UNIVERSITÄT LEIPZIG

Forschungsseminar CSS – Supervised & Unsupervised ML

SR 423, 19.11.2024

Felix Lennert, M.Sc.

OUTLINE

- Intro
- Supervised ML
  - Motivation
  - “Text Regression”
  - The Procedure
- Unsupervised ML; Topic Modeling
  - Value for Social Sciences
  - LDA
  - In Practice
  - Evaluation Strategy

BEFORE WE START

- Today is going to be about the logic behind and the steps you researchers have to do when using supervised & unsupervised text classification
- Caveat:
  - models based on the bag of words-assumption (which we will use today) are becoming increasingly outdated
  - new models are there and incredible, but they remain black boxes we cannot open
  - yet they are fairly user-friendly, about 5 lines of code (and a lot of waiting time depending on your computer)
  - and: the training and evaluation process is basically the same

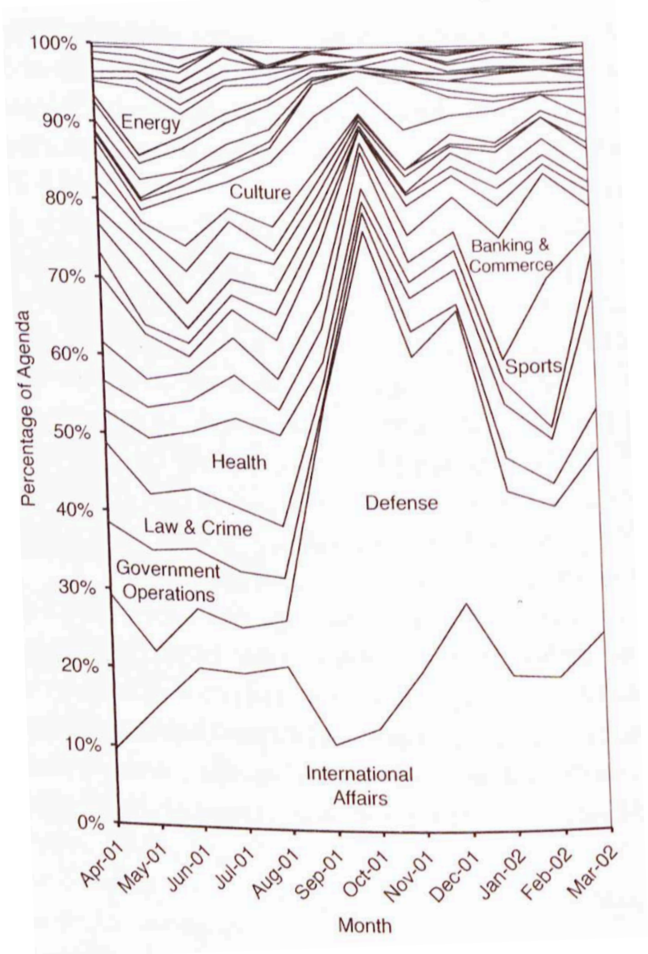
RECAP: MEASUREMENT USING TEXT DATA

- Text mining is often about “producing data” – a (numerical) **summary** of the documents in question
- With the methods we’re using today, these produced data can look like…
  - A discrete label from binary classification (e.g., “positive/negative”, being about a certain topic, “sexist/non-sexist”)
  - A discrete label from multinomial classification (e.g., multiple topics, authors)
  - A continuous value (sentiment, probability of having a certain label, ideological scaling)

$\Longrightarrow$ We can then eventually use these values/label counts to test hypotheses

RECAP: MEASUREMENT USING TEXT DATA

- Example: labels counted over time
- International politics frames that made it to NYT headlines in 2001 (Boydstun 2013; taken from Grimmer et al. 2022)



RECAP: MEASUREMENT USING TEXT DATA

- Example: using classification accuracy as **continuous indicator** for speech polarization

