Methods workshop: Automated Text Analysis with R

Pablo Barberá
Center for Data Science
New York University

May 19, 2016

materials: github.com/pablobarbera/eui-text-workshop

Text as data

Text as data


Kukuriku
koalicija
Plan 21


the TRUE NORTH
STRONG AND FREE
Stephen Harper's plan for Canadians
Conservative


Et retfærdigt
Danmark


ΕΚΛΟΓΙΚΗ ΔΙΑΚΗΡΥΞΗ
ΣΥΡΙΖΑ
ισχυρός
στη Βουλή
και στους αγώνες
Είναι στο χέρι μας!
ΣΥΡΙΖΑ
ΣΥΝΑΣΠΙΣΜΟΣ ΡΙΖΟΣΠΑΣΤΙΚΗΣ ΑΡΙΣΤΕΡΑΣ
www.syriza.gr

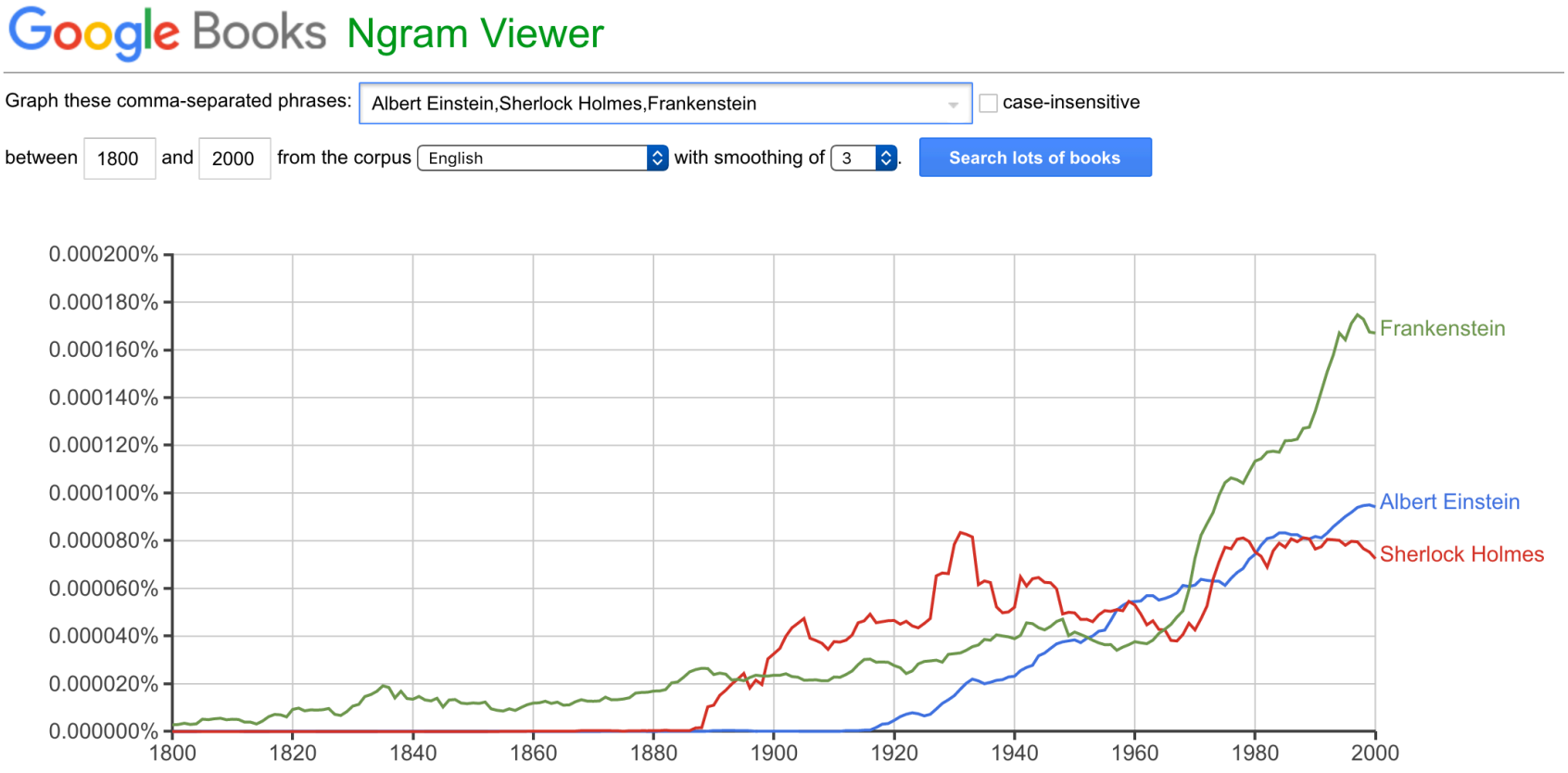

SPÖ
WAHLMANIFEST


GRÜNE NEUE
GESELLSCHAFTS-
VERTRAG
KLIMA • ARBEIT • GERECHTIGKEIT • FREIHEIT


WIR HABEN DIE KRAFT
GEMEINSAM
FÜR UNSER LAND.
REGIERUNGSPROGRAMM 2009 – 2013
CDU CSU


Sozial und
Demokratisch.
Anpacken. Für Deutschland.
Das Regierungsprogramm der SPD
SPD

Text as data


Google Books Ngram Viewer
Graph these comma-separated phrases: Albert Einstein,Sherlock Holmes,Frankenstein
case-insensitive
between 1800 and 2000 from the corpus English with smoothing of 3
Search lots of books
0.000200%
0.000180%
0.000160%
0.000140%
0.000120%
0.000100%
0.000080%
0.000060%
0.000040%
0.000020%
0.000000%
1800
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
Frankenstein
Albert Einstein
Sherlock Holmes

Text as data


facebook
XING
Pinterest
Gefällt mir
Instagram
twitter
LinkedIn
flickr
skype
YouTube

