Auto-IA Workshop HMTM Hannover

Automatische Sprachverarbeitung

LEIBNIZ-INSTITUT
FÜR MEDIENFORSCHUNG
HANS-BREDOW-INSTITUT

Automatic Content Analysis

Topic Modeling

Gregor Wiedemann | g.wiedemann@leibniz-hbi.de
Media Research Methods Lab
Leibniz-Institute for Media Research | Hans-Bredow-Institut

Andreas Niekler | aniekler@informatik.uni-leipzig.de
Abteilung Automatische Sprachverarbeitung, Institut für Informatik
Universität Leipzig

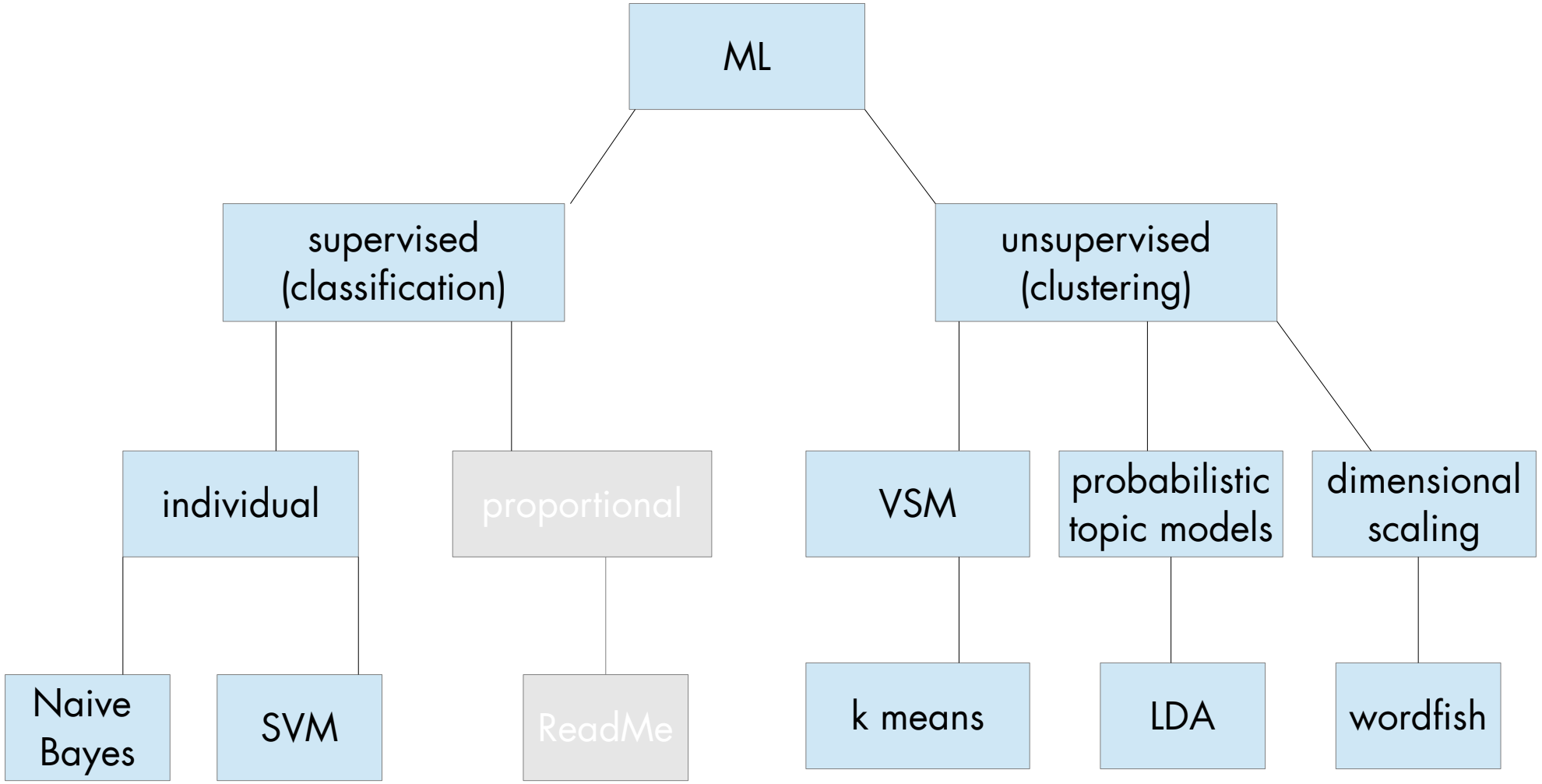
Machine Learning

- Definitions
  - "[Giving] computers the ability to learn without being explicitly programmed" – Arthur Samuel
  - "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" – Tom M. Mitchell
- Common foundations
  - Probability Theory
  - Decision Theory (making "good" decisions based on data)
  - Optimization (optimize model of data based on decision objective)
- independent of application domain