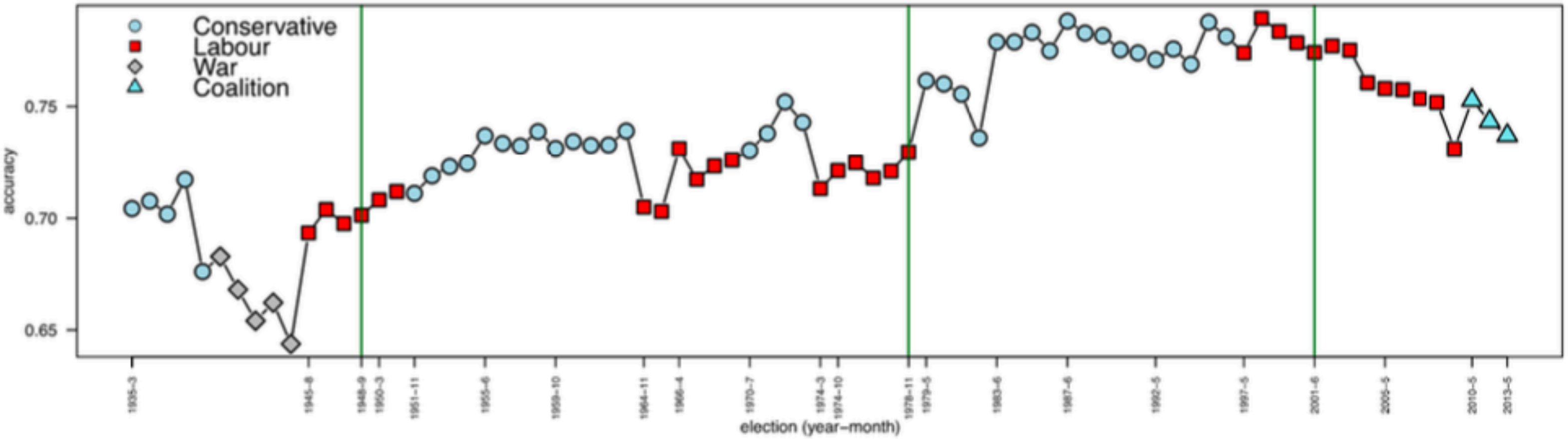


Figure 3. Estimates of parliamentary polarization, by session. Election dates mark *x*-axis. Estimated change points are [green] vertical lines.

HOW TO PRODUCE THESE DATA?

Most basic approach: read the text

1. Develop a coding scheme (based on prior theory)
2. Read text, decide on annotation based on coding scheme
3. Do it for all your documents
4. …
5. …there is plenty of text available now, so it takes forever…
6. **Consider different career paths over and over again as this process sucks so bad**

⟹ Luckily, there are computational tools we can harness to take away some of the pain

⟹ **MACHINE LEARNING**

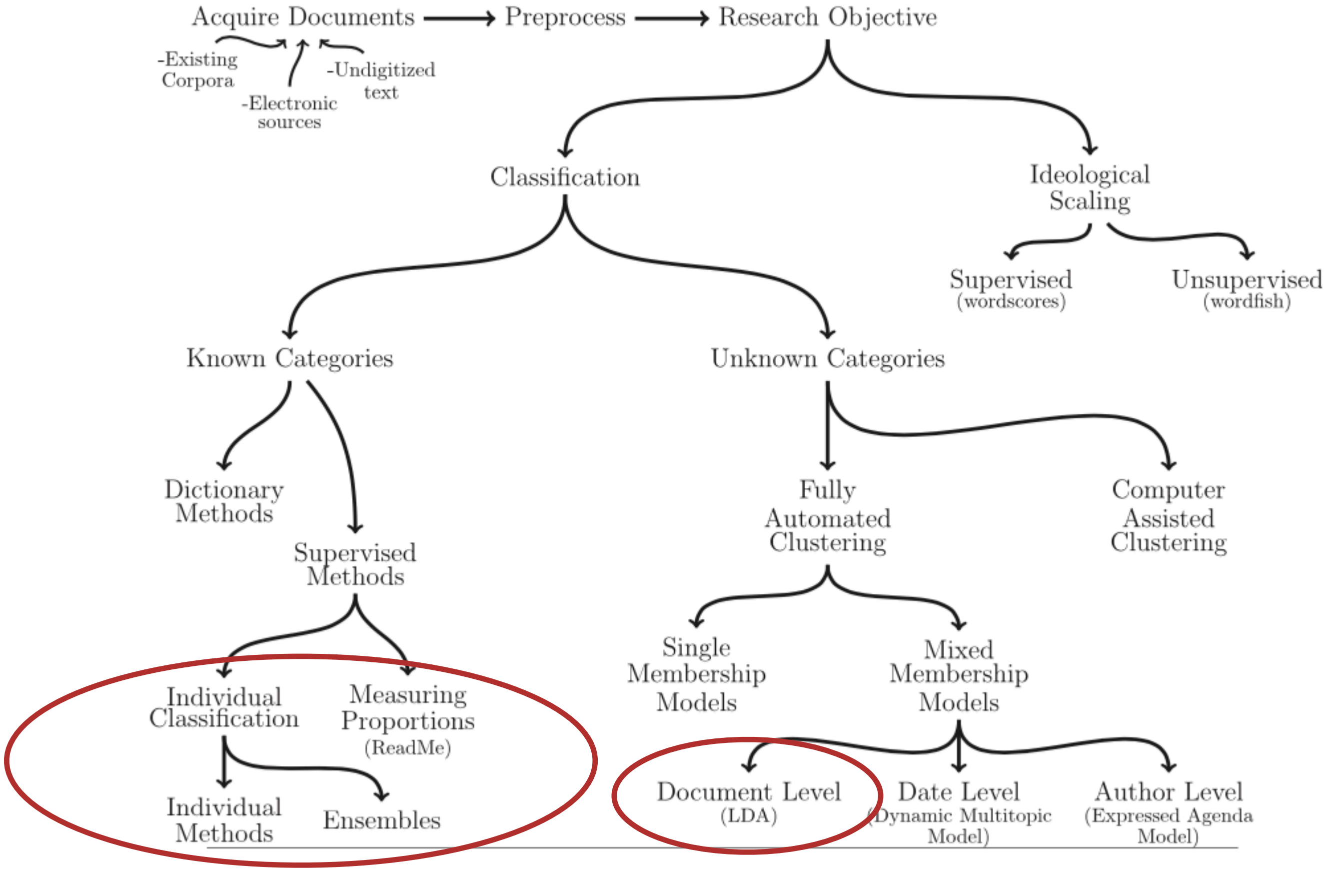
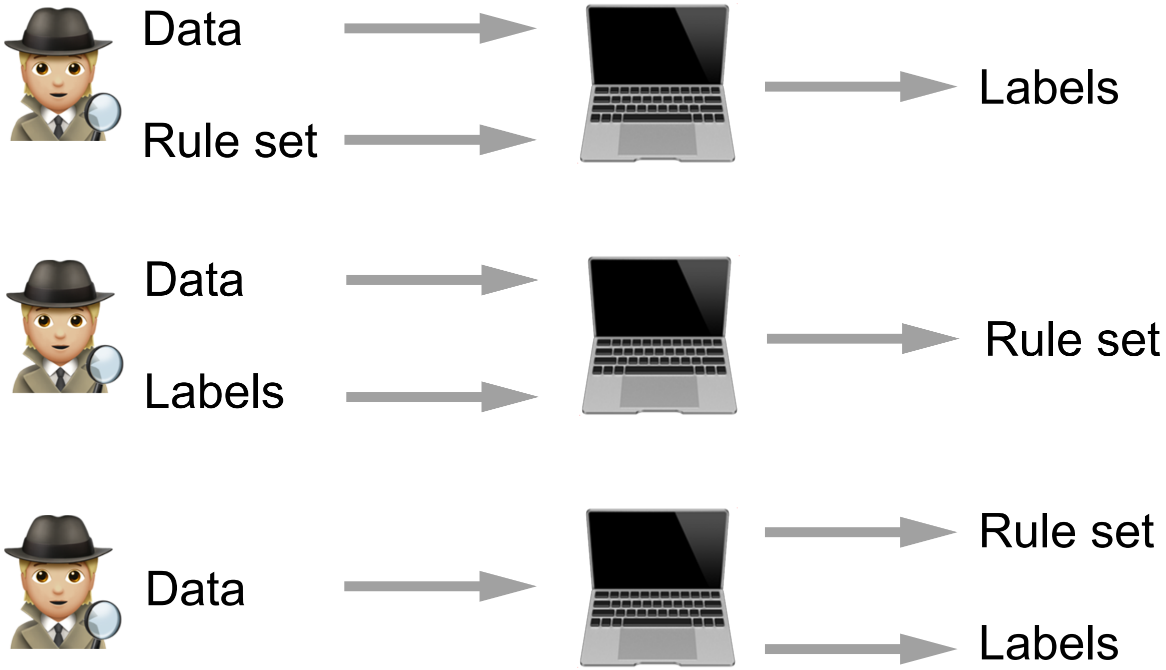


