaining speed: https://github.com/trinker/sentimentr
- **SentimentAnalysis** – comes with additional lexicons (e.g. finance-specific lexicons): https://www.rdocumentation.org/packages/SentimentAnalysis
- **Sentiword** – very large lexicon: https://github.com/aesuli/SentiWordNet
- **SentiWS** – German language sentiment dictionary: https://wortschatz.uni-leipzig.de/de/download#sentiWSDownload

R Session



Supervised Learning

3. Topic Modeling

Topic modeling intuition

- Imagine a library with millions of scientific articles on different topics like “climate change,” “artificial intelligence,” and “genetics.”
- Topic modeling is like having a librarian who reads all these articles and automatically organizes them into groups based on their main themes.

- The librarian **doesn’t know the topics beforehand**. They analyze the words in each article (like “carbon emissions,” “neural networks,” or “DNA sequencing”) to figure out the **hidden topics**.
- This process is like grouping books together based on shared keywords. Articles with similar words are likely to be about the same topic.

- The result is a **set of topics**, each **represented by a list of relevant words**.

Topic modeling: LDA (1)

- **Latent Dirichlet Allocation (LDA)** goes back to Blei, Ng, and Jordan (2003; see also Blei 2012) (simplest topic model)
- **Finds hidden topics**: LDA automatically discovers underlying themes in text data
- **Mix-and-match**: Sees documents as combinations of different topics
- **Uses probabilities**: Calculates the likelihood of words, topics, and documents being related
- **Widely applied**: Used for tasks like analyzing customer reviews, classifying research papers, and recommending news articles
- LDA: “the imaginary random process by which the model assumes the documents arose” (Blei 2012:78)
- LDA = classic model but relevance challenged by LLMs

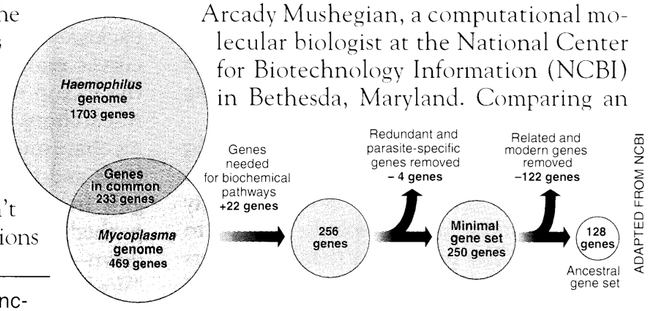
Topic modeling: LDA (2)

- **Simple intuition**: Documents exhibit multiple topics.

