Data management

Introduction to Text Data

Malka Guillot (& Elliott Ash)

Text as Data

- Text data is a sequence of characters called **documents**.
- The set of documents is the **corpus**.
- Text data is **unstructured**:
  - the information we want is mixed together with (lots of) information we don't.
- All text data approaches will throw away some information:
  - The trick is figuring out how to retain valuable information.

1. **Read text documents as data:**
    - Convert texts to features – words, phrases, syntactic/semantic relations.
    - Feature selection / dimension reduction to exclude irrelevant information.

1. **Read text documents as data:**
   - Convert texts to features – words, phrases, syntactic/semantic relations.
   - Feature selection / dimension reduction to exclude irrelevant information.
2. **Dictionary methods for targeted studies:**
   - e.g. sentiment analysis

1. **Read text documents as data:**
    - Convert texts to features – words, phrases, syntactic/semantic relations.
    - Feature selection / dimension reduction to exclude irrelevant information.
2. **Dictionary methods for targeted studies:**
    - e.g. sentiment analysis
3. **Unsupervised learning techniques for interpreting corpora:**
    - topic models, document embeddings

1. **Read text documents as data:**
    - Convert texts to features – words, phrases, syntactic/semantic relations.
    - Feature selection / dimension reduction to exclude irrelevant information.
2. **Dictionary methods for targeted studies:**
    - e.g. sentiment analysis
3. **Unsupervised learning techniques for interpreting corpora:**
    - topic models, document embeddings
4. **Supervised learning with text:**
    - applying regressors and classifiers to text features.

1. **Read text documents as data:**
    - Convert texts to features – words, phrases, syntactic/semantic relations.
    - Feature selection / dimension reduction to exclude irrelevant information.
2. **Dictionary methods for targeted studies:**
    - e.g. sentiment analysis
3. **Unsupervised learning techniques for interpreting corpora:**
    - topic models, document embeddings
4. **Supervised learning with text:**
    - applying regressors and classifiers to text features.
5. **Word embedding for isolating dimensions of language:**
    - Analyze values, attitudes, and ideology

Outline

Outline

[4]:
```python
from sklearn.datasets import fetch_20newsgroups
data = fetch_20newsgroups() # object is a dictionary
data.keys()
```

[4]:
```
dict_keys(['data', 'filenames', 'target_names', 'target', 'DESCR'])
```

Data Set Characteristics:

[5]:
```python
print(data['DESCR'])
```

```
.. _20newsgroups_dataset:

The 20 newsgroups text dataset
------------------------------

The 20 newsgroups dataset comprises around 18000 newsgroups posts on
20 topics split in two subsets: one for training (or development)
and the other one for testing (or for performance evaluation). The split
between the train and test set is based upon a messages posted before
and after a specific date.
```

```
[6]: W, y = data.data, data.target
     n_samples = y.shape[0]
     n_samples
```

```
[6]: 11314
```

```
[7]: y[:10] # news story categories
```

```
[7]: array([ 7,  4,  4,  1, 14, 16, 13,  3,  2,  4])
```

```
[8]: doc = W[0]
     doc
```

```
[8]: "From: lerxst@wam.umd.edu (where's my thing)\nSubject: WHAT car is this!?\nNntp
     -Posting-Host: rac3.wam.umd.edu\nOrganization: University of Maryland, College
     Park\nLines: 15\n\n I was wondering if anyone out there could enlighten me on t
     his car I saw\nthe other day. It was a 2-door sports car, looked to be from the
     late 60s/\nearly 70s. It was called a Bricklin. The doors were really small. In
     addition,\nthe front bumper was separate from the rest of the body. This is \na
     ll I know. If anyone can tellme a model name, engine specs, years\nof productio
     n, where this car is made, history, or whatever info you\nhave on this funky lo
     oking car, please e-mail.\n\nThanks,\n- IL\n   ---- brought to you by your neig
     hborhood Lerxst ----\n\n\n\n\n"
```

```
df = pd.DataFrame(W,columns=['text'])
df['topic'] = y
df.head()
```

	text	topic
0	From: lerxst@wam.umd.edu (where's my thing)\nS...	7
1	From: guykuo@carson.u.washington.edu (Guy Kuo)...	4
2	From: twillis@ec.ecn.purdue.edu (Thomas E Will...	4
3	From: jgreen@amber (Joe Green)\nSubject: Re: W...	1
4	From: jcm@head-cfa.harvard.edu (Jonathan McDow...	14

Corpus cleaning

- Pre-Processing Steps:
  - Remove HTML markup, extra white space, and unicode

Corpus cleaning

- Pre-Processing Steps:
  - Remove HTML markup, extra white space, and unicode
- But HTML markup is often valuable:
  - HTML markup for section header names.
  - e.g., legal database web sites often have HTML tags for citations to other cases.

Corpus cleaning

- Pre-Processing Steps:
  - Remove HTML markup, extra white space, and unicode
- But HTML markup is often valuable:
  - HTML markup for section header names.
  - e.g., legal database web sites often have HTML tags for citations to other cases.
- Other cleaning steps:
  - page numbers
  - hyphenations at line breaks
  - table of contents, indexes, etc.
- These are all corpus-specific, so inspect ahead of time.

OCR (Optical Character Recognition)

- Your data might be in PDF's or images. Needs to be converted to text
- The best solution (that I know of) is ABBYY FineReader, which is expensive but might be available at your university library.
- My colleague Joe Sutherland at Columbia has a nice open-source package for OCR:
  - `https://github.com/jlsutherland/doc2text`

Other Languages

- All of the tools that we discuss in this class are available in many languages.
  - See, e.g., `https://spacy.io/usage/models`.
- Can also translate (e.g., API links to google translate and DeepL).
- The machine learning models are language-independent.

What counts as a document?

The unit of analysis (the "document") will vary depending on your question.

- needs to be fine enough to fit the relevant metadata variation
- should not be finer – would make dataset more high-dimensional without empirical benefit.

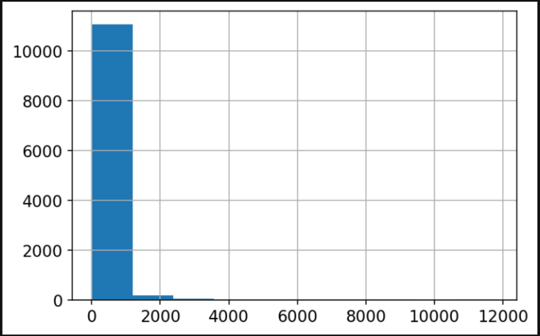
Outline

Count words per document.

[13]:
```python
def get_words_per_doc(txt):
    # split text into words and count them.
    return len(txt.split())

# apply to our data
df['num_words'] = df['text'].apply(get_words_per_doc)
df['num_words'].hist()
```

[13]: <AxesSubplot:>

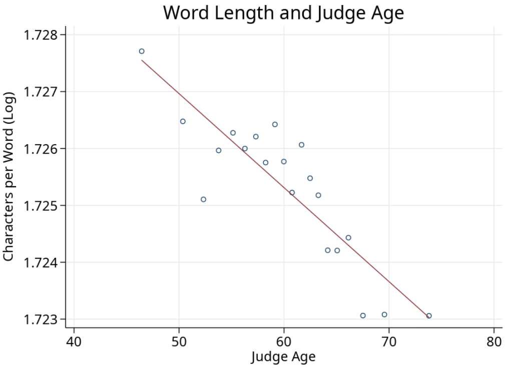


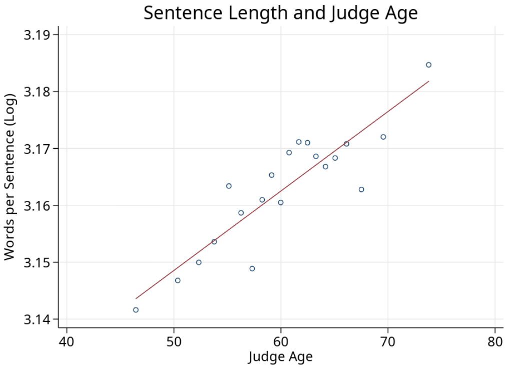
Judge Age and Writing Style

Ash, Goessmann, and MacLeod (2021)

Judge Age and Writing Style

Ash, Goessmann, and MacLeod (2021)





Optimal Legal Complexity (Katz and Bommarito 2014)

- More legal detail is needed to properly specify rules and target incentives to activities and groups.
  - but there are costs to understanding/following/maintaining complex laws, so there is a trade off.

Optimal Legal Complexity (Katz and Bommarito 2014)

- More legal detail is needed to properly specify rules and target incentives to activities and groups.
  - but there are costs to understanding/following/maintaining complex laws, so there is a trade off.
- Katz and Bommarito measure complexity/detail from the text – number of words for code title, and also *word entropy* $\approx$ diversity of the vocabulary.

Optimal Legal Complexity (Katz and Bommarito 2014)

- More legal detail is needed to properly specify rules and target incentives to activities and groups.
  - but there are costs to understanding/following/maintaining complex laws, so there is a trade off.
- Katz and Bommarito measure complexity/detail from the text – number of words for code title, and also *word entropy* $\approx$ diversity of the vocabulary.

Five largest and smallest titles by token count

Title	Tokens	Tokens per section
Public Health and Welfare (Title 42)	2,732,251	369.22
Internal Revenue Code (Title 26)	1,016,995	487.07
Conservation (Title 16)	947,467	200.48
Commerce and Trade (Title 15)	773,819	336.88
Agriculture (Title 7)	751,579	274.00
President (Title 3)	7,564	120.06
Intoxicating Liquors (Title 27)	6,515	144.78
Flag and Seal, Seat of Govt. and the States (Title 4)	5,598	119.11
General Provisions (Title 1)	3,143	80.59
Arbitration (Title 9)	2,489	80.29

Optimal Legal Complexity (Katz and Bommarito 2014)

- More legal detail is needed to properly specify rules and target incentives to activities and groups.
  - but there are costs to understanding/following/maintaining complex laws, so there is a trade off.
- Katz and Bommarito measure complexity/detail from the text – number of words for code title, and also *word entropy* ≈ diversity of the vocabulary.

Five largest and smallest titles by token count

Title	Tokens	Tokens per section
Public Health and Welfare (Title 42)	2,732,251	369.22
Internal Revenue Code (Title 26)	1,016,995	487.07
Conservation (Title 16)	947,467	200.48
Commerce and Trade (Title 15)	773,819	336.88
Agriculture (Title 7)	751,579	274.00
President (Title 3)	7,564	120.06
Intoxicating Liquors (Title 27)	6,515	144.78
Flag and Seal, Seat of Govt. and the States (Title 4)	5,598	119.11
General Provisions (Title 1)	3,143	80.59
Arbitration (Title 9)	2,489	80.29

Five highest and lowest titles by word entropy

Title	Word entropy
Commerce and Trade (Title 15)	10.80
Public Health and Welfare (Title 42)	10.79
Conservation (Title 16)	10.75
Navigation and Navigable Waters (Title 33)	10.67
Foreign Relations and Intercourse (Title 22)	10.67
Intoxicating Liquors (Title 27)	9.01
President (Title 3)	8.89
National Guard (Title 32)	8.50
General Provisions (Title 1)	8.49
Arbitration (Title 9)	8.24

Outline

Overview of Dictionary-Based Methods

- Dictionary-based text methods use a pre-selected list of words or phrases to analyze a corpus.
  - use regular expressions for this task (see notebook)

Overview of Dictionary-Based Methods

- Dictionary-based text methods use a pre-selected list of words or phrases to analyze a corpus.
  - use regular expressions for this task (see notebook)
- Corpus-specific: counting sets of words or phrases across documents
  - (e.g., number of times a judge says "justice" vs "efficiency")

Overview of Dictionary-Based Methods

- Dictionary-based text methods use a pre-selected list of words or phrases to analyze a corpus.
  - use regular expressions for this task (see notebook)
- Corpus-specific: counting sets of words or phrases across documents
  - (e.g., number of times a judge says "justice" vs "efficiency")
- General dictionaries: WordNet, LIWC, MFD, etc.

Measuring uncertainty in macroeconomy

Baker, Bloom, and Davis (QJE 2016)

Measuring uncertainty in macroeconomy

Baker, Bloom, and Davis (QJE 2016)

For each newspaper on each day since 1985, submit the following query:

1. Article contains “uncertain” OR “uncertainty”, AND
2. Article contains “economic” OR “economy”, AND
3. Article contains “congress” OR “deficit” OR “federal reserve” OR “legislation” OR “regulation” OR “white house”

Normalize resulting article counts by total newspaper articles that month.

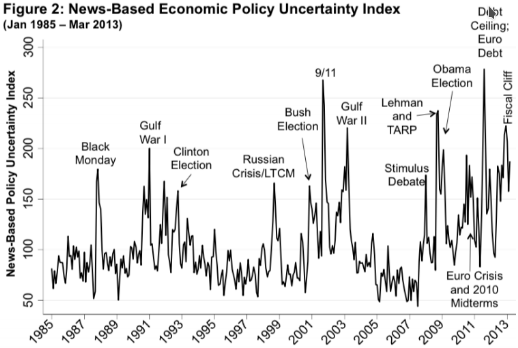
Measuring uncertainty in macroeconomy

Baker, Bloom, and Davis (QJE 2016)

For each newspaper on each day since 1985, submit the following query:

1. Article contains "uncertain" OR "uncertainty", AND
2. Article contains "economic" OR "economy", AND
3. Article contains "congress" OR "deficit" OR "federal reserve" OR "legislation" OR "regulation" OR "white house"

Normalize resulting article counts by total newspaper articles that month.



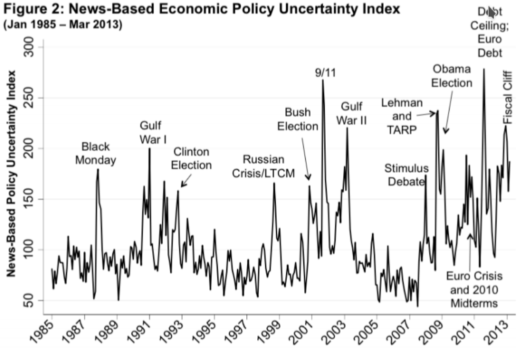
Measuring uncertainty in macroeconomy

Baker, Bloom, and Davis (QJE 2016)

For each newspaper on each day since 1985, submit the following query:

1. Article contains "uncertain" OR "uncertainty", AND
2. Article contains "economic" OR "economy", AND
3. Article contains "congress" OR "deficit" OR "federal reserve" OR "legislation" OR "regulation" OR "white house"

Normalize resulting article counts by total newspaper articles that month.



- but see Keith et al (2020), showing some big problems with this measure (`https://arxiv.org/abs/2010.04706`).

Sentiment Analysis

Extract a “tone” dimension – positive, negative, neutral

- standard approach is lexicon-based, but they fail easily: e.g., “good” versus “not good” versus “not very good”

Sentiment Analysis

Extract a "tone" dimension – positive, negative, neutral

- standard approach is lexicon-based, but they fail easily: e.g., "good" versus "not good" versus "not very good"
- flair's pre-trained sentiment model uses a context-sensitive neural net

Sentiment Analysis

Extract a "tone" dimension – positive, negative, neutral

- standard approach is lexicon-based, but they fail easily: e.g., "good" versus "not good" versus "not very good"
- flair's pre-trained sentiment model uses a context-sensitive neural net
- Off-the-shelf scores designed for online writing – may not work for legal text, for example.
    - Hamilton et al (2016) and Zorn and Rice (2019) show how to make domain-specific sentiment lexicons using word embeddings (more on this later).

