Structural topic models for enriching quantitative text analysis

Carsten Schwemmer, U of Bamberg
Cornelius Puschmann, Hans Bredow Institute

July 17th, 2019
IC²S², Amsterdam

RStudio Notebook:
t1p.de/stm-ic2s2

About us

Carsten Schwemmer https://www.carstenschwemmer.com/ @c_schwemmer

Carsten is a PhD candidate in CSS and lecturer for the Chair of Political Sociology at the University of Bamberg. He is interested in NLP, data mining and the development of research software.

Cornelius Puschmann http://cbpuschmann.net/ @cbpuschmann

Cornelius is a Senior Researcher at the Leibniz Institute for Media Research who studies online hate speech and the role of algorithms for the selection of media content.

About you

What's your background?

- R or Python?
- PhD student, postdoc, faculty?
- Social science, computer science, other fields?
- Prior experience with topic modeling?

Structure of this tutorial

1. Refresher on formal background of topic models (Cornelius)
2. Considerations for preprocessing and feature selection (Cornelius)
3. Introducing structural topic models and parameter tuning (Cornelius)
4. Model validation and interactively exploring STM models (Carsten)
5. Estimating and interpreting STM prevalence and content effects (Carsten)
6. Open coding session (Carsten/Cornelius)

Code and data for this tutorial

Code

- stm_ic2s2 Github contains an RStudio Notebook
  - either run the code as we move through Carsten's demonstration
  - or follow along the HTML version
- R libraries: *tidyverse, stm, stminsights, quanteda, rmarkdown*

Data

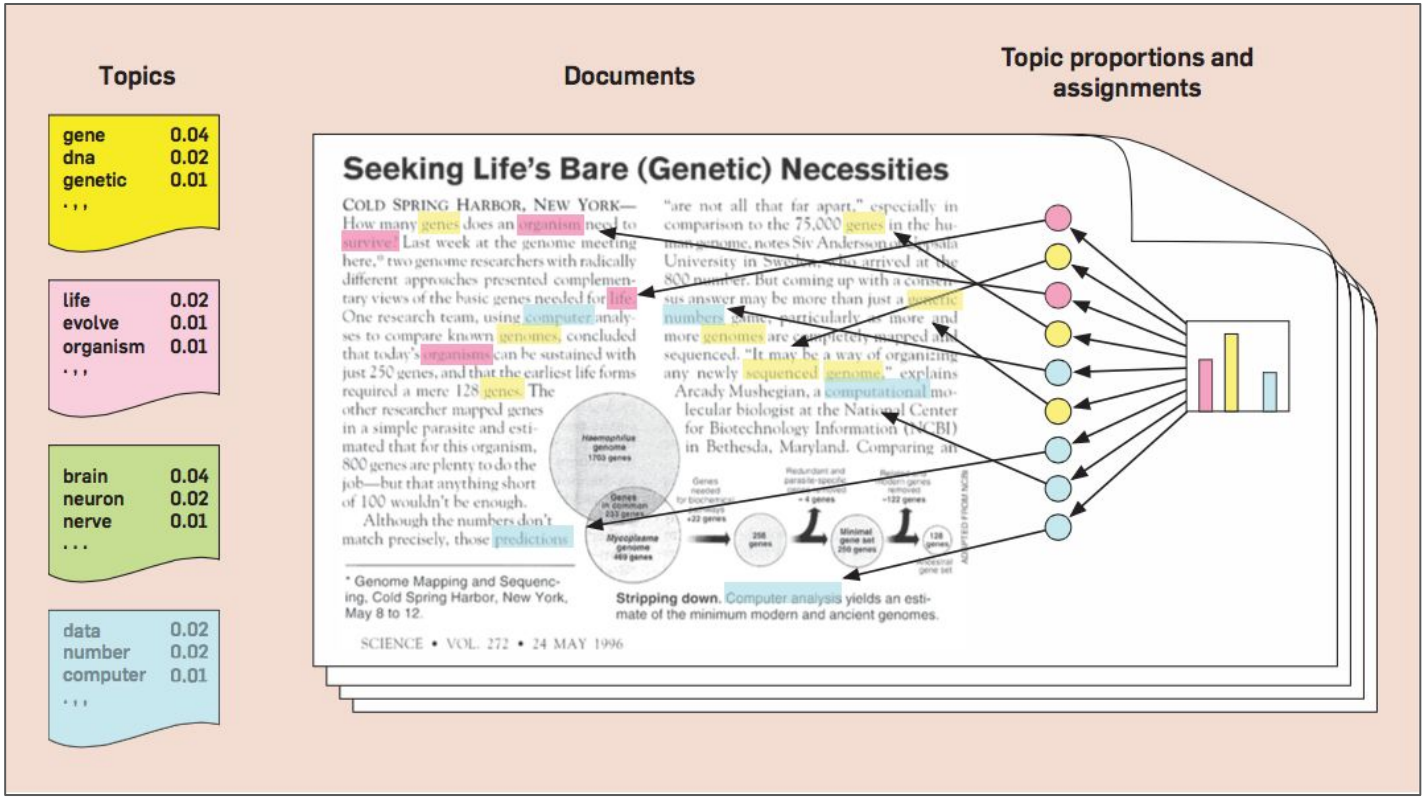
- DonorsChoose.org dataset from Kaggle: https://www.kaggle.com/c/donorschoose-application-screening (we use an abridged version)

DonorsChoose.org

Support a classroom. Build a future.

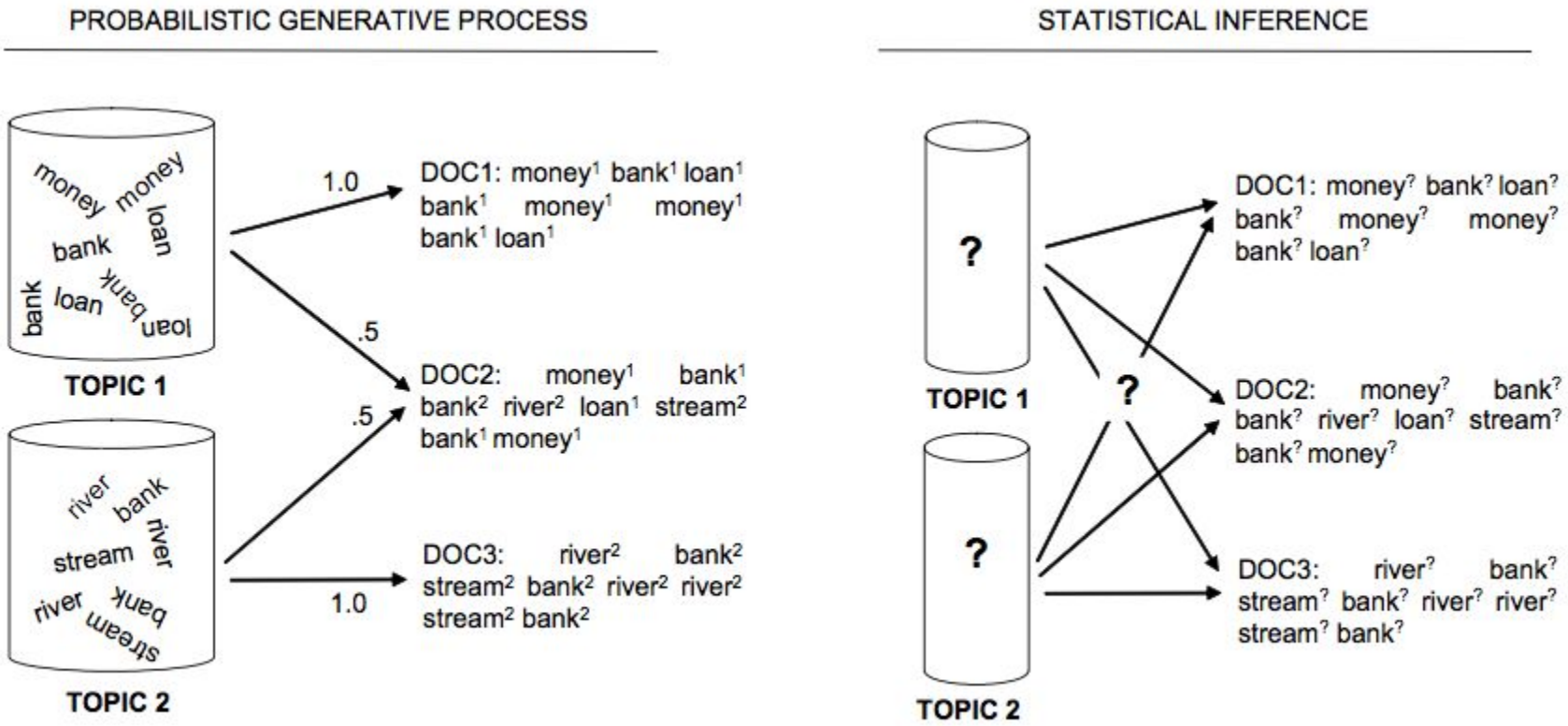
Theoretical Background: How do topic models work?