Fig. 1 An overview of text as data methods.

inspired by Ash (2018)

HOW TO PRODUCE THESE DATA?

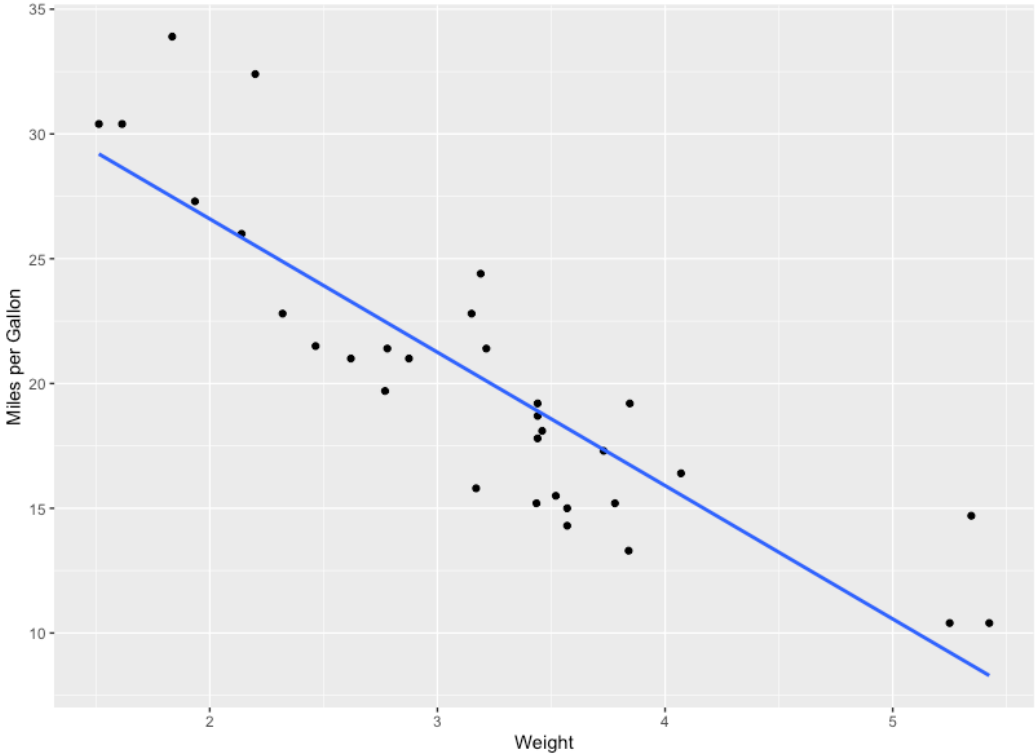


Dictionary-based analysis
Computer applies rules

Supervised ML
Computer learns relationship ("rules") between data and answers

Unsupervised ML
Computer suggests rules and answers based on patterns in data

EVEN OLS IS MACHINE LEARNING IF YOU WILL



$$MPG = \beta_0 + \beta_1 Weight + \epsilon$$

inspired by Ash (2018)