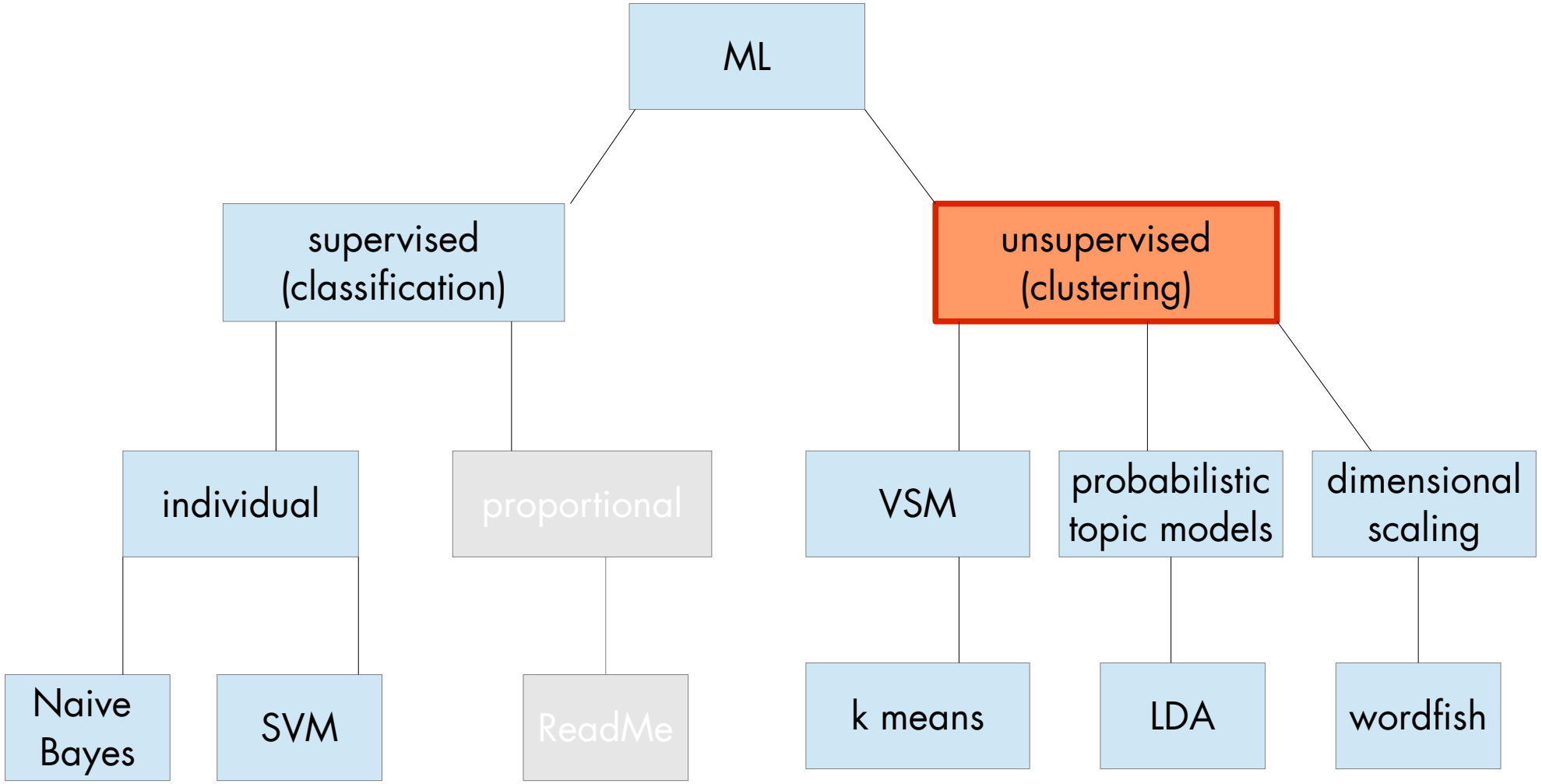
Application domains of ML

- Autonomous driving
- Medical diagnosis
- Network security
- …
- Textmining and NLP:
  - Linguistic preprocessing: POS-Tagging, Parsing
  - Modelling Language Semantic
    - category classification
    - clusters of (latent) meaning

Machine learning for NLP



Machine learning for NLP



Clustering

Automatische Sprachverarbeitung

- Cluster analysis:
  - „task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)“ [Wikipedia]
- Motivation in content analysis:
  - identifying similar / (quasi-)duplicate documents / parts of text
  - splitting a collection of documents into groups
    - thematic
    - time periods
  - identifying groups of related terms
  - placing documents on an ideological scale

Topic Modeling

Topic Models

- Class of probabilistic graphical models which infer semantic coherences from large text collections as latent variables
  - Latent variables = topics
- Background assumptions
  - Bag-of-words model
  - Generative process (1. document author, 2. draw topic mixture, 3. draw terms from topics → document)
- 2 posterior distributions
  - θ : Documents = mixtures of topics
  - β : Topics = probability distributions over terms

Variety of models (Blei 2012): e.g.

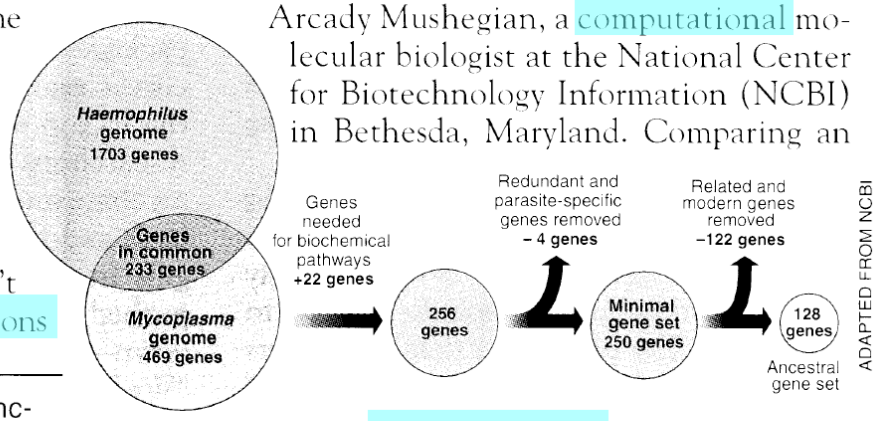
- Latent Dirichlet Allocation (Blei et al. 2003)
- Correlated Topic Models (Blei & Lafferty 2007)
- Hierarchical Dirichlet Process (Teh et al. 2006)
- Author-Topic Model (Rosen-Zvi et al. 2004)
- …