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



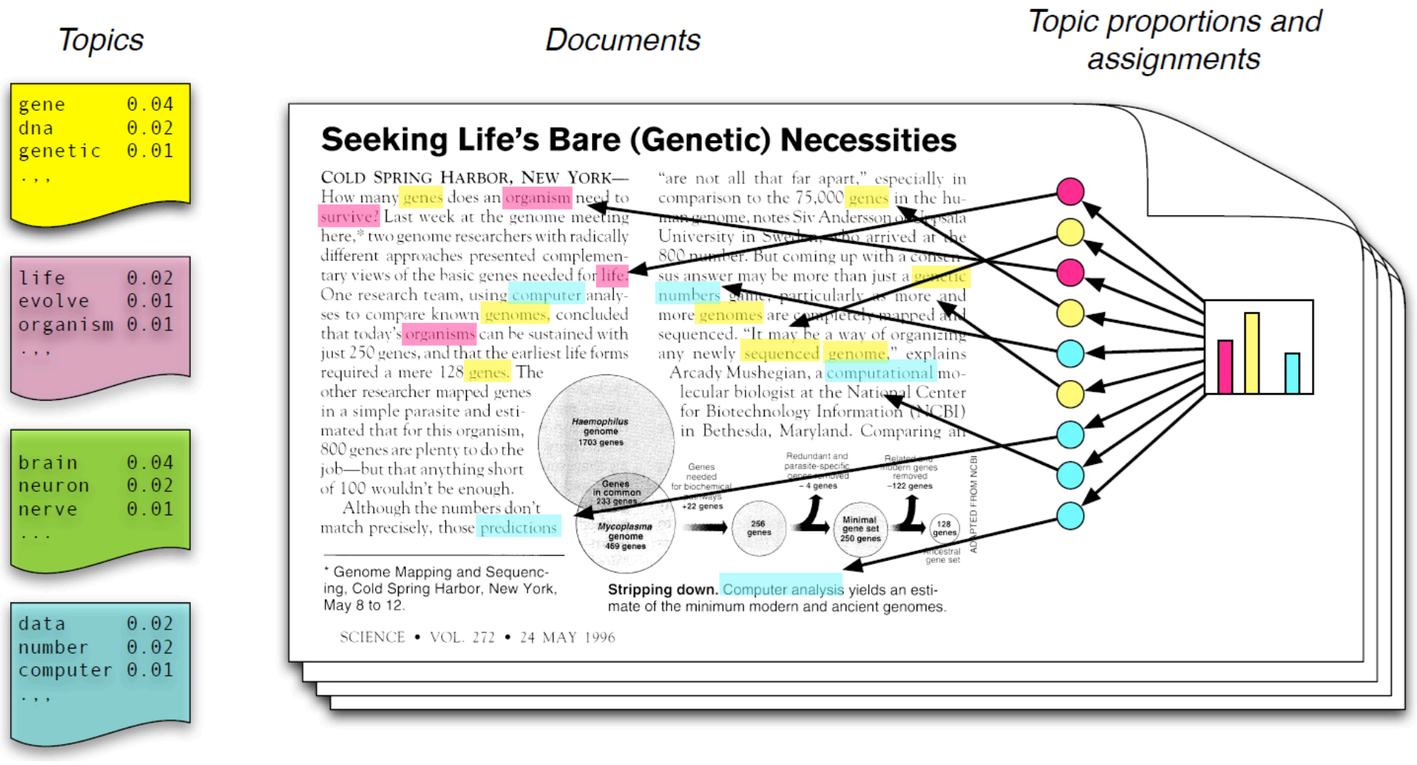
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

"This article, entitled"Seeking Life's Bare (Genetic) Necessities," is about using data analysis to determine the number of genes an organism needs to survive (in an evolutionary sense)." (Blei 2012:78)

Topic modeling: LDA (3)