Sentiment Analysis

[16]:
```python
# Dictionary-Based Sentiment Analysis

from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
polarity = sid.polarity_scores(doc)
print(polarity)
```

```
{'neg': 0.012, 'neu': 0.916, 'pos': 0.072, 'compound': 0.807}
```

General Dictionaries

- WordNet: English word database: 118K nouns, 12K verbs, 22K adjectives, 5K adverbs. Synonym sets (synsets) are a group of near-synonyms, plus a gloss (definition).
  - also contains information on antonyms (opposites), holonyms/meronyms (part-whole).

General Dictionaries

- WordNet: English word database: 118K nouns, 12K verbs, 22K adjectives, 5K adverbs. Synonym sets (synsets) are a group of near-synonyms, plus a gloss (definition).
  - also contains information on antonyms (opposites), holonyms/meronyms (part-whole).
- Function words (e.g. *for*, *rather*, *than*)
  - also called stopwords
  - can be used to get at non-topical dimensions, identify authors.

General Dictionaries

- WordNet: English word database: 118K nouns, 12K verbs, 22K adjectives, 5K adverbs. Synonym sets (synsets) are a group of near-synonyms, plus a gloss (definition).
  - also contains information on antonyms (opposites), holonyms/meronyms (part-whole).
- Function words (e.g. *for*, *rather*, *than*)
  - also called stopwords
  - can be used to get at non-topical dimensions, identify authors.
- LIWC (pronounced “Luke”): Linguistic Inquiry and Word Counts
  - 2300 words 70 lists of category-relevant words, e.g. “emotion”, “cognition”, “work”, “family”, “positive”, “negative” etc.

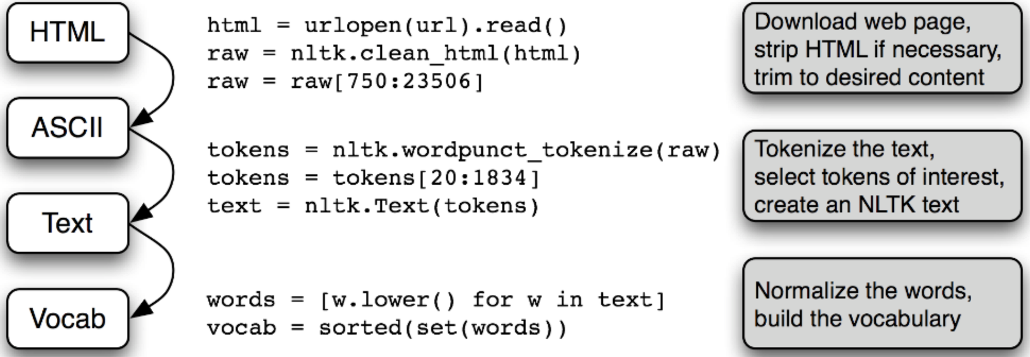
General Dictionaries

- WordNet: English word database: 118K nouns, 12K verbs, 22K adjectives, 5K adverbs. Synonym sets (synsets) are a group of near-synonyms, plus a gloss (definition).
  - also contains information on antonyms (opposites), holonyms/meronyms (part-whole).
- Function words (e.g. *for*, *rather*, *than*)
  - also called stopwords
  - can be used to get at non-topical dimensions, identify authors.
- LIWC (pronounced "Luke"): Linguistic Inquiry and Word Counts
  - 2300 words 70 lists of category-relevant words, e.g. "emotion", "cognition", "work", "family", "positive", "negative" etc.
- Mohammad and Turney (2011):
  - code 10,000 words along four emotional dimensions: joy–sadness, anger-fear, trust-disgust, anticipation-surprise
- Warriner et al (2013):
  - code 14,000 words along three emotional dimensions: valence, arousal, dominance.

Outline

Goals of Featurization

- The goal: produce features that are
  - **predictive** in the learning task
  - **interpretable** by human investigators
  - **tractable** enough to be easy to work with



Pre-processing

- An important piece of the "art" of text analysis is deciding what data to throw out.
  - Uninformative data add noise and reduce statistical precision.
  - They are also computationally costly.

Pre-processing

- An important piece of the "art" of text analysis is deciding what data to throw out.
  - Uninformative data add noise and reduce statistical precision.
  - They are also computationally costly.
- Pre-processing choices can affect down-stream results, especially in unsupervised learning tasks (Denny and Spirling 2017).
  - some features are more interpretable

Pre-processing

- An important piece of the "art" of text analysis is deciding what data to throw out.
  - Uninformative data add noise and reduce statistical precision.
  - They are also computationally costly.
- Pre-processing choices can affect down-stream results, especially in unsupervised learning tasks (Denny and Spirling 2017).
  - some features are more interpretable
- Standard pre-processing steps:
  - drop capitalization, punctuation, numbers, stopwords (e.g. "the", "such")
  - remove word stems (e.g., "taxes" and "taxed" become "tax")

Say we want to convert a corpus D to a matrix X:

- In the "bag-of-words" representation, a row of X is just the frequency distribution over words in the document corresponding to that row.

Say we want to convert a corpus D to a matrix X:

- In the "bag-of-words" representation, a row of X is just the frequency distribution over words in the document corresponding to that row.

More generally:

- **Document counts**: number of documents where a token appears.
- **Term counts**: number of total appearances of a token in corpus.

Say we want to convert a corpus D to a matrix X:

- In the "bag-of-words" representation, a row of X is just the frequency distribution over words in the document corresponding to that row.

More generally:

- **Document counts**: number of documents where a token appears.
- **Term counts**: number of total appearances of a token in corpus.
- **Term frequency**:

$$\text{Term Frequency in document } k = \frac{\text{Term count in document } k}{\text{Total tokens in document } k}$$

Building a vocabulary

- An important featurization step is to build a vocabulary of words:
  - Compute document frequencies for all words
  - Inspect low-frequency words and determine a minimum document threshold.
    - e.g., 10 documents, or .25% of documents.

Building a vocabulary

- An important featurization step is to build a vocabulary of words:
  - Compute document frequencies for all words
  - Inspect low-frequency words and determine a minimum document threshold.
    - e.g., 10 documents, or .25% of documents.
- Can also impose more complex thresholds, e.g.:
  - appears twice in at least 20 documents
  - appears in at least 3 documents in at least 5 years

Building a vocabulary

- An important featurization step is to build a vocabulary of words:
  - Compute document frequencies for all words
  - Inspect low-frequency words and determine a minimum document threshold.
    - e.g., 10 documents, or .25% of documents.
- Can also impose more complex thresholds, e.g.:
  - appears twice in at least 20 documents
  - appears in at least 3 documents in at least 5 years
- Assign numerical identifiers to tokens to increase speed and reduce disk usage.

TF-IDF Weighting

- TF/IDF: "Term-Frequency / Inverse-Document-Frequency."
- The formula for word w in document k:

$$\underbrace{\frac{\text{Count of } w \text{ in } k}{\text{Total word count of } k}}_{\text{Term Frequency}} \times \underbrace{\log\left(\frac{\text{Number of documents in } D}{\text{Count of documents containing } w}\right)}_{\text{Inverse Document Frequency}}$$

TF-IDF Weighting

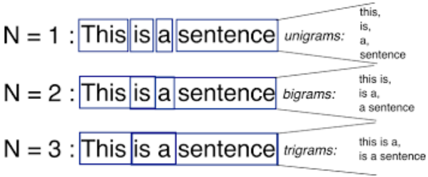
- TF/IDF: "Term-Frequency / Inverse-Document-Frequency."
- The formula for word w in document k:

$$\underbrace{\frac{\text{Count of } w \text{ in } k}{\text{Total word count of } k}}_{\text{Term Frequency}} \times \underbrace{\log\left(\frac{\text{Number of documents in } D}{\text{Count of documents containing } w}\right)}_{\text{Inverse Document Frequency}}$$

- The formula up-weights relatively rare words that do not appear in all documents.
  - These words are probably more distinctive of topics or differences between documents.
  - Example: A document contains 100 words, and the word appears 3 times in the document. The TF is .03. The corpus has 100 documents, and the word appears in 10 documents. the IDF is $\log(100/10) \approx 2.3$, so the TF-IDF for this document is $.03 \times 2.3 = .07$. Say the word appears in 90 out of 100 documents: Then the IDF is 0.105, with TF-IDF for this document equal to .003.

N-grams

- N-grams are phrases, sequences of words up to length N.
  - bigrams, trigrams, quadgrams, etc.



- capture information and familiarity from local word order.
  - e.g. "estate tax" vs "death tax"

scikit-learn's TfidfVectorizer

https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

scikit-learn's TfidfVectorizer

https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

```
>>> from sklearn.feature_extraction.text import TfidfVectorizer
>>> vectorizer = TfidfVectorizer()
>>> vectorizer.fit_transform(corpus)
<4x9 sparse matrix of type '<... 'numpy.float64'>'
    with 19 stored elements in Compressed Sparse ... format>
```

- `corpus` is a sequence of strings, e.g. pandas data-frame columns.
- pre-processing options: strip accents, lowercase, drop stopwords,
- n-grams: can produce phrases up to length n (words or characters).
- vocab options: min/max frequency, vocab size
- post-processing: binary, l2 norm, (smoothed) idf weighting, etc

Filtering the Vocabulary

- N-grams will blow up your feature space: filtering out uninformative n-grams is necessary.
  - Google Developers recommend vocab size $= m =$20,000; I have gotten good performance from $m =$2,000.

Filtering the Vocabulary

- N-grams will blow up your feature space: filtering out uninformative n-grams is necessary.
  - Google Developers recommend vocab size = m =20,000; I have gotten good performance from m =2,000.

1. Drop phrases that appear in few documents, or in almost all documents.

Filtering the Vocabulary

- N-grams will blow up your feature space: filtering out uninformative n-grams is necessary.
  - Google Developers recommend vocab size = m =20,000; I have gotten good performance from m =2,000.

1. Drop phrases that appear in few documents, or in almost all documents.
2. filter on parts of speech (keep nouns, adjectives, and verbs).

Filtering the Vocabulary

- N-grams will blow up your feature space: filtering out uninformative n-grams is necessary.
  - Google Developers recommend vocab size $= m =$20,000; I have gotten good performance from $m =$2,000.

1. Drop phrases that appear in few documents, or in almost all documents.
2. filter on parts of speech (keep nouns, adjectives, and verbs).
3. filter on pointwise mutual information to get collocations (Ash JITE 2017, pg. 2)

Filtering the Vocabulary

- N-grams will blow up your feature space: filtering out uninformative n-grams is necessary.
  - Google Developers recommend vocab size = m =20,000; I have gotten good performance from m =2,000.

1. Drop phrases that appear in few documents, or in almost all documents.
2. filter on parts of speech (keep nouns, adjectives, and verbs).
3. filter on pointwise mutual information to get collocations (Ash JITE 2017, pg. 2)
4. supervised feature selection: select phrases that are predictive of outcome.

Feature selection using univariate comparisions

- χ^2 is a fast feature selection routine for classification tasks
  - features must be non-negative
  - works on sparse matrices
  - works on multi-class problems

```
#%% Univariate feature selection using chi2
from sklearn.feature_selection import SelectKBest, chi2,
select = SelectKBest(chi2, k=10)
Y = df['topic']==1
X_new = select.fit_transform(X, Y)
```

Feature selection using univariate comparisions

- χ^2 is a fast feature selection routine for classification tasks
  - features must be non-negative
  - works on sparse matrices
  - works on multi-class problems

```
#%% Univariate feature selection using chi2
from sklearn.feature_selection import SelectKBest, chi2,
select = SelectKBest(chi2, k=10)
Y = df['topic']==1
X_new = select.fit_transform(X, Y)
```

- With negative predictors:
  - use `f_classif`.

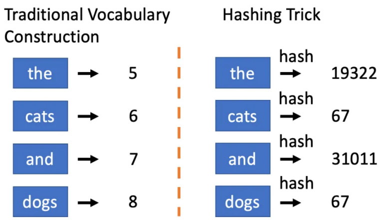
Feature selection using univariate comparisions

- χ^2 is a fast feature selection routine for classification tasks
  - features must be non-negative
  - works on sparse matrices
  - works on multi-class problems

```
#%% Univariate feature selection using chi2
from sklearn.feature_selection import SelectKBest, chi2,
select = SelectKBest(chi2, k=10)
Y = df['topic']==1
X_new = select.fit_transform(X, Y)
```

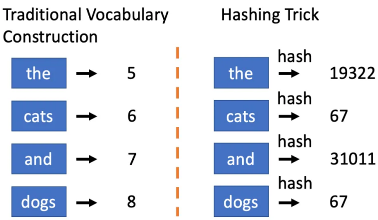
- With negative predictors:
  - use `f_classif`.
- For regression tasks:
  - use `f_regression` or OLS coefficients.

Hashing Vectorizer



- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and outputs an integer in some range (e.g. 1 to 10,000).

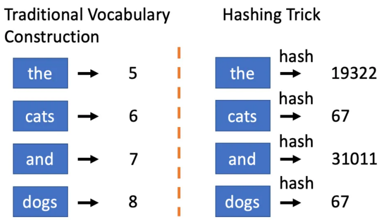
Hashing Vectorizer



- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and outputs an integer in some range (e.g. 1 to 10,000).

```
>>> from sklearn.feature_extraction.text import HashingVectorizer
>>> hv = HashingVectorizer(n_features=10)
>>> hv.transform(corpus)
<4x10 sparse matrix of type '<... 'numpy.float64'>'
    with 16 stored elements in Compressed Sparse ... format>
```

Hashing Vectorizer



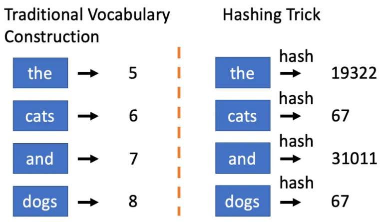
- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and outputs an integer in some range (e.g. 1 to 10,000).

```
>>> from sklearn.feature_extraction.text import HashingVectorizer
>>> hv = HashingVectorizer(n_features=10)
>>> hv.transform(corpus)
<4x10 sparse matrix of type '<... 'numpy.float64'>'
    with 16 stored elements in Compressed Sparse ... format>
```

Pros:

- can have arbitrarilly small feature space
- handles out-of-vocabulary words – any word or n-gram gets assigned to an arbitrary integer based on the hash function.

Hashing Vectorizer



- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and outputs an integer in some range (e.g. 1 to 10,000).

```
>>> from sklearn.feature_extraction.text import HashingVectorizer
>>> hv = HashingVectorizer(n_features=10)
>>> hv.transform(corpus)
<4x10 sparse matrix of type '<... 'numpy.float64'>'
    with 16 stored elements in Compressed Sparse ... format>
```

Pros:

- can have arbitrarilly small feature space
- handles out-of-vocabulary words – any word or n-gram gets assigned to an arbitrary integer based on the hash function.

Cons:

- harder to interpret features, at least not directly – but the eli5 implementation keeps track of the mapping
- collisions – n-grams will randomly be paired with each other in the feature map.
  - usually innocuous, but could sum outputs of two hashing functions to minimize this.

Named Entity Recognition

- refers to the task of identifying named entities such as "ETH Zurich" and "Marie Curie", which can be used as tokens.