HOW DOES IT LOOK FOR TEXT – “TEXT REGRESSION”

Objective: to learn a model that maps an outcome Y to the features W'

$$Y_i = \beta W_i' + \epsilon_i$$

$\Rightarrow$ Requires labeled documents

$\Rightarrow$ Features (words) are treated as predictors

$\Rightarrow$ Algorithms will not accept words – we use word counts (alternatives: “one-hot encoding” (1 if word is present in document, 0 if not), tf-idf values, embedding vectors)

HOW DOES IT LOOK FOR TEXT – “TEXT REGRESSION”

Objective: to learn a model that maps an outcome Y to the features W'

$\Longrightarrow$ Eventually, predictions can be made on unseen documents

$\Longrightarrow$ Different approaches/algorithms exist – which one to choose depends on

computational capabilities and desired outcome (i.e., discrete label – binary or multinomial – or continuous value)

SUPERVISED LEARNING WITH TEXT – THE PROCESS

- Choose a set of documents (corpus)
- Annotate a sub-set of the corpus
- Split the annotated set into training and test set (for validity assessment)
- Preprocess the documents

  ⇒ e.g., tokenization (also: bi- and trigrams), weighting, stemming/lemmatization, etc. – whatever works best
- Train a classifier on training set

  ⇒ tuning with cross-validation
- Evaluate classifier using test set and confusion matrix
- If sufficient, apply it to unlabeled data

(for a hands-on guide, see Barberá et al. 2021)

CHOICE OF CORPUS

- Must fit the question
- Usual approach: keyword-based search (e.g., using regular expressions) $\Longrightarrow$ has its own pitfalls though, see Barberá et al. (2021) and King, Lam, and Roberts (2017)

CHARACTERISTICS OF A GOOD ANNOTATED SET (GRIMMER ET AL. 2022, P. 190)

- **Objective–intersubjective**: categories are *objectively* measured; researchers have a *shared understanding* of them
- **A priori**: codebook is derived from theory
- **Reliable**: annotation process is repeatable across coders – will yield same results
- **Valid**: concept of interest is clearly measured
- **Generalizable**: the training set is a representative sample of the underlying texts (and also the final population)
- **Replicable**: approaches should replicate with same and different data

ANNOTATION OF TRAINING AND TEST SET OF CORPUS

Step 1: Randomly sample documents from corpus

- Sample should be representative (e.g., if corpus spans a long time period, has different authors, etc.)
- Usually, algorithm can only derive rules for terms it has seen

ANNOTATION OF TRAINING AND TEST SET OF CORPUS

Step 2: Define your codebook

- Usually: rules depend on your theory
  - They need to be stated explicitly (in paper and/or appendix)
  - Ideally, you find examples from the data for each rule
    ⇨ To guide your reader

    ⇨ But also for yourself
- Sometimes, codebooks are already available (e.g., from related studies)

ANNOTATION OF TRAINING AND TEST SET OF CORPUS

Step 3: Get other coders/get ready to annotate multiple times

- Needed to assess the reliability of the coding process
  ⇒ Either between raters

  ⇒ If only one rater exists: multiple timepoints
- Also a test for the codebook
- Finally, agreement between coders needs to be assessed
- Ideally: make a test run with a set that will be later discarded to ensure that concepts are understood; discuss cases of disagreement
- More on this: Barberá et al. (2021)

ANNOTATION OF TRAINING AND TEST SET OF CORPUS

Step 4: Determine training and test set

- Training set: used to train the model
- Test set: used to evaluate performance
- Usual split: 80/20
- Important: classes should be equally represented in training and test set (can be mitigated using upsampling or downsampling)

PREPROCESSING

- No one-fits-all solution
- *recipes* and the *tune package* make it easy to experiment a bit
- Common steps:
  - Using bi- and trigrams
  - Weighting by TF or TF-IDF
  - Stemming/Lemmatization
  - Removal of rare/common words or stopwords (feature reduction)

TRAINING THE CLASSIFIER

Step 1: Choose a classifier

⟹ Depends on question and computational capabilities

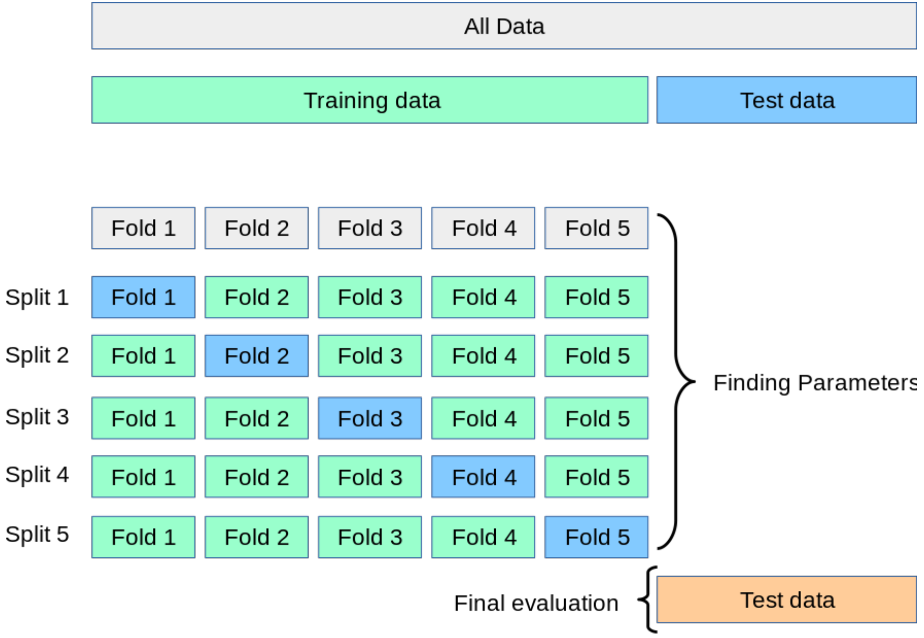
- Do you want to predict continuous or categorical value?
- Will you run the models on a server or your own laptop?