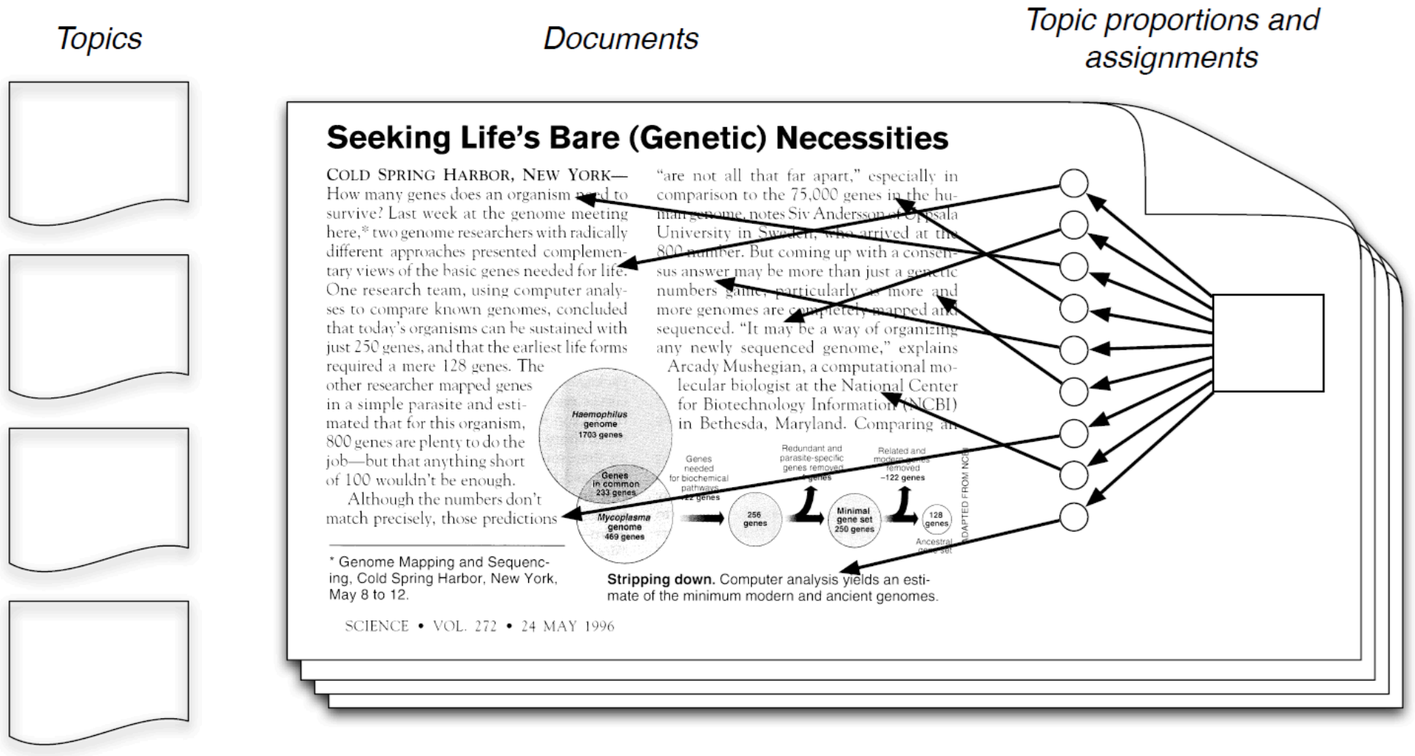


Blei (2012)

- Each **topic** is a distribution over words (e.g., *genetics topic*)
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Topic modeling: LDA (4)

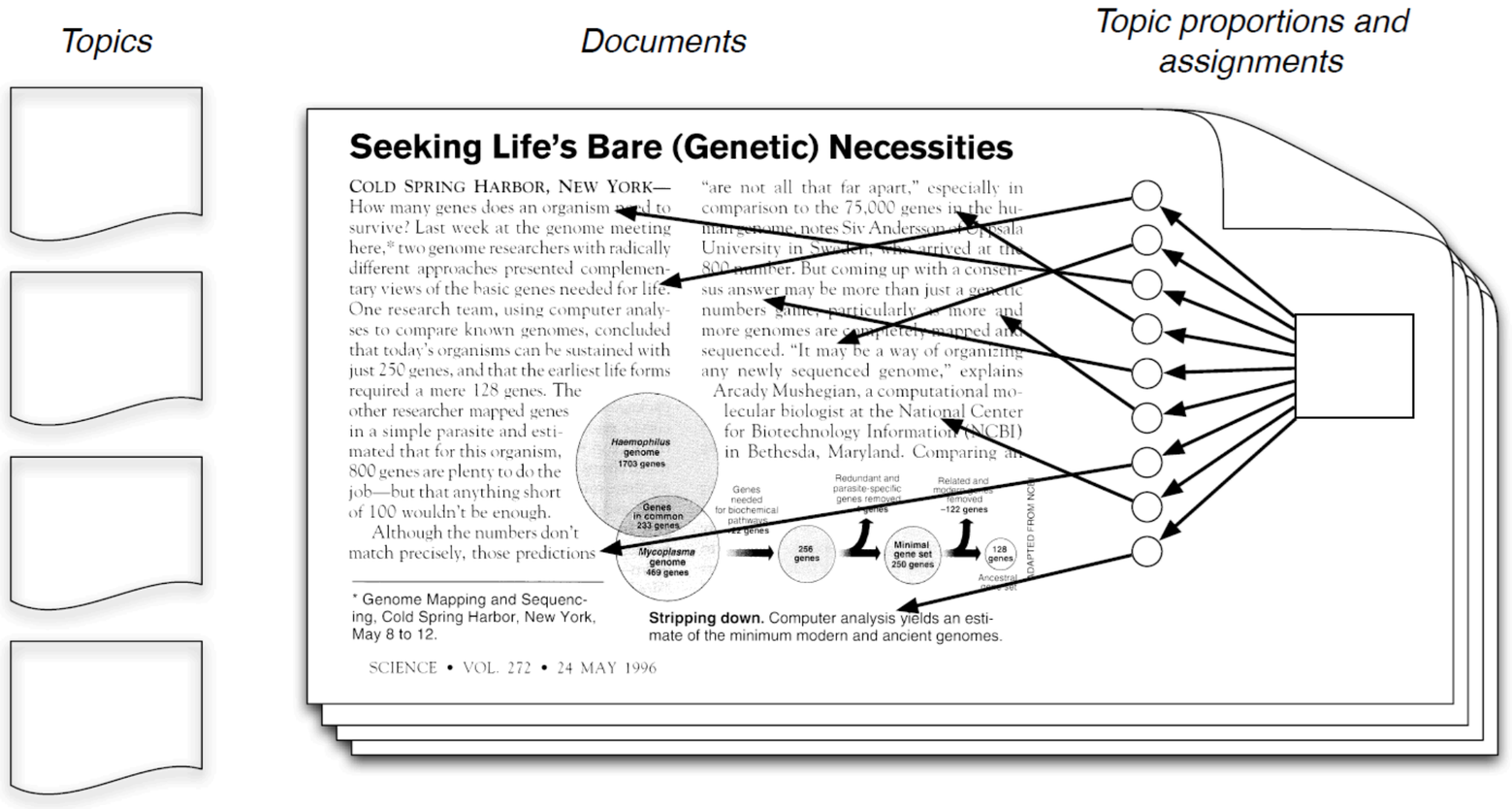
- In reality, we only observe the documents
- The topic structure (*topics*, *per-document topic distributions/proportions*, and the *per-document per-word topic assignments*) are **hidden variables**