[PER John Smith] , president of [ORG McCormik Industries] visited his niece [PER Paris] in [LOC Milan], reporters say .

```python
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")

for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

Parts of speech

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - **Content**: noun (NN), verb (VB), adjective (JJ), adverb (RB)
  - **Function**: determinant (DT), preposition (IN), conjunction (CC), pronoun (PR).

Parts of speech

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - **Content**: noun (NN), verb (VB), adjective (JJ), adverb (RB)
  - **Function**: determinant (DT), preposition (IN), conjunction (CC), pronoun (PR).
- Parts of speech vary in their informativeness for various functions:
  - For categorizing **topics**, nouns are usually most important
  - For **sentiment**, adjectives are usually most important.

A decent baseline for featurization

A decent baseline for featurization

- Tag parts of speech: keep nouns, verbs, and adjectives.
- Drop stopwords, capitalization, punctuation.
- Run snowball stemmer to drop word endings.

A decent baseline for featurization

- Tag parts of speech: keep nouns, verbs, and adjectives.
- Drop stopwords, capitalization, punctuation.
- Run snowball stemmer to drop word endings.
- Make bigrams from the tokens.
- drop bigrams appearing in more than half of documents, then take top 10,000 bigrams by term frequency.

A decent baseline for featurization

- Tag parts of speech: keep nouns, verbs, and adjectives.
- Drop stopwords, capitalization, punctuation.
- Run snowball stemmer to drop word endings.
- Make bigrams from the tokens.
- drop bigrams appearing in more than half of documents, then take top 10,000 bigrams by term frequency.
- Represent documents as tf-idf frequencies over these bigrams.

Application: What Drives Media Slant?

Gentzkow and Shapiro (2010)

Application: What Drives Media Slant?

Gentzkow and Shapiro (2010)

- Corpora:
  - news text from large sample of US daily newspapers.
  - congressional text is 2005 Congressional Record.

Application: What Drives Media Slant?

Gentzkow and Shapiro (2010)

- Corpora:
  - news text from large sample of US daily newspapers.
  - congressional text is 2005 Congressional Record.
- Pre-process text, stripping away prepositions, conjunctions, pronouns, and common words
  - get bigrams and trigrams

Application: What Drives Media Slant?

Gentzkow and Shapiro (2010)

- Corpora:
  - news text from large sample of US daily newspapers.
  - congressional text is 2005 Congressional Record.
- Pre-process text, stripping away prepositions, conjunctions, pronouns, and common words
  - get bigrams and trigrams
- Identify polarizing phrases using χ^2 metric.

TABLE I

MOST PARTISAN PHRASES FROM THE 2005 *CONGRESSIONAL RECORD*[a]

Panel A: Phrases Used More Often by Democrats

Two-Word Phrases		
private accounts	Rosa Parks	workers rights
trade agreement	President budget	poor people
American people	Republican party	Republican leader
tax breaks	change the rules	Arctic refuge
trade deficit	minimum wage	cut funding
oil companies	budget deficit	American workers
credit card	Republican senators	living in poverty
nuclear option	privatization plan	Senate Republicans
war in Iraq	wildlife refuge	fuel efficiency
middle class	card companies	national wildlife
Three-Word Phrases		
veterans health care	corporation for public broadcasting	cut health care
congressional black caucus	additional tax cuts	civil rights movement
VA health care	pay for tax cuts	cuts to child support
billion in tax cuts	tax cuts for people	drilling in the Arctic National
credit card companies	oil and gas companies	victims of gun violence
security trust fund	prescription drug bill	solvency of social security
social security trust	caliber sniper rifles	Voting Rights Act
privatize social security	increase in the minimum wage	war in Iraq and Afghanistan
American free trade	system of checks and balances	civil rights protections
central American free	middle class families	credit card debt

TABLE I—*Continued*

Panel B: Phrases Used More Often by Republicans

Two-Word Phrases		
stem cell	personal accounts	retirement accounts
natural gas	Saddam Hussein	government spending
death tax	pass the bill	national forest
illegal aliens	private property	minority leader
class action	border security	urge support
war on terror	President announces	cell lines
embryonic stem	human life	cord blood
tax relief	Chief Justice	action lawsuits
illegal immigration	human embryos	economic growth
date the time	increase taxes	food program
Three-Word Phrases		
embryonic stem cell	Circuit Court of Appeals	Tongass national forest
hate crimes legislation	death tax repeal	pluripotent stem cells
adult stem cells	housing and urban affairs	Supreme Court of Texas
oil for food program	million jobs created	Justice Priscilla Owen
personal retirement accounts	national flood insurance	Justice Janice Rogers
energy and natural resources	oil for food scandal	American Bar Association
global war on terror	private property rights	growth and job creation
hate crimes law	temporary worker program	natural gas natural
change hearts and minds	class action reform	Grand Ole Opry
global war on terrorism	Chief Justice Rehnquist	reform social security

[a]The top 60 Democratic and Republican phrases, respectively, are shown ranked by χ^2_{pl}. The phrases are classified as two or three word after dropping common "stopwords" such as "for" and "the." See Section 3 for details and see Appendix B (online) for a more extensive phrase list.

Consumers drive media slant (GS 2010)

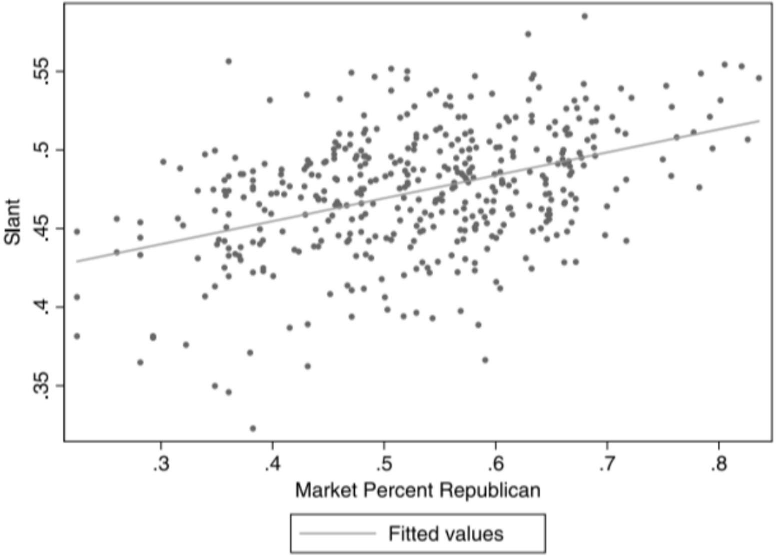


FIGURE 4.—Newspaper slant and consumer ideology. The newspaper slant index against Bush's share of the two-party vote in 2004 in the newspaper's market is shown.

Outline

Text Re-Use

- Text Re-Use algorithms (like "Smith-Waterman") measure similarity by finding and counting shared sequences in two texts above some minimum length, e.g. 10 words.
  - useful for plagiarism detection, for example.
- precise but slow
  - shortcut: look at proportion of shared (hashed) 5-grams across texts

Cosine Similarity

- We represent each document i as a vector x_i, for example x_i = term counts or x_i = IDF-weighted term frequencies.

Cosine Similarity

- We represent each document i as a vector x_i, for example $x_i =$ term counts or $x_i =$ IDF-weighted term frequencies.
- Each document is a non-negative vector in an n_x-space, where $n_x =$ vocabulary size.
  - that is, documents are rays, and similar documents have similar vectors.

Cosine Similarity

- We represent each document i as a vector x_i, for example $x_i =$ term counts or $x_i =$ IDF-weighted term frequencies.
- Each document is a non-negative vector in an n_x-space, where $n_x =$ vocabulary size.
  - that is, documents are rays, and similar documents have similar vectors.
- Can measure similarity between documents i and j by the cosine of the angle between x_i and x_j :
  - With perfectly collinear documents (that is, $x_i = \alpha x_j$, $\alpha > 0$), $\cos(0) = 1$
  - For orthogonal documents (no words in common), $\cos(\pi/2)=0$

Cosine Similarity

- We represent each document i as a vector x_i, for example x_i = term counts or x_i = IDF-weighted term frequencies.
- Each document is a non-negative vector in an n_x-space, where n_x = vocabulary size.
    - that is, documents are rays, and similar documents have similar vectors.
- Can measure similarity between documents i and j by the cosine of the angle between x_i and x_j :
    - With perfectly collinear documents (that is, $x_i = \alpha x_j$, $\alpha > 0$), $\cos(0) = 1$
    - For orthogonal documents (no words in common), $\cos(\pi/2)=0$

Cosine similarity is computable as the normalized dot product between the vectors:

$$\text{cos\_sim}(x_1, x_2) = \frac{x_1 \cdot x_2}{||x_1||||x_2||}$$

```
from sklearn.metrics.pairwise import
cosine_similarity
# between two vectors:
sim = cosine_similarity(x, y)[0,0]
# between all rows of a matrix:
sims = cosine_similarity(X)
```

Burgess et al, "Legislative Influence Detectors"

- Compare bill texts across states in two-step process:
  (1) find candidates using elasticsearch (tf-idf similarlity);
  (2) compare candidates using text reuse score.

Burgess et al, "Legislative Influence Detectors"

- Compare bill texts across states in two-step process:
  (1) find candidates using elasticsearch (tf-idf similarlity);
  (2) compare candidates using text reuse score.

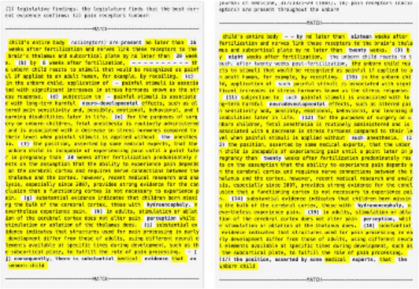


Figure 10: Match between Scott Walker's bill and a highly similar bill from Louisiana. For a detailed view, please visit http://dssg.uchicago.edu/lid/.

Burgess et al, “Legislative Influence Detectors”

- Compare bill texts across states in two-step process:
  (1) find candidates using elasticsearch (tf-idf similarlity);
  (2) compare candidates using text reuse score.

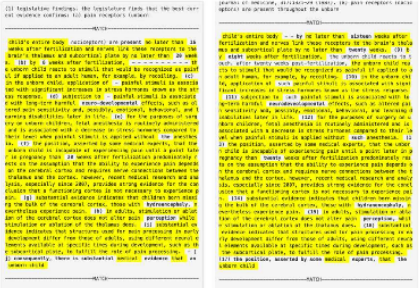


Figure 10: Match between Scott Walker's bill and a highly similar bill from Louisiana. For a detailed view, please visit http://dssg.uchicago.edu/lid/.

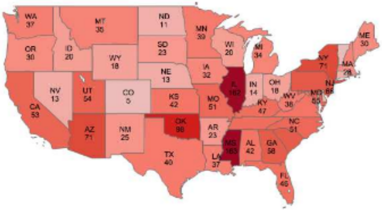


Figure 7: Introduced bills by state from ALEC model legislation

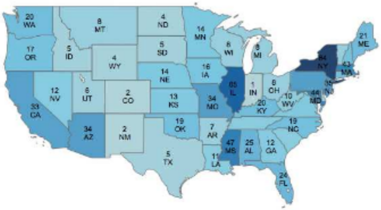


Figure 8: Introduced bills by state from ALICE model legislation

ABSTRACT

State legislatures introduce at least 45,000 bills each year. However, we lack a clear understanding of who is actually writing those bills. As legislators often lack the time and staff to draft each bill, they frequently copy text written by other states or interest groups.

However, existing approaches to detect text reuse are slow, biased, and incomplete. Journalists or researchers who want to know where a particular bill originated must perform a largely manual search. Watchdog organizations even hire armies of volunteers to monitor legislation for matches. Given the time-consuming nature of the analysis, journalists and researchers tend to limit their analysis to a subset of topics (e.g. abortion or gun control) or a few interest groups.

This paper presents the Legislative Influence Detector (LID). LID uses the Smith-Waterman local alignment algorithm to detect sequences of text that occur in model legislation and state bills. As it is computationally too expensive to run this algorithm on a large corpus of data, we use a search engine built using Elasticsearch to limit the number of comparisons. We show how LID has found 45,405 instances of bill-to-bill text reuse and 14,137 instances of model-legislation-to-bill text reuse. LID reduces the time it takes to manually find text reuse from days to seconds.

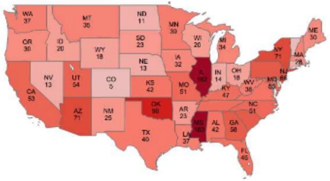


Figure 7: Introduced bills by state from ALEC model legislation

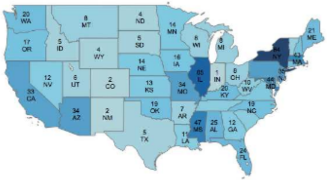


Figure 8: Introduced bills by state from ALICE model legislation

1. What is the research question?
2. Why is it important?
3. What is the problem solved?
4. What is being measured?
5. How does the measurement help answer the research question?

Text analysis of patent innovation

Kelly, Papanikolau, Seru, and Taddy (AERI 2020)

"Measuring technological innovation over the very long run"

- Data:
  - 9 million patents since 1840, from U.S. Patent Office and Google Scholar Patents.
  - date, inventor, backward citations
  - text (abstract, claims, and description)

Text analysis of patent innovation

Kelly, Papanikolau, Seru, and Taddy (AERI 2020)

"Measuring technological innovation over the very long run"

- Data:
  - 9 million patents since 1840, from U.S. Patent Office and Google Scholar Patents.
  - date, inventor, backward citations
  - text (abstract, claims, and description)
- Text pre-processing:
  - drop HTML markup, punctuation, numbers, capitalization, and stopwords.
  - remove terms that appear in less than 20 patents.
  - 1.6 million words in vocabulary.

Measuring Patent Similarity

- Each patent i = x_i= TF-IDF word features (vector with 1.6m entries)
- Compute (roughly) TF-IDF cosine similarity ρ_{ij} between patents i and j.
  - 9m$\times$9m similarity matrix = 30TB of data.
  - enforce sparsity by setting similarity $< .05$ to zero (93.4% of pairs).

Measuring Patent Similarity

- Each patent i = x_i= TF-IDF word features (vector with 1.6m entries)
- Compute (roughly) TF-IDF cosine similarity ρ_{ij} between patents i and j.
  - 9m$\times$9m similarity matrix = 30TB of data.
  - enforce sparsity by setting similarity $< .05$ to zero (93.4% of pairs).
- Validation:
  - For pairs with higher ρ_{ij}, patent j more likely to cite patent i.
  - Within technology class (assigned by patent office), similarity is higher than across class.

- "Novelty" is defined by dissimilarity (negative similarity) to previous patents:

$$\text{Novelty}_j = -\sum_{i \in B(j)} \rho_{ij}$$

where $B(j)$ is the set of previous patents (in, e.g., last 20 years).

- "Novelty" is defined by dissimilarity (negative similarity) to previous patents:

$$\text{Novelty}_j = -\sum_{i \in B(j)} \rho_{ij}$$

  where $B(j)$ is the set of previous patents (in, e.g., last 20 years).

- "Impact" is defined as similarity to subsequent patents:

$$\text{Impact}_i = \sum_{j \in F(i)} \rho_{ij}$$

  where $F(i)$ is the set of future patents (in, e.g., next 100 years).

- "Novelty" is defined by dissimilarity (negative similarity) to previous patents:

$$\text{Novelty}_j = -\sum_{i \in B(j)} \rho_{ij}$$

where $B(j)$ is the set of previous patents (in, e.g., last 20 years).

- "Impact" is defined as similarity to subsequent patents:

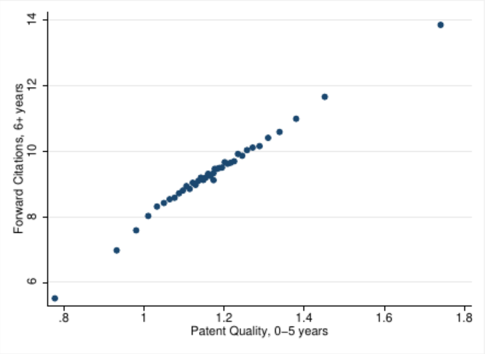
$$\text{Impact}_i = \sum_{j \in F(i)} \rho_{ij}$$

where $F(i)$ is the set of future patents (in, e.g., next 100 years).

- A patent has high **quality** if it is **novel** and **impactful**:

$$\log \text{Quality}_k = \log \text{Impact}_k + \log \text{Novelty}_k$$

- Higher quality patents get more cites:



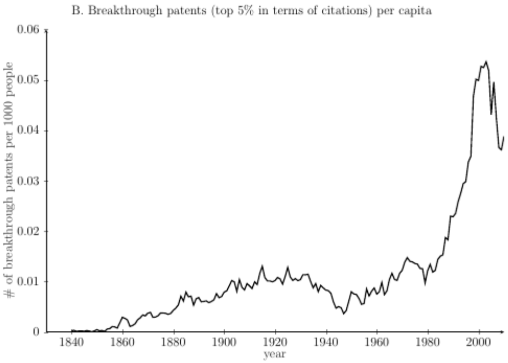
Most Innovative Firms

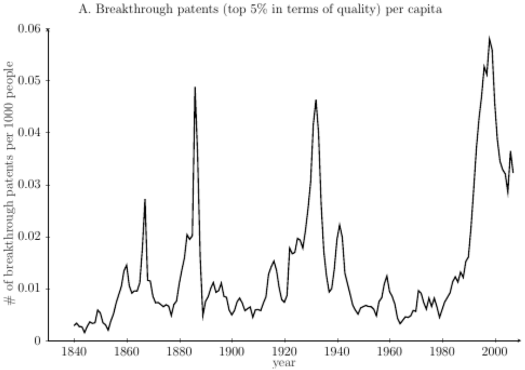
Kelly, Papanikolau, Seru, and Taddy (2018)

Assignee	First Year	# Breakthroughs
General Electric	1872	3,457
Westinghouse Electric Co.	1889	1,762
Eastman Kodak Co.	1890	2,244
Western Electric Co.	1899	1,222
AT&T (includes Bell Labs)	1899	5,645
Standard Oil Co.	1900	1,212
Dow Chemical Co.	1902	1,235
Du Pont	1905	3,353
International Business Machines	1908	14,913
American Cyanamid Co.	1909	690
Universal Oil Products Co.	1919	590
RCA	1920	3,222
Monsanto Company (inc. Monsanto Chemicals)	1921	902
Honeywell International, inc.	1928	872
General Aniline & Film Corp.	1929	1,181
Massachusetts Institute of Technology	1935	504
Philips	1939	1145
Texas Instruments	1960	2,088
Xerox	1961	2,198
Applied Materials	1971	510
Digital Equipment	1971	1,101
Hewlett-Packard Co.	1971	2,661
Intel	1971	2,629
Motorola, inc.	1971	4,129
Regents of the University of California	1971	823
United States Navy	1945	791
NCR	1973	737
Advanced Micro Devices	1974	1,195
Apple Computer	1978	864

Breakthrough patents: citations vs quality

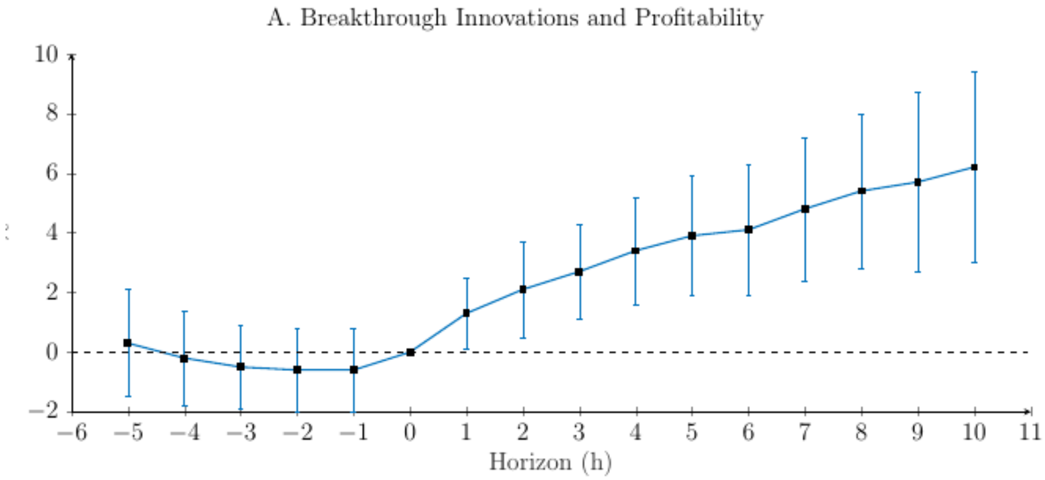
Kelly, Papanikolau, Seru, and Taddy (2018)





Breakthrough patents and firm profits

Kelly, Papanikolau, Seru, and Taddy (2018)



Outline

Machine Learning with Text Data

- We have a corpus (or dataset) D of $n_D \geq 1$ documents (or data points), whose features can be represented as a matrix of vectors $\boldsymbol{x}$ with $n_x \geq 1$ features.

Machine Learning with Text Data

- We have a corpus (or dataset) D of $n_D \geq 1$ documents (or data points), whose features can be represented as a matrix of vectors $\boldsymbol{x}$ with $n_x \geq 1$ features.
- Each document has an associated outcome or label $\boldsymbol{y}$ with dimensions $n_y \geq 1$

Machine Learning with Text Data

- We have a corpus (or dataset) D of $n_D \geq 1$ documents (or data points), whose features can be represented as a matrix of vectors $\boldsymbol{x}$ with $n_x \geq 1$ features.
- Each document has an associated outcome or label $\boldsymbol{y}$ with dimensions $n_y \geq 1$
- Some documents are unlabeled $\rightarrow$ we would like to train a model to machine-classify them.

XGBoost

- Feurer et al (2018) find that XGBoost beats a sophisticated AutoML procedure with grid search over 15 classifiers and 18 data preprocessors.
- A good starting point for any machine learning task.