Overview of text as data methods

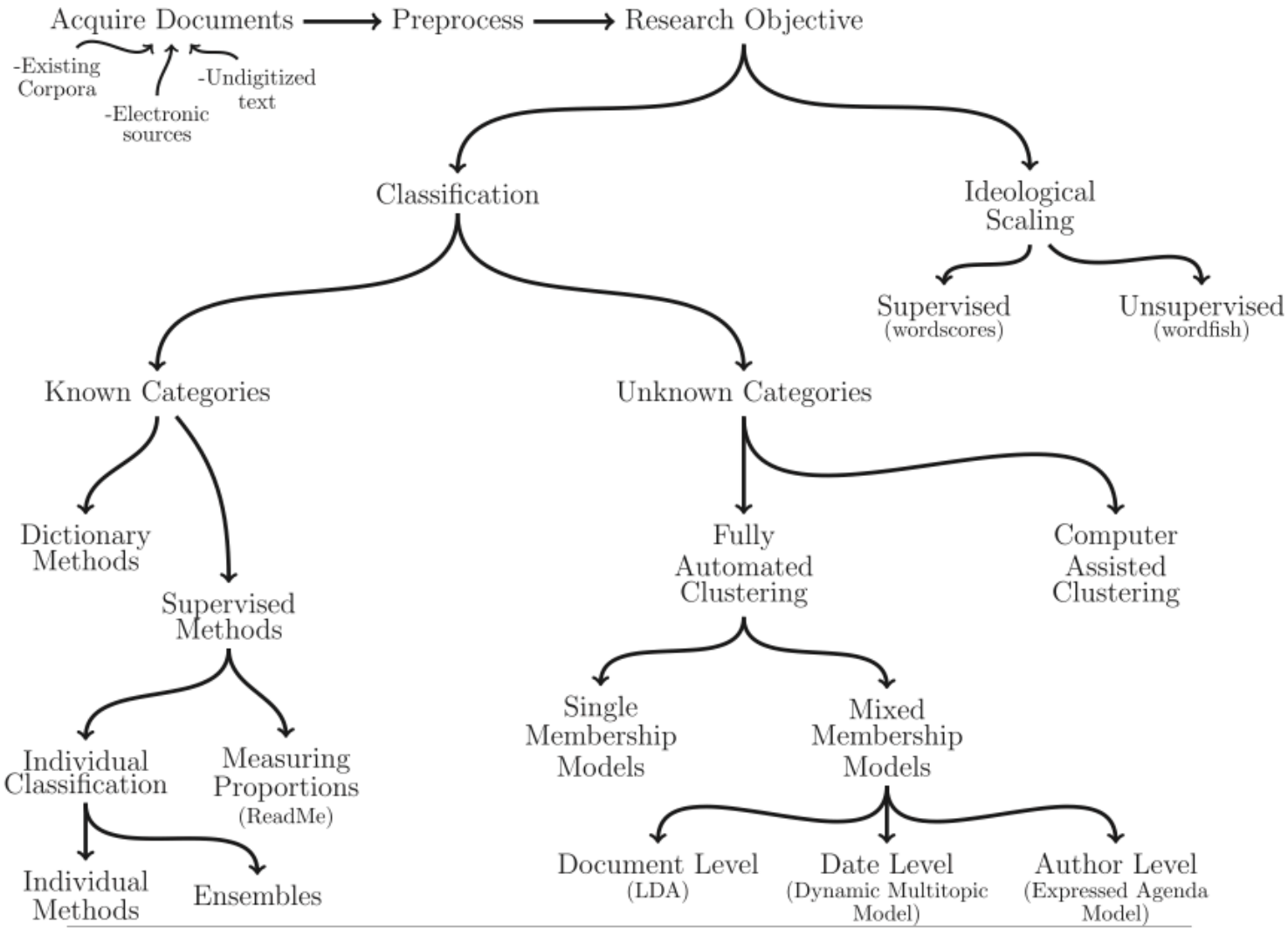


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

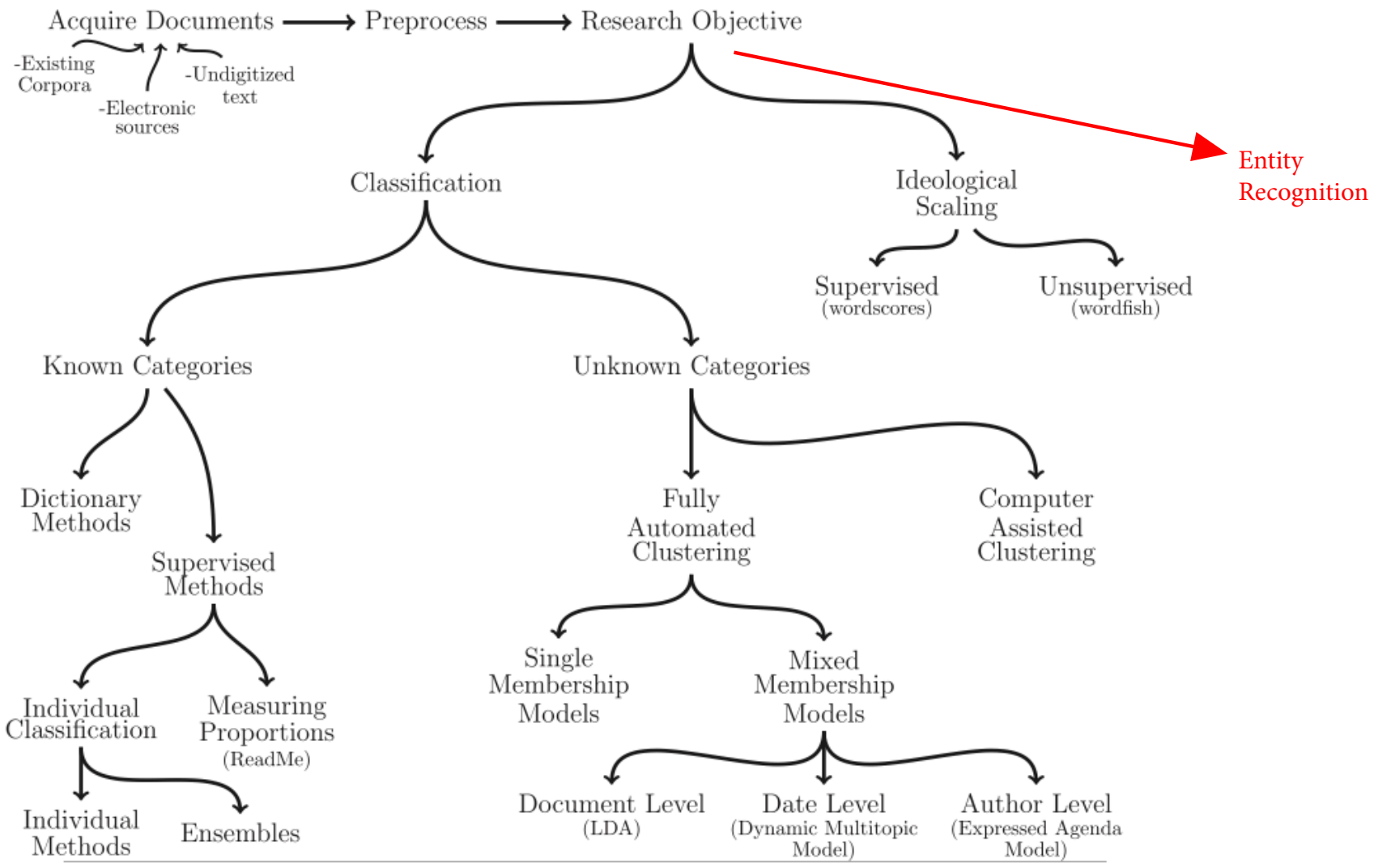


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

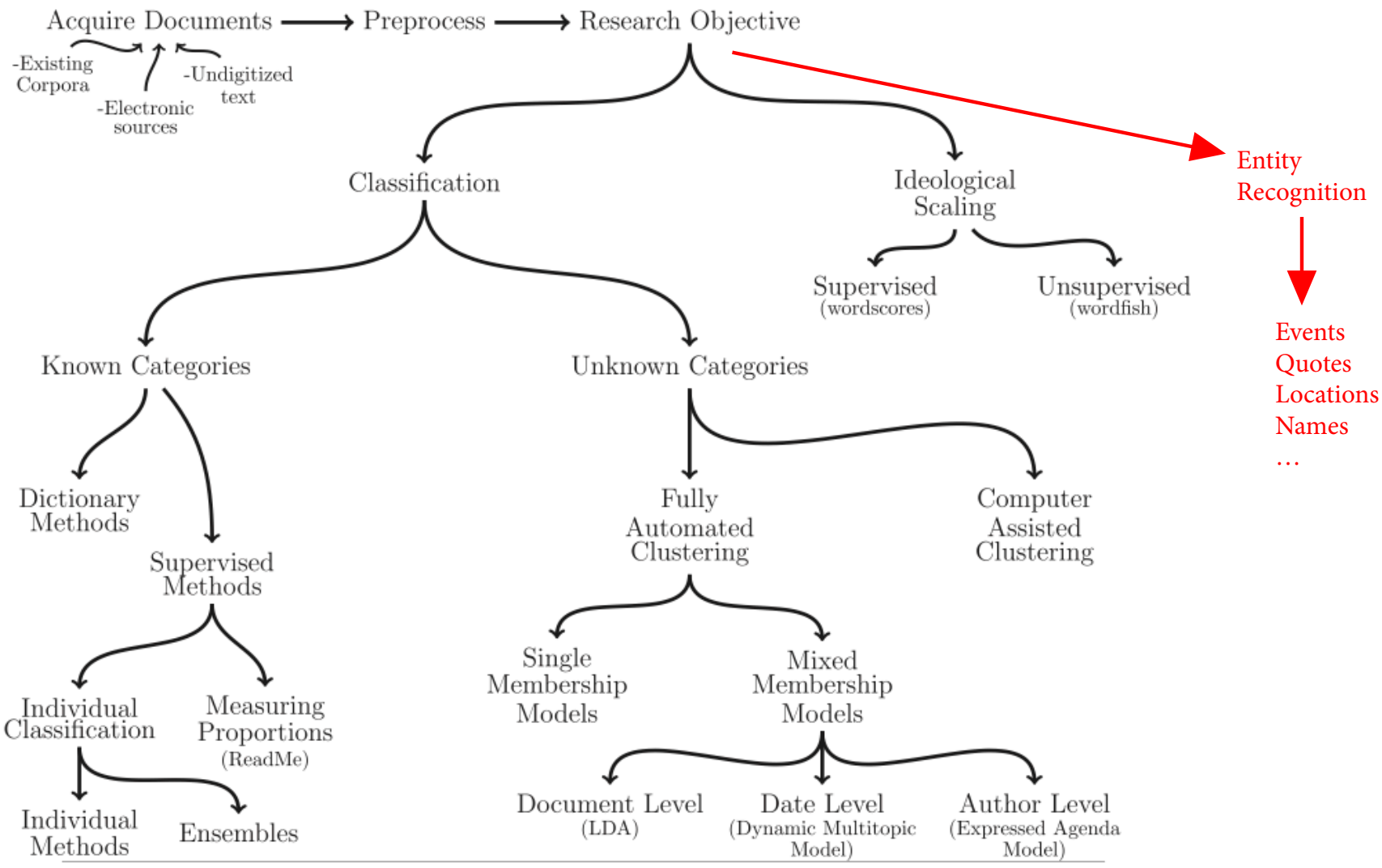


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