Latent Dirichlet Allocation



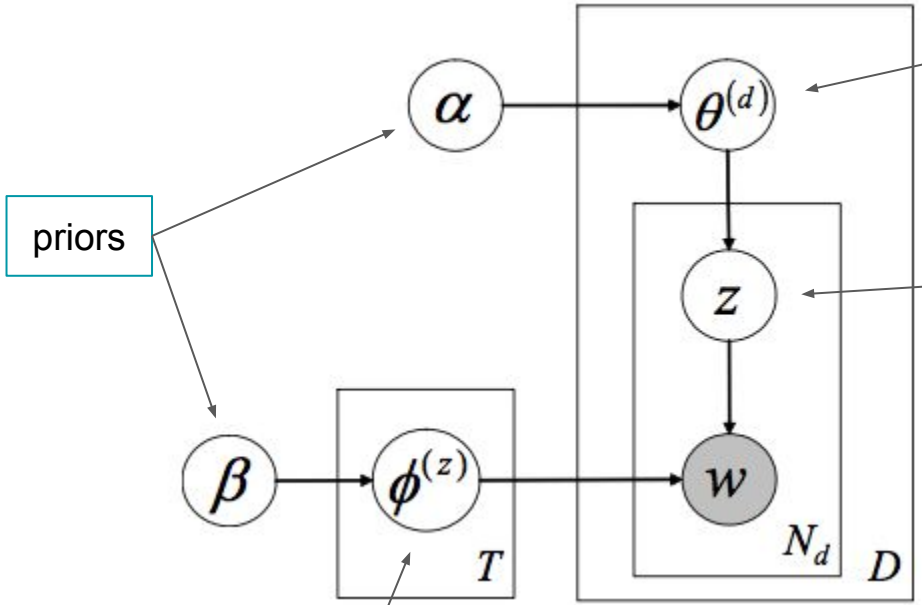
(Blei 2012, Comm. ACM)

Topics: Generation vs. Inference



(Steyvers & Griffiths 2007)

Graphical model



topic mixture per document

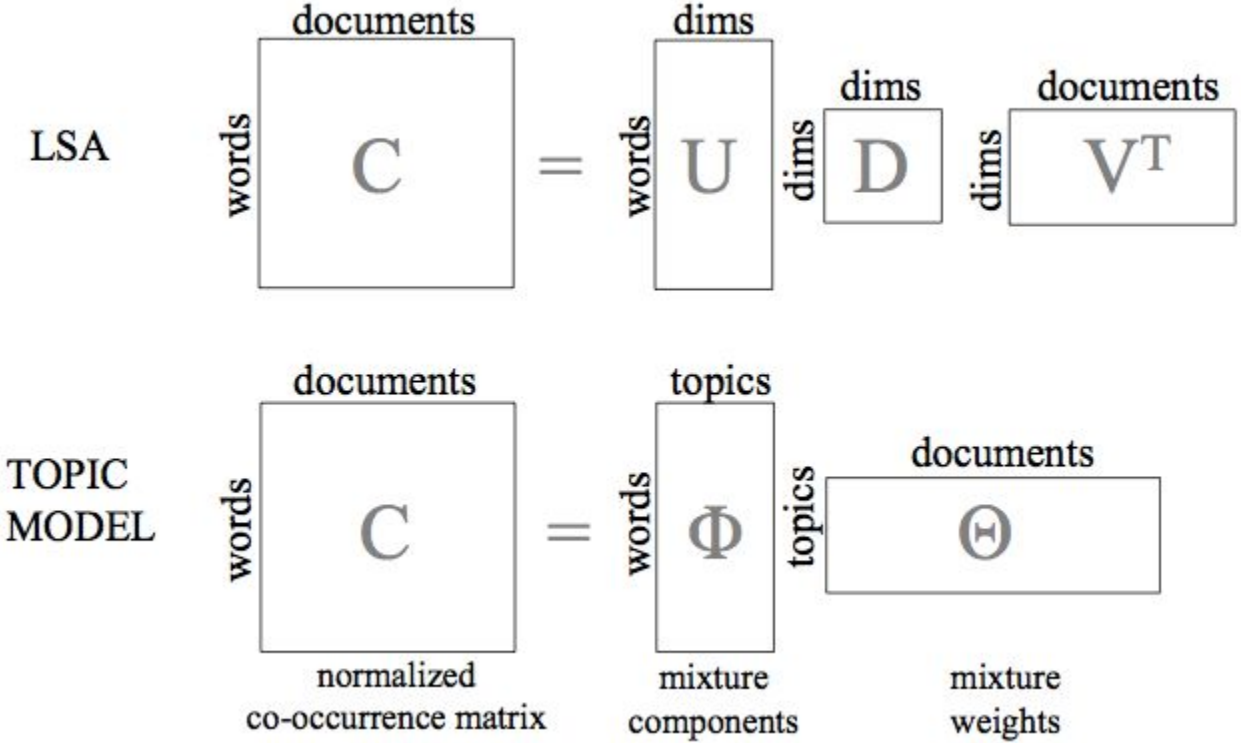
priors

topic

topics (= distributions over words)

(Steyvers & Griffiths 2007)

Matrix factorization



(Steyvers & Griffiths 2007)

Learning an LDA topic model

Idea:

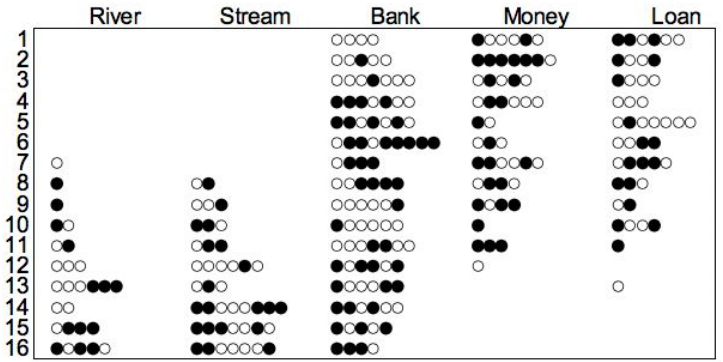
- Estimate the posterior distribution *z* (of words to topics) by *Gibbs sampling*
- Then approximate ϕ (PHI) and θ (THETA)