```python
from xgboost import XGBClassifier
model = XGBClassifier()

model.fit(X_train, y_train,
          early_stopping_rounds=10,
          eval_metric="logloss",
          eval_set=[(X_eval, y_eval)]
          )

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

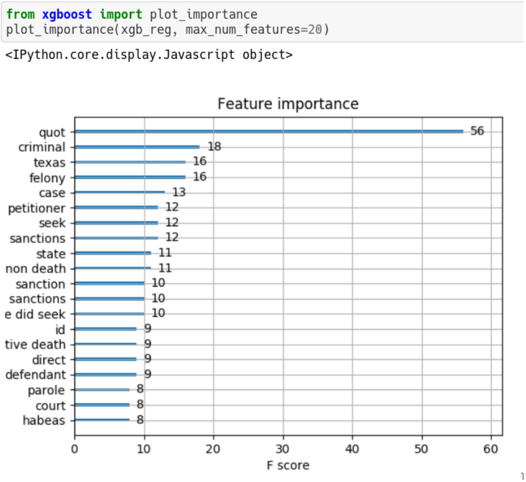
- easy to use
- actively developed
- efficient / parallelizable
- provides model explanations
- takes sparse matrices as input

Interpreting Tree Ensembles

XGBoost's Feature Importance Metric:

- At each decision node, compute **information gain** for feature j **(change in predicted probability)**.
- Average across all nodes for each j.

Ranks predictors by their relative contributions.




```
from xgboost import plot_importance
plot_importance(xgb_reg, max_num_features=10)
```

Objectives of Machine Learning Project

1. **What is the question or problem?**

Objectives of Machine Learning Project

1. **What is the question or problem?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata

Objectives of Machine Learning Project

1. **What is the question or problem?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata
3. Machine learning:
   - Select a model and train it.
   - Fine-tune hyperparameters for out-of-sample fit.
   - Interpret predictions using model explanation methods.

Objectives of Machine Learning Project

1. **What is the question or problem?**
2. Corpus and Data:
   - obtain, clean, preprocess, and link.
   - Produce descriptive visuals and statistics on the text and metadata
3. Machine learning:
   - Select a model and train it.
   - Fine-tune hyperparameters for out-of-sample fit.
   - Interpret predictions using model explanation methods.
4. Empirical analysis
   - Produce statistics or predictions with the trained model.
   - **Answer the question / solve the problem.**

Application: Predicting Political Party from Text

Andrew Peterson and Arthur Spirling, "Classification accuracy as a substantive quantity of interest: Measuring polarization in Westminster systems," ***Political Analysis*** **(2018).**

Application: Predicting Political Party from Text

Andrew Peterson and Arthur Spirling, "Classification accuracy as a substantive quantity of interest: Measuring polarization in Westminster systems," ***Political Analysis*** **(2018).**

- Machine Learning Problem:
  - Corpus $D = 3.5$M U.K. parliament speeches, 1935-2013.

Application: Predicting Political Party from Text

Andrew Peterson and Arthur Spirling, "Classification accuracy as a substantive quantity of interest: Measuring polarization in Westminster systems," *Political Analysis* (2018).

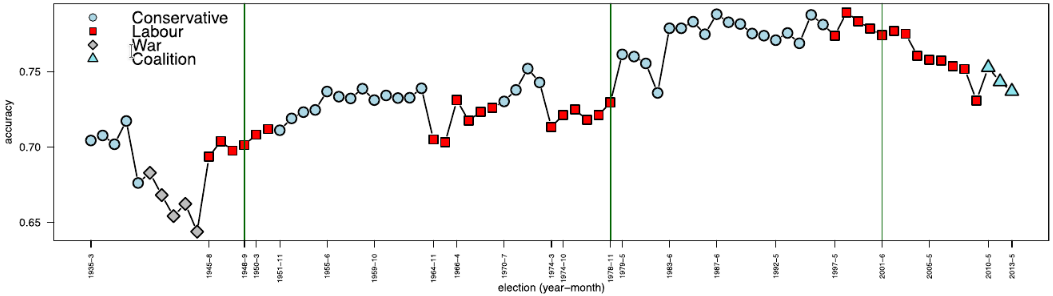
- Machine Learning Problem:
  - Corpus $D =$ 3.5M U.K. parliament speeches, 1935-2013.
  - Label $Y =$ party of speaker (Conservative or Labour)

Application: Predicting Political Party from Text

Andrew Peterson and Arthur Spirling, "Classification accuracy as a substantive quantity of interest: Measuring polarization in Westminster systems," *Political Analysis* (2018).

- Machine Learning Problem:
  - Corpus D = 3.5M U.K. parliament speeches, 1935-2013.
  - Label Y = party of speaker (Conservative or Labour)

In years that classifier is more accurate, speech is more polarized:



A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.

A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.
2. Train a machine learning model predict outcome y:
   - For classification, regularized logistic regression or xgboost classifier.
   - For regression, use elastic net or xgboost regressor.

A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.
2. Train a machine learning model predict outcome y:
   - For classification, regularized logistic regression or xgboost classifier.
   - For regression, use elastic net or xgboost regressor.
3. Use cross-validation grid search in training set to select model hyperparameters.

A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.
2. Train a machine learning model predict outcome y:
   - For classification, regularized logistic regression or xgboost classifier.
   - For regression, use elastic net or xgboost regressor.
3. Use cross-validation grid search in training set to select model hyperparameters.
4. Evaluate model in held-out test set:
   - For classification, use F1 score and confusion matrix.
   - For regression, use R squared and calibration plot.

A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.
2. Train a machine learning model predict outcome y:
   - For classification, regularized logistic regression or xgboost classifier.
   - For regression, use elastic net or xgboost regressor.
3. Use cross-validation grid search in training set to select model hyperparameters.
4. Evaluate model in held-out test set:
   - For classification, use F1 score and confusion matrix.
   - For regression, use R squared and calibration plot.
5. Interpret the model predictions:
   - for gradient boosting, use feature importance ranking.
   - for linear models, examine coefficients
   - look at highest and lowest ranked documents for $\hat{y}$

A baseline for machine learning using text

1. Take tf-idf-weighted POS-filtered bigrams (from above) as inputs X.
2. Train a machine learning model predict outcome y:
   - For classification, regularized logistic regression or xgboost classifier.
   - For regression, use elastic net or xgboost regressor.
3. Use cross-validation grid search in training set to select model hyperparameters.
4. Evaluate model in held-out test set:
   - For classification, use F1 score and confusion matrix.
   - For regression, use R squared and calibration plot.
5. Interpret the model predictions:
   - for gradient boosting, use feature importance ranking.
   - for linear models, examine coefficients
   - look at highest and lowest ranked documents for $\hat{y}$
6. Answer the research question!

Outline

Topic Models in Social Science

- Core methods for topic models were developed in computer science and statistics
  - summarize unstructured text
  - use words within document to infer subject
  - useful for dimension reduction

Topic Models in Social Science

- Core methods for topic models were developed in computer science and statistics
  - summarize unstructured text
  - use words within document to infer subject
  - useful for dimension reduction
- Social scientists use topics as a form of measurement
  - how observed covariates drive trends in language
  - tell a story not just about what, but how and why

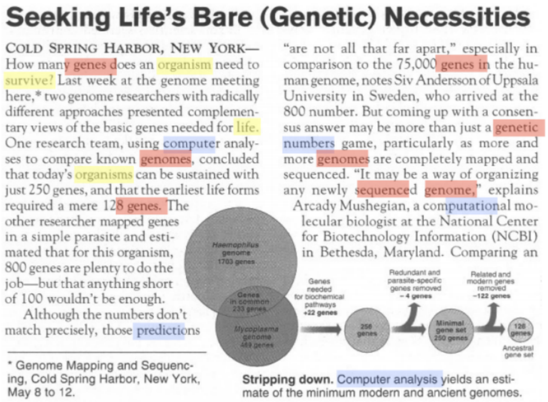
Topic Models in Social Science

- Core methods for topic models were developed in computer science and statistics
  - summarize unstructured text
  - use words within document to infer subject
  - useful for dimension reduction
- Social scientists use topics as a form of measurement
  - how observed covariates drive trends in language
  - tell a story not just about what, but how and why
  - **topic models are more interpretable** than other dimension reduction methods, such as PCA.

- **Latent Dirichlet Allocation (LDA):**
  - Each topic is a distribution over words.
  - Each document is a distribution over topics.

- **Latent Dirichlet Allocation (LDA):**
  - Each topic is a distribution over words.
  - Each document is a distribution over topics.
- Input: $N \times M$ document-term count matrix X
- Assume: there are K topics (tunable hyperparameter, use coherence).
- Like PCA or NMF, LDA works by factorizing X into:
  - an $N \times K$ document-topic matrix
  - an $K \times M$ topic-term matrix.

- **Latent Dirichlet Allocation (LDA):**
  - Each topic is a distribution over words.
  - Each document is a distribution over topics.
- Input: $N \times M$ document-term count matrix X
- Assume: there are K topics (tunable hyperparameter, use coherence).
- Like PCA or NMF, LDA works by factorizing X into:
  - an $N \times K$ document-topic matrix
  - an $K \times M$ topic-term matrix.



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

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

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

Image from Hanna Wallach

```python
# creating the term dictionary
from gensim import corpora
dictionary = corpora.Dictionary(doc_clean)
```

Converting list of documents (corpus) into Document Term Matrix using dictionary prepared above.

```python
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
```

```python
# train LDA with 10 topics and print
from gensim.models.ldamodel import LdaModel

lda = LdaModel(doc_term_matrix, num_topics=10,
               id2word = dictionary, passes=3)
lda.show_topics(formatted=False)
```

Using an LDA Model

Once trained, can easily get topic proportions for a corpus.

- for any document – doesn't have to be in training corpus.
- main topic is the highest-probability topic

Using an LDA Model

Once trained, can easily get topic proportions for a corpus.

- for any document – doesn't have to be in training corpus.
- main topic is the highest-probability topic
- documents with highest share in a topic can work as representative documents for the topic.

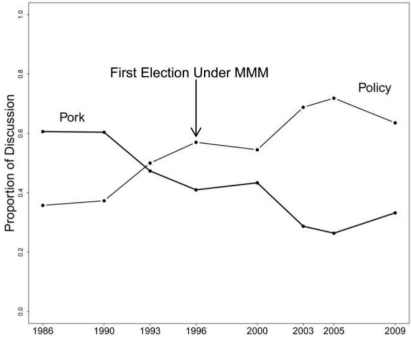
Using an LDA Model

Once trained, can easily get topic proportions for a corpus.