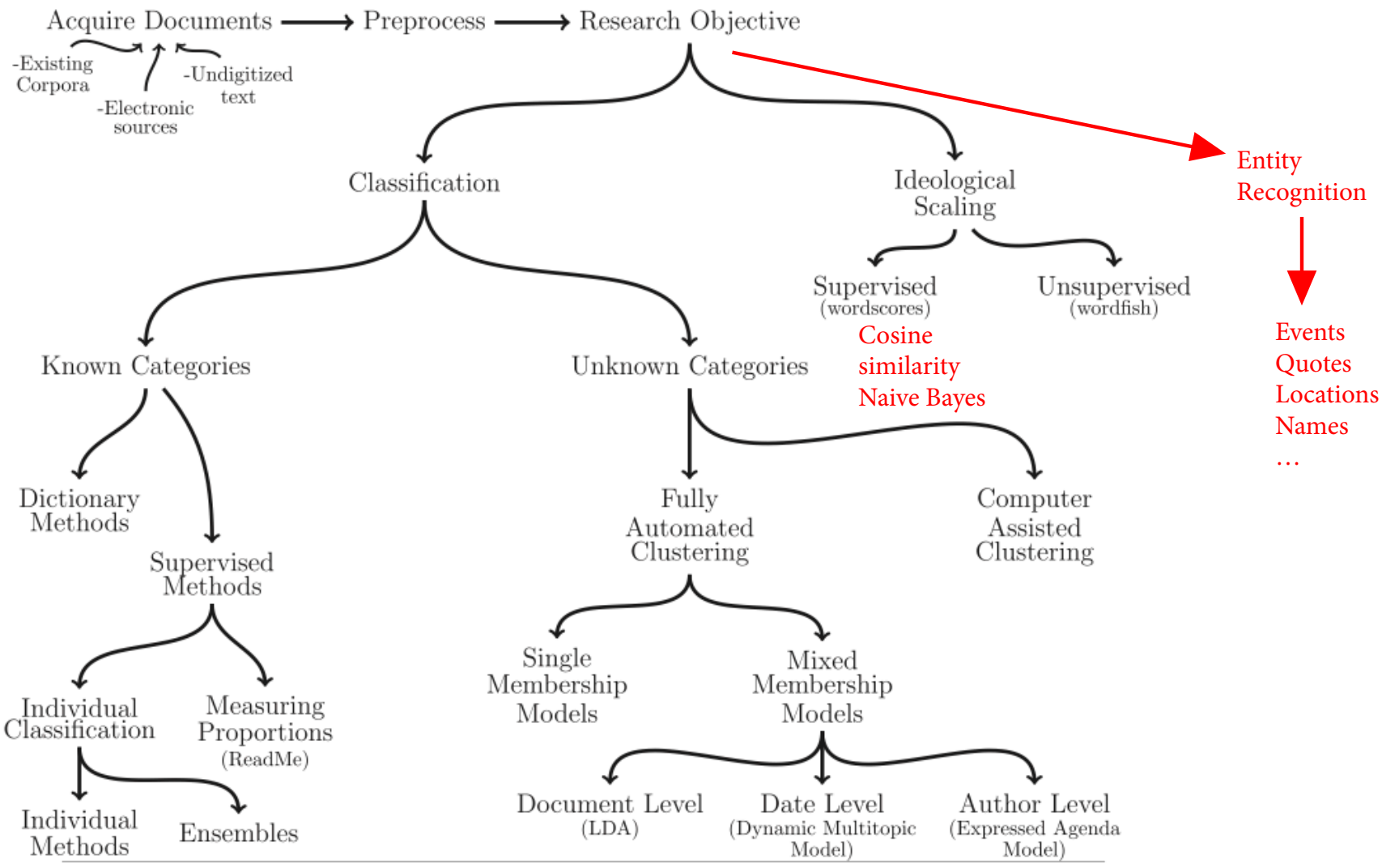


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

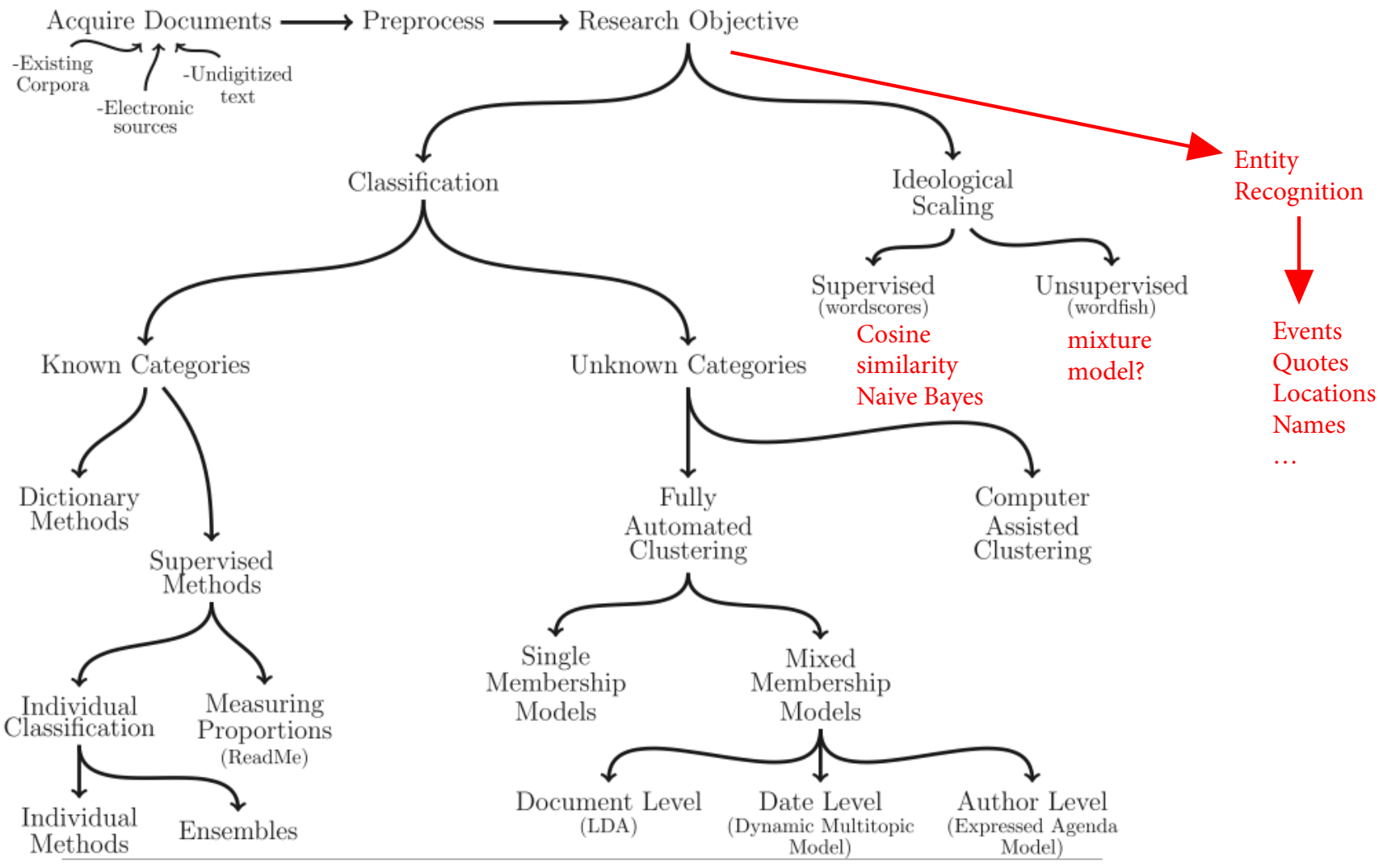


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

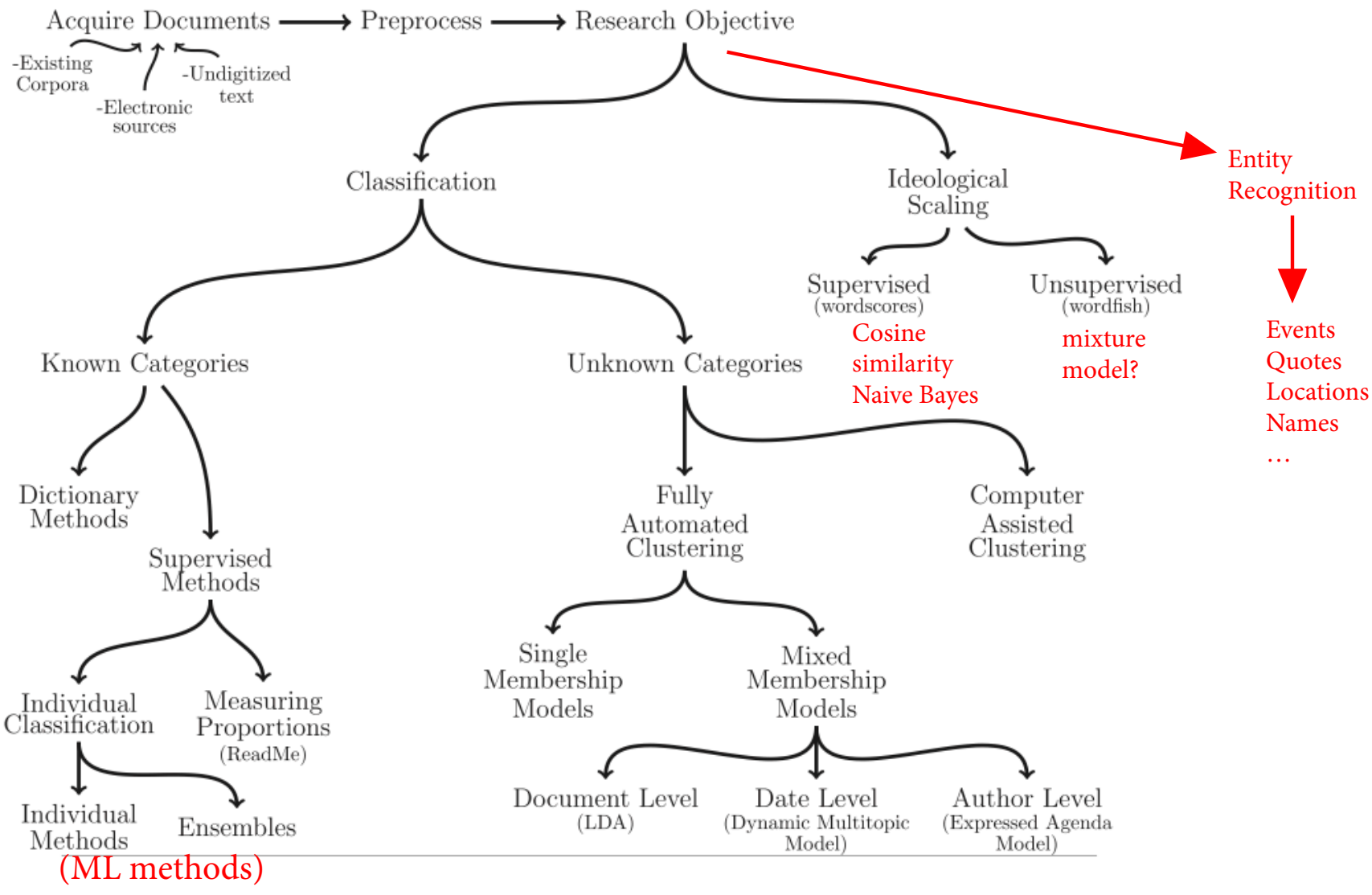


Fig. 1 in Grimmer and Stewart (2013)

