Class 7: Dictionaries and Topic Modelling

Topic 2: Text

Computational Analysis of Text, Audio, and Images, Fall 2023

Aarhus University

Mathias Rask (mathiasrask@ps.au.dk)

Aarhus University

Today's Menu

Dictionaries

Topic Models

Lab

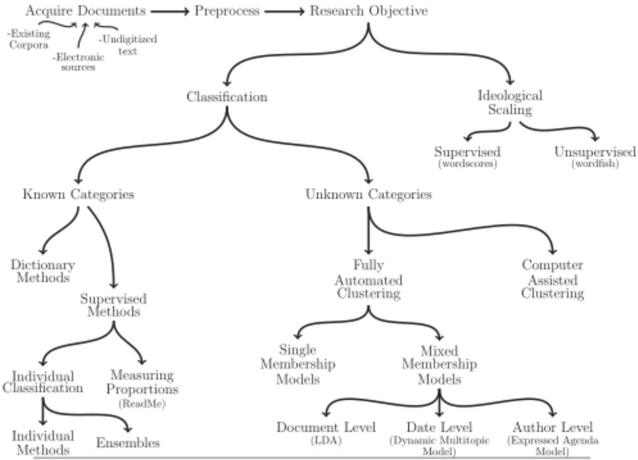
What Can We Do With Text?

1. Prediction
   - Hate-speech in tweets
   - Partisanship
2. Meaning
   - Actor-variation
   - Time-variation
3. Language use
   - Similarity
   - Complexity
4. Content
   - Topics
   - Word counts
5. Measurement
   - Positions (i.e. scaling)
   - Sentiment
   - Emotions

Four Guiding Principles (Grimmer and Stewart, 2013)

1. All models for text are wrong, but *some* are useful
2. Models augment humans but do not replace humans
3. Validation is key
4. Quantitative text analysis is dimensionality reduction

Methods



Acquire Documents
Preprocess
Research Objective
-Existing Corpora
-Electronic sources
-Undigitized text
Classification
Ideological Scaling
Supervised (wordscores)
Unsupervised (wordfish)
Known Categories
Unknown Categories
Dictionary Methods
Supervised Methods
Fully Automated Clustering
Computer Assisted Clustering
Individual Classification
Measuring Proportions (ReadMe)
Single Membership Models
Mixed Membership Models
Individual Methods
Ensembles
Document Level (LDA)
Date Level (Dynamic Multitopic Model)
Author Level (Expressed Agenda Model)

Table of Contents

Rule-based Measurement

Dictionaries are widely used in political science and are basically about counting words in a set $\{\}$
$\rightsquigarrow$ A generalization of counting individual words

We can use dictionaries for two purposes:

- Content: If certain words $\{w_1, w_2, \ldots, w_J\}$ are present in $\mathcal{D}_i \rightsquigarrow$ contains C
    - Example: Talking about immigration
    - Words: [udlænding, asylansøger, familiesammenføring]
- Measurement: If certain words $\{w_1, w_2, \ldots, w_N\}$ are present in $\mathcal{D}_i \rightsquigarrow$ signalling of L
    - Example: Use aggressive language
    - Words: [had, idiot, dum, fatsvag, dompap]

Counting Words

Applying a dictionary is straightforward.

Assume we have our corpus $\mathcal{C}$ with documents $\mathcal{D}_i$ with $i \in \{1, \ldots, N\}$.

For each document $\mathcal{D}_i$, the dictionary score is:

$$\text{score}_{\mathcal{D}_i} = \frac{\sum_{j=1}^{J} W_{ij}}{n_i}$$

where:

- W_{ij} is a vector of 0 and 1 indicating whether a dictionary word j appears in $\mathcal{D}_i$
  - $\sum_{j=1}^{J} W_{ij} = |A \cap B|$
- n_i is the total number of words in $\mathcal{D}_i$

⇝ Why do we normalize by n_i?

⇝ Note that we can also add a time-dimension. What does our score then look like?

Exercise

Assume the dictionary:

Key:	Aggression
Values:	stupid dishonest lier idiot ignorant hate fight battle

and the document $\mathcal{D}$: "That statement is as barbaric as it is downright stupid; it is nothing more than an ignorant, cruel, and deliberate misconception to hide behind."

1. Compute the dictionary score $\frac{\sum_{j=1}^{J} W_{ij}}{n_i}$ with $\boldsymbol{n_i}$ being the number of unique words (**14**)
2. What's the upper and lower bound of the aggressiveness scores?

Sentiment Analysis

A dictionary-based analysis is often equivalent to a sentiment analysis of text: Positive or negative use of language.

- Different from policy positions, but often highly correlated

Example: Silva and Proksch (2022)

- Using a sentiment dictionary to compute a measure of positions of MPs expressed in tweets (X's?) about EU
- How do they measure sentiment about the EU in parliamentary speeches?

How to Get Dictionaries

Dictionaries can have two different origins:

1. Pre-package (i.e. pretrained)
   - e.g. AFINN, LIWC, ANEW, LSD, https://github.com/cjhutto/vaderSentiment
2. Domain-specific
   - e.g. EU-related words (Silva and Proksch, 2022) such as “Brussels”, “Europ”, etc.

$\rightsquigarrow$ How does this relates to questions about *recall* and *precision*?