Blei (2012)

Topic modeling: LDA (5)

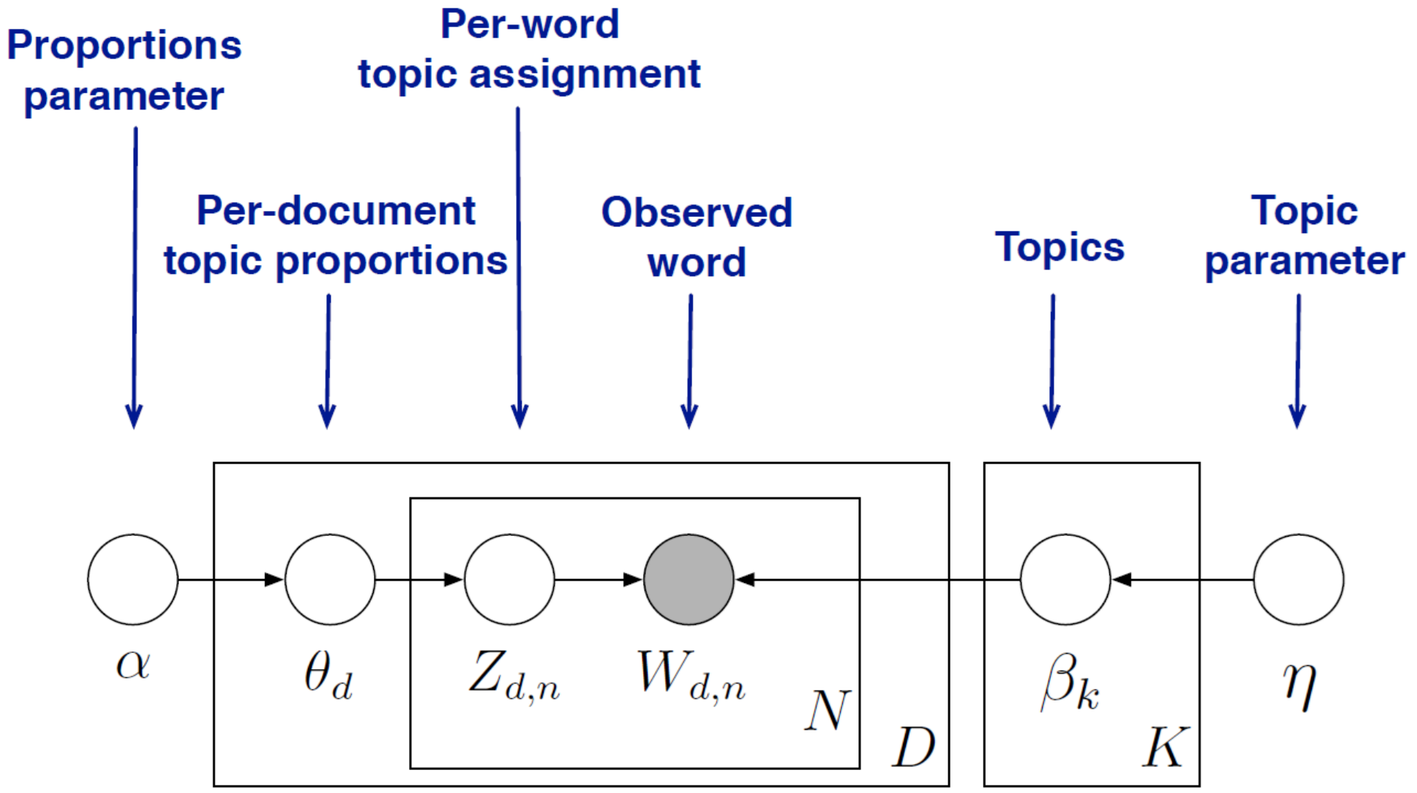
- Goal is to infer hidden topic structure/variables (compute distributions)
  - Calculate: $p(topics, proportions, assignments|document)$