Overview of text as data methods

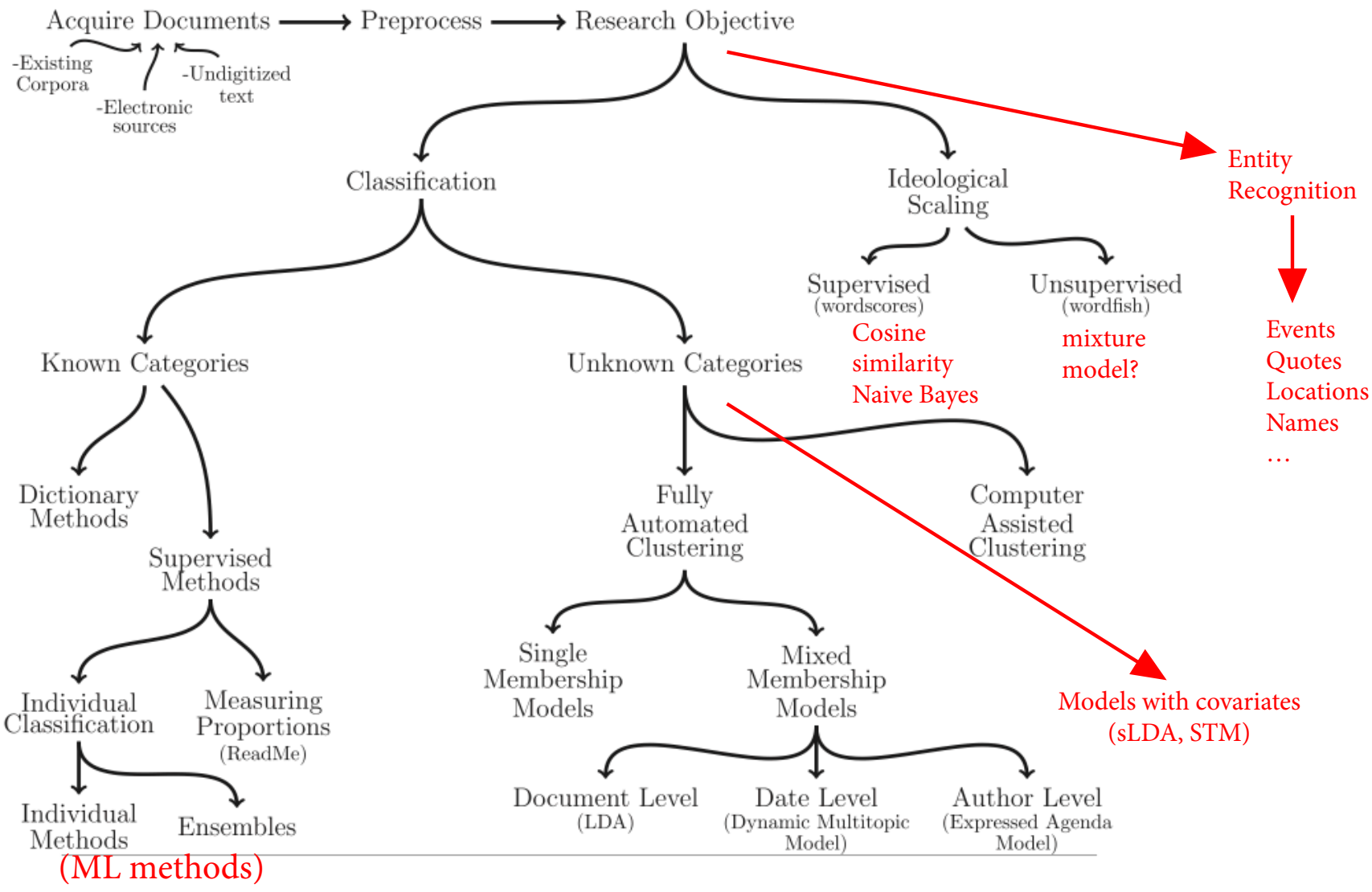


Fig. 1 in Grimmer and Stewart (2013)

From words to numbers

1. Bag-of-words assumption

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization
   - Part-of-speech tagging

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization
   - Part-of-speech tagging
3. Document-term matrix

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization
   - Part-of-speech tagging
3. Document-term matrix
   - **W**: matrix of N documents by M unique words

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization
   - Part-of-speech tagging
3. Document-term matrix
   - **W**: matrix of N documents by M unique words
   - W_{im}= number of times m-th words appears in i-th document.

From words to numbers

1. Bag-of-words assumption
2. Pre-processing text
   - Capitalization, cleaning digits/URLs, removing stopwords and sparse words, etc.
   - Stemming / lemmatization
   - Part-of-speech tagging
3. Document-term matrix
   - **W**: matrix of N documents by M unique words
   - W_{im}= number of times m-th words appears in i-th document.
   - Usually large matrix, but sparse (so it fits in memory)

From words to numbers

From words to numbers

1. Preprocess text:

"@MEPcandidate thank you and congratulations, you're the best #EP2014"

"@MEPcandidate You're an idiot, I would never vote for you"

From words to numbers

From words to numbers

1. Preprocess text: lowercase,

"@mepcandidate thank you and congratulations, you're the best #ep2014"

"@mepcandidate you're an idiot, i would never vote for you"

From words to numbers

From words to numbers

1. Preprocess text: lowercase, remove stopwords and punctuation,

"@mepcandidate thank ~~you~~ ~~and~~ congratulations~~,~~ you're ~~the~~ best #ep2014"

"@mepcandidate you're ~~an~~ idiot~~,~~ ~~i~~ ~~would~~ never vote ~~for~~ ~~you~~"

From words to numbers

From words to numbers

1. Preprocess text: lowercase, remove stopwords and punctuation, stem,

   "@ thank congratul~~ations~~, you'r~~e~~ best #ep2014"

   "@ you'r~~e~~ idiot never vote"

From words to numbers

From words to numbers

1. Preprocess text: lowercase, remove stopwords and punctuation, stem, tokenize into unigrams and bigrams (bag-of-words assumption)

   [@, thank, congratul, you'r, best, #ep2014, @ thank, thank congratul, congratul you'r, you'r best, best, best #ep2014]

   [@, you'r, idiot, never, vote, @ you'r, you'r idiot, idiot never, never vote]

From words to numbers

From words to numbers

1. Preprocess text: lowercase, remove stopwords and punctuation, stem, tokenize into unigrams and bigrams (bag-of-words assumption)

   [@, thank, congratul, you'r, best, #ep2014, @ thank, thank congratul, congratul you'r, you'r best, best, best #ep2014]

   [@, you'r, idiot, never, vote, @ you'r, you'r idiot, idiot never, never vote]

2. Document-term matrix:
   - **W**: matrix of N documents by M unique words
   - W_{im}= number of times m-th words appears in i-th document.

	@	thank	congratul	you'r	#ep2014	@ thank	…	M words
Document 1	1	1	1	1	1	1	…	
Document 2	1	0	0	1	0	0	…	
…								
Document n n	0	1	1	0	0	0	…	

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier
  - Validation/Test set: used to validate the classifier

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier
  - Validation/Test set: used to validate the classifier
- Method to extrapolate from hand coding to unlabeled documents (classifier):

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier
  - Validation/Test set: used to validate the classifier
- Method to extrapolate from hand coding to unlabeled documents (classifier):
  - SVM, Naive Bayes, regularized regression, BART, ensemble methods...

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier
  - Validation/Test set: used to validate the classifier
- Method to extrapolate from hand coding to unlabeled documents (classifier):
  - SVM, Naive Bayes, regularized regression, BART, ensemble methods...
- Approach to validate classifier: cross-validation

Supervised machine learning

Goal: classify documents into pre existing categories.
e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews... **What we need**:

- Hand-coded dataset (labeled), to be split into:
  - Training set : used to train the classifier
  - Validation/Test set: used to validate the classifier
- Method to extrapolate from hand coding to unlabeled documents (classifier):
  - SVM, Naive Bayes, regularized regression, BART, ensemble methods...
- Approach to validate classifier: cross-validation
- Performance metric to choose best classifier and avoid overfitting: confusion matrix, AUC, accuracy, precision, recall...

Regularized regression

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\beta, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \beta' \boldsymbol{x}_i \right)^2 \\
\widehat{\beta} &= \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Source: Grimmer, 2014, "Text as Data" course week 15

Regularized regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

where:

- $\beta_0 \rightsquigarrow$ intercept
- $\lambda \rightsquigarrow$ penalty parameter

Source: Grimmer, 2014, "Text as Data" course week 15

Performance metrics

Confusion matrix:

	Actual Label	
Classification (algorithm)	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

$$F_{\text{Liberal}} = \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}$$

Source: Grimmer, 2014, "Text as Data" course week 14

Cross-validation

Intuition:

- Create K training and test sets ("folds") within training set.

Cross-validation

Intuition:

- Create K training and test sets ("folds") within training set.
- For each k in K, run classifier and estimate performance in test set within fold.

Cross-validation

Intuition:

- Create K training and test sets ("folds") within training set.
- For each k in K, run classifier and estimate performance in test set within fold.
- Pick value of λ (regularization parameter) that gives better performance

Cross-validation

Intuition:

- Create K training and test sets ("folds") within training set.
- For each k in K, run classifier and estimate performance in test set within fold.
- Pick value of λ (regularization parameter) that gives better performance
- Train classifier with full dataset and compute performance on test set

Cross-validation

Intuition:

- Create K training and test sets ("folds") within training set.
- For each k in K, run classifier and estimate performance in test set within fold.
- Pick value of λ (regularization parameter) that gives better performance
- Train classifier with full dataset and compute performance on test set
- Why? Avoids overfitting.