Methodological Considerations

Dictionaries are important tools due to their easy implementation: we can get far with low resources.

- Word reduction is an important preprocessing step to relax word dependency
- Word ambiguity can be an issue
- Denominating by totals is crucial
- Do dictionaries travel across contexts/domains? (i.e. generalizability)
- How can we validate our dictionaries? – Dictionaries require front-end work

Table of Contents

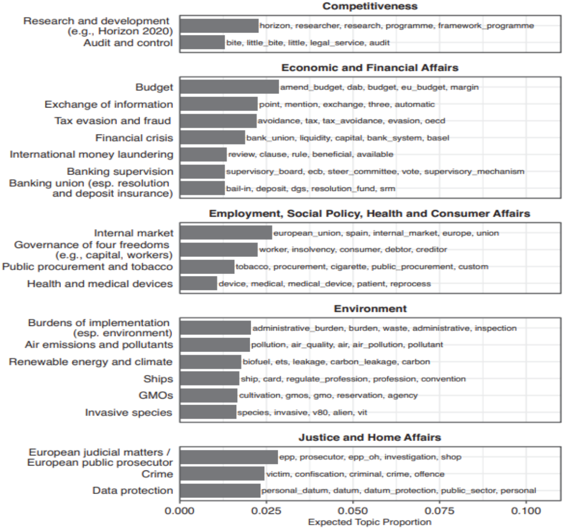
Topic Models

Topic models allow us to cluster similar documents $\mathcal{D}_i$ in a corpus $\mathcal{C}$ *together* $\rightsquigarrow$ clustering!

- We *already* have learned the necessary tools...
  - Dictionary-based content identification
  - Supervised learning

  $\rightsquigarrow$ Why do we need yet another method?
- Topic models permit *unsupervised* learning – automatic discovery of latent “topics” $k \in \{1, \ldots, K\}$
  - Most popular approach is Latent Dirichlet Allocation (LDA), which assumes a mixture model:
    - Documents can contain multiple topics
    - Words can belong to multiple topics

Wratil et al. (2022): Policy-Specific Topics in Council Deliberations



Topic Models as Probabilistic Language Models

What is a language model?

- A model that describes the generation of language as probabilities
  - Given words Q, what is the probability that word q belongs to the same topic k?
- A language model is represented by a probability distribution over words in $\mathcal{V}$
- Chat-GPT is a *large language model* (LLM), but topic models are also language models
  - For each topic k, we estimate a probability distribution over the words (i.e. k distributions)
  - For each document $\mathcal{D}$, we estimate a probability distribution over the topics (i.e. $|\mathcal{D}|$ distributions)

$\rightsquigarrow$ These probabilities are computed simultaneously

LDA

- More than 43,000 citations on Google Scholar!!! (Blei, 2012)
- We start by choosing K – the number of topics in $\mathcal{C}$
- Assumptions:
  - Each topic k is a mixture of words
  - Each document $\mathcal{D}_i$ is a mixture of topics
- Outputs:
  - Document-topic distribution: $|\mathcal{C}| \times K$ matrix
    - ▷ $|\mathcal{C}| = 10,000$ and $K = 40$: $10,000 \times 40$ matrix
    - ▷ A document D_i is a probability distribution over K topics:
    - ▷ $\sum_{k=1}^{K} \theta_{\mathcal{D}_i k} = 1$
    - ▷ $\theta_{\mathcal{D}_i k}$ denotes the probability of a topic k occurring in document $\mathcal{D}_i$
  - Word-topic distribution $|\mathcal{V}| \times K$ matrix

Advantages and Disadvantages

Advantages:

- Automatically finds substantively “clusters” of words
- These clusters often form *somewhat* coherent topics
- Scalable without the need for manual labeling

Disadvantages:

- Sensitive to K
- Post-hoc interpretation and mapping
    - A common approach is to manually map topics to the target concepts after fitting a model
- One topic might itself be a mixture of topics
- Many topics are often incoherent and redundant

$\rightsquigarrow$ Preprocessing is an important step!

LDA Backends

LDA is the foundation of (better?) more advanced approaches:

- Structural Topic Model (Roberts *et al.*, 2014)
- Seeded LDA (Watanabe and Baturo, 2023)
- BERTopic

Table of Contents

See you next week!

Topic 2: Text

Computational Analysis of Text, Audio, and Images, Fall 2023

Aarhus University

References i

[1] J. Grimmer and B. M. Stewart, "Text as data: The promise and pitfalls of automatic content analysis methods for political texts," *Political analysis*, vol. 21, no. 3, pp. 267–297, 2013.

[2] B. C. Silva and S.-O. Proksch, "Politicians unleashed? political communication on twitter and in parliament in western europe," *Political science research and methods*, vol. 10, no. 4, pp. 776–792, 2022.

[3] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012. DOI: `http://doi.acm.org/10.1145/2133806.2133826`.

[4] M. E. Roberts *et al.*, "Structural topic models for open-ended survey responses," *American journal of political science*, vol. 58, no. 4, pp. 1064–1082, 2014.

[5] K. Watanabe and A. Baturo, "Seeded sequential lda: A semi-supervised algorithm for topic-specific analysis of sentences," *Social Science Computer Review*, p. 08 944 393 231 178 605, 2023.