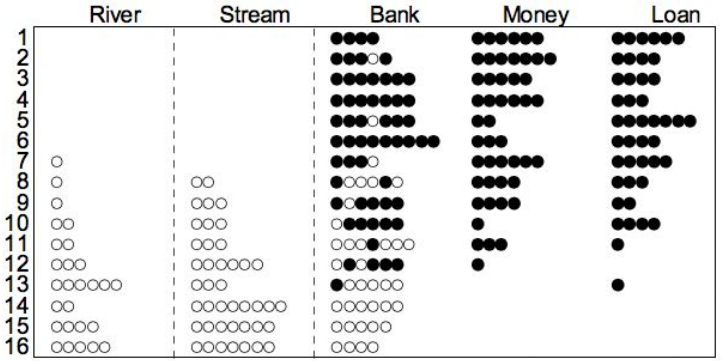
Procedure:

- Randomly assign words to topics
- Consider each word w_i in turn; determine a probability for each topic, given all the other words and topics in the corpus; pick a new topic for w_i
- Finally, obtain ϕ (PHI) and θ (THETA) from word-topic and topic-document count matrices

Example

- 16 documents (generated from 2 topics)
- Random assignment of words to topics (top panel)
- After 64 iterations, topic structure becomes clear (bottom panel)





(Steyvers & Griffiths 2007)

From text to features: preprocessing, tokens, n-grams

From text to features: Let’s think about it!

A text is a long string of characters, but we want numerical features

- Represent each text as a vector of word frequencies (“bag of words”) → term-document matrix

However:

- Not all words are important (stopword removal)
- Some words might be more important than others (raw frequencies vs. tf-idf scores)
- Words that are *almost* the same shouldn’t be treated differently (case, punctuation, stemming)
- What about two-word expressions like ‘White House’? (ngrams)

Preprocessing your data

- Stop word removal is typically an aspect of preprocessing
- Other steps may include
  - removing punctuation, numbers, separators, symbols, URLs, ...
  - tokenization
  - stemming
  - tagging
  - parsing
- Short stop word lists may include only a handful of high frequency terms (*the, to, and, of, ...*) extensive ones may include 200-300 terms (if you go far beyond this, you should start thinking about a thesaurus)

Why remove stopwords?

- The underlying assumption when removing tokens is that documents contain “noise”, i.e. material that is not conducive to the analysis
- But it is generally difficult for humans to anticipate in advance which words will be important for
- Words in a document != words in isolation
- Words in a document can be
  - semantically meaningful
  - reliably indicate a particular speaker or context

Information compression

Determining what goes onto your stop word list

- Usually these are function words such as articles, pronouns and conjunctions
- Nouns verbs and adjectives are typically the word classes that are retained
- It may also be favorable to remove very high frequency common nouns, as well as search terms that you used to generate a corpus
- Stop words can be filtered relying on a list which may be compiled in several ways:
    - Using a manually compiled list, or list from the Web
    - Using a corpus of common English words such as COCA or Web 1T 5-gram
    - Using heuristic procedures such as TDF-IF

Caveats of manual lists

The main drawback of manually compiled lists is that they are insensitive to frequency variation between corpora

- Humans are bad at guessing which words are highly frequent and normally distributed
- Humans are also bad at guessing what particular words may be meaningful in a given context

This problem doesn’t go away when using standard language corpora (COCA, Web 1T 5-gram) because your corpus may be different

TF-IDF

- *Term frequency–inverse document frequency* (TF-IDF, Spärck Jones, 1972) is a metric commonly used in information retrieval (IF), for example in recommender systems
- Among the most popular term-weighting schemes
- Weighs the frequency of a term within a particular document (IDF) in relation to its frequency within the entire corpus (TF)

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

$$\mathrm{tf}(''\mathsf{example}'', d_1) = \frac{0}{5} = 0$$

$$\mathrm{tf}(''\mathsf{example}'', d_2) = \frac{3}{7} \approx 0.429$$

$$\mathrm{idf}(''\mathsf{example}'', D) = \log\left(\frac{2}{1}\right) = 0.301$$

$$\mathrm{tfidf}(''\mathsf{example}'', d_1) = \mathrm{tf}(''\mathsf{example}'', d_1) \times \mathrm{idf}(''\mathsf{example}'', D) = 0 \times 0.301 = 0$$

$$\mathrm{tfidf}(''\mathsf{example}'', d_2) = \mathrm{tf}(''\mathsf{example}'', d_2) \times \mathrm{idf}(''\mathsf{example}'', D) = 0.429 \times 0.301 \approx 0.13$$

Alternative: Use a dictionary or thesaurus

- When using a dictionary or thesaurus, all words in your corpus are collapsed into a particular lemma
- This may be useful for hyponymous relations (*France, Spain, US* = *country*) or when a particular word field identifies a concept reliably (see *Christmas* in the example)
- Popular examples: Lexicoder Policy Areas, LIWC, WordStat

```
mycorpus <- corpus_subset(data_corpus_inaugural, Year>1900)
mydict <- dictionary(list(christmas = c("Christmas", "Santa", "holiday"),
                          opposition = c("Opposition", "reject", "notincorpus"),
                          taxing = "taxing",
                          taxation = "taxation",
                          taxregex = "tax*",
                          country = "america"))
head(dfm(mycorpus, dictionary = mydict))
#> Document-feature matrix of: 30 documents, 6 features (71.7% sparse).
#> (showing first 6 documents and first 6 features)
#>                 christmas opposition taxing taxation taxregex country
#> 1901-McKinley           0          2      0        1        1       0
#> 1905-Roosevelt          0          0      0        0        0       0
#> 1909-Taft               0          1      0        4        6       4
#> 1913-Wilson             0          0      0        1        1       0
#> 1917-Wilson             0          0      0        0        0       2
#> 1921-Harding            0          0      0        1        2      15
```

Tuning of the topic number for optimal model fit

What is the optimal k for a given topic model?

There is no definitive single answer, as topics in topic modeling are generative

More topics = **fine-grained analysis**

- Pro: Able to capture “blips” in the data, such as a particular event in a social media corpus
- Con: Lack of focus
- Con: Topics tend to be redundant

Fewer topics = **coarse analysis**

- Pro: Able to capture “broad strokes”, such as recurring themes in a news corpus
- Con: Lack of detail
- Con: Lack of nuance