Step 2: Train classifier(s) using training set

- Use different specifications of training set
- Use different classifiers

Step 3: Cross-validate and tune different specifications to find optimal solution

CROSS-VALIDATION



https://scikit-learn.org/stable/modules/cross_validation.html

FINAL EVALUATION

How well does the classifier compare to gold standard data?

Example: Sentiment Analysis

PREDICTED VALUES \ ACTUAL VALUES	**POSITIVE**	**NEGATIVE**
POSITIVE	TRUE POSITIVE	FALSE POSITIVE
NEGATIVE	FALSE NEGATIVE	TRUE NEGATIVE

FINAL EVALUATION

How well does the classifier compare to gold standard data?

Accuracy: $\frac{TP + FN}{TP + FP + FP + FN}$ – how many predictions are correct (reasonable if labels are balanced!)

Precision: $\frac{TP}{TP + FP}$ – how many positive predictions are correct

Recall/Sensitivity: $\frac{TP}{TP + FN}$ – how many actual positives are predicted properly

F1-score: $2 \times \frac{Precision \times Recall}{Precision + Recall}$ – harmonic mean of precision and recall

TOPIC MODELING'S VALUE FOR SOCIAL SCIENTISTS (DIMAGGIO ET AL. 2013)

A good approach for distance-reading should fulfill four requirements

- *explicitness* – others should be able to replicate it
- *automation* – as data sets become larger
- *inductive* – shall not rely on researcher's priors too much
- *take into account context* – terms can mean different things in different contexts (*relationality* of meaning)

TOPIC MODELING'S VALUE FOR SOCIAL SCIENTISTS

Topic models

- *organize* documents into topics based on their content, i.e., the words they contain
- *organize* terms into topics based on their co-appearance

- documents are a mixture of topics
- topics are a mixture of words
- words can appear in multiple topics

TOPIC MODELING'S VALUE FOR SOCIAL SCIENTISTS (DIMAGGIO ET AL. 2013)

A good approach for distance-reading should fulfill four requirements

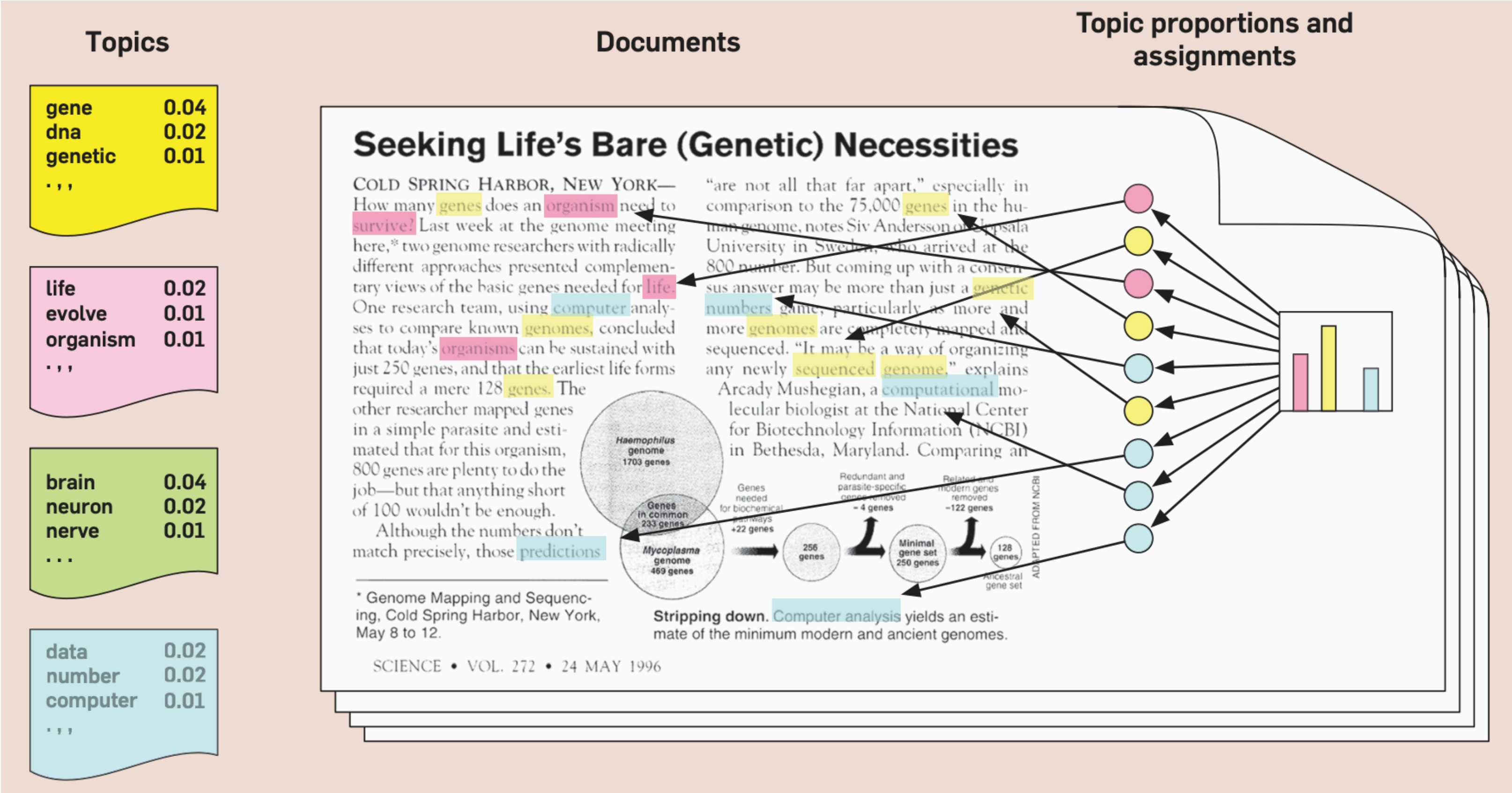
- *explicitness* – others should be able to replicate it
- *automation* – as data sets become larger
- *inductive* – shall not rely on researcher's priors too much
- *take into account context* – terms can mean different things in different contexts (*relationality* of meaning)

