ECPR Methods Summer School: Automated Collection of Web and Social Data

Pablo Barberá

School of International Relations
University of Southern California
pablobarbera.com

Networked Democracy Lab
www.netdem.org

Course website:
github.com/pablobarbera/ECPR-SC103

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


SPÖ
WAHLMANIFEST


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


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

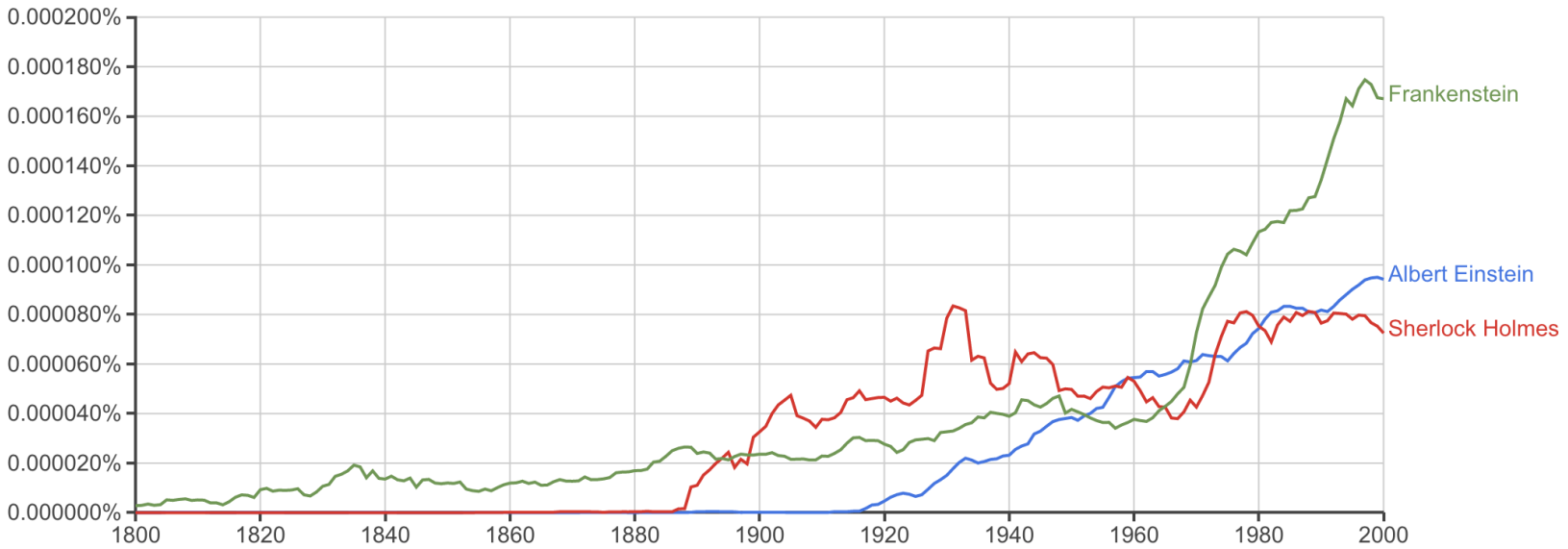

Sozial und
Demokratisch.
Anpacken. Für Deutschland.
Das Regierungsprogramm der SPD
SPD

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



Text as data


facebook
XING
Pinterest
Instagram
Gefällt mir
twitter
LinkedIn
flickr
skype
YouTube