- for any document – doesn't have to be in training corpus.
- main topic is the highest-probability topic
- documents with highest share in a topic can work as representative documents for the topic.

Can then use the topic proportions as variables in a social science analysis.

- e.g., Catalinac (2016) shows that after a Japanese political reform that reduced intraparty competition, candidate platforms reduced local pork and increased national policy.



Topic modeling Federal Reserve Bank transcripts

Hansen, McMahon, and Prat (QJE 2017)

- Analyze speech transcripts from FOMC (Federal Open Market Committee).
  - private discussions among committee members at Federal Reserve (U.S. Central Bank)
  - 150 meetings, 20 years, 26,000 speeches, 24,000 unique words.

Topic modeling Federal Reserve Bank transcripts

Hansen, McMahon, and Prat (QJE 2017)

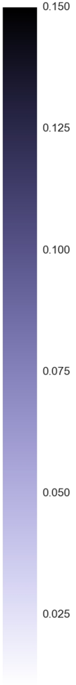
- Analyze speech transcripts from FOMC (Federal Open Market Committee).
  - private discussions among committee members at Federal Reserve (U.S. Central Bank)
  - 150 meetings, 20 years, 26,000 speeches, 24,000 unique words.
- Pre-processing:
  - drop stopwords, stems; vocab = 10,000 words

Topic modeling Federal Reserve Bank transcripts

Hansen, McMahon, and Prat (QJE 2017)

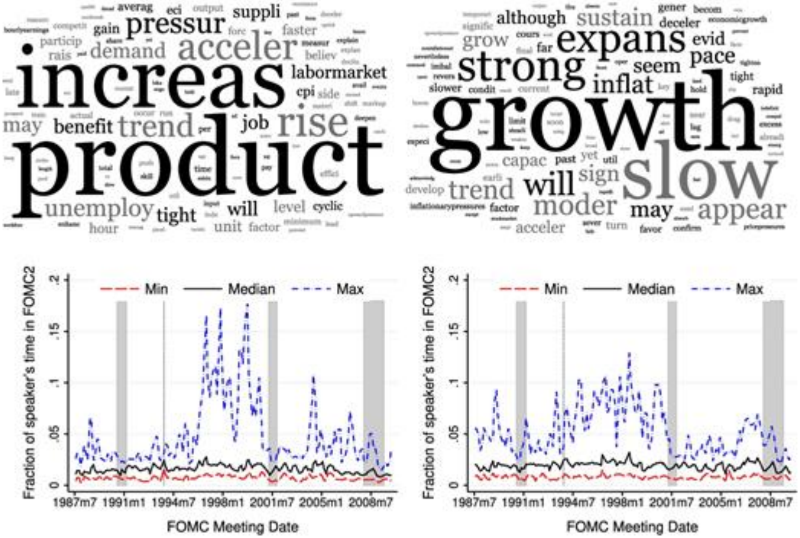
- Analyze speech transcripts from FOMC (Federal Open Market Committee).
  - private discussions among committee members at Federal Reserve (U.S. Central Bank)
  - 150 meetings, 20 years, 26,000 speeches, 24,000 unique words.
- Pre-processing:
  - drop stopwords, stems; vocab $=$ 10,000 words
- LDA:
  - $K = 40$ topics selected for interpretability / topic coherence.

													Pro-cyclicality
Topic0[1]	product	increas	wage	price	cost	labor	rise	acceler	inflat	pressur	trend	compens	0.024
Topic1[1,2]	growth	slow	economi	continu	expans	strong	trend	inflat	will	recent	slowdown	moder	0.023
Topic2[2]	inflat	expect	core	measur	higher	path	slack	gradual	continu	remain	view	suggest	0.017
Topic3[1]	percent	year	quarter	growth	month	rate	last	next	state	averag	california	employ	0.007
Topic4	number	data	look	chang	measur	use	point	show	revis	estim	gdp	actual	0.007
Topic5[1,2]	polici	inflat	monetarpol	need	time	can	monetari	move	tighten	view	action	believ	0.005
Topic6[2]	rate	term	expect	real	lower	increas	rise	level	declin	short	nomin	year	0.005
Topic7	statement	word	chang	meet	languag	discuss	issu	want	read	sentenc	view	use	0.005
Topic8[2]	chairman	support	mr	direct	recommend	agre	asymmetr	prefer	symmetr	move	toward	favor	0.004
Topic9[1]	employ	continu	growth	job	nation	region	seem	state	manufactur	greenbook	busi	bit	0.004
Topic10	dollar	unitedstates	export	countri	import	foreign	japan	growth	abroad	trade	develop	currenc	0.003
Topic11	model	use	simul	shock	effect	scenario	nairu	differ	rule	chang	baselin	altern	0.003
Topic12[2]	risk	may	balanc	seem	side	uncertainti	possibl	economi	probabl	reason	upsid	much	0.003
Topic13	forecast	greenbook	staff	project	differ	assumpt	littl	assum	somewhat	lower	end	period	0.002
Topic14	period	committe	consist	econom	run	maintain	futur	read	slightli	stabil	expect	develop	0.002
Topic15	invest	incom	spend	capit	household	consum	busi	hous	consumpt	sector	stock	stockmarket	0.002
Topic16[1]	month	report	increas	survey	expect	indic	remain	continu	last	recent	data	activ	0.002
Topic17[1]	project	forecast	year	quarter	expect	will	percent	revis	anticip	growth	next	recent	0.002
Topic18	question	ask	issu	let	want	answer	rais	discuss	don	start	without	okay	0.001
Topic19	peopl	talk	lot	much	comment	around	differ	number	realli	look	thing	hear	0.001
Topic20	presid	ye	governor	parri	stern	vice	hoenig	minehan	kelley	jordan	moskow	mcteer	0.001
Topic21	move	can	evid	signific	stage	inde	will	issu	economi	may	quit	clearli	0.001
Topic22[2]	chairman	thank	mr	time	meet	laughter	comment	let	will	point	call	may	0.0
Topic23[1]	year	panel	line	shown	right	chart	expect	project	percent	middl	left	next	0.0
Topic24	district	nation	area	continu	sector	construct	manufactur	report	activ	region	economi	remain	0.0
Topic25	know	someth	happen	right	thing	want	look	sure	can	realli	anyth	els	0.0
Topic26[1,2]	polici	might	committe	market	may	tighten	eas	risk	action	staff	possibl	potenti	-0.001
Topic27	year	continu	product	price	level	industri	will	sale	increas	auto	last	district	-0.001
Topic28[1]	inventori	product	sale	level	order	will	sector	come	good	quarter	much	adjust	-0.001
Topic29	price	oil	increas	energi	effect	import	suppli	product	demand	will	market	oilprices	-0.002
Topic30	term	might	point	can	sens	run	short	probabl	time	longer	tri	someth	-0.002
Topic31	seem	may	time	certainli	bit	littl	quit	much	far	perhap	better	might	-0.003
Topic32	money	aggreg	borrow	seem	rang	reserv	rate	target	time	altern	suggest	million	-0.003
Topic33[2]	move	market	point	will	fundsrate	rate	basispoints	need	fed	today	basi	time	-0.004
Topic34[1]	report	busi	compani	year	contact	firm	sale	worker	expect	plan	director	industri	-0.004
Topic35	will	fiscal	ta	budget	cut	govern	effect	billion	state	spend	deficit	year	-0.005
Topic36	will	economi	world	rather	problem	believ	can	situat	much	seem	view	good	-0.008
Topic37	realli	look	side	thing	lot	problem	concern	littl	pretti	situat	kind	much	-0.012
Topic38	bank	credit	market	loan	financi	debt	lend	fund	concern	financ	problem	spread	-0.018
Topic39[1,2]	economi	weak	recoveri	recess	confid	eas	neg	econom	will	turn	declin	period	-0.059



Pro-Cyclical Topics

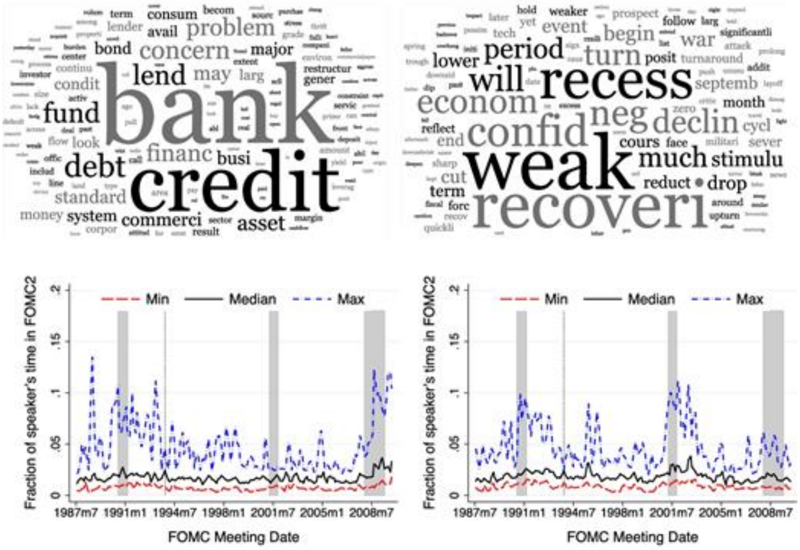
Hansen, McMahon, and Prat (QJE 2017)



(A) Topic 0 'Productivity'   (B) Topic 1 'Growth'

Counter-Cyclical Topics

Hansen, McMahon, and Prat (QJE 2017)



(A) Topic 38 'Financial sector'

(B) Topic 39 'Economic weakness'

Effect of Transparency

Hansen, McMahon, and Prat (QJE 2017)

- In 1993, there was an unexpected transparency shock where transcripts became public.

Effect of Transparency

Hansen, McMahon, and Prat (QJE 2017)

- In 1993, there was an unexpected transparency shock where transcripts became public.
- Increasing transparency results in:
  - higher discipline / technocratic language (probably beneficial)
  - higher conformity (probably costly)
- Highlights tradeoffs from transparency in bureaucratic organizations.

Structural Topic Model = LDA + Metadata

Roberts, Stewart, and Tingley

STM provides two ways to include contextual information:

- Topic prevalence can vary by metadata
  - e.g. Republicans talk about military issues more then Democrats

Structural Topic Model = LDA + Metadata

Roberts, Stewart, and Tingley

STM provides two ways to include contextual information:

- Topic prevalence can vary by metadata
  - e.g. Republicans talk about military issues more then Democrats
- Topic content can vary by metadata
  - e.g. Republicans talk about military issues more patriotically than Democrats.

Structural Topic Model = LDA + Metadata

Roberts, Stewart, and Tingley

STM provides two ways to include contextual information:

- Topic prevalence can vary by metadata
  - e.g. Republicans talk about military issues more then Democrats
- Topic content can vary by metadata
  - e.g. Republicans talk about military issues more patriotically than Democrats.

- Structural topic model is not a prediction model:
  - it will tell you which topics or features correlate with an outcome, but it will not provide an in-sample or out-of-sample prediction for an outcome

Structural Topic Model = LDA + Metadata

Roberts, Stewart, and Tingley

STM provides two ways to include contextual information:

- Topic prevalence can vary by metadata
  - e.g. Republicans talk about military issues more then Democrats
- Topic content can vary by metadata
  - e.g. Republicans talk about military issues more patriotically than Democrats.

- Structural topic model is not a prediction model:
  - it will tell you which topics or features correlate with an outcome, but it will not provide an in-sample or out-of-sample prediction for an outcome
- The main implementation is in R. gensim has a light-weight version called "author topic model".

Outline

Word2Vec & GloVe

- "Word embeddings" often refer to Word2Vec or GloVe – these are particular (popular) models for producing word embeddings.
  - the goal: represent the meaning of words by the neighboring words – their **contexts**.

Word2Vec & GloVe

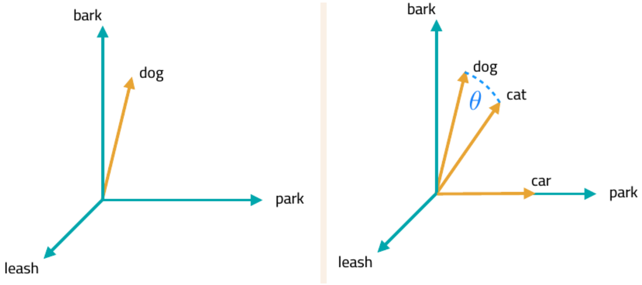
- "Word embeddings" often refer to Word2Vec or GloVe – these are particular (popular) models for producing word embeddings.
  - the goal: represent the meaning of words by the neighboring words – their **contexts**.
  - rather than predicting some metadata (such as classifying topic labels) they predict the co-occurence of neighboring words.

Word2Vec & GloVe

- "Word embeddings" often refer to Word2Vec or GloVe – these are particular (popular) models for producing word embeddings.
  - the goal: represent the meaning of words by the neighboring words – their **contexts**.
  - rather than predicting some metadata (such as classifying topic labels) they predict the co-occurence of neighboring words.
- "You shall know a word by the company it keeps":
  - "He filled the **wampimuk**, passed it around and we all drunk some."
  - "We found a little, hairy **wampimuk** sleeping behind the tree."

Word Similarity

- Once words are represented as vectors $\{v_1 = \boldsymbol{M}_{[w_1,:]},\ v_2 = \boldsymbol{M}_{[w_2,:]}, \ldots\}$, we can use linear algebra to understand the relationships between words:
  - Words that are geometrically close to each other are similar: e.g. "dog" and "cat":



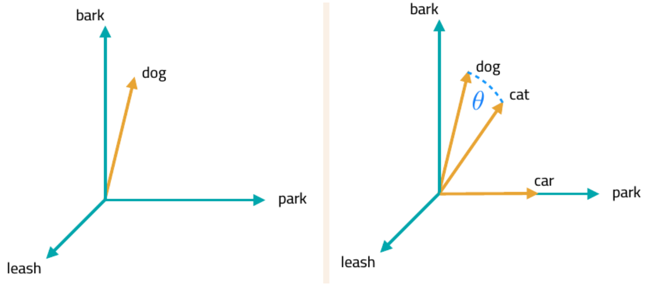
- The standard metric for comparing vectors is cosine similarity:

$$\cos\theta = \frac{v_1 \cdot v_2}{||v_1|| \, ||v_2||}$$

  - alternatives include e.g. Jaccard similarity (Goldberg 2017)

Word Similarity

- Once words are represented as vectors $\{v_1 = \boldsymbol{M}_{[w_1,:]},\ v_2 = \boldsymbol{M}_{[w_2,:]}, \ldots\}$, we can use linear algebra to understand the relationships between words:
  - Words that are geometrically close to each other are similar: e.g. "dog" and "cat":



- The standard metric for comparing vectors is cosine similarity:

$$\cos\theta = \frac{v_1 \cdot v_2}{||v_1||||v_2||}$$

  - alternatives include e.g. Jaccard similarity (Goldberg 2017)
- Thanks to linearity, can compute similarities between groups of words by averaging the groups.

Word2Vec

- When people mention "word2vec", they are usually talking about a particular word-embedding model with good performance on a range of analogy and prediction tasks.