Latent Dirichlet Allocation (LDA)

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



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

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

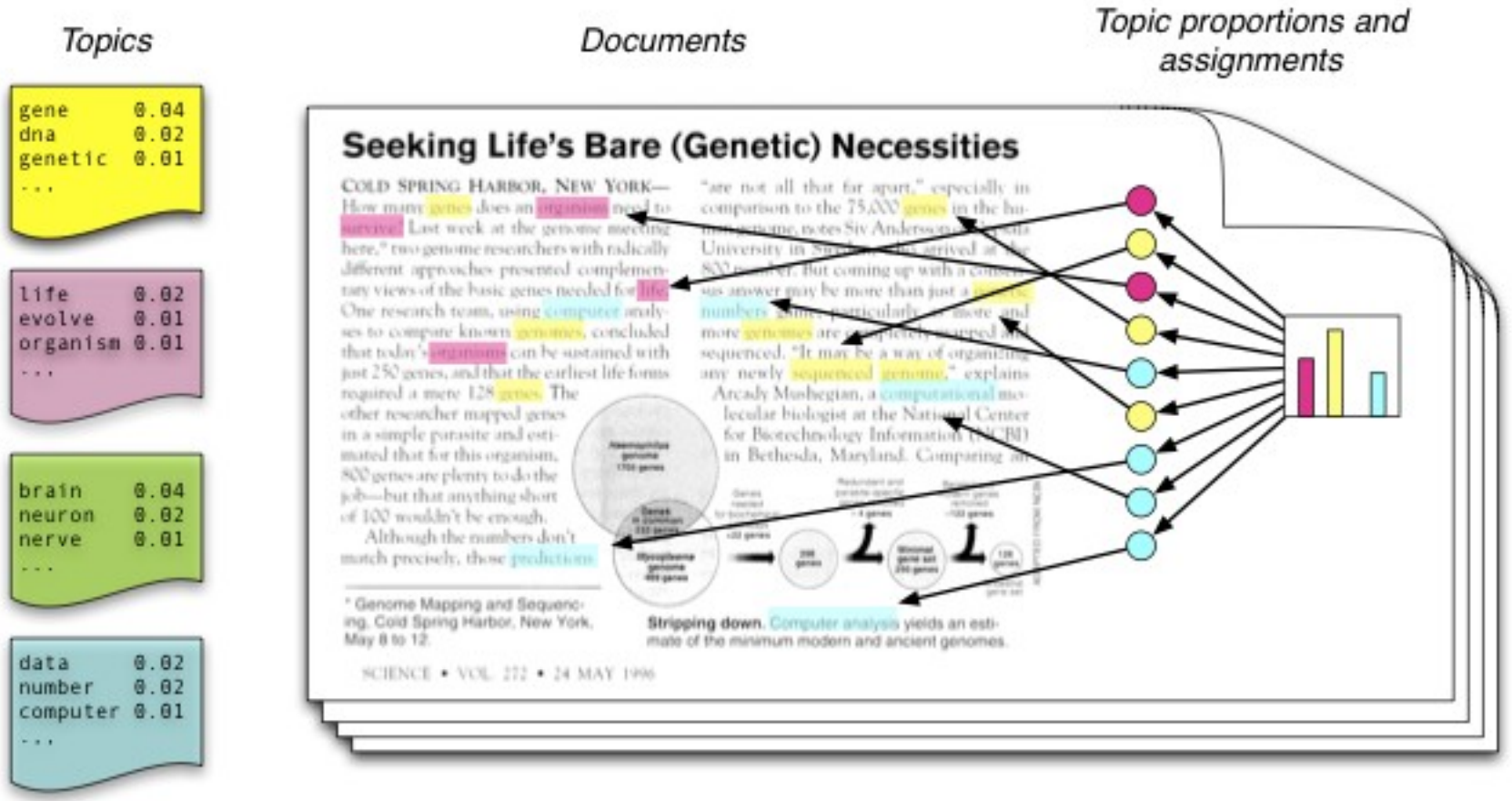
SCIENCE • VOL. 272 • 24 MAY 1996

Assumptions:

1. Documents are mixtures of topics

2. Topics are distributions over terms

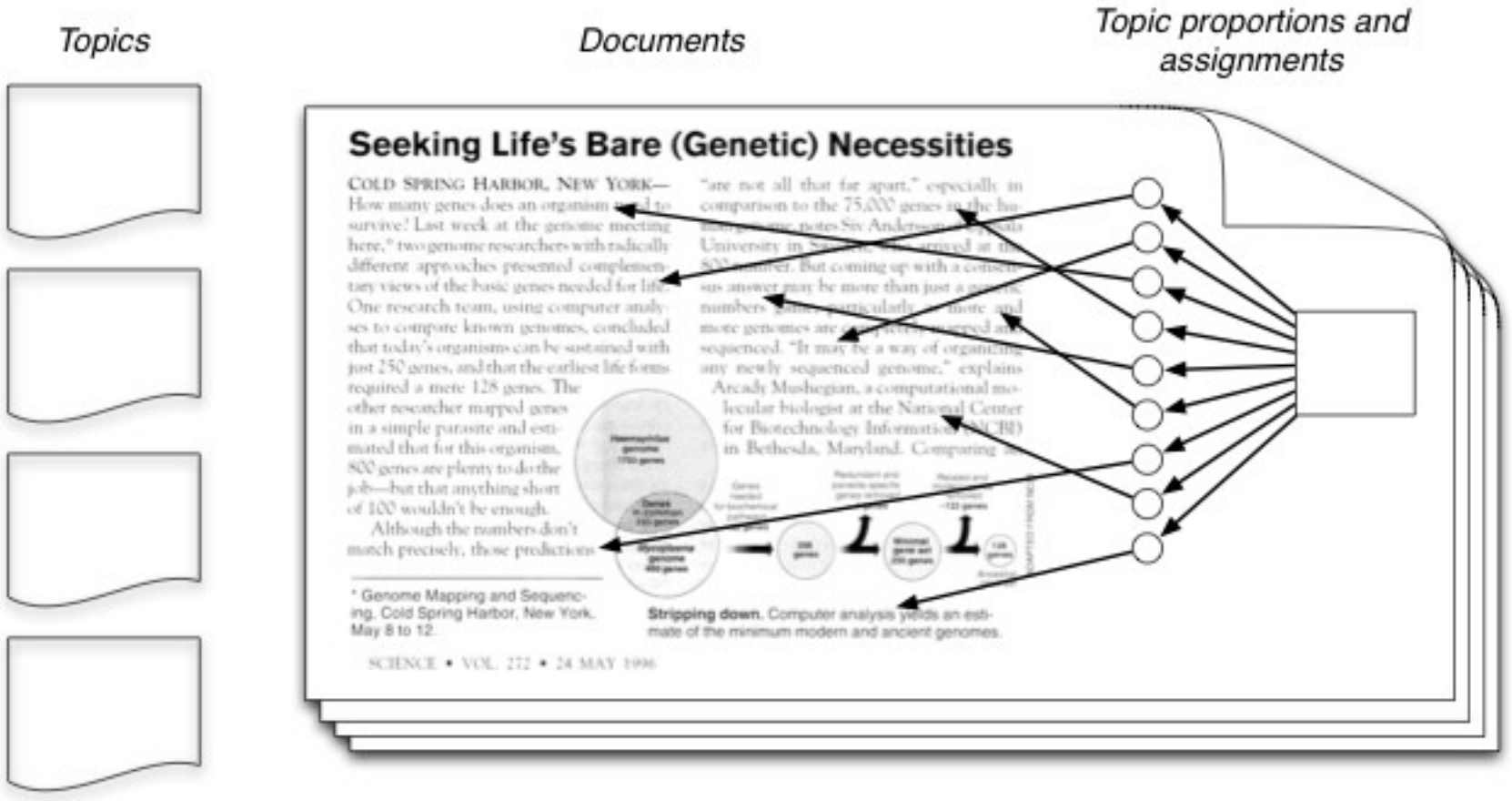
Latent Dirichlet Allocation (LDA)



Model assumptions

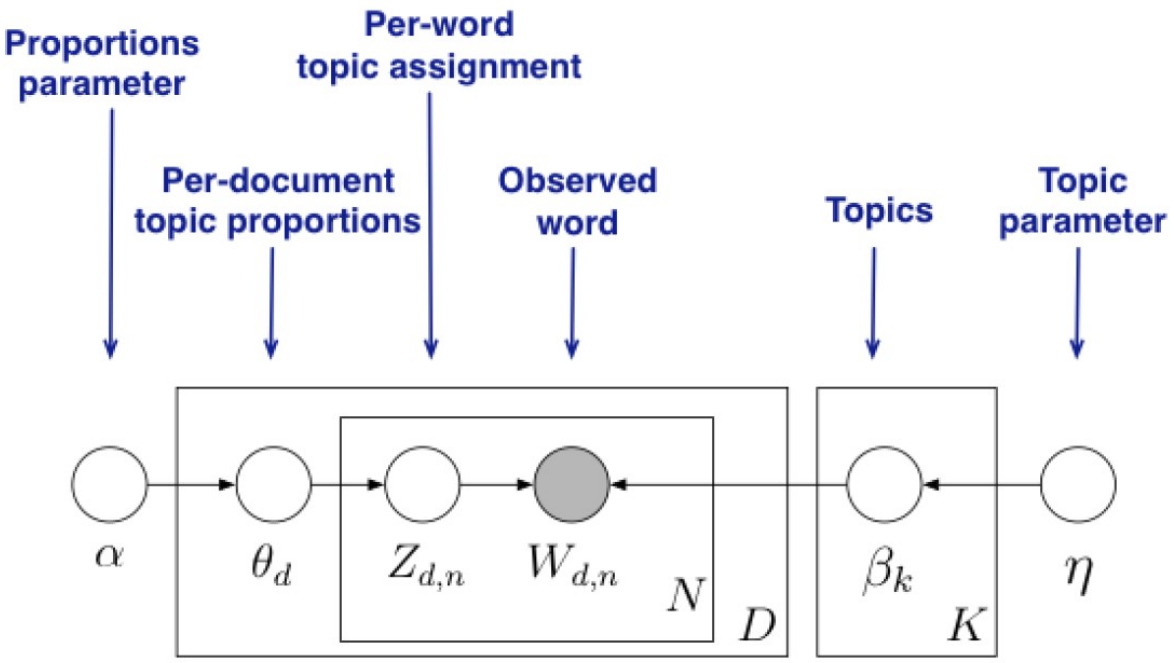
- every topic is represented by a distribution over terms
- every document is represented by a distribution over topics
- generative process: for every document → draw a topic distribution → for each topic → draw a term