Blei (2012)

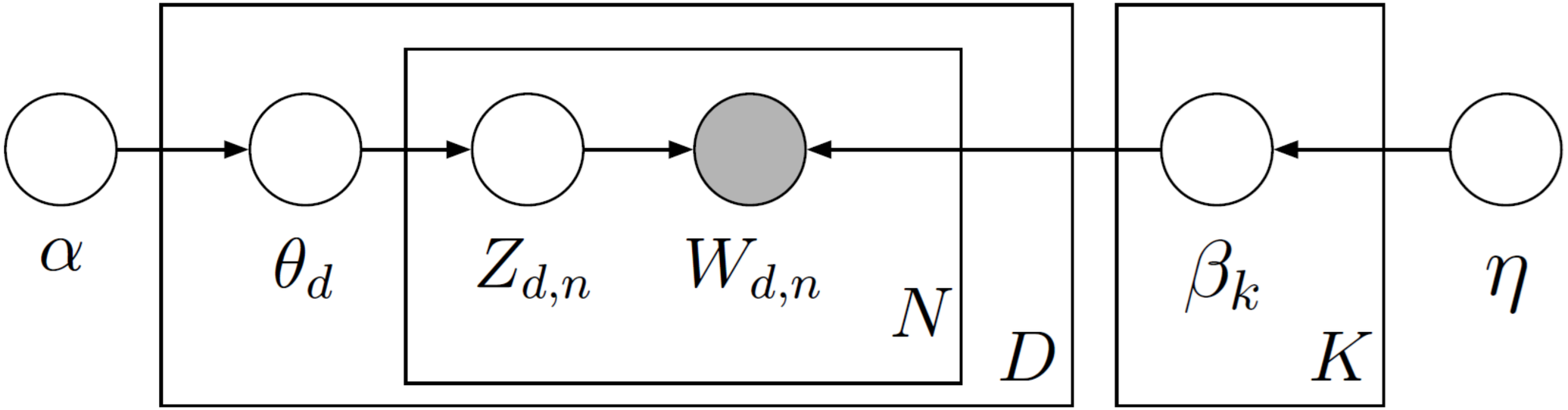
Topic modeling: LDA (6)

- Graphical model (Blei 2012:4): Each node is a random variable (observed = gray, hidden = transparent)
  - Data arises from generative process that includes *hidden variables* (process defines a *joint probability distribution*)
  - **Data analysis**: use joint distribution to compute *conditional distribution* (also called *posterior distribution*) of hidden variables given the observed variables