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:
  - $P_{rm} = \frac{F_{rm}}{\sum_r F_{rm}} \rightarrow$ (Prob. we are reading r if we observe m)

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:
  - $P_{rm} = \frac{F_{rm}}{\sum_r F_{rm}} \rightarrow$ (Prob. we are reading r if we observe m)
  - Wordscore $S_{md} = \sum_r (P_{rm} \times A_{rd})$

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:
  - $P_{rm} = \frac{F_{rm}}{\sum_r F_{rm}} \rightarrow$ (Prob. we are reading r if we observe m)
  - Wordscore $S_{md} = \sum_r (P_{rm} \times A_{rd})$
- Scores for "virgin" texts:

Wordscores (Laver, Benoit, Garry, 2003, APSR)

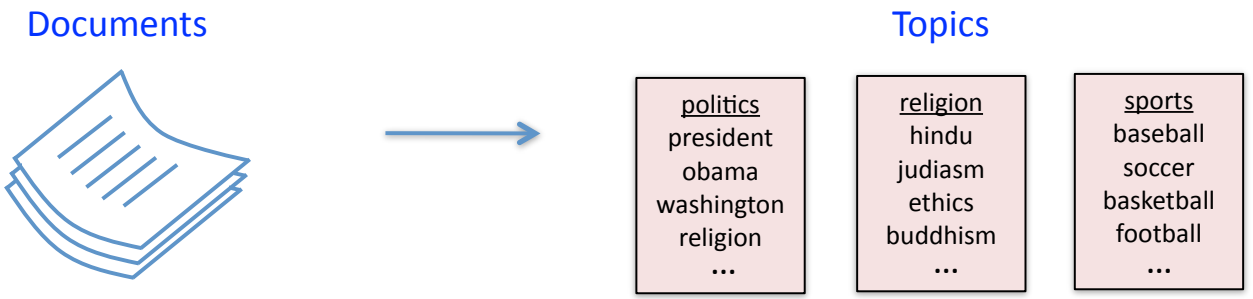
- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:
  - $P_{rm} = \frac{F_{rm}}{\sum_r F_{rm}} \rightarrow$ (Prob. we are reading r if we observe m)
  - Wordscore $S_{md} = \sum_r (P_{rm} \times A_{rd})$
- Scores for "virgin" texts:
  - $S_{vd} = \sum_w (F_{vm} \times S_{md}) \rightarrow$ (weighted average of scored words)

Wordscores (Laver, Benoit, Garry, 2003, APSR)

- Goal: estimate positions on a latent ideological scale
- Data = document-term matrix $\mathbf{W_R}$ for set of "reference" texts, each with known A_{rd}, a policy position on dimension d.
- Compute $\mathbf{F}$, where F_{rm} is relative frequency of word m over the total number of words in document r.
- Scores for individual words:
  - $P_{rm} = \frac{F_{rm}}{\sum_r F_{rm}} \rightarrow$ (Prob. we are reading r if we observe m)
  - Wordscore $S_{md} = \sum_r (P_{rm} \times A_{rd})$
- Scores for "virgin" texts:
  - $S_{vd} = \sum_w (F_{vm} \times S_{md}) \rightarrow$ (weighted average of scored words)
  - $S^*_{vd} = (S_{vd} - \overline{S_{vd}}) \left(\frac{SD_{rd}}{SD_{vd}} \right) + \overline{S_{vd}} \rightarrow$ Rescaled scores.

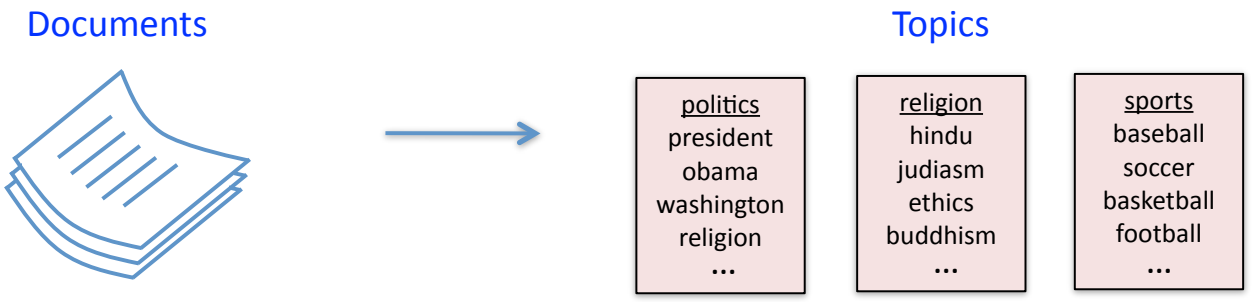
Latent Dirichlet allocation (LDA)

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents

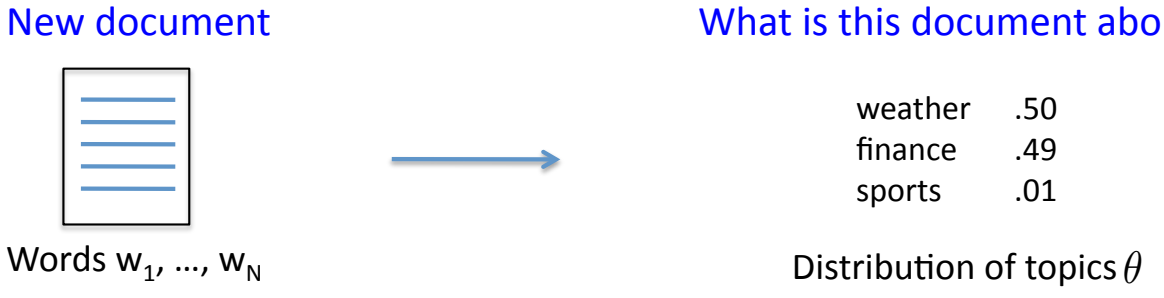


Latent Dirichlet allocation (LDA)

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents

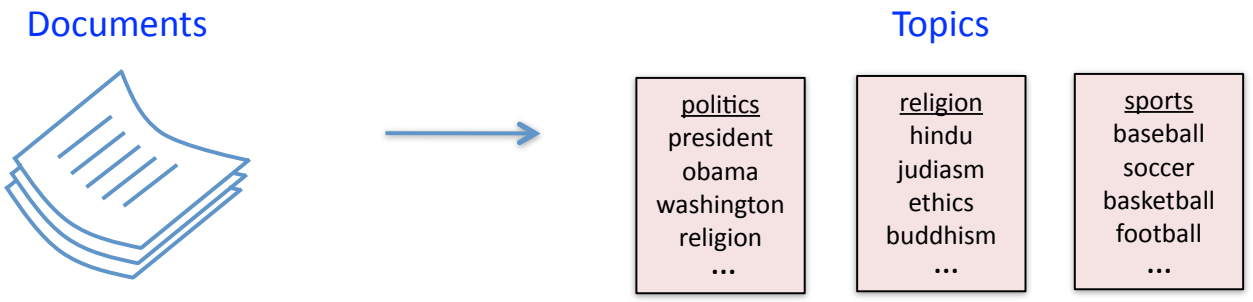


- Many applications in information retrieval, document summarization, and classification

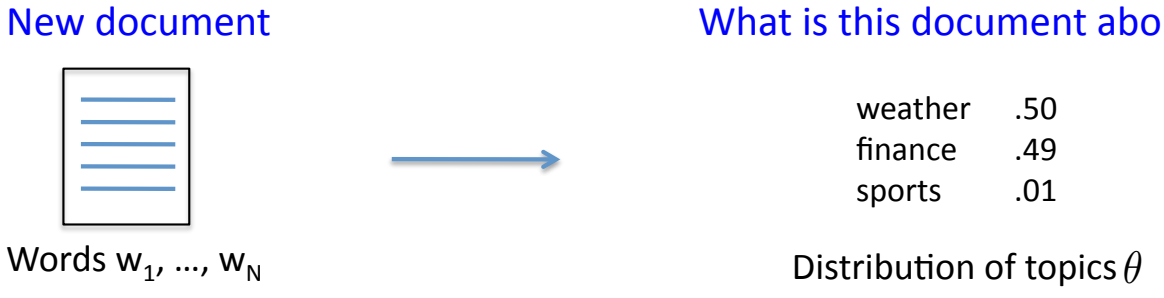


Latent Dirichlet allocation (LDA)

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents



- Many applications in information retrieval, document summarization, and classification



- LDA is one of the simplest and most widely used topic models

Latent Dirichlet Allocation

- Document = random mixture over latent topics
- Topic = distribution over n-grams

Probabilistic model with 3 steps:

1. Choose $\theta_i \sim \text{Dirichlet}(\alpha)$
2. Choose $\beta_k \sim \text{Dirichlet}(\delta)$
3. For each word in document i:
   - Choose a topic $z_m \sim \text{Multinomial}(\theta_i)$
   - Choose a word $w_{im} \sim \text{Multinomial}(\beta_{i,k=z_m})$

where:
α=parameter of Dirichlet prior on distribution of topics over docs.
θ_i=topic distribution for document i
δ=parameter of Dirichlet prior on distribution of words over topics
β_k=word distribution for topic k