⇒ parameters are explicit

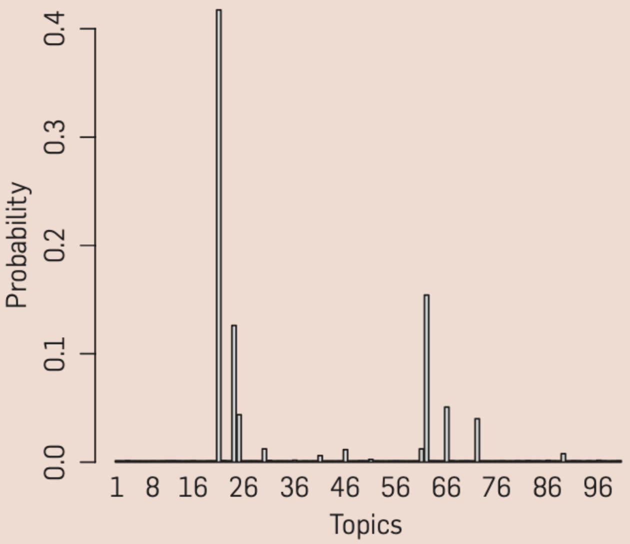
⇒ computer does the work

⇒ unsupervised

⇒ words can belong to different topic



Blei 2012, p. 78



"Genetics"	**"Evolution"**	**"Disease"**	**"Computers"**
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Blei 2012, p. 79

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

Topic models assume the following data generation process

- author decides on length of text
- author decides on topics
- author draws words from vocabulary of topics

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

Example: 5 sentences, 2 topics

- I like to eat broccoli and bananas.
- I ate a banana and spinach smoothie for breakfast.
- Chinchillas and kittens are cute.
- My sister adopted a kitten yesterday.
- Look at this cute hamster munching on a piece of broccoli.

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

Example: 5 sentences, 2 topics

- I like to eat broccoli and bananas. ⇒ 100% food
- I ate a banana and spinach smoothie for breakfast. ⇒ 100% food
- Hamsters and kittens are cute. ⇒ 100% adorable animals
- My sister adopted a kitten yesterday. ⇒ 100% adorable animals
- Look at this cute hamster munching on a piece of broccoli. ⇒ 50% adorable animals, 50% food

⇒ IDEA OF LDA: topics are mixture of words, documents mixture of topics (and of words)

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

Example: 5 sentences, 2 topics

- I like to **eat broccoli** and **bananas**. ⇒ 100% food
- I **ate** a **banana** and **spinach smoothie** for **breakfast**. ⇒ 100% food
- *Hamsters* and *kittens* are *cute*. ⇒ 100% adorable animals
- My sister *adopted* a *kitten* yesterday. ⇒ 100% adorable animals
- Look at this *cute hamster* **munching** on a piece of **broccoli**. ⇒ 50% adorable animals, 50% food

Problem: For the computer, all the words look the same

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

- assign a **topic t** at random to each **word w** in each **document d**

$\Rightarrow$ number of topics (*k*) is chosen **<u>before</u>**

- go through each **word w** in each **document d**
- assume that all the other assigned topics (to the words) are correct
- compute $p(t \mid d)$= the proportion of **words w** in **document d** that are currently assigned to **topic t**
- compute $p(w \mid t)$= the **proportion of w** being **assigned to t** (over all documents)
- new **topic distribution for w**: $p(t \mid d) \times p(w \mid t)$
- …repeat until a steady state is achieved