Latent Dirichlet Allocation (LDA)



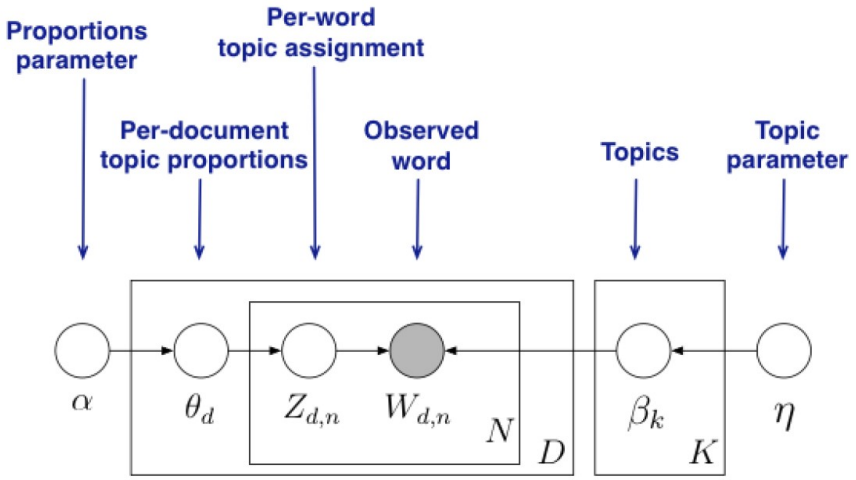
- in reality we see only the documents
- topics as latent variables have to be inferred from the observations (terms in documents)
- → calculate the distribution: *p(topics, proportions, assignments | document)*

LDA as graphical model



$$p(\beta, \theta, \mathbf{z}, \mathbf{w}) = \left(\prod_{i=1}^{K} p(\beta_i|\eta) \right) \left(\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

LDA as graphical model



- from joint distribution P(β,θ,**z**,**w**), posterior distribution can be inferred: P(β,θ,**z**|**w**)
- i.e. Topic model inference resulting in
  - 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
- results can be used for information retrieval, time series, collection filtering, …

Inference

$$p(\beta, \theta, \mathbf{z}, \mathbf{w}) = \left(\prod_{i=1}^{K} p(\beta_i|\eta) \right) \left(\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

- Analytically intractable → approximate solution needed
- Hard task in topic modeling: computationally tractable inference on model parameters; e.g.
  - Gibbs Sampling
  - Variational inference
- Good news: there are some fast and reliable implementations, that do the math for you :)

- Crucial parameters influencing model outcomes:
  - K – number of topics to be inferred
  - α – prior for topic–document distributions
  - η – prior for term–topic distributions
- Inference is based on sampling → results are not entirely deterministic