Assessing similarity by clustering/metric comparison

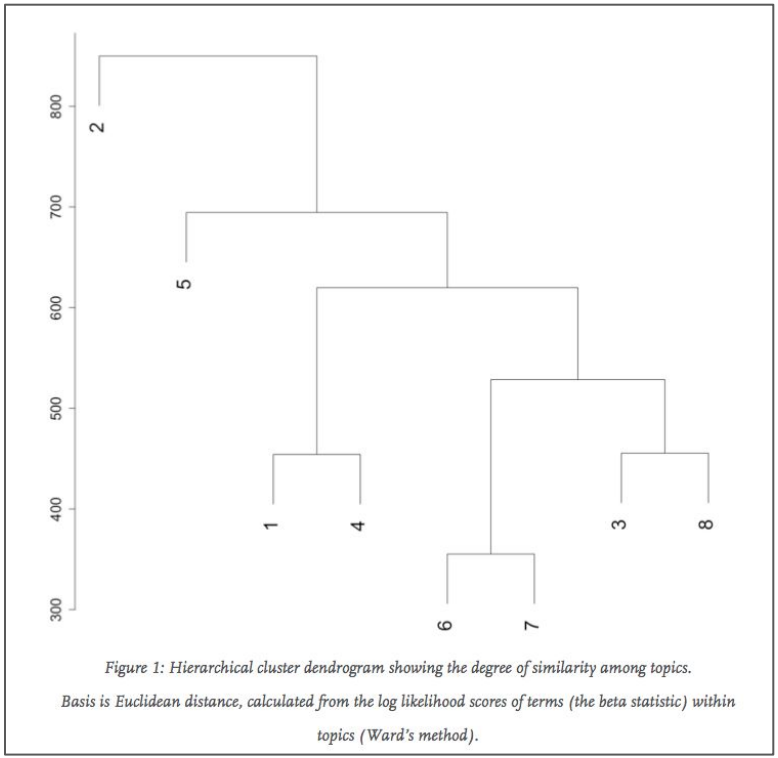


Figure 1: Hierarchical cluster dendrogram showing the degree of similarity among topics. Basis is Euclidean distance, calculated from the log likelihood scores of terms (the beta statistic) within topics (Ward's method).

(Puschmann & Scheffler, 2016, p. 9)

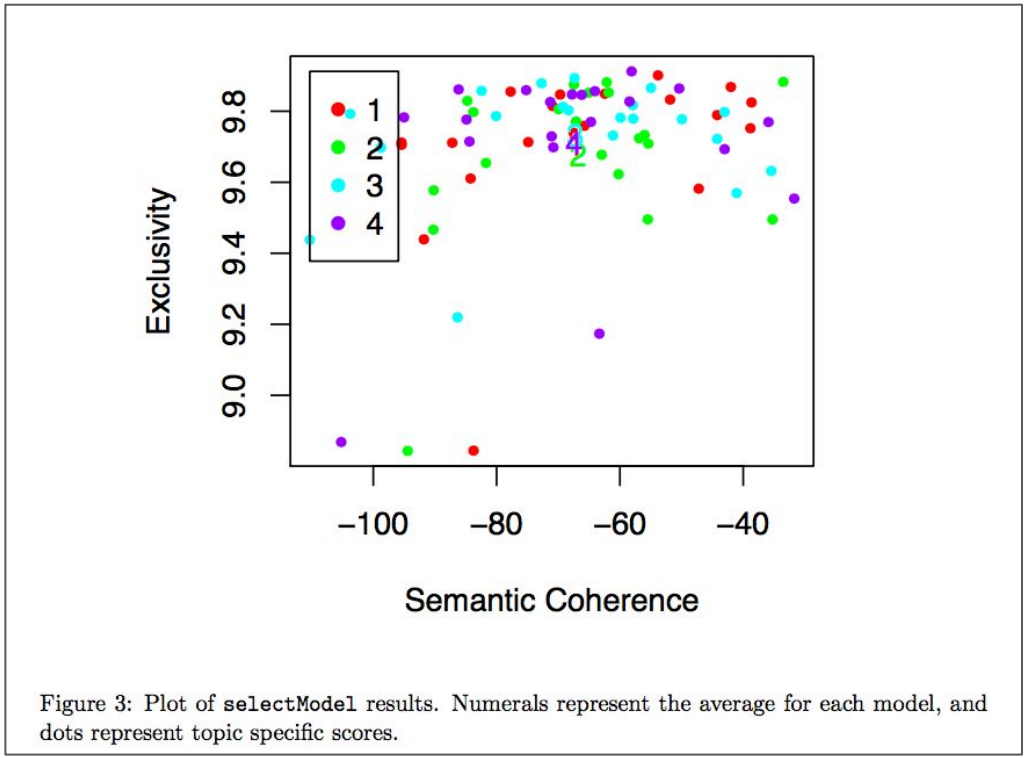
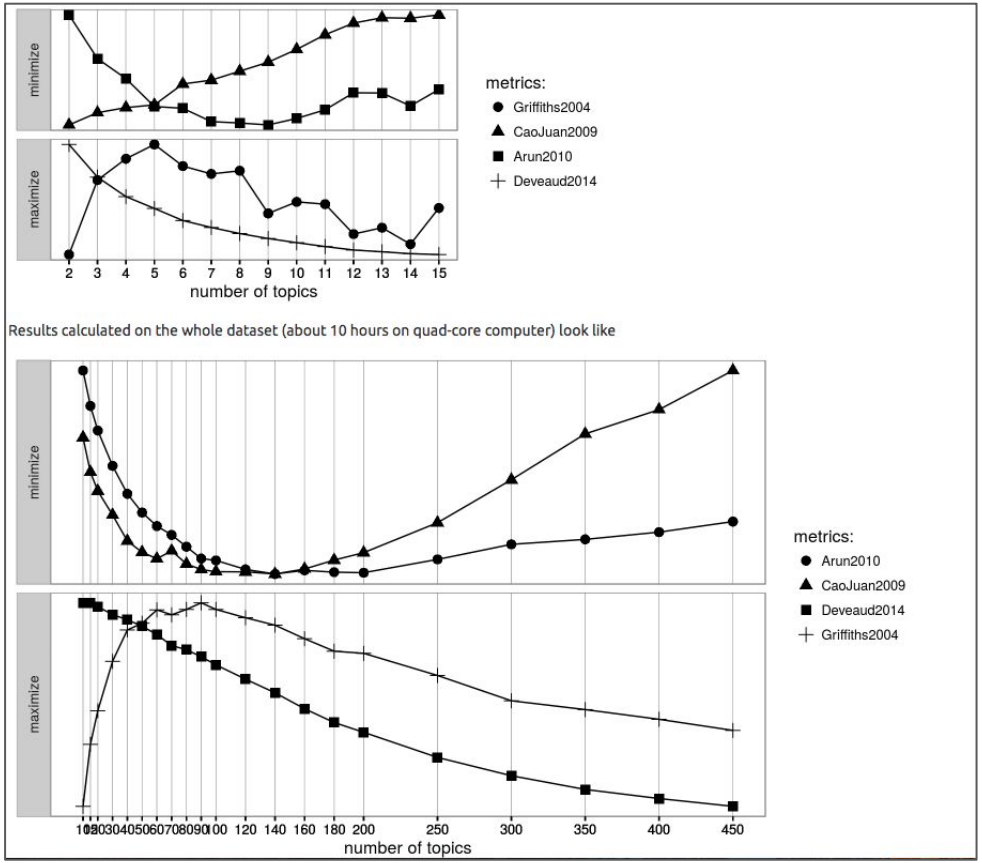


Figure 3: Plot of `selectModel` results. Numerals represent the average for each model, and dots represent topic specific scores.