Word2Vec

- When people mention "word2vec", they are usually talking about a particular word-embedding model with good performance on a range of analogy and prediction tasks.
- How does it learn the meaning of the word "fox"?
  - By comparing true instances of the word fox ("The quick brown **fox** jumps over the lazy dog")
  - to fake (randomly sampled) ones ("The prescription of **fox** is advised for this diagnosis")

Word2Vec

- When people mention "word2vec", they are usually talking about a particular word-embedding model with good performance on a range of analogy and prediction tasks.
- How does it learn the meaning of the word "fox"?
  - By comparing true instances of the word fox ("The quick brown **fox** jumps over the lazy dog")
  - to fake (randomly sampled) ones ("The prescription of **fox** is advised for this diagnosis")
- Word2Vec learns embedding vectors for the target word ("fox") and context words (neighbors of "fox") to distinguish true from false samples.

```python
# train the model
from gensim.models import Word2Vec
w2v = Word2Vec(sentences,  # list of tokenized sentences
               workers = 8, # Number of threads to run in parallel
               size=300,  # Word vector dimensionality
               min_count =  25, # Minimum word count
               window = 5, # Context window size
               sample = 1e-3, # Downsample setting for frequent words
               )

# done training, so delete context vectors
w2v.init_sims(replace=True)

w2v.save('w2v-vectors.pkl')
```

```python
w2v.wv.most_similar('man') # most similar words
```

```
[('christ', 0.7512136697769165),
 ('woman', 0.7265682220458984),
 ('jesus', 0.7187944650650024),
 ('satan', 0.6972118616104126),
 ('lord', 0.6948500275611877),
 ('god', 0.6891006231307983),
```

GloVe Embeddings

- Pennington et al (2014) (GloVe = Global Vectors) take a different (non-neural-net) approach.
- Input: C_{ij} = local co-occurrence counts between words $i, j \in \{1, ..., n_w\}$ within some co-occurence window, e.g. ten words.

GloVe Embeddings

- Pennington et al (2014) (GloVe = Global Vectors) take a different (non-neural-net) approach.
- Input: C_{ij} = local co-occurrence counts between words $i, j \in \{1, ..., n_w\}$ within some co-occurence window, e.g. ten words.

Learn word vectors $\boldsymbol{w} = (w_1, ..., w_i, ..., w_{n_w})$, where $w_i \in (-1, 1)^{n_E}$, to solve

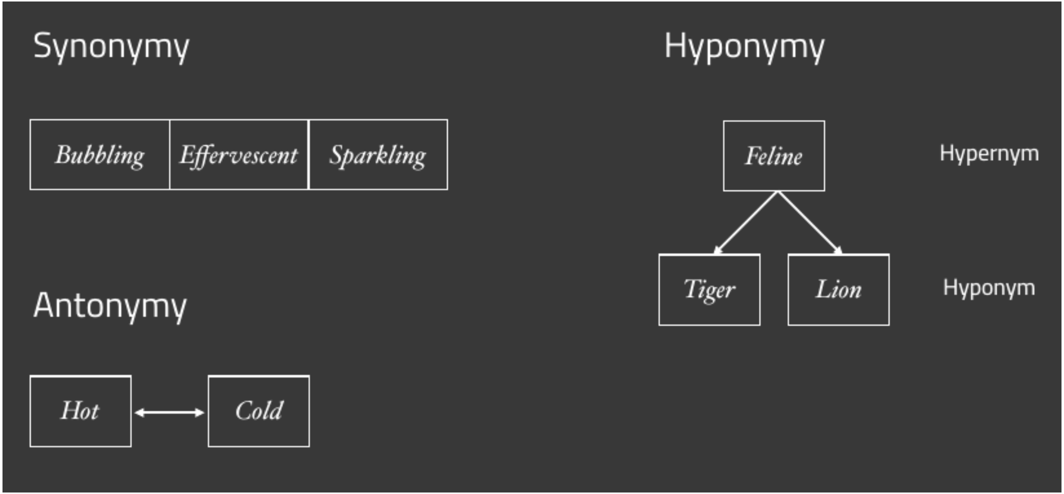
$$\min_{\boldsymbol{w}} \sum_{i,j} f\left(C_{ij}\right) \left(w_i^T w_j - \log\left(C_{ij}\right)\right)^2$$

where $f(\cdot)$ is weighting function to down-weight frequent words.

- Minimizes **squared difference** between**:**
  - **dot product of word vectors,** $w_i^T w_j$
  - **empirical co-occurrence,** $\log\left(C_{ij}\right)$
- Intuitively: words that co-occur should have high correlation (dot product)

Word Embeddings Encode Linguistic Relations

Word Embeddings Encode Linguistic Relations



Similarity vs. Relatedness (Budansky and Hirst, 2006)

- Semantic **similarity**: words sharing salient attributes / features
  - synonymy (car / automobile)
  - hypernymy (car / vehicle)
  - co-hyponymy (car / van / truck)

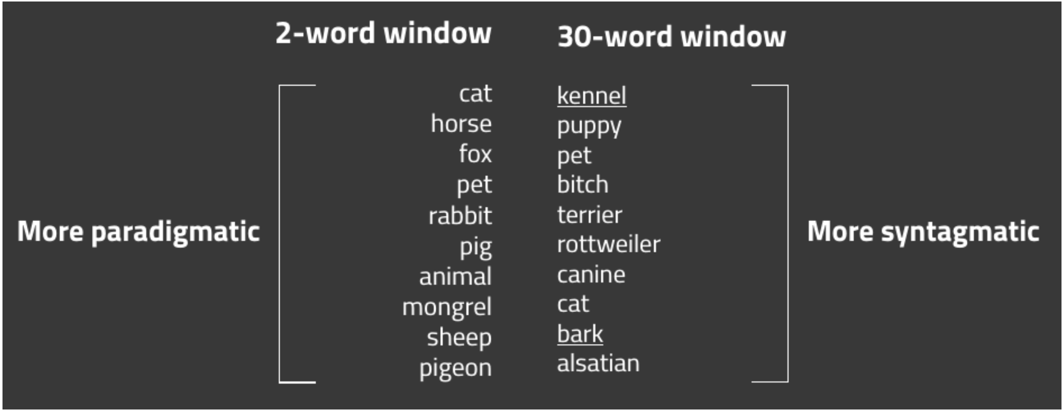
Similarity vs. Relatedness (Budansky and Hirst, 2006)

- Semantic **similarity**: words sharing salient attributes / features
  - synonymy (car / automobile)
  - hypernymy (car / vehicle)
  - co-hyponymy (car / van / truck)
- Semantic **relatedness**: words semantically associated without necessarily being similar
  - function (car / drive)
  - meronymy (car / tire)
  - location (car / road)
  - attribute (car / fast)

Similarity vs. Relatedness (Budansky and Hirst, 2006)

- Semantic **similarity**: words sharing salient attributes / features
  - synonymy (car / automobile)
  - hypernymy (car / vehicle)
  - co-hyponymy (car / van / truck)
- Semantic **relatedness**: words semantically associated without necessarily being similar
  - function (car / drive)
  - meronymy (car / tire)
  - location (car / road)
  - attribute (car / fast)
- Word embeddings will recover one or both of these relations, depending on how contexts and associated are constructed.

Most similar words to “dog”, depending on context window size



- Small windows pick up substitutable words; large windows pick up topics.

The black sheep problem

- The trivial or obvious features of a word are not mentioned in standard corpora.

The black sheep problem

- The trivial or obvious features of a word are not mentioned in standard corpora.
- For example, although most sheep are white, you rarely see the phrase “white sheep”.
  - so word2vec tells you sim(black,sheep) $>$ sim(white,sheep).

The black sheep problem

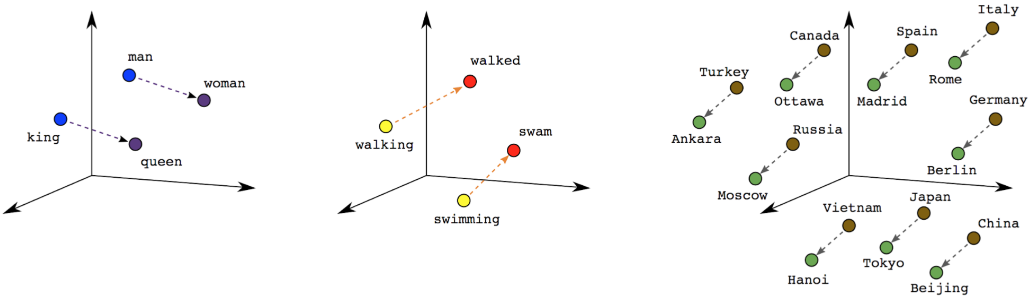
- The trivial or obvious features of a word are not mentioned in standard corpora.
- For example, although most sheep are white, you rarely see the phrase “white sheep”.
  - so word2vec tells you sim(black,sheep) $>$ sim(white,sheep).
- This is really important when we will use embeddings to anayze beliefs/attitudes.

The black sheep problem

- The trivial or obvious features of a word are not mentioned in standard corpora.
- For example, although most sheep are white, you rarely see the phrase “white sheep”.
  - so word2vec tells you sim(black,sheep) $>$ sim(white,sheep).
- This is really important when we will use embeddings to anayze beliefs/attitudes.
- Relatedly, antonyms are often rated similarly, have to be careful with that.

Vector Directions ↔ Meaning

- Intriguingly, word2vec algebra can depict conceptual, analogical relationships between words:



Male-Female Verb Tense Country-Capital

Word Embeddings for Analogies

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

Word Embeddings for Analogies

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- More generally: The analogy $a_1 : b_1 :: a_2 : b_2$ can be solved (that is, find b_2 given a_1, b_1, a_2) by

$$\arg\max_{b_2 \in V} \cos(b_2, a_2 - a_1 + b_1)$$

  where V excludes (a_1, b_1, a_2).

Word Embeddings for Analogies

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- More generally: The analogy $a_1 : b_1 :: a_2 : b_2$ can be solved (that is, find b_2 given a_1, b_1, a_2) by
$$\arg\max_{b_2 \in V} \cos(b_2, a_2 - a_1 + b_1)$$
where V excludes (a_1, b_1, a_2).
- Often works better with normalized vectors (so that one long vector doesn't wash out the others)

Word Embeddings for Analogies

$$vec(king) - vec(man) + vec(woman) \approx vec(queen)$$

- More generally: The analogy $a_1 : b_1 :: a_2 : b_2$ can be solved (that is, find b_2 given a_1, b_1, a_2) by
$$\arg\max_{b_2 \in V} \cos(b_2, a_2 - a_1 + b_1)$$
where V excludes (a_1, b_1, a_2).
- Often works better with normalized vectors (so that one long vector doesn't wash out the others)
- Levy and Goldberg (2014) recommend the following "CosMul" metric which tends to perform better:
$$\arg\max_{b_2 \in V} \frac{\cos(b_2, a_2)\cos(b_2, b_1)}{\cos(b_2, a_1) + \epsilon}$$
  - requires normalized, non-negative vectors (can transform using $(x + 1)/2$)
  - ϵ is a small smoothing parameter.

Tokenizing for Word Embeddings

- drop capitalization
- punctuation is optional
- don't drop stopwords/function-words
- add special tokens for start of sentence and end of sentence
- for out-of-vocab words, substitute a special token or replace with part-of-speech tag

Can cluster word embeddings to produce topics

Cluster #	Top 10 Words
174	complicate, depend, crucial, illustrate, elusive, focus, important, straightforward, elide, critical
134	implausible, problematic, exaggeration, skeptical, ascribe, discredit, contradictory, weak, exaggerate, supportable
75	reverse, AFFIRM, affirm, vacate, reversed, REMANDED, forego, foregoing, forgoing, remands
70	importation, import, ecstasy, marihuana, illicit, opium, distilled, export, phencyclidine, narcotic
178	perverse, sensible, tempt, unlikely, unwise, anomalous, would, easy, costly, attractive
32	phrase, meaning, word, synonymous, language, interpret, noun, wording, verb, adjective
169	circumscribe, endow, unfettered, vest, unlimited, boundless, broad, constrain, exercise, unbounded
85	hundred, thousand, many, million, huge, massive, large, enormous, most, dozen
28	emphasis, bracket, alteration, citation, footnote, italic, ellipsis, petcitation, idcitation, punctuation
138	logo, symbol, stylized, imprint, emblem, grille, prefix, lettering, suffix, crosshair
181	wilful, carelessness, recklessness, careless, intentional, willful, conscious, reckless, unintentional, wantonness
158	rigorous, demanding, heightened, reasonableness, rigid, heighten, objective, deferential, flexible, particular
55	agreement, contract, contractual, promise, novation, repudiate, guaranty, enforceable, novate, repurchase
197	summation, admonish, sidebar, prosecutor, admonishment, mistrial, curative, questioning, remark, recess
120	scrivener, typographical, reversible, plain, harmless, clerical, invited, clear, requiresthe, instructional
15	adjudicatory, adjudicative, adversarial, judicial, rulemaking, decisionmaking, administrative, meaningful, rulemake, agency

Clustered word embeddings in judicial opinions, from Ash and Nikolaus (2020)

Pre-trained word embeddings

- In many settings (e.g. a small corpus), better to use pre-trained embeddings.

```
import spacy
en = spacy.load('en_core_web_lg') # higher-quality vectors (but 800MB)
apple = en('apple')
apple.vector[:10] # vector for 'apple'

[158]: array([-0.36391 ,  0.43771 , -0.20447 , -0.22889 , -0.14227 ,  0.27396 ,
       -0.011435, -0.18578 ,  0.37361 ,  0.75339 ], dtype=float32)

[159]: apple.similarity(apple)

[159]: 1.0

[166]: orange = en('orange')
       apple.similarity(orange)

[166]: 0.5618917538704213
```

- e,g, spaCy's GloVe embeddings:
  - one million vocabulary entries, 300-dimensional vectors, trained on the Common Crawl corpus
- Can initialize models with pre-trained embeddings, can fine-tune as needed.

Implicit attitudes (Caliskan, Bryson, and Narayanan 2017)

Implicit attitudes (Caliskan, Bryson, and Narayanan 2017)

"Attitudes that affect our understanding, actions, and decisions in an unconscious manner" (Kirnan institute, OSU)

Implicit attitudes (Caliskan, Bryson, and Narayanan 2017)

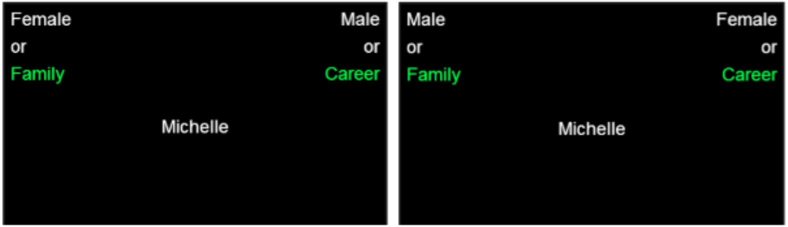
"Attitudes that affect our understanding, actions, and decisions in an unconscious manner" (Kirnan institute, OSU)

- Generally measured using Implicit Association Tests (IATs)

Implicit attitudes (Caliskan, Bryson, and Narayanan 2017)

"Attitudes that affect our understanding, actions, and decisions in an unconscious manner" (Kirnan institute, OSU)

- Generally measured using Implicit Association Tests (IATs)
- Subjects asked to assign words to categories (Greenwald et al. 1998)

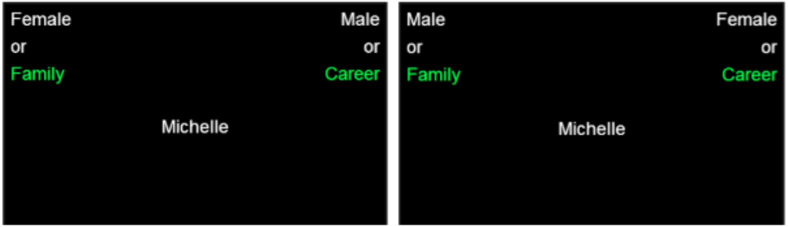


- Comparing reaction times across trials with different word pairs:
  - subjects tend to be slower and more error-prone in assignments against stereotype (e.g. "Michelle" goes to "Female or Career").