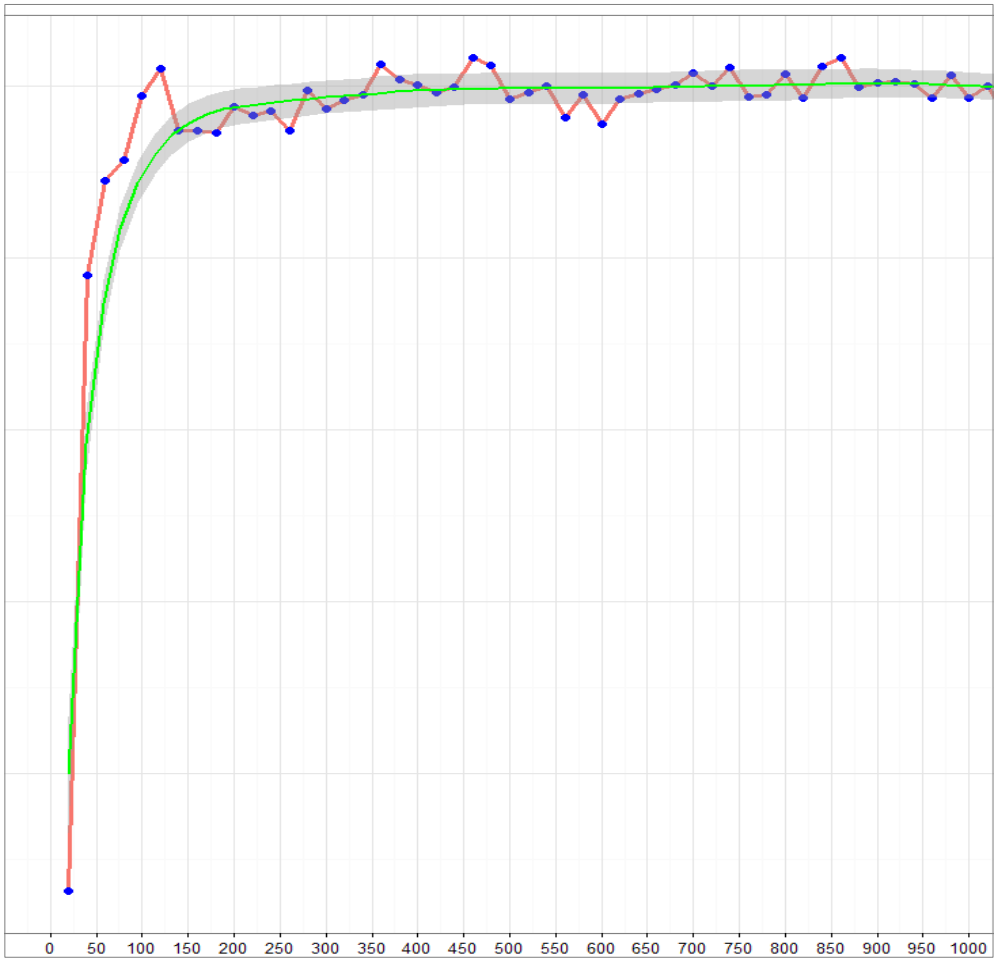
Inference – Gibbs Sampling

Automatische Sprachverarbeitung

Data: words $\mathbf{w} \in$ documents $\mathbf{d}$
Result: Topic assignments $\mathbf{z}$
initialize $\mathbf{z}$ randomly
foreach *iteration* **do**
  foreach *word* w **do**
    foreach *topic* k **do**
      $\theta_{d_w,k} \propto \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V} n^{(t)}_{k,\neg i} + \beta_t} (n^{(k)}_{m,\neg i} + \alpha_k)$
    end
    topic $\leftarrow$ sample from $mult(\theta_{d_w})$
    z[w]$\leftarrow$ topic
    update counts according to new assignment
  end
end
return z

Inference – Gibbs Sampling

$$p(\beta, \theta, \mathbf{z}, \mathbf{w}) = \left(\prod_{i=1}^{K} p(\beta_i|\eta) \right) \left(\prod_{d=1}^{D} p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$



- More interations = better likelihood of the model
- Optimum almost reached at 200 – 300 iterations – This is called „burnin“ of the sampler
- Likelihood does implicate the quality of the model for your research – it is an optimum w.r.t. the model assumptions
- Rule of thumb: 1000 – 2000 iterations

Inference – Gibbs Sampling

$$\propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{(t)} + \beta_t} (n_{m,\neg i}^{(k)} + \alpha_k)$$

Gibbs algorithm samples assignments for each word

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t},$$

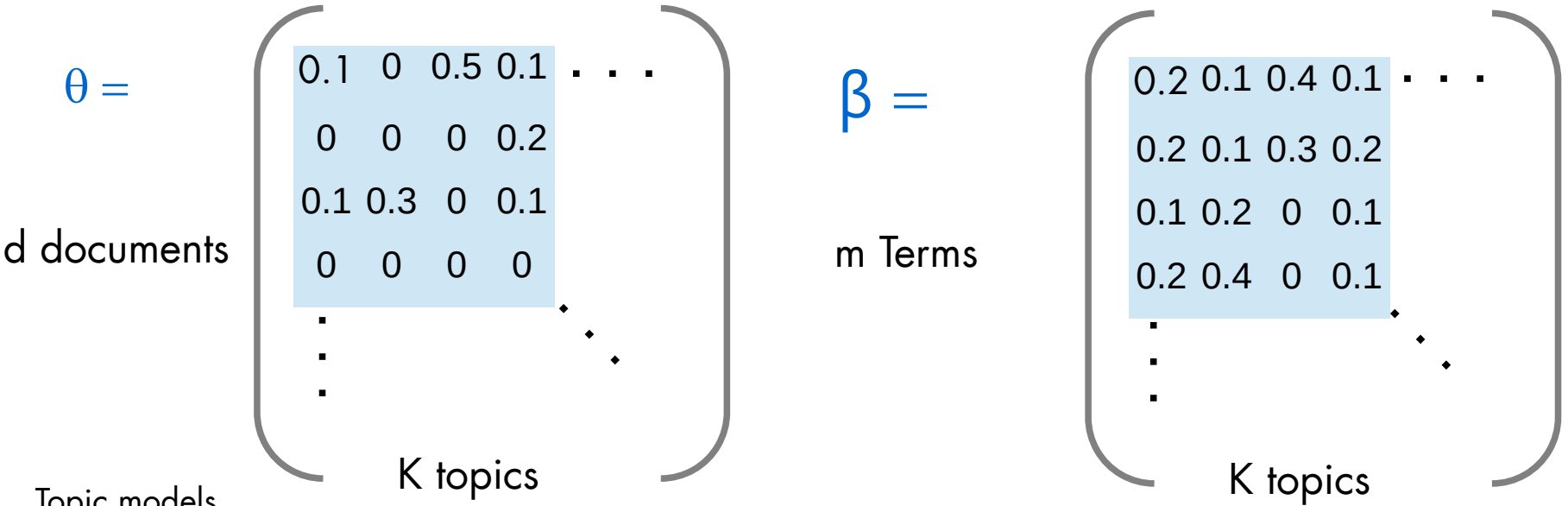
$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k}.$$

!!! Sometimes β is referenced as Φ because of the hyperparameters redundant name.

Samples must be converted to parameters θ and β by accuiring the statistics from the assigments.