(Roberts, Stewart & Tingley, 2016, p. 13)

Heuristics for picking optimal *k* in LDAtuning for R

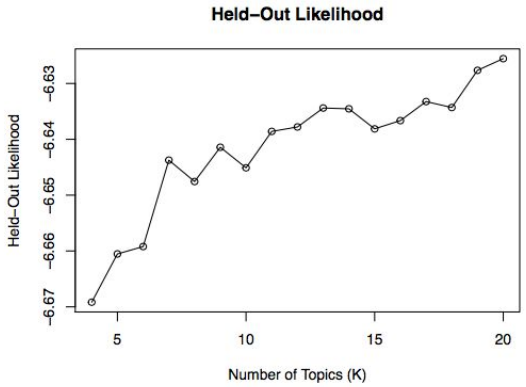
- Package ldatuning for R developed by Nikita Murzintcev
- Compatible with popular *topicmodels* package
- Implements four different metrics for determining optimal *k,* papers describing metrics are conveniently provided with the package

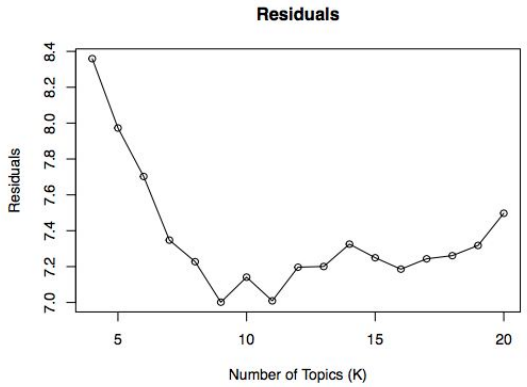


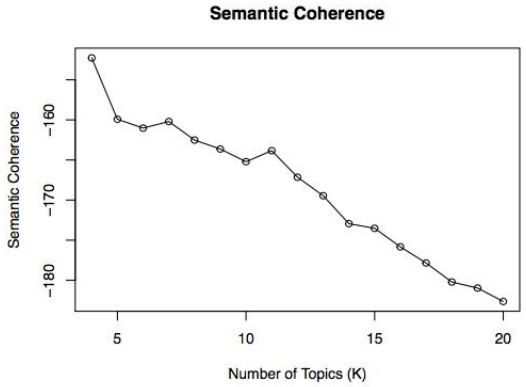
Heuristics for picking optimal *k* in STM for R

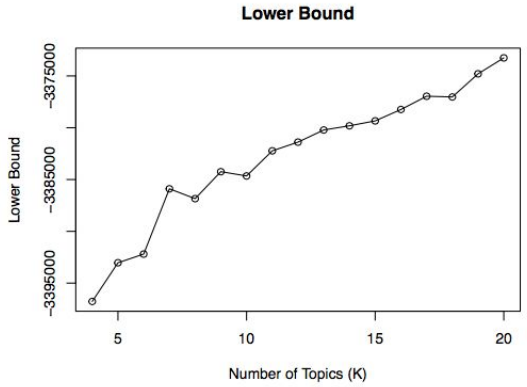
Another set of metrics is provided with the STM package for R:

1. held out likelihood (Wallach et al. 2009)
2. residual analysis (Taddy 2012)
3. semantic coherence (Mimno et al. 2011)
4. lower bound convergence (Roberts et al. 2016)









What is the optimal *k* for a given topic model?

- There is no single correct answer to the question of how many topics to model for a given collection (Grimmer & Stewart 2013; Roberts, Stewart & Tingley, 2016)
- Left and right plateaus in the metric distributions seem favorable choices (left = course analysis, right = fine-grained analysis)
- Further alternatives
  - Model a large number of topics but discard some as “junk”
  - Distinguish between topics and issues/themes, the latter of which are collections of topics
  - Let humans judge the validity of topics (Stier et al, 2017)

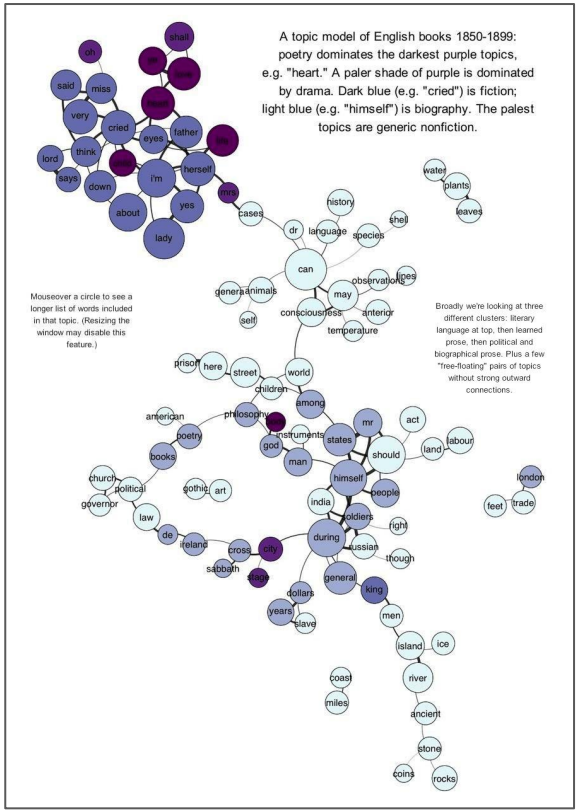
Visualizing and interactively exploring topic models

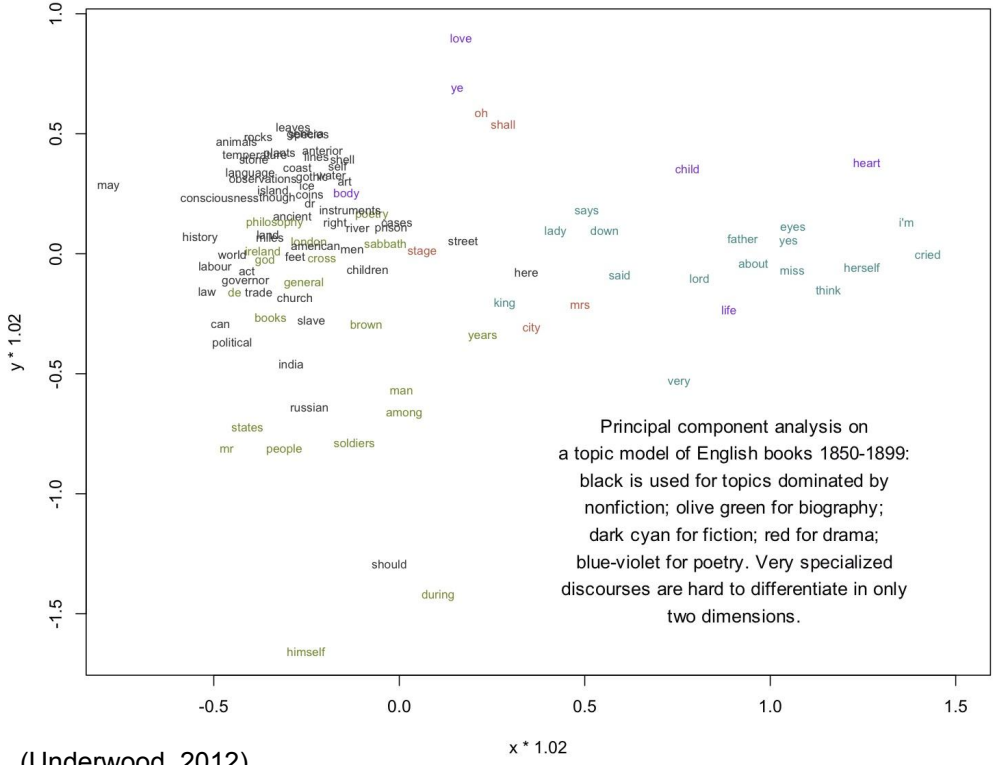
Advantages of applying visualization to topic models

- Topic models are difficult to interpret for humans in purely statistical terms
- Interpretation through keyword lists (usually terms with a high likelihood of association) are also limited, because the strength of topic models lies in their ability to discriminate (rather than to describe)
- Visualization can among other things be used to better understand:
  - Topic similarity
  - Document similarity (*)
  - Topic share distributions
  - Topic-document (*) contrasts