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

- assign a **topic t** at random to each **word w** in each **document d** assign a **topic t** at random to each **word w** in each **document d**

here: k=2

	broccoli	banana(s)	munching	hamster	kitten	spinach	smoothie	cute
S 1	1	2						
S 2		2				1	1	
S 3			2		1			2
S 4					2			
S 5	2			2				2
S …								

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

LDA takes as input the documents and the assumed number of topics

It aims to learn the proportion α of each topic t in a document

- Learning process:
  - go through each **word w** in each **document d**
  - assume that all the other assigned topics (to the words) are correct
  - compute **p(topic t | document d)** = the proportion of **words in document d** that are currently assigned to **topic t** ($\Rightarrow$ if a word appears in a document, it is likely to be of the same topic)
  - compute **p(word w | topic t)** = the **proportion of w** being **assigned to t** (over all documents)
  - new **topic distribution for w**: **p(topic t | document d) * p(word w | topic t)**
  - …repeat until a steady state is achieved

- go through each **word w** in each **document d**
- assume that all the other assigned topics (to the words) are correct
- compute $p(t \mid d)$= the proportion of **words w** in **document d** that are currently assigned to **topic t**

	broccoli	banana(s)	munching	hamster	kitten	spinach	smoothie	cute
S 1	**p(T=2|S 1)=1**	2						
S 2		2				1	1	
S 3			2		1			2
S 4					2			
S 5	2			2				2
S …								

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

- compute $p(w\,|\,t)$ = the **proportion of w** being **assigned to t** (over all documents)

$\Rightarrow p(w = broccoli\,|\,t = 1) = 0$

$\Rightarrow p(w = broccoli\,|\,t = 2) = 1$

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

- go through each **word w** in each **document d**
- assume that all the other assigned topics (to the words) are correct
- compute $p(t \mid d)$= the proportion of **words w** in **document d** that are currently assigned to **topic t**
- compute $p(w \mid t)$= the **proportion of w** being **assigned to t** (over all documents)
- new **topic distribution for w**: $p(t \mid d) \times p(w \mid t)$

$$\Rightarrow p(broccoli, t = 1) = 0 \times 0 = 0$$

$$\Rightarrow p(broccoli, t = 2) = p(t = 2 \mid d = s_1) \times p(w = broccoli \mid t = 2) = 1$$

STYLIZED APPROACH (TAKEN FROM CHEN 2013)

- go through each **word w** in each **document d**
- assume that all the other assigned topics (to the words) are correct
- compute $p(t \mid d)$= the proportion of **words w** in **document d** that are currently assigned to **topic t**
- compute $p(w \mid t)$= the **proportion of w** being **assigned to t** (over all documents)
- new **topic distribution for w**: $p(t \mid d) \times p(w \mid t)$

- **…repeat until a steady state is achieved**

STYLIZED APPROACH

In the end, the topic model will give us two coefficients:

- γ (gamma), document-topic probability: the proportion of words in a document coming from a topic
- β (beta), term-topic probability: the probability of a term coming from a topic

UNSUPERVISED LEARNING WITH TEXT – THE PROCESS

- Choose a set of documents (corpus) and a number of topics *k*
  ⇒ usually *k* is not known a priori – estimation by training multiple models and comparing different measures
- Preprocess the documents
  ⇒ e.g., tokenization (also: bi- and trigrams), stemming/lemmatization, remove frequent words, etc. – for ramifications, see Denny and Spirling (2018)
- Learn topic model
- Make sense of topics

CHOICE OF CORPUS

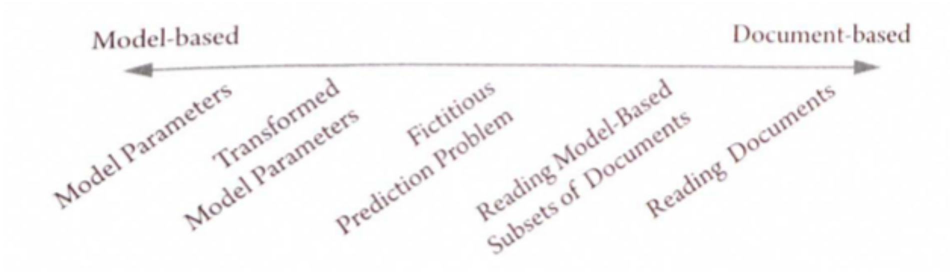
- Not as important here, model searches for structure
- Documents should have a certain length (since model assumes documents to be a *mixture of topics*)
  ⟹ for short texts, e.g., Tweets, specific “single-membership” models exist

CHOOSING K

- “One of the most difficult questions in Unsupervised Learning” (Grimmer and Stewart 2013: 19)
- No straightforward thing to do
- Solution: train many models and calculate evaluation scores for them (using R package “ldatuning”, or “stm::searchK()”)

MAKING SENSE OF TOPICS

- LDA gives you two values:
  - the probability that a word belongs to a topic, β
  - the probability that a document belongs to a topic, γ
- Goal: to give topics labels