Blei (2012), Graphical model

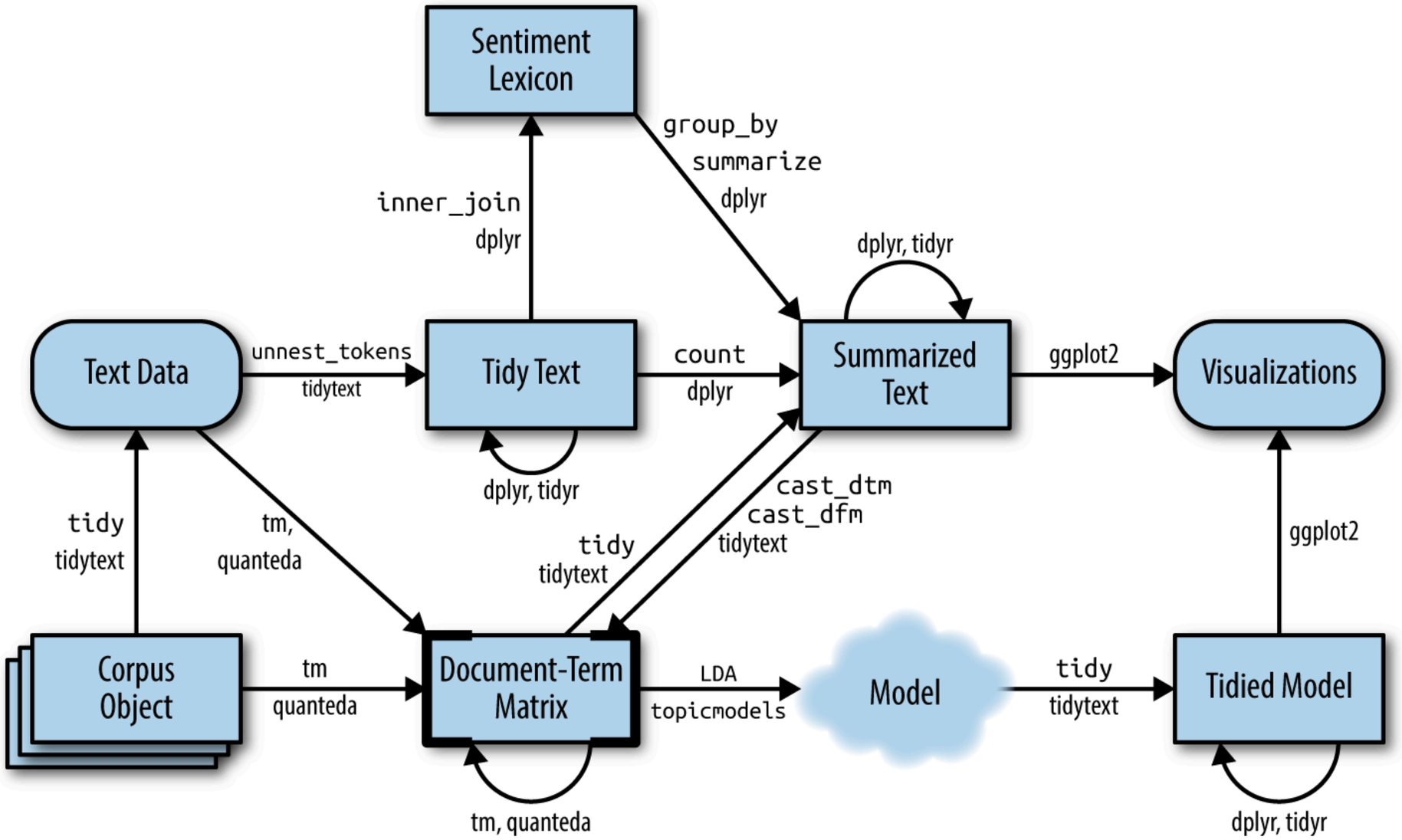
Topic modeling: LDA (7)



Blei (2012)

- Topic model aims to infer $P(\beta, \theta, \mathbf{z}|\mathbf{w})$:
  - assignments $z_{d,n}$ of topics to all words in each document
  - distribution of topics θ_d in each document
  - distribution of terms β_k in each of the K topics
- Goals of inference process:
  - Assign terms to only a few possible topics per document
  - Assign high probability to few terms in one topic
- Parameters influencing model outcomes:
  - K – number of topics to be inferred
    - Re-fit with different K's and interpret
  - α – prior for topic–document distributions
  - η – prior for term–topic distributions
- Estimation: compute *conditional distribution* of the topic structure given observed documents (see Blei (2012) for discussion of algorithms)