    * can be either a document or metadata describing the document

Topic similarity -- network and PCA/MDS of topics





(Underwood, 2012)

Topic intensity over time and topic explorer

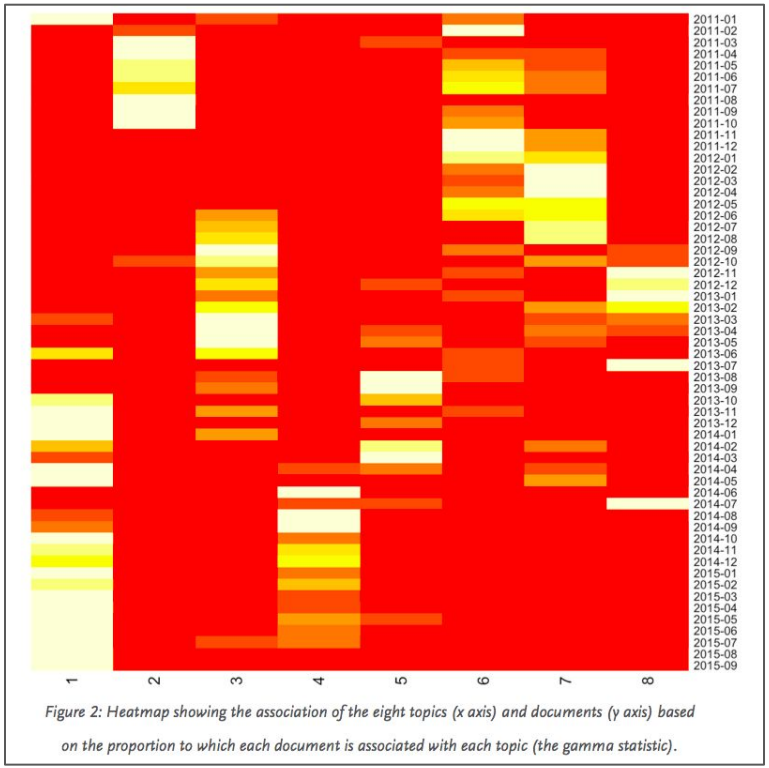
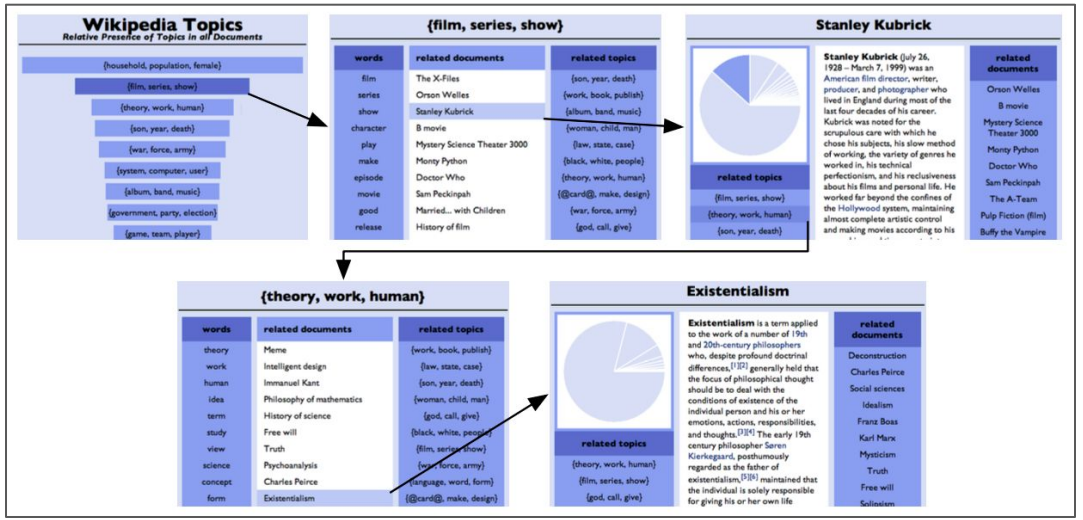


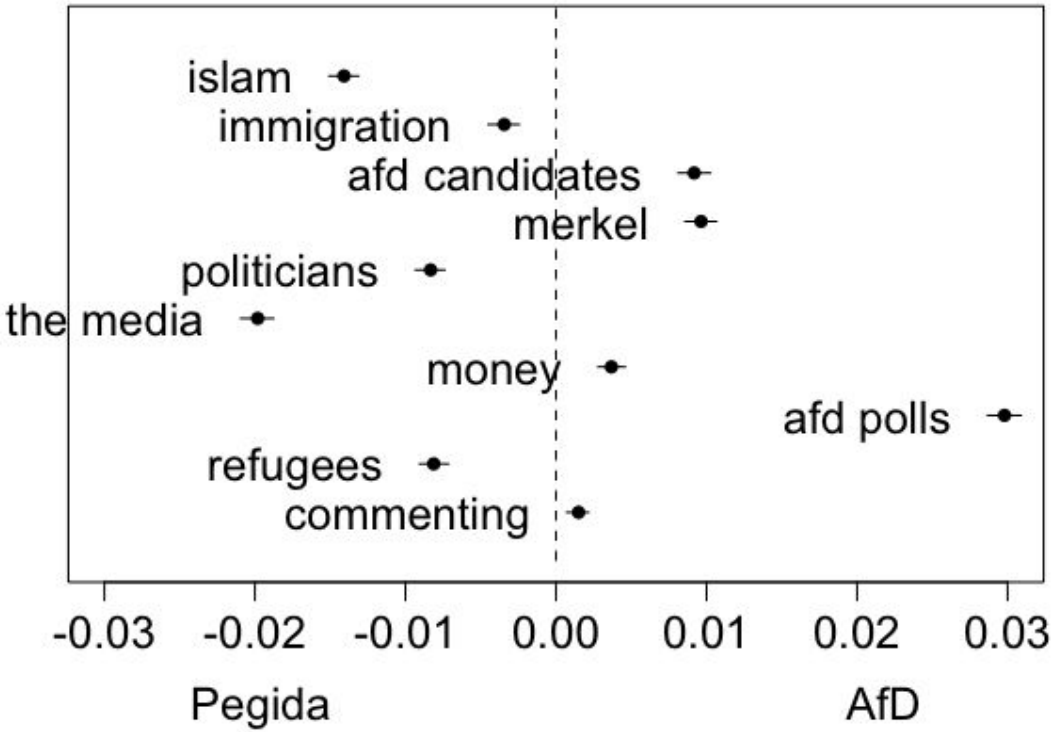
Figure 2: Heatmap showing the association of the eight topics (x axis) and documents (y axis) based on the proportion to which each document is associated with each topic (the gamma statistic).