Plate notation

Plate notation

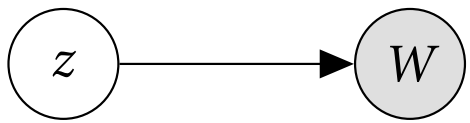


Plate notation

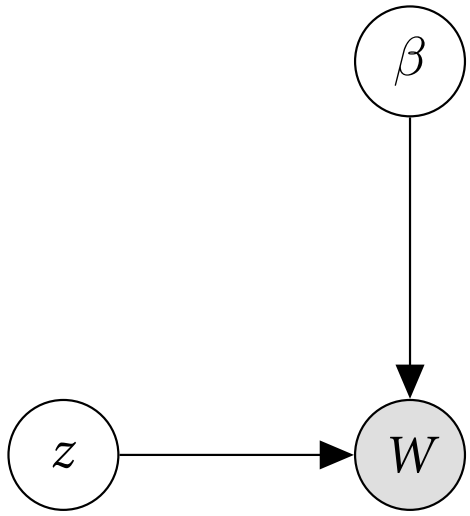


Plate notation

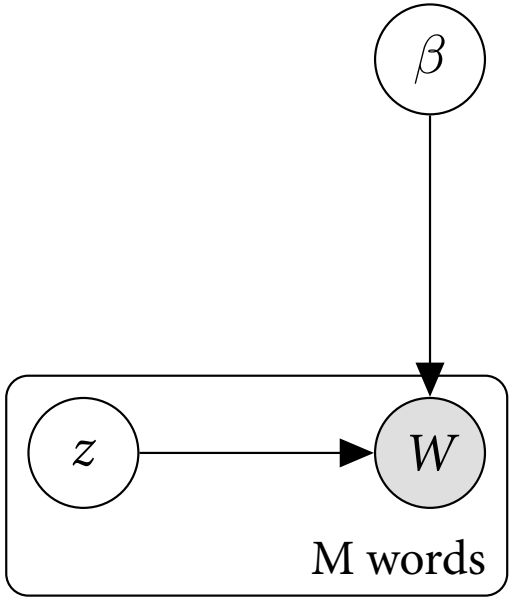


Plate notation

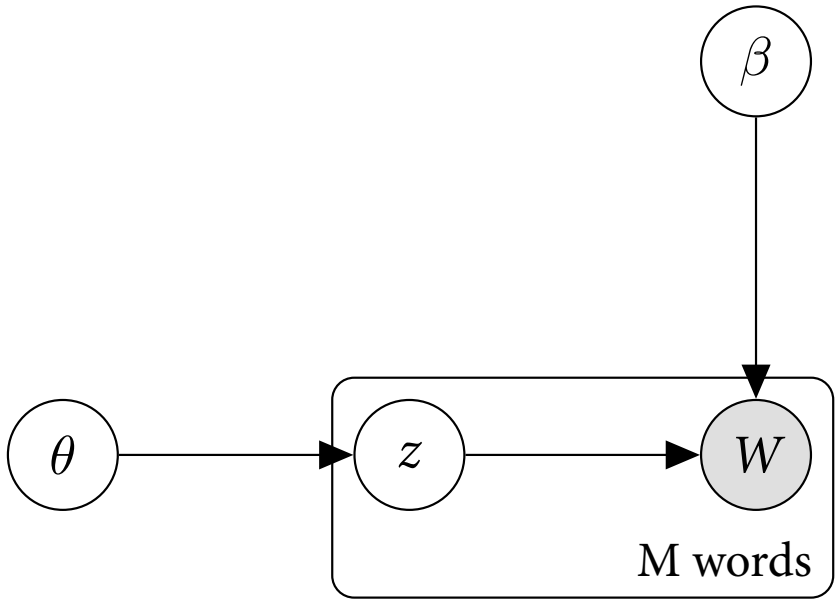


Plate notation

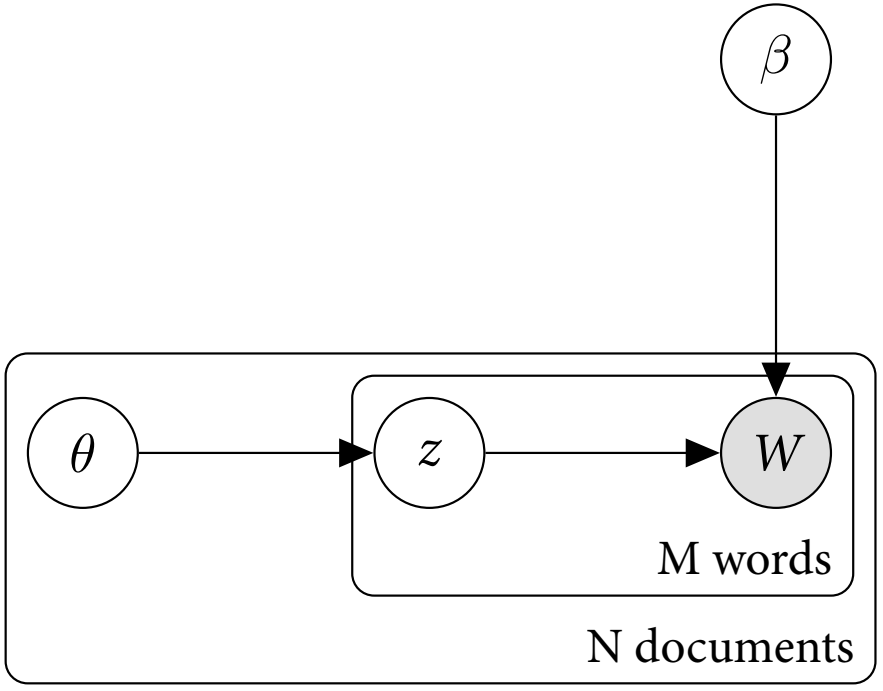
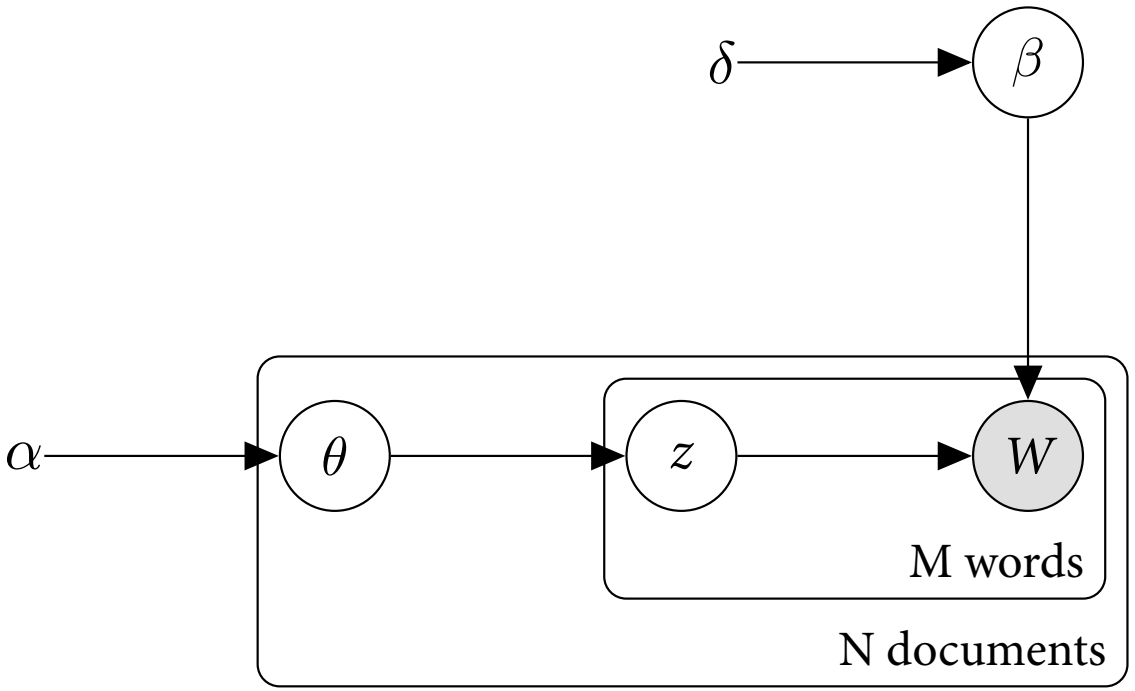


Plate notation



Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?
   - Do they depart from existing measures where they should depart?

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?
   - Do they depart from existing measures where they should depart?
3. Predictive validity

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?
   - Do they depart from existing measures where they should depart?
3. Predictive validity
   - Does variation in topic usage correspond with expected events?

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?
   - Do they depart from existing measures where they should depart?
3. Predictive validity
   - Does variation in topic usage correspond with expected events?
4. Hypothesis validity

Validation

From Quinn et al, AJPS, 2010:

1. Semantic validity
   - Do the topics identify coherent groups of tweets that are internally homogenous, and are related to each other in a meaningful way?
2. Convergent/discriminant construct validity
   - Do the topics match existing measures where they should match?
   - Do they depart from existing measures where they should depart?
3. Predictive validity
   - Does variation in topic usage correspond with expected events?
4. Hypothesis validity
   - Can topic variation be used effectively to test substantive hypotheses?

Eexample: topics in US legislators' tweets

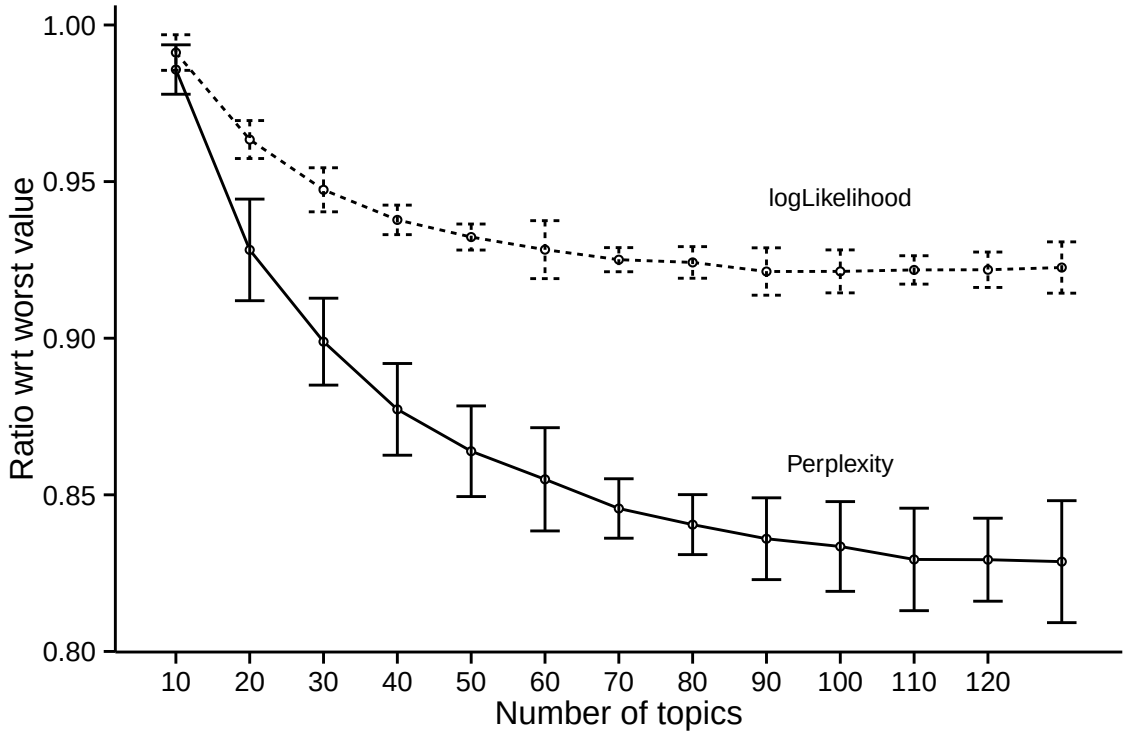
- Data: 651,116 tweets sent by US legislators from January 2013 to December 2014, collected by the SMaPP lab.
- 2,920 documents = 730 days $\times$ 2 chambers $\times$ 2 parties
- Why aggregating? Applications that aggregate by author or day outperform tweet-level analyses (Hong and Davidson, 2010)
- $K = 100$ topics (more on this later)
- Validation: http://j.mp/lda-congress-demo