Overview of text as data methods

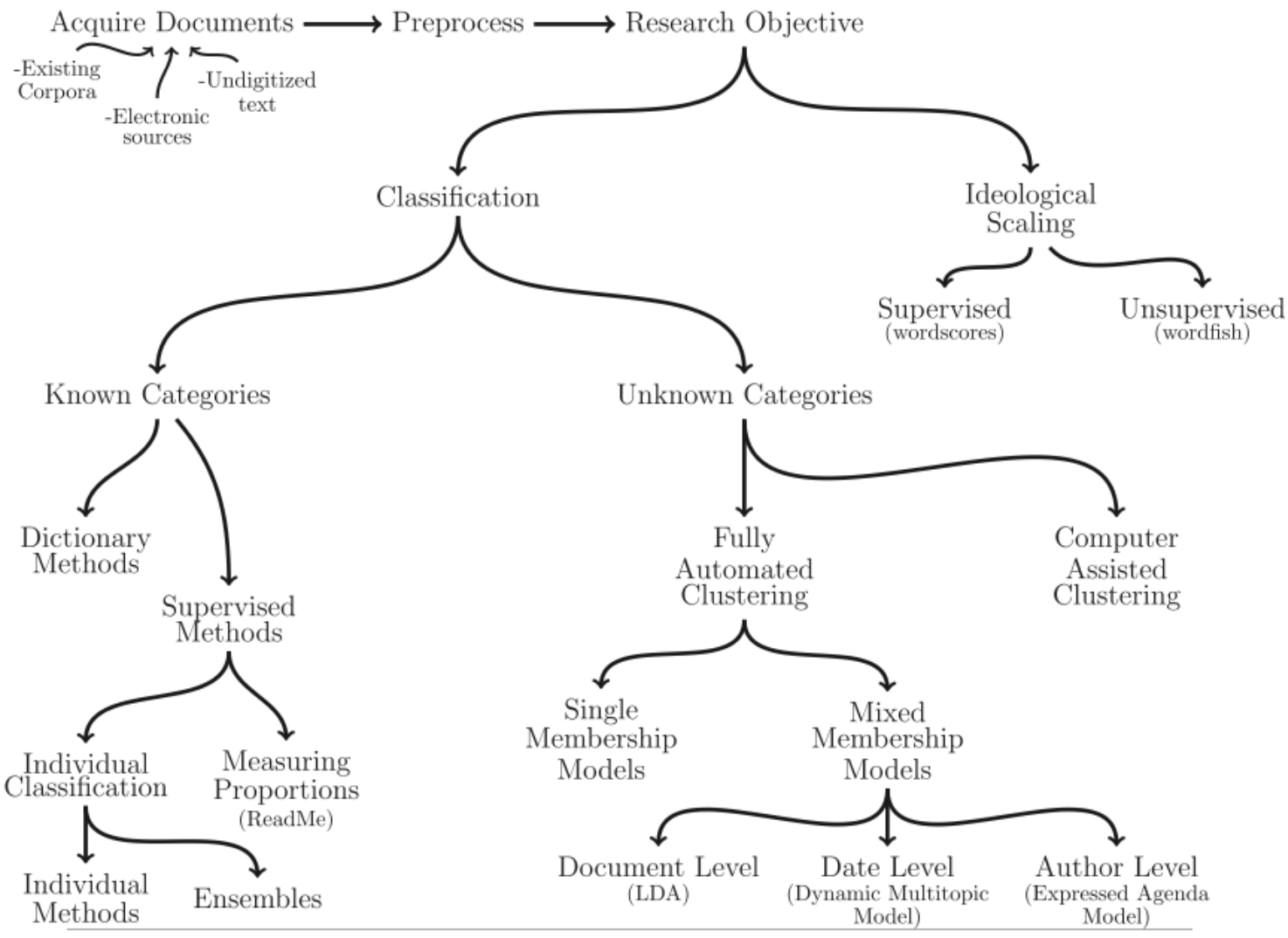


Fig. 1 in Grimmer and Stewart (2013)

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 n-grams
   - w_{im}= number of times m-th n-gram 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	0	1	1	0	0	0	...	

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- Validate, validate, validate.

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- Validate, validate, validate.
  - Check sensitivity of results to exclusion of specific words

Dictionary methods

Classifying documents when categories are known using dictionaries:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- Validate, validate, validate.
  - Check sensitivity of results to exclusion of specific words
  - Code a few documents manually and see if dictionary prediction aligns with human coding of document