(Puschmann & Scheffler, 2016, p. 9)

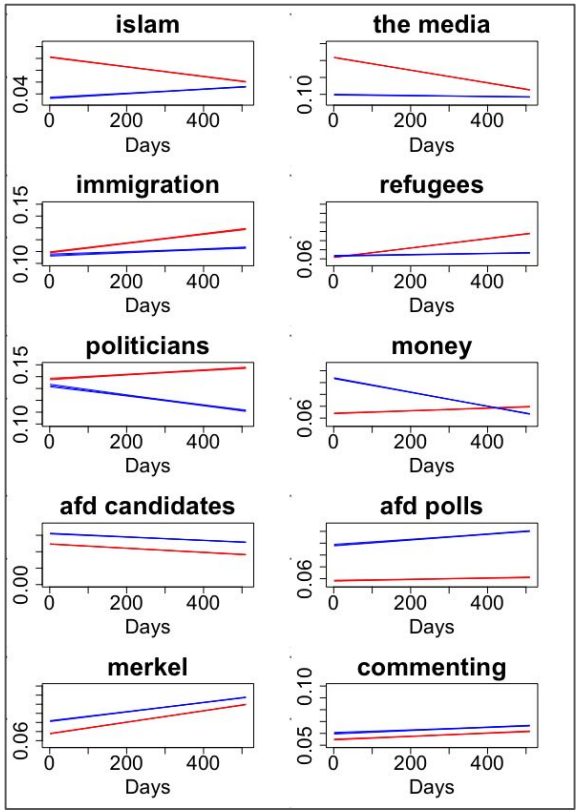


(Chaney & Blei, 2012, p. 420)

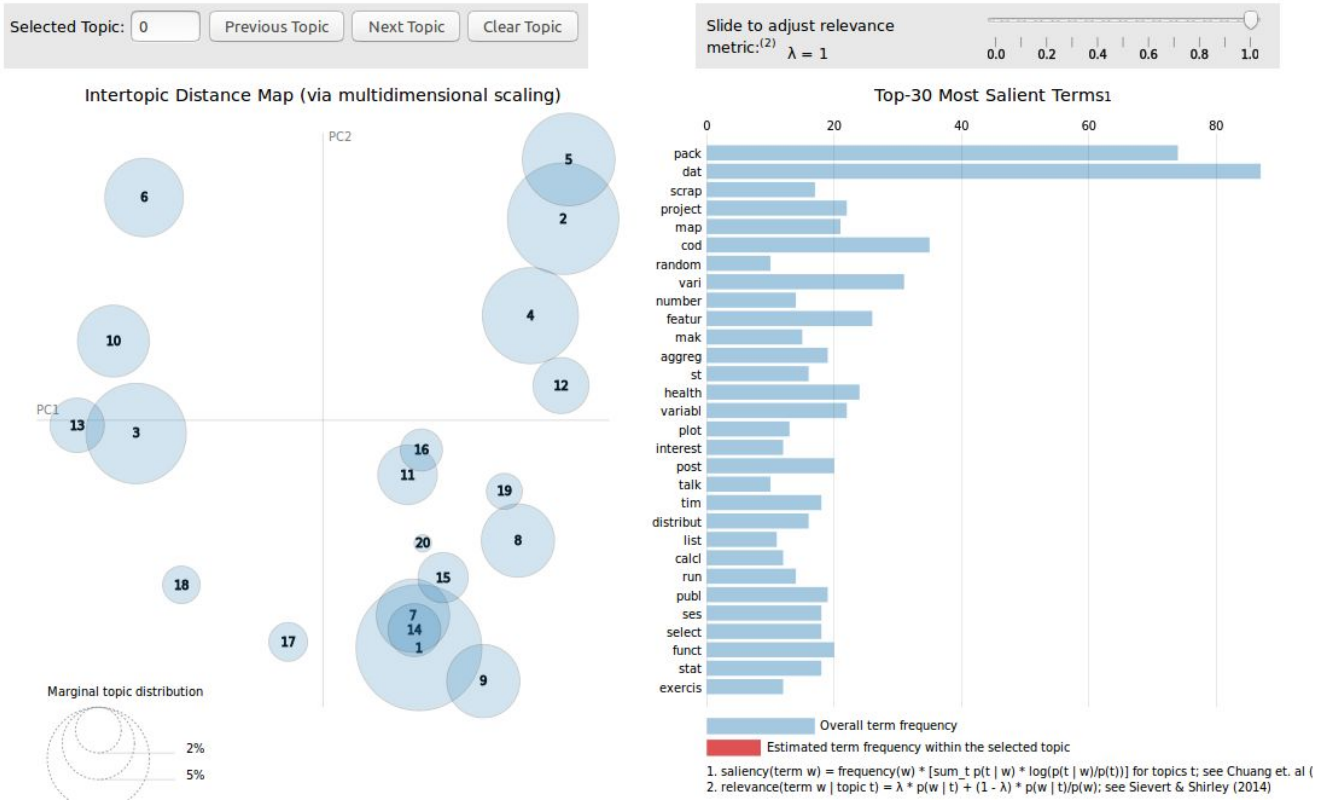
Topic prevalence contrast on two Facebook pages



(Puschmann, Ausserhofer and Slerka, forthcoming)

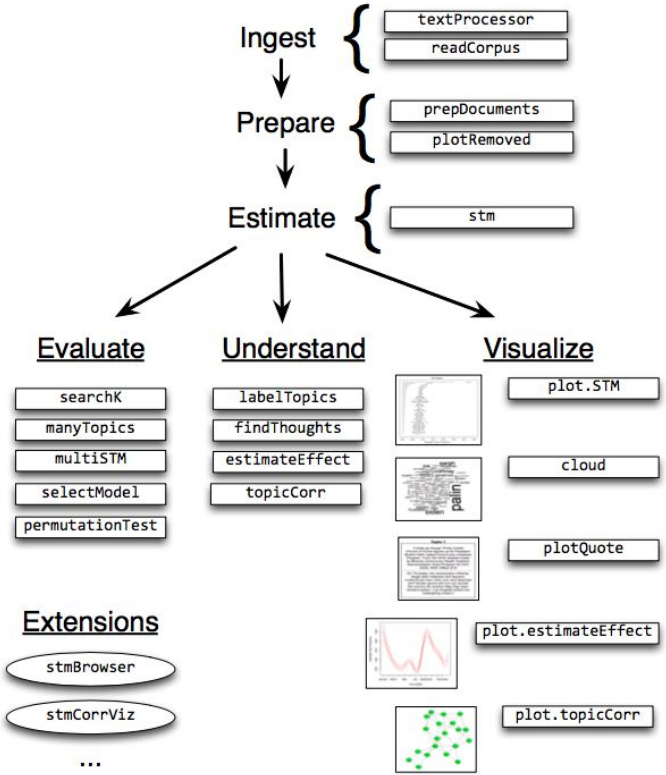


LDAvis for R and pyLDAvis



Introducing structural topic models

The *stm* package for R



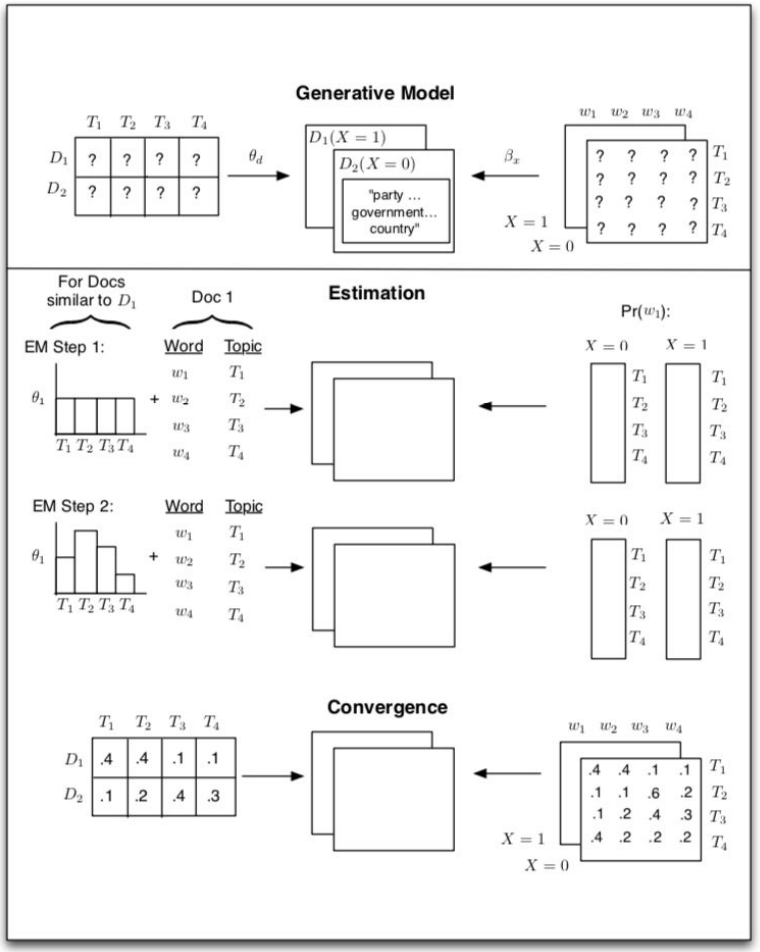
(Roberts, Stewart & Tingley, 2016)