Implicit attitudes (Caliskan, Bryson, and Narayanan 2017)

"Attitudes that affect our understanding, actions, and decisions in an unconscious manner" (Kirnan institute, OSU)

- Generally measured using Implicit Association Tests (IATs)
- Subjects asked to assign words to categories (Greenwald et al. 1998)



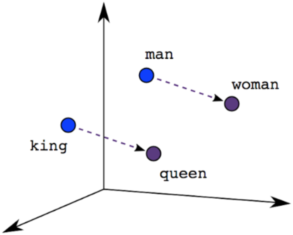
- Comparing reaction times across trials with different word pairs:
  - subjects tend to be slower and more error-prone in assignments against stereotype (e.g. "Michelle" goes to "Female or Career").
  - IAT score = difference in reaction time between stereotype-consistent and stereotype-inconsistent rounds.

Caliskan, Bryson, and Narayanan (2017)

- "We replicated a spectrum of known biases, as measured by the Implicit Association Test, using a widely used, purely statistical machine-learning model trained on a standard corpus of text from the World Wide Web. . . "

Caliskan, Bryson, and Narayanan (2017)

- “We replicated a spectrum of known biases, as measured by the Implicit Association Test, using a widely used, purely statistical machine-learning model trained on a standard corpus of text from the World Wide Web. . . “

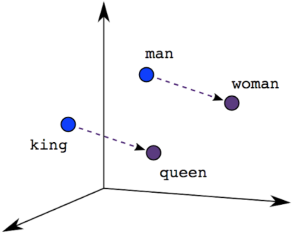


Analogies

- king : queen :: man : woman
- walked : walking :: swam : swimming

Caliskan, Bryson, and Narayanan (2017)

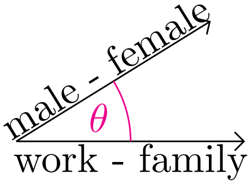
- "We replicated a spectrum of known biases, as measured by the Implicit Association Test, using a widely used, purely statistical machine-learning model trained on a standard corpus of text from the World Wide Web. . . "



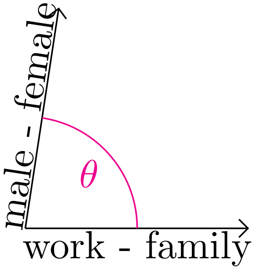
Analogies

- king : queen :: man : woman
- walked : walking :: swam : swimming
- **man : programmer :: woman : homemaker**
- **he : physician :: she : nurse**

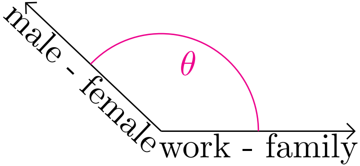
Measuring Gender Stereotypes using Cosine Similarity



(a)



(b)



(c)

Example Stimuli

- Targets:
  - **Flowers**: aster, clover, hyacinth, marigold, poppy, azalea, crocus, iris, orchid, rose, bluebell, daffodil, lilac, pansy, tulip, buttercup, daisy, lily, peony, violet, carnation, gladiola, magnolia, petunia, zinnia.
  - **Insects**: ant, caterpillar, flea, locust, spider, bedbug, centipede, fly, maggot, tarantula, bee, cockroach, gnat, mosquito, termite, beetle, cricket, hornet, moth, wasp, blackfly, dragonfly, horsefly, roach, weevil.

Example Stimuli

- Targets:
  - **Flowers**: aster, clover, hyacinth, marigold, poppy, azalea, crocus, iris, orchid, rose, bluebell, daffodil, lilac, pansy, tulip, buttercup, daisy, lily, peony, violet, carnation, gladiola, magnolia, petunia, zinnia.
  - **Insects**: ant, caterpillar, flea, locust, spider, bedbug, centipede, fly, maggot, tarantula, bee, cockroach, gnat, mosquito, termite, beetle, cricket, hornet, moth, wasp, blackfly, dragonfly, horsefly, roach, weevil.
- Attributes:
  - **Pleasant**: caress, freedom, health, love, peace, cheer, friend, heaven, loyal, pleasure, diamond, gentle, honest, lucky, rainbow, diploma, gift, honor, miracle, sunrise, family, happy, laughter, paradise, vacation.
  - **Unpleasant**: abuse, crash, filth, murder, sickness, accident, death, grief, poison, stink, assault, disaster, hatred, pollute, tragedy, divorce, jail, poverty, ugly, cancer, kill, rotten, vomit, agony, prison.

Results

- Pleasant vs. Unpleasant?
  - Flowers vs. Insects
  - Musical instruments vs. weapons.

Results

- Pleasant vs. Unpleasant?
  - Flowers vs. Insects
  - Musical instruments vs. weapons.
  - European-American names vs. African-American names

Results

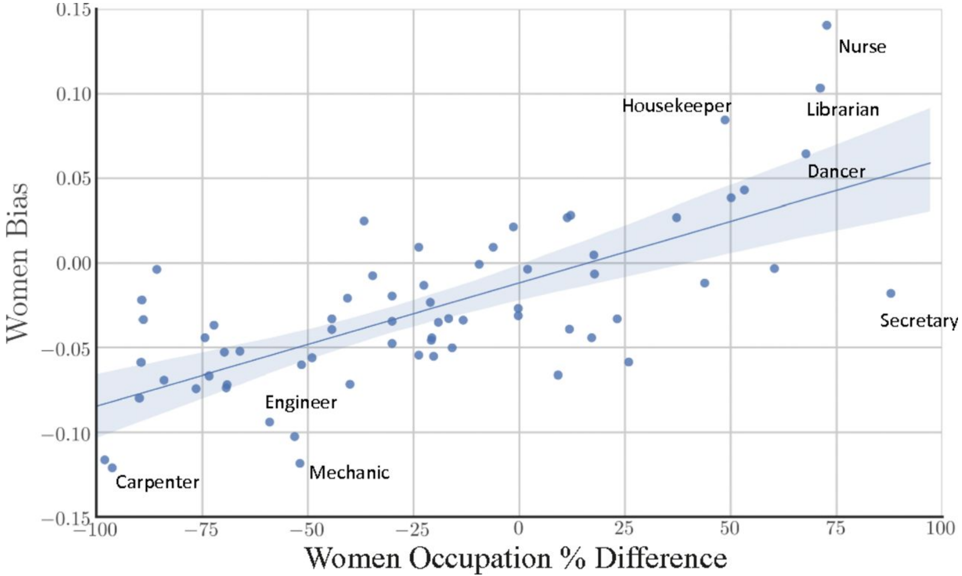
- Pleasant vs. Unpleasant?
  - Flowers vs. Insects
  - Musical instruments vs. weapons.
  - European-American names vs. African-American names
- Male names vs. Female names:

Results

- Pleasant vs. Unpleasant?
  - Flowers vs. Insects
  - Musical instruments vs. weapons.
  - European-American names vs. African-American names
- Male names vs. Female names:
  - Career words (e.g. professional, corporation, ...) vs. family words (e.g. home, children, ...)
  - Math/science words vs arts words

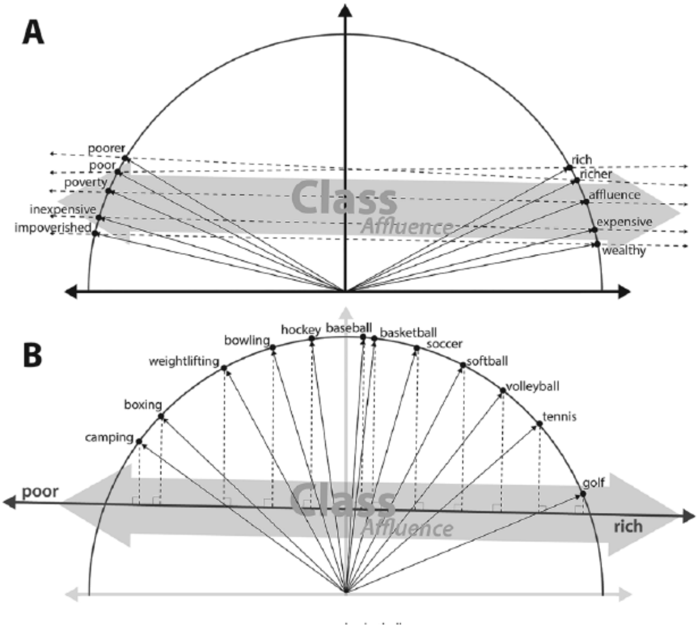
What do we learn from this?

Garg, Schiebinger, Jurafsky, and Zou (PNAS 2018)



Women's occupation relative percentage vs. embedding bias in Google News vectors.

Kozlowski, Evans, and Taddy (ASR 2019)



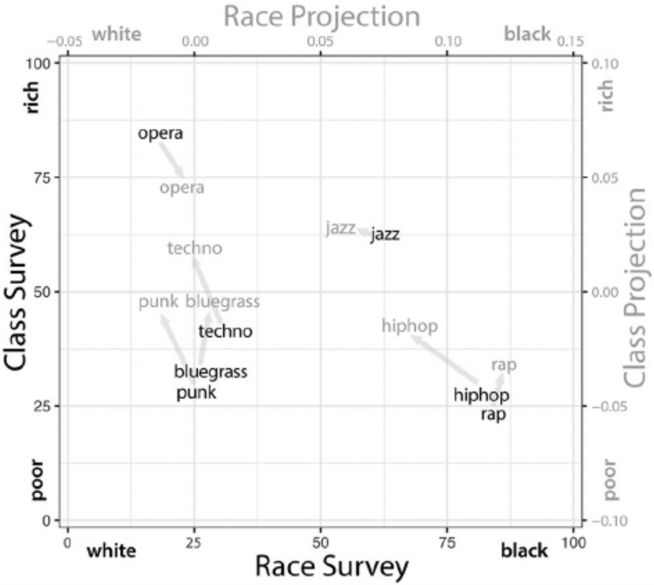


Figure 3. Projection of Music Genres onto Race and Class Dimensions of the Google News Word Embedding (Gray) and Average Survey Ratings for Race and Class Associations (Black)

Time Series Analysis of Affluence

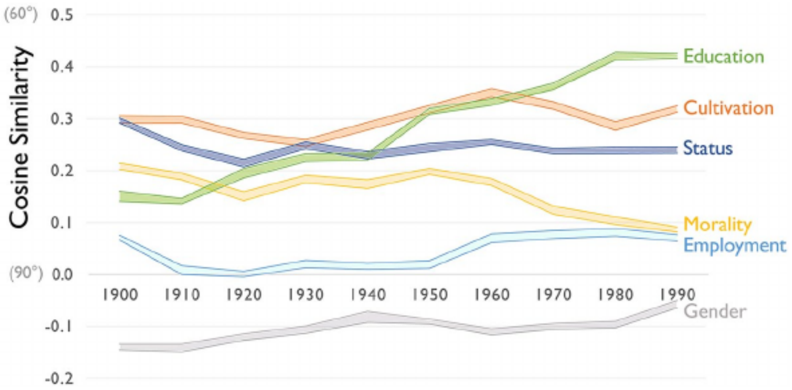


Figure 5. Cosine Similarity between the Affluence Dimension and Six Other Cultural Dimensions of Class by Decade; 1900 to 1999 Google Ngrams Corpus
Note: Bands represent 90 percent bootstrapped confidence intervals produced by subsampling.

"Among the 10 nouns most highly projecting on the affluence dimension in the first decade of the twentieth century are "fragrance," "perfume," "jewels," and "gems," ..."

Measuring stereotypical beliefs in the judiciary (Ash, Chen, and Ornaghi 2021)

- We do not have IAT scores for sitting judges

Measuring stereotypical beliefs in the judiciary (Ash, Chen, and Ornaghi 2021)

- We do not have IAT scores for sitting judges
- Proposed solution: proxy for IAT using large amounts of written text: **judicial opinions**.

Measuring stereotypical beliefs in the judiciary (Ash, Chen, and Ornaghi 2021)

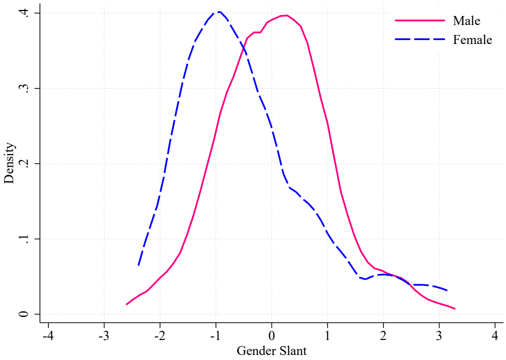
- We do not have IAT scores for sitting judges
- Proposed solution: proxy for IAT using large amounts of written text: **judicial opinions**.

Adjectives associated with

Male and **Female**

in **judicial opinion text**.

Gender Slant, by Judge Gender



Distribution of the slant measure (cosine similarity between the gender and career-family dimensions), by judge gender. (p=0.012)

Does Gender Stereotyping Matter? (Ash, Chen, and Ornaghi 2021)

Does Gender Stereotyping Matter? (Ash, Chen, and Ornaghi 2021)

1. It matters for **decisions**: More stereotyped judges tend to vote against expanding women's rights.

Does Gender Stereotyping Matter? (Ash, Chen, and Ornaghi 2021)

1. It matters for **decisions**: More stereotyped judges tend to vote against expanding women's rights.

2. It matters for **treatment of colleagues**: More stereotyped judges more likely to reverse female judges and less likely to cite them.

Does Gender Stereotyping Matter? (Ash, Chen, and Ornaghi 2021)

1. It matters for **decisions**: More stereotyped judges tend to vote against expanding women's rights.

2. It matters for **treatment of colleagues**: More stereotyped judges more likely to reverse female judges and less likely to cite them.

3. It reshapes the **language of the law**, which could influence culture and society.

Outline

Vectorizing Documents

- Quantitative analysis of language requires that documents be transformed to numbers – that is, vectors.
- We started with the baseline approach: documents become sparse vectors of token counts/frequencies.

Vectorizing Documents

- Quantitative analysis of language requires that documents be transformed to numbers – that is, vectors.
- We started with the baseline approach: documents become sparse vectors of token counts/frequencies.
  - high-dimensionality can cause issues, but sparsity mitigates.
  - can use documents of arbitrary length
  - can capture local word order with n-grams, but long-run word order is lost.

From Word Vectors to Document Vectors

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- The "continuous bag of words" representation for document D is the sum, or the average (potentially weighted by a_w), of the vectors $\vec{w}$ for each word w in ahe document.
  - word vectors $\vec{w}$ constructed using Word2Vec or GloVe (pre-trained or trained on the corpus).
  - "Document" could be sentence, paragraph, section, etc.

Document Vectors

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens:
  - drop stopwords
  - filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)

Document Vectors

$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens:
  - drop stopwords
  - filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
- Token weighting:
  - set a_w to weight words by inverse term frequency or inverse document frequency (that is, up-weight rare/informative words)

Document Vectors

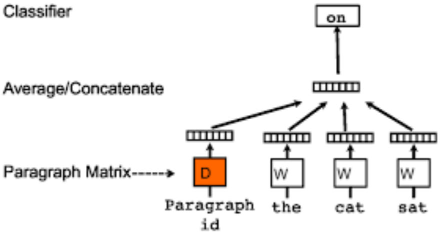
$$\vec{D} = \sum_{w \in D} a_w \vec{w}$$

- Can filter tokens:
  - drop stopwords
  - filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
- Token weighting:
  - set a_w to weight words by inverse term frequency or inverse document frequency (that is, up-weight rare/informative words)
  - Arora, Liang, and Ma (2016) provide a "tough to beat baseline", the SIF-weighted ("smoothed inverse frequency") average of the vectors:

    $$a_w = \frac{\alpha}{\alpha + p_w}$$

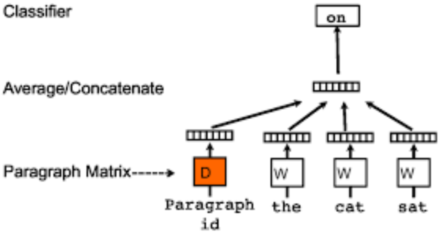
    where p_w is the probability (frequency) of the word and $\alpha = .001$ is a smoothing parameter.