Choosing the number of topics

- Choosing K is "one of the most difficult questions in unsupervised learning" (Grimmer and Stewart, 2013, p.19)

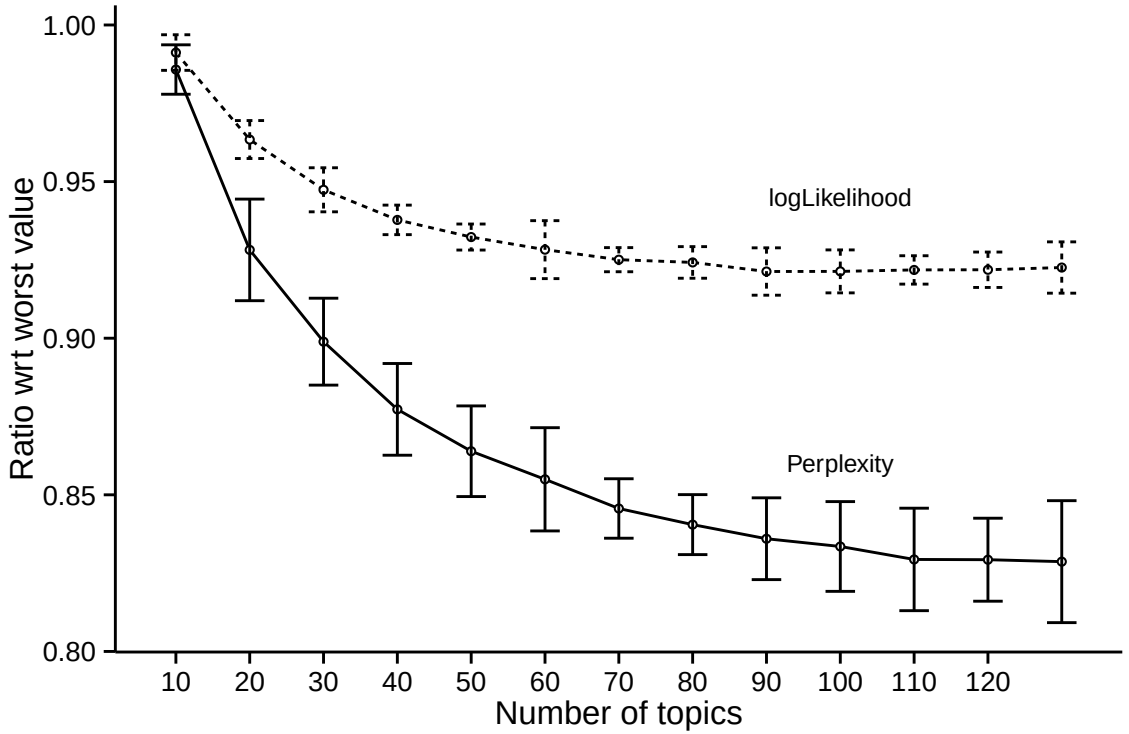
Choosing the number of topics

- Choosing K is "one of the most difficult questions in unsupervised learning" (Grimmer and Stewart, 2013, p.19)
- We chose $K = 100$ based on cross-validated model fit.



Choosing the number of topics

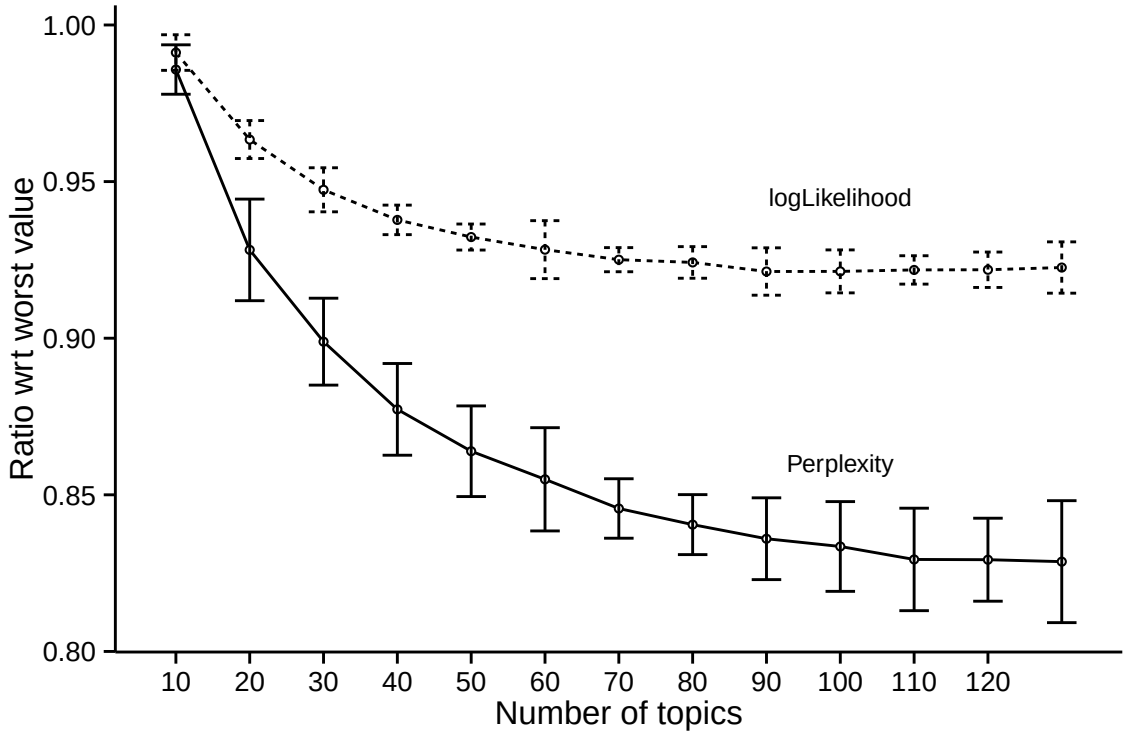
- Choosing K is "one of the most difficult questions in unsupervised learning" (Grimmer and Stewart, 2013, p.19)
- We chose $K = 100$ based on cross-validated model fit.



- **BUT**: "there is often a negative relationship between the best-fitting model and the substantive information provided".

Choosing the number of topics

- Choosing K is "one of the most difficult questions in unsupervised learning" (Grimmer and Stewart, 2013, p.19)
- We chose $K = 100$ based on cross-validated model fit.



- **BUT**: "there is often a negative relationship between the best-fitting model and the substantive information provided".
- GS propose to choose K based on "substantive fit."

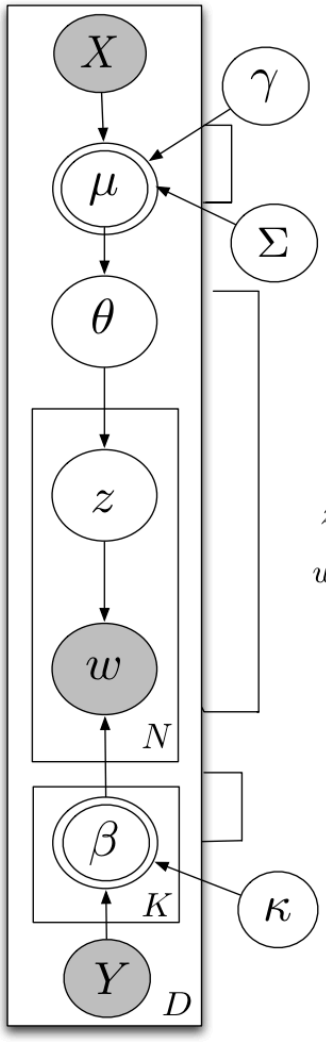
Extensions of LDA

1. Structural topic model (Roberts et al, 2014, AJPS)
2. Dynamic topic model (Blei and Lafferty, 2006, ICML; Quinn et al, 2010, AJPS)
3. Hierarchical topic model (Griffiths and Tenembaun, 2004, NIPS; Grimmer, 2010, PA)

Why?

- Substantive reasons: incorporate specific elements of DGP into estimation
- Statistical reasons: structure can lead to better topics.