Topic modeling in R



Silge (2017)

Sentiment analysis and topic modeling: ChatGPT (OpenAI)

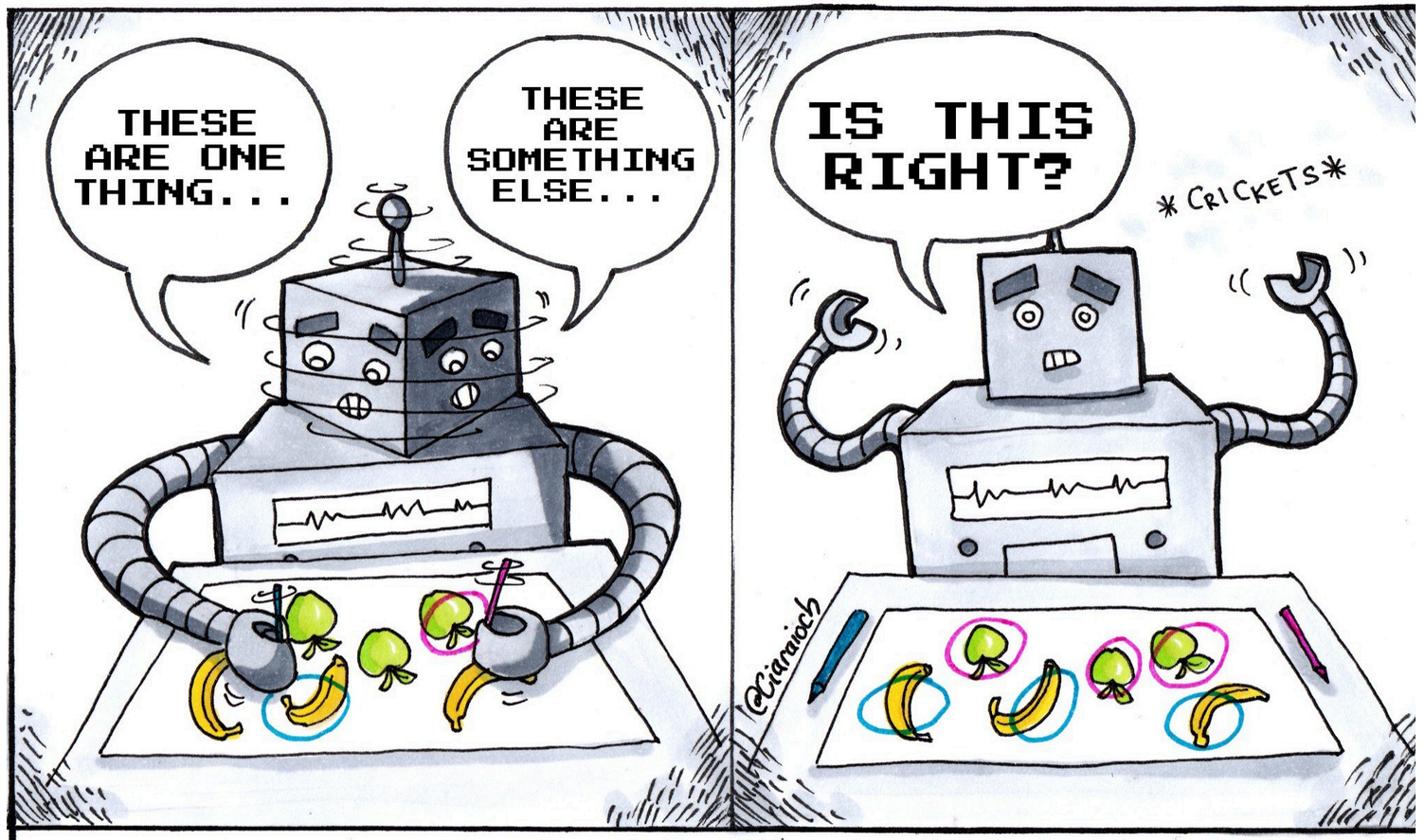
- Nowadays, LLMs are taking over..

```
install.packages("chatgpt")
# Sys.setenv(OPENAI_API_KEY = "XX-XXXXXXXXXXXXXXXXXXXXXX")
library(chatgpt)
cat(ask_chatgpt("Analyze the following text, focusing on both sentiment and key topics:

The recent election was a disaster. While the opposition focused on fear-mongering and divisive rhetoric, the incumbe
barely addressed the real issues affecting ordinary people. Their campaign promises were empty, and their focus on ec
growth ignored the widening inequality in our society. This result leaves me deeply concerned about the future of our

Provide a breakdown of:

Overall sentiment: Is the sentiment expressed primarily positive, negative, or neutral? Explain your reasoning.
Specific targets: Are there any particular individuals, groups, or entities towards which the sentiment is directed?
Key topics: What are the main topics or themes discussed in the text?

Please provide short answers."))

# Overall sentiment: The sentiment expressed is negative, with feelings of disappointment and concern.
# Specific targets: The sentiment is directed towards the opposition and the incumbent party for their actions during
```