Grimmer, Roberts, and Stewart 2022: 160

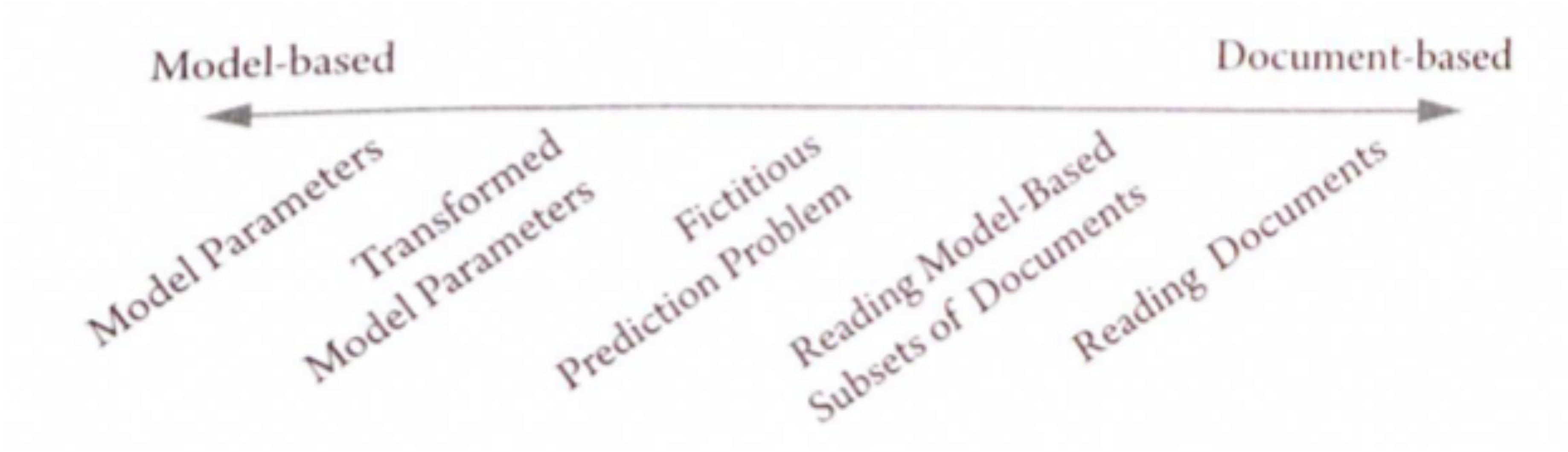
MAKING SENSE OF TOPICS

- Goal: to give topics labels
- Look at most prevalent terms contained in topics
  - *apophenia* (seeing patterns in random sets)
  - confirmation bias (seeing what you want to see)
- Read documents that consist mainly of words drawn from topics
  - tedious

⇒ but, remember the rules of text mining: VALIDATE VALIDATE VALIDATE

⇒ in this case: ensure that your topics constitute what you think they do

MAKING SENSE OF TOPICS



Grimmer, Roberts, and Stewart 2022: 160

EXTENSION: STRUCTURAL TOPIC MODELS

LDA comes with a bunch of limitations:

- Only takes text into account (no document covariates) – **topics are learned taking covariates into account**
- Topic-word distribution is stationary, cannot vary between documents (Republicans and Democrats may talk about the same topics but use different terms) – **different documents may contain the same topic but use different lingo**
- Topics are treated as independent from each other – **topics are allowed to be correlated**

⟹ Structural Topic Models mitigate these shortcomings

EXTENSION: SEEDED TOPIC MODELS

LDA comes with a bunch of limitations:

- Topics may actually be known in the beginning
- However: if LDA doesn't find the topic, this doesn't work
- Solution: **define ("seed") topics before – assign certain terms to topics**

⟹ Seeded topic model

RESULT

- Finally, you have added a new label to your document, namely its topic distribution
- You can use this label as a dependent as well as an independent variable for further inference

REMARKS

These methods are great and robust, but (unfortunately) will be outdated in the near future: transfer learning using large language models is going to replace them – for more on this, wait for TAD IV

The Augmented Social Scientist: Using Sequential Transfer Learning to Annotate Millions of Texts with Human-Level Accuracy

Salomé Do[1,2], Étienne Ollion[3], and Rubing Shen[2,3]



REFERENCES

- Barberá, Pablo, Amber E. Boydstun, Suzanna Linn, Ryan McMahon, and Jonathan Nagler. 2021. “Automated Text Classification of News Articles: A Practical Guide.” *Political Analysis* 29(1):19–42.
- Denny, Matthew J. and Arthur Spirling. 2018. “Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It.” Political Analysis 26(2):168–89.
- DiMaggio, Paul, Manish Nag, and David Blei. 2013. “Exploiting Affinities Between Topic Modeling and the Sociological Perspective on Culture: Application to Newspaper Coverage of U.S. Government Arts Funding.” *Poetics* 41(6):570–606.
- Grimmer, Justin, Margaret Roberts, and Brandon Stewart. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton: Princeton University Press.
- Hurtado Bodell, Miriam, Måns Magnusson, and Marc Keuschnigg. n.d. “Seeded Topic Models in Digital Archives: Analyzing the Swedish Understanding of Immigration, 1945–2019.” OSF Preprint.
- Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, and Edoardo M. Airoldi. 2013. “The Structural Topic Model and Applied Social Science.” in *NIPS 2013 Workshop on Topic Models*.

MERCI

Felix Lennert
Institut für Soziologie

felix.lennert@uni-leipzig.de
www.uni-leipzig.de