Inference

- 2 mutually exclusive goals of inference process
  - assign terms to as little as possible topics in one document
  - assign high probability to as little as possible terms in one topic
- interdependency:
  - assign all terms in one document to one topic only → missing 2. goal
  - assigning high probability to few terms in each topic → missing 1. goal
- optimal alignment of these goals results in semantically coherent groups of terms and interpretable numbers of topics in documents



LDA examples

Example: Wortschatz data source, 100 million sentences in 2010

Topic 12
people
haiti
air
flight
day
hours
airport
weather
morning
port
night
earthquake
early
thursday
fire

[Jähnichen 2015]

LDA examples

Example: Wortschatz data source, 100 million sentences in 2010

Topic 23
oil
bp
gulf
spill
gas
company
coast
water
mexico
drilling
million
day
disaster
louisiana
barrels

[Jähnichen 2015]

LDA examples

Example: Wortschatz data source, 100 million sentences in 2010

Topic 48
space
nasa
station
shuttle
program
earth
planet
center
rocket
moon
international
mission
launch
mars
star
search

[Jähnichen 2015]

LDA examples

Topic 5	Topic 11	Topic 19	Topic 30	Topic 86
apple	food	fish	bank	summer
iphone	eat	river	financial	fall
phone	restaurant	water	banks	spring
mobile	fresh	animals	debt	year
google	meat	lake	crisis	early
users	wine	dog	market	winter
ipad	chicken	fishing	investment	late
company	add	dogs	capital	tourism
software	small	species	money	season
computer	water	boat	fund	trips
web	cheese	animal	billion	garden
video	vegetables	hunting	company	spirit
internet	rice	bass	investors	tour
access	lunch	waters	government	ll
devices	foods	forest	companies	travel
microsoft	menu	wildlife	based	families

[Jähnichen 2015]

Sorting Terms

Automatische Sprachverarbeitung

Standard Term Scoring

$$relevance(w|t) = sort(p(w|t))$$

Scoring by Chang, J., Chang, M.J.: Package “lda”.
Citeseer (2010).

$$relevance(w|t) = p(w,t) \cdot \left(\log p(w|t) - \frac{1}{K} \sum_{t'} \log p(w|t') \right)$$

Scoring by Sievert, C., Shirley, K.E.: LDAvis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. pp. 63–70 (2014).

$$relevance(w|t) = \lambda \cdot \log p(w|t) + (1-\lambda) \cdot \log \frac{p(w|t)}{p(w)}.$$

Sorting Topics

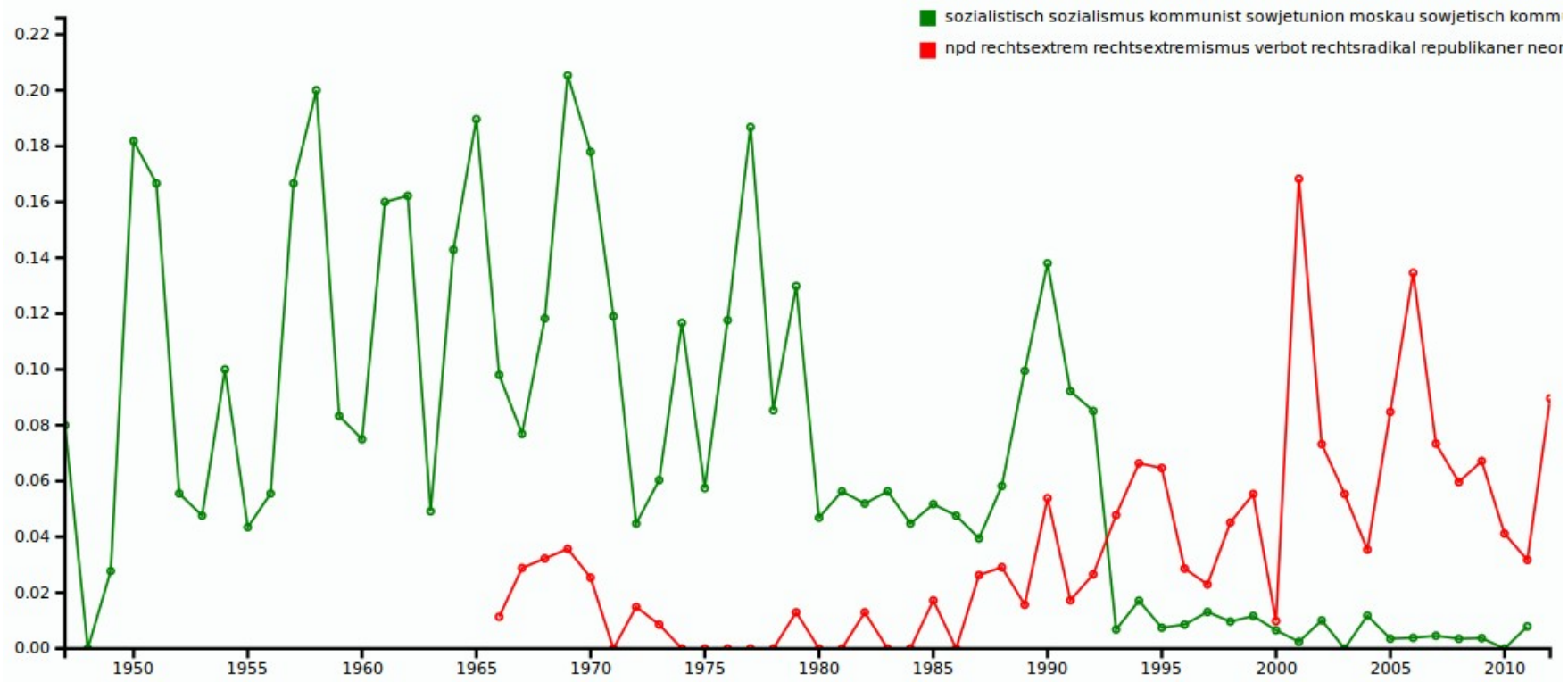
1. Approach: We sort the topics by their probability among the whole collection:

$$p(z) = \frac{\sum_D p(z|d)}{\sum_D \sum_{z=1}^{K} p(z|d)}$$

2. Approach: We count how often a topic appears as the primary topic within the documents. This method is called Rank-1.

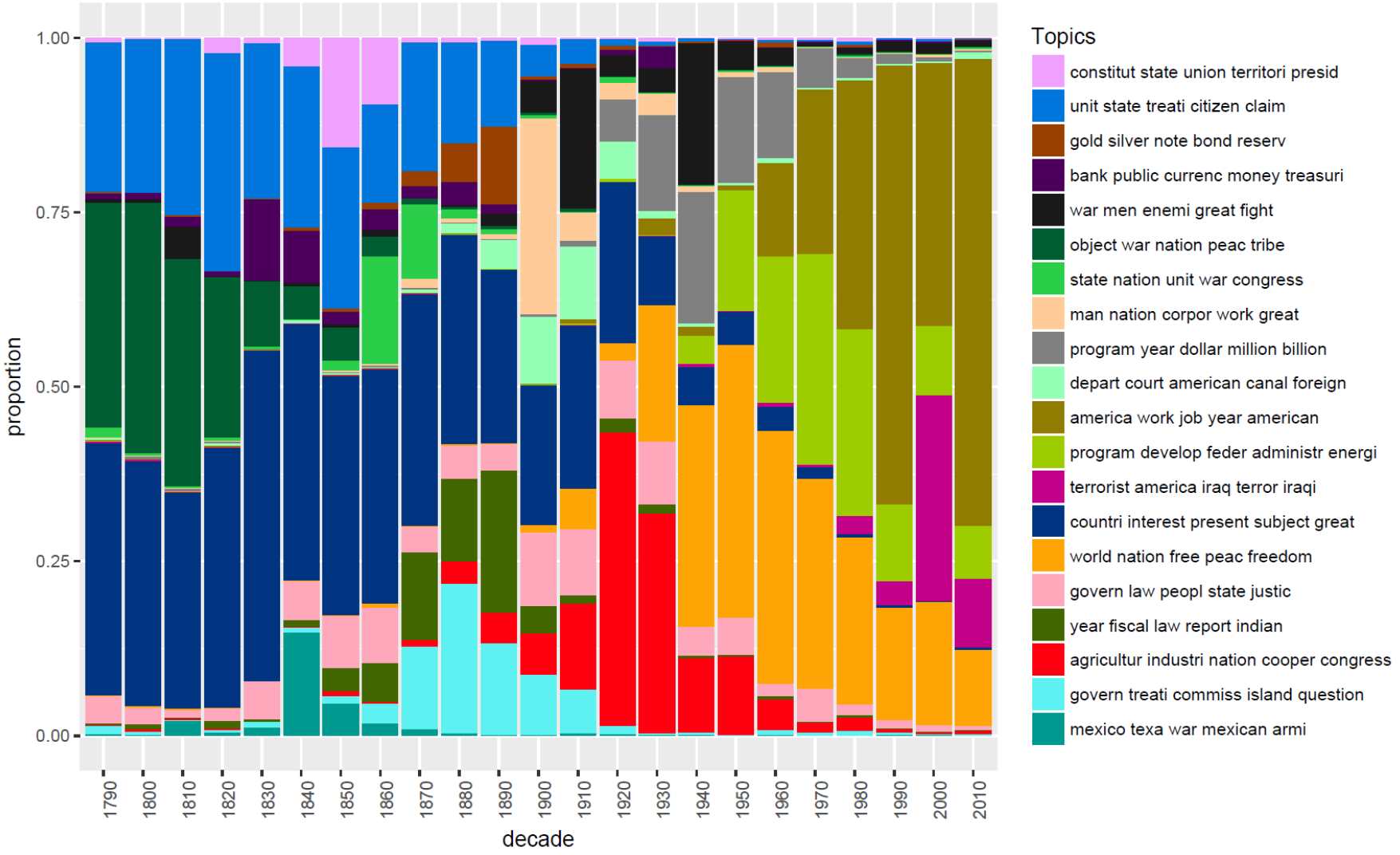
$$count(z) = \sum_{d=1}^{D} \arg\max_{z' \in K} p(z'|d) \begin{cases} 1, & \text{if } z' = z, \\ 0, & \text{otherwise.} \end{cases}$$

LDA examples



Time series: frequencies (relative to collection size) of documents containing minimum 25% share of topic k ($\theta_{d,k} > 0.25$)

LDA examples



Applying topic models in CA

Automatische Sprachverarbeitung

- Topic models can be employed to
  - split collections into thematic clusters
  - identify documents related to certain topics
  - track topic quantities over time (aggregation of document topic probabilities)
  - identify similar documents based on similarity of topic distributions rather than term frequencies (cp. k-means on DTM)
  - identify relevant vocabulary within a collection
- Analysts need to
  - choose appropriate K
  - interprete resulting topics
  - eventually optimize mode parameters (α, η priors)
  - incorporate result data into their workflows

Possible alterations

- Filter POS-Types: Concentrate on NN(S) / NNP(S)
- Add extra vocabulary
  - time stamps, categories, dictionary terms, onthology categoritzation
- Initialize to predefined assignments – forced topic id's for words
  - seeded initialization
  - initialization by co-occurrence clusters of words
- Problem: Inner document statistics are dependent on empirical data within the document - Short documents cannot be modelled very well
  - Twitter: Try to concat tweets of the same user or hashtag to longer artificial documents – depends on the intendet topical structures