Premises:

- a “one stop-shop” for topic modeling
- social scientific analysis (rather than information retrieval) in mind
- iterative work process
- thorough validation
- website: www.structuraltopicmodel.com

Model architecture

- Similar to LDA, *stm* combines a generative model and a sequential estimation process
- Process stops when model converges or max. number of iterations is reached
- Differences with LDA
  - optional Spectral initialization
  - declaration of covariates via the ‘prevalence’ argument
  - standard R formula notation
  - verbose step-by-step reporting on model fitting process



Typical *stm* workflow

#	Step	Function(s)	Input	Output
1	Estimate	stm()	A DFM (for example from quanteda) or other form of document term matrix	An STM model
2	Evaluate	searchK() selectModel()	A list of documents and a vocabulary	Return a set of heuristics to determine model fit
3	Understand	labelTopics() estimateEffect()	An STM model and corpus metadata	A list of topic labels and effect estimation values
4	Visualize	plot.STM() plot.estimateEffect()	An STM model and corpus metadata	Plots that show topic share and prevalence

Interrogation

Topic 3

Here's video of the ad we reported on below that the Obama campaign is running in Ohio responding to the earlier Swift–Boating spot tying Obama to former Weatherman Bill Ayers... With all our pr

---

As noted here and elsewhere, the words 'William Ayers' appeared nowhere in yesterday's debate, despite the fact that the McCain campaign hinted for days that McCain would go hard at Obama's association

Topic 20

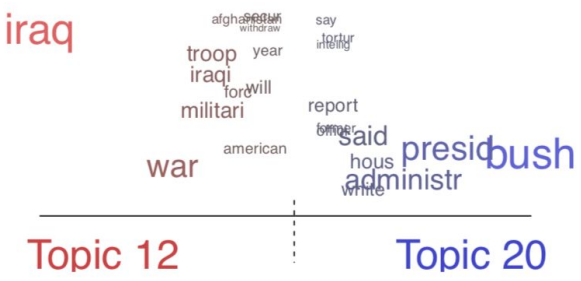
Waxman calls for release of FBI interviews with Bush and Cheney. In a letter to Attorney General Michael Mukasey today, Rep. Henry Waxman (D–CA), the Chairman of the Hous e Committee on Oversight

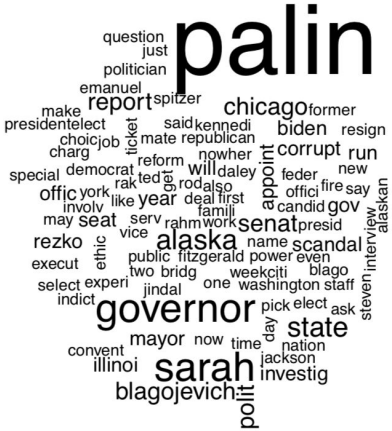
---

Report: Bush 'Personally Directed' Gonzales To Strong–Arm Ashcroft At His BedsideIn his May 2007 testimony, describing the infamous strong–arming of John Ashcroft done by Andy Card and Alberto

```
R> labelTopics(poliblogPrevFit, c(3, 7, 20))

Topic 3 Top Words:
        Highest Prob: obama, barack, campaign, biden, polit, will, debat
        FREX: ayer, barack, obama, wright, biden, jeremiah, joe
        Lift: oct, goolsbe, ayerss, ayr, bernadin, ayer, annenberg
        Score: oct, obama, barack, ayer, wright, campaign, biden
Topic 7 Top Words:
        Highest Prob: palin, governor, sarah, state, alaska, polit, senat
        FREX: blagojevich, palin, sarah, rezko, alaska, governor, gov
        Lift: jindal, blagojevich, juneau, monegan, blago, burri, wasilla
        Score: monegan, palin, blagojevich, sarah, alaska, rezko, governor
Topic 20 Top Words:
        Highest Prob: bush, presid, administr, said, hous, white, report
        FREX: cheney, tortur, cia, administr, interrog, bush, perino
        Lift: addington, fratto, perino, mcclellan, feith, plame, cheney
        Score: addington, bush, tortur, perino, cia, cheney, administr
```





Useful extensions/related packages

quanteda

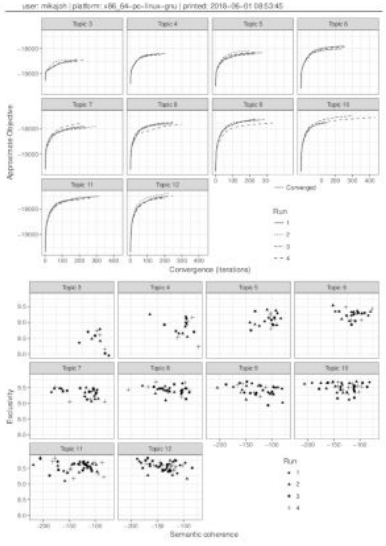
(general text processing framework)

stmprinter

(print dashboard of topics to PDF)

stminsights

(interactively inspect STM models)



Things to consider

- Distribution of topic scores
  - Topic scores often exhibit highly skewed distributions with potentially extreme values
    - use cutoff value and recode to 0/1 (not present/present);
    - or only consider highest-scoring topic
- High number of topics *k*
  - PCA or MDS
  - Human annotators to merge topics
  - Drop uninteresting and boilerplate topics

Thank you!
Now for the code...

Bibliography

Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis, 21(3), 267–297. https://doi.org/10.1093/pan/mps028

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. Handbook of latent semantic analysis, 427(7), 424-440.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010, June). Automatic evaluation of topic coherence. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 100-108). Association for Computational Linguistics.

Chaney, A. J. B., & Blei, D. M. (2012). Visualizing Topic Models. In ICWSM '12: Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (pp. 419–422). Dublin: AAAI Press.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (pp. 288-296).

Puschmann, C., & Scheffler, T. (2016). Topic modeling for media and communication research : A short primer (HIIG Discussion Paper Series No. 2016–5). Berlin. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2836478

Roberts, M. E., Stewart, B. M., & Tingley, D. (2016). stm: R Package for Structural Topic Models. *Journal of Statistical Software,* VV(Ii).

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004, July). The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (pp. 487-494). AUAI Press.

Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". *Journal of Documentation*. 28: 11–21. doi:10.1108/eb026526

Stier, S., Posch, L., Bleier, A., & Strohmaier, M. (2017). When populists become popular: comparing Facebook use by the right-wing movement Pegida and German political parties. *Information, Communication & Society,* 4462(May), 1–24. https://doi.org/10.1080/1369118X.2017.1328519