Structural topic model



Topic Prevalence:

$$\mu_{d,k} = X_d\gamma_k$$
$$\gamma_k \sim \mathcal{N}(0, \sigma_k^2)$$
$$\sigma_k^2 \sim \text{Gamma}(s^\gamma, r^\gamma)$$

Language Model:

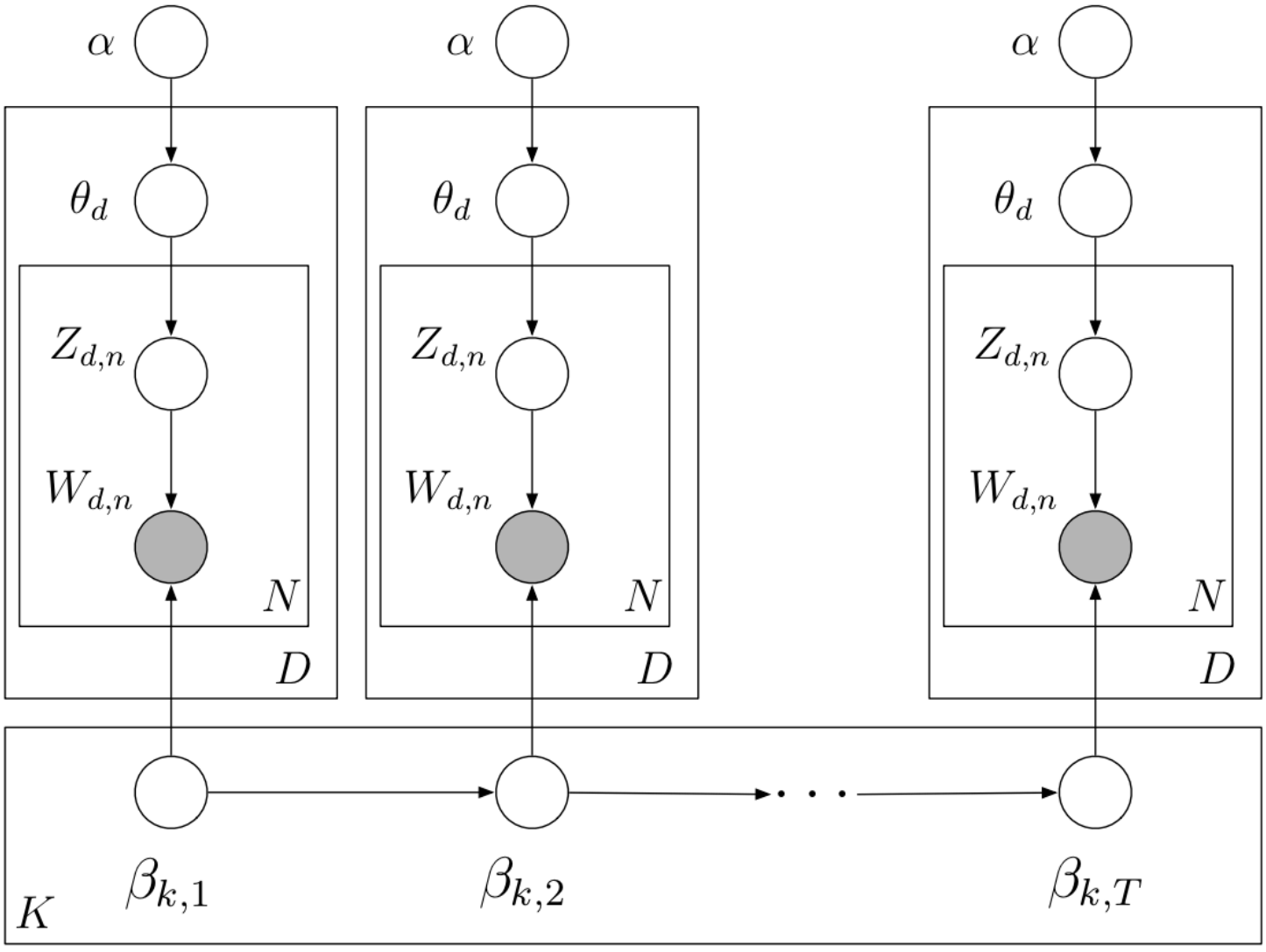
$$\theta_d \sim \text{LogisticNormal}(\mu_d, \Sigma)$$
$$z_{d,n} \sim \text{Mult}(\theta_d)$$
$$w_{d,n} \sim \text{Mult}(\beta_d^{k=z_{d,n}})$$

Topical Content:

$$\beta_{d,v}^k \propto \exp(m_v + \kappa_v^{\cdot,k} + \kappa_v^{y,\cdot} + \kappa_v^{y,k})$$
$$\kappa_v^{y,k} \sim \text{Laplace}(0, \tau_v^{y,k})$$
$$\tau_v^{y,k} \sim \text{Gamma}(s^\kappa, r^\kappa)$$

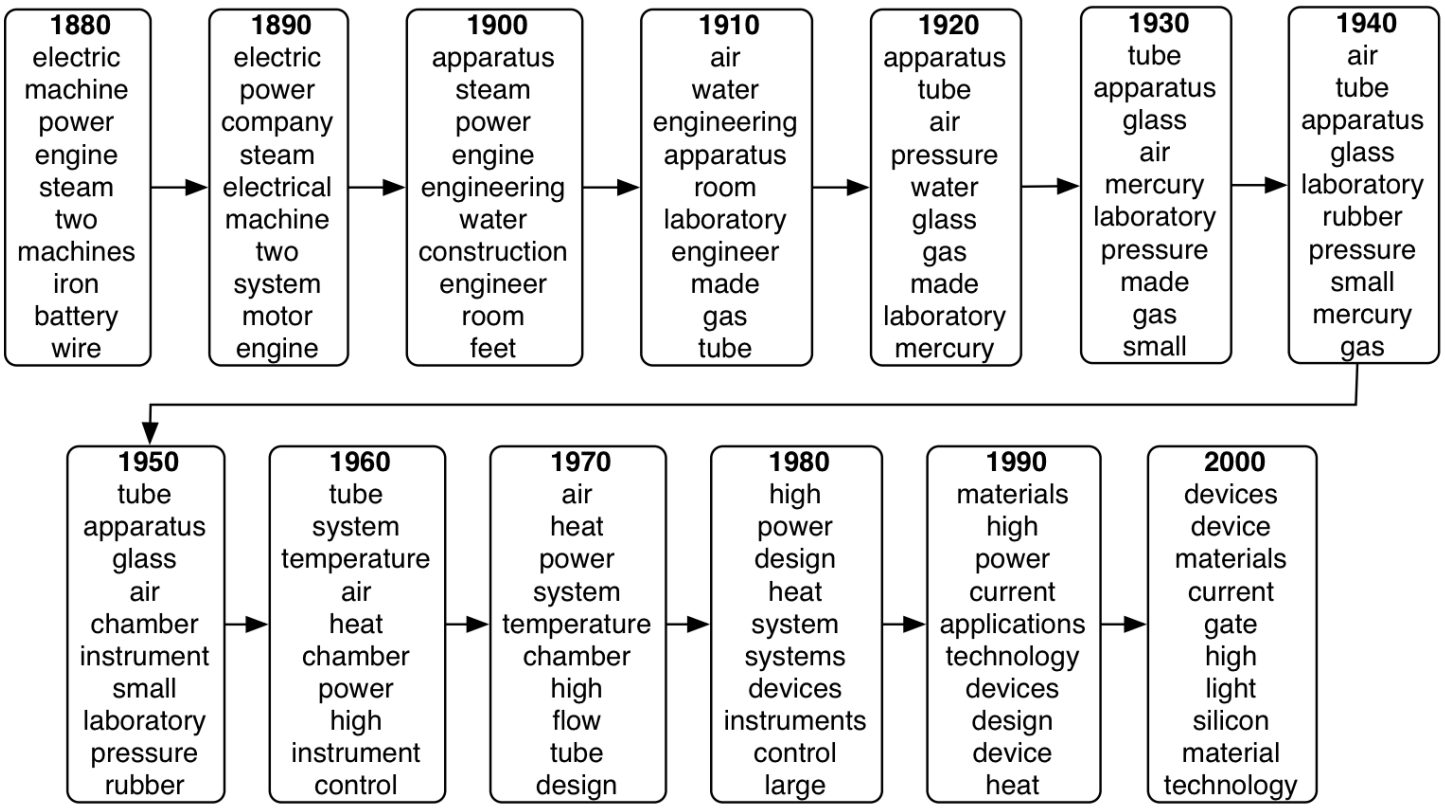
- **Prevalence**: Prior on the mixture over topics is now document-specific, and can be a function of covariates (documents with similar covariates will tend to be about the same topics)
- **Content**: distribution over words is now document-specific and can be a function of covariates (documents with similar covariates will tend to use similar words to refer to the same topic)

Dynamic topic model



Source: Blei, "Modeling Science"

Dynamic topic model



Source: Blei, "Modeling Science"

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m
  β_m is discrimination parameter of word m

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m
  β_m is discrimination parameter of word m

- Estimation using EM algorithm.

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m
  β_m is discrimination parameter of word m

- Estimation using EM algorithm.
- Identification:

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m
  β_m is discrimination parameter of word m

- Estimation using EM algorithm.
- Identification:
  - Unit variance restriction for θ_i

Wordfish (Slapin and Proksch, 2008, AJPS)

- Goal: unsupervised scaling of ideological positions
- Ideology of politician i, θ_i is a position in a latent scale.
- Word usage is drawn from a Poisson-IRT model:

$$W_{im} \sim \text{Poisson}(\lambda_{im})$$
$$\lambda_{im} = exp(\alpha_i + \psi_m + \beta_m \times \theta_i)$$

- where:

  α_i is "loquaciousness" of politician i
  ψ_m is frequency of word m
  β_m is discrimination parameter of word m

- Estimation using EM algorithm.
- Identification:
  - Unit variance restriction for θ_i
  - Choose a and b such that $\theta_a > \theta_b$