Sentiment analysis and topic modeling: Gemini (Google)

```
install.packages("gemini.R")
library(gemini.R)
# setAPI("#########") # Set API key

gemini("Analyze the following text, focusing on both sentiment and key topics:

The recent election was a disaster. While the opposition focused on fear-mongering and divisive rhetoric, the incumbe
barely addressed the real issues affecting ordinary people. Their campaign promises were empty, and their focus on ec
growth ignored the widening inequality in our society. This result leaves me deeply concerned about the future of our

Provide a breakdown of:

Overall sentiment: Is the sentiment expressed primarily positive, negative, or neutral? Explain your reasoning.
Specific targets: Are there any particular individuals, groups, or entities towards which the sentiment is directed?
Key topics: What are the main topics or themes discussed in the text?

Please provide short answers.")

# "**Overall sentiment:** Strongly negative. The author uses words like \"disaster\" \"fear-mongering\" \"empty\"
```

R Session



Unsupervised Learning

References

Blei, David M. 2012. “Probabilistic Topic Models.” *Communications of the ACM* 55(4):77–84.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. “Latent Dirichlet Allocation.” *Journal of Machine Learning Research* 3(Jan):993–1022.

Bunt, Harry, Paola Merlo, and Joakim Nivre. 2010. *Trends in Parsing Technology: Dependency Parsing, Domain Adaptation, and Deep Parsing*. Vol. 43. Springer Science & Business Media.

Chiche, Alebachew, and Betselot Yitagesu. 2022. “Part of Speech Tagging: A Systematic Review of Deep Learning and Machine Learning Approaches.” *Journal of Big Data* 9(1):10.

Cui, Meiji, Li Li, Zhihong Wang, and Mingyu You. 2017. “A Survey on Relation Extraction.” Pp. 50–58 in *Knowledge graph and semantic computing. Language, knowledge, and intelligence: Second china conference, CCKS 2017, chengdu, china, august 26–29, 2017, revised selected papers 2*. Springer.

Goyal, Archana, Vishal Gupta, and Manish Kumar. 2018. “Recent Named Entity Recognition and Classification Techniques: A Systematic Review.” *Computer Science Review* 29:21–43.

Grimmer, Justin, and Brandon M. Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Polit. Anal.* 21(3):267–97.

Sen, Indira, Fabian Flöck, and Claudia Wagner. 2020. “On the Reliability and Validity of Detecting Approval of Political Actors in Tweets.” Pp. 1413–26 in *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*.

Silge, Julia. 2017. *Text Mining with r : A Tidy Approach*. First edition. Beijing, China.

Welbers, Kasper, Wouter Van Atteveldt, and Kenneth Benoit. 2017. “Text Analysis in r.” *Communication Methods and Measures* 11(4):245–65.

Xiang, Wei, and Bang Wang. 2019. “A Survey of Event Extraction from Text.” *IEEE Access* 7:173111–37.