Doc2Vec (Le and Mikolov)



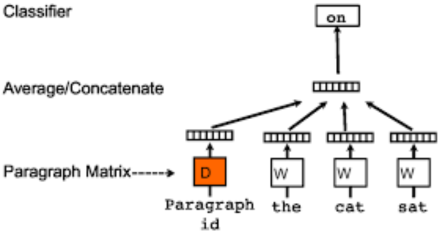
- Doc2Vec generalizes Word2Vec to documents:
  - predict a word using both the immediate neighbors, as well as **a bag-of-words representation of the whole document**.

Doc2Vec (Le and Mikolov)



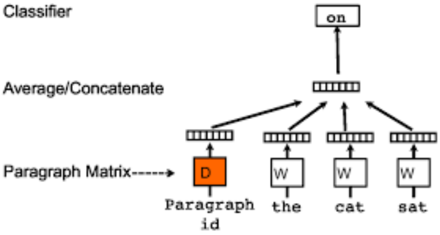
- Doc2Vec generalizes Word2Vec to documents:
  - predict a word using both the immediate neighbors, as well as **a bag-of-words representation of the whole document**.
- In Doc2Vec, both words **and documents** are assigned a learned vector representation through an embedding layer.

Doc2Vec (Le and Mikolov)



- Doc2Vec generalizes Word2Vec to documents:
  - predict a word using both the immediate neighbors, as well as **a bag-of-words representation of the whole document**.
- In Doc2Vec, both words **and documents** are assigned a learned vector representation through an embedding layer.
- Just as directions in word space encode semantic information about the words, directions in document space encode topical information about the documents.

Doc2Vec (Le and Mikolov)



- Doc2Vec generalizes Word2Vec to documents:
  - predict a word using both the immediate neighbors, as well as **a bag-of-words representation of the whole document**.
- In Doc2Vec, both words **and documents** are assigned a learned vector representation through an embedding layer.
- Just as directions in word space encode semantic information about the words, directions in document space encode topical information about the documents.
- In topic models, each dimension has a topical interpretation; in document embeddings, a direction (might) have a topical interpretation.

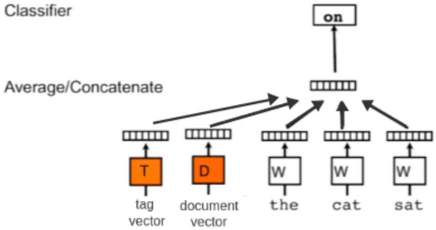
Doc2Vec in gensim

```
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
doc_iterator = [TaggedDocument(doc, [i]) for i, doc in enumerate(docs)]
d2v = Doc2Vec(doc_iterator,
                min_count=10, # minimum word count
                window=10,    # window size
                vector_size=200, # size of document vector
                sample=1e-4,
                negative=5,
                workers=4, # threads
                #dbow_words = 1 # uncomment to get word vectors too
                max_vocab_size=1000) # max vocab size
```

- can train both document vectors and word vectors.
- can get similarity between documents, and use clustering to get groups of related documents.

Tagged Documents for Classifier Features

- Can add additional non-unique document "tags"; these will be embedded separately from the unique doc ID:




```
In [168]: tagged_docs[3]

Out[168]: TaggedDocument(words=['aftershere', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen', 'design', 'beforeher
          e', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen', 'design', 'afterhere', 'project', 'finishing', 'stag
          es', 'home', 'decor', 'kitchen', 'design', 'beforehere', 'project', 'finishing', 'stages', 'home', 'decor', 'kitchen',
           'design'], tags=['Remodeling & Renovating', 'SENT_3'])
```

- will improve performance if using the embeddings to classify the tag.

Doc2Vec on Wikipedia

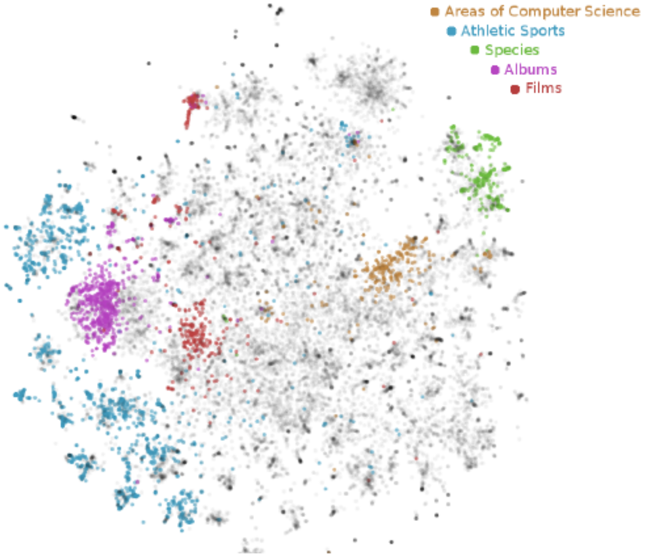


Figure 3: Visualization of Wikipedia paragraph vectors using t-SNE.

Table 5: arXiv nearest neighbours to “Distributed Representations of Sentences and Documents” using Paragraph Vectors.

Title	Cosine Similarity
Evaluating Neural Word Representations in Tensor-Based Compositional Settings	0.771
Polyglot: Distributed Word Representations for Multilingual NLP	0.764
Lexicon Infused Phrase Embeddings for Named Entity Resolution	0.757
A Convolutional Neural Network for Modelling Sentences	0.747
Distributed Representations of Words and Phrases and their Compositionality	0.740
Convolutional Neural Networks for Sentence Classification	0.735
SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation	0.735
Exploiting Similarities among Languages for Machine Translation	0.731
Efficient Estimation of Word Representations in Vector Space	0.727
Multilingual Distributed Representations without Word Alignment	0.721

Table 2: Wikipedia nearest neighbours

(a) Wikipedia nearest neighbours to "Lady Gaga" using Paragraph Vectors. All articles are relevant.

Article	Cosine Similarity
Christina Aguilera	0.674
Beyonce	0.645
Madonna (entertainer)	0.643
Artpop	0.640
Britney Spears	0.640
Cyndi Lauper	0.632
Rihanna	0.631
Pink (singer)	0.628
Born This Way	0.627
The Monster Ball Tour	0.620

(b) Wikipedia nearest neighbours to "Lady Gaga" - "American" + "Japanese" using Paragraph Vectors. Note that Ayumi Hamasaki is one of the most famous singers, and one of the best selling artists in Japan. She also has an album called "Poker Face" in 1998.

Article	Cosine Similarity
Ayumi Hamasaki	0.539
Shoko Nakagawa	0.531
Izumi Sakai	0.512
Urbangarde	0.505
Ringo Sheena	0.503
Toshiaki Kasuga	0.492
Chihiro Onitsuka	0.487
Namie Amuro	0.485
Yakuza (video game)	0.485
Nozomi Sasaki (model)	0.485

Outline

Beyond Word Order

- The models we have seen so far have counted words and phrases, or embedded sequences
  - the only language structure used is the ordering of words.

Beyond Word Order

- The models we have seen so far have counted words and phrases, or embedded sequences
  - the only language structure used is the ordering of words.
- How to identify whether the defendant was negligent?
  - "The negligent defendant"
  - "The defendant was negligent"
  - "The defendant, a driver, was negligent"

Beyond Word Order

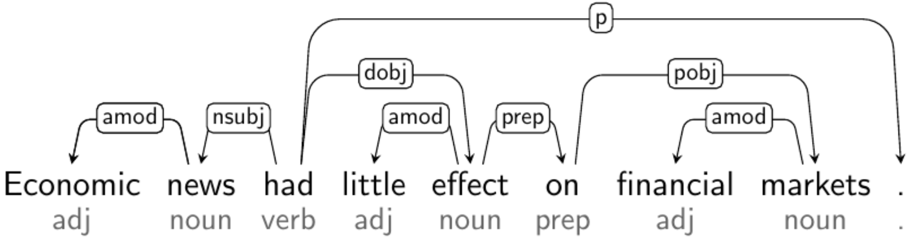
- The models we have seen so far have counted words and phrases, or embedded sequences
  - the only language structure used is the ordering of words.
- How to identify whether the defendant was negligent?
  - "The negligent defendant"
  - "The defendant was negligent"
  - "The defendant, a driver, was negligent"
- Syntactic and semantic parsing will do this.

Dependency Grammar

- The basic idea:
  - **Syntactic structure** consists of **words**, linked by binary symmetric relations called **dependencies**.
  - Dependencies identify the grammatical relations between words.

Dependency Grammar

- The basic idea:
  - **Syntactic structure** consists of **words**, linked by binary symmetric relations called **dependencies**.
  - Dependencies identify the grammatical relations between words.



- Dependency structures represent grammatical relations between words in a sentence:
  - head-dependent relations (directed arcs)
    - functional categories (arc labels)
    - structural categories (parts-of-speech)

dependencies in spaCy

```
for sent in doc.sents:
    print(sent)
    print(sent.root)
    print([(w, w.dep_) for w in sent.root.children])
    print()
```

```
Science cannot solve the ultimate mystery of nature.
solve
[(Science, 'nsubj'), (can, 'aux'), (not, 'neg'), (mystery, 'dobj'), (., 'punct')]

And that is because, in the last analysis, we ourselves are a part of the mystery
that we are trying to solve.
is
[(And, 'cc'), (that, 'nsubj'), (are, 'advcl'), (., 'punct')]
```

dependencies in spaCy

```
for sent in doc.sents:
    print(sent)
    print(sent.root)
    print([(w, w.dep_) for w in sent.root.children])
    print()
```

```
Science cannot solve the ultimate mystery of nature.
solve
[(Science, 'nsubj'), (can, 'aux'), (not, 'neg'), (mystery, 'dobj'), (., 'punct')]

And that is because, in the last analysis, we ourselves are a part of the mystery
that we are trying to solve.
is
[(And, 'cc'), (that, 'nsubj'), (are, 'advcl'), (., 'punct')]
```

- For production, use spaCy processing pipelines (https://spacy.io/usage/processing-pipelines)
  - customizable and parallelizable

Unsupervised Discovery of Gendered Language

- This paper builds on the "gender bias" NLP papers by adding in syntactic information:

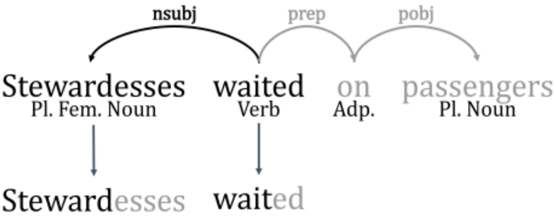


Figure 2: An example sentence with its labeled dependency parse (top) and lemmatized words (bottom).

Unsupervised Discovery of Gendered Language

- This paper builds on the "gender bias" NLP papers by adding in syntactic information:

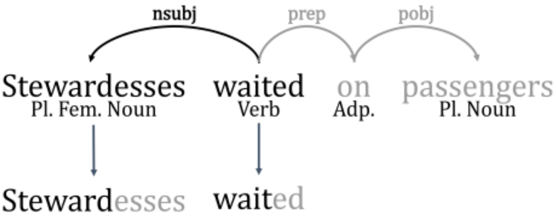


Figure 2: An example sentence with its labeled dependency parse (top) and lemmatized words (bottom).

- Corpus: dependency parse of 3.5 million books from Goldberg and Orwant (2013).
  - 37 million noun-adjective pairs
  - 41-million subject-verb pairs
  - 14 million verb-object pairs

Extracting gendered language

- Hoyle et al (2019) extract the set of adjectives and verbs attached to nouns that are predictive of the gender of the noun.
  - they use a regularized latent variable model
  - the resulting metric is (almost) proportional to PMI.

Extracting gendered language

- Hoyle et al (2019) extract the set of adjectives and verbs attached to nouns that are predictive of the gender of the noun.
  - they use a regularized latent variable model
  - the resulting metric is (almost) proportional to PMI.
- Interpreting the dimensions:
  - categorize adjectives/verbs by sentiment (positive, negative, neutral)

Extracting gendered language

- Hoyle et al (2019) extract the set of adjectives and verbs attached to nouns that are predictive of the gender of the noun.
  - they use a regularized latent variable model
  - the resulting metric is (almost) proportional to PMI.
- Interpreting the dimensions:
  - categorize adjectives/verbs by sentiment (positive, negative, neutral)
  - categorize adjectives/verbs as related to the body and emotions.

Gendered Adjectives

$\tau_{\text{MASC-POS}}$		$\tau_{\text{MASC-NEG}}$		$\tau_{\text{MASC-NEU}}$		$\tau_{\text{FEM-POS}}$		$\tau_{\text{FEM-NEG}}$		$\tau_{\text{FEM-NEU}}$	
Adj.	Value	Adj.	Value	Adj.	Value	Adj.	Value	Adj.	Value	Adj.	Value
faithful	2.3	unjust	2.4	german	1.9	pretty	3.3	horrible	1.8	virgin	2.8
responsible	2.2	dumb	2.3	teutonic	0.8	fair	3.3	destructive	0.8	alleged	2.0
adventurous	1.9	violent	1.8	financial	2.6	beautiful	3.4	notorious	2.6	maiden	2.8
grand	2.6	weak	2.0	feudal	2.2	lovely	3.4	dreary	0.8	russian	1.9
worthy	2.2	evil	1.9	later	1.6	charming	3.1	ugly	3.2	fair	2.6
brave	2.1	stupid	1.6	austrian	1.2	sweet	2.7	weird	3.0	widowed	2.4
good	2.3	petty	2.4	feudatory	1.8	grand	2.6	harried	2.4	grand	2.1
normal	1.9	brutal	2.4	maternal	1.6	stately	3.8	diabetic	1.2	byzantine	2.6
ambitious	1.6	wicked	2.1	bavarian	1.5	attractive	3.3	discontented	0.5	fashionable	2.5
gallant	2.8	rebellious	2.1	negro	1.5	chaste	3.3	infected	2.8	aged	1.8
mighty	2.4	bad	1.9	paternal	1.4	virtuous	2.7	unmarried	2.8	topless	3.9
loyal	2.1	worthless	1.6	frankish	1.8	fertile	3.2	unequal	2.4	withered	2.9
valiant	2.8	hostile	1.9	welsh	1.7	delightful	2.9	widowed	2.4	colonial	2.8
courteous	2.6	careless	1.6	ecclesiastical	1.6	gentle	2.6	unhappy	2.4	diabetic	0.7
powerful	2.3	unsung	2.4	rural	1.4	privileged	1.4	horrid	2.2	burlesque	2.9
rational	2.1	abusive	1.5	persian	1.4	romantic	3.1	pitiful	0.8	blonde	2.9
supreme	1.9	financial	3.6	belted	1.4	enchanted	3.0	frightful	0.5	parisian	2.7
meritorious	1.5	feudal	2.5	swiss	1.3	kindly	3.2	artificial	3.2	clad	2.5
serene	1.4	false	2.3	finnish	1.1	elegant	2.8	sullen	3.1	female	2.3
godlike	2.3	feeble	1.9	national	2.2	dear	2.2	hysterical	2.8	oriental	2.2
noble	2.3	impotent	1.7	priestly	1.8	devoted	2.0	awful	2.6	ancient	1.7
rightful	1.9	dishonest	1.6	merovingian	1.6	beauteous	3.9	haughty	2.6	feminist	2.9
eager	1.9	ungrateful	1.5	capetian	1.4	sprightly	3.2	terrible	2.4	matronly	2.6
financial	3.3	unfaithful	2.6	prussian	1.4	beloved	2.5	damned	2.4	pretty	2.5
chivalrous	2.6	incompetent	1.7	racial	0.9	pleasant	1.8	topless	3.5	asiatic	2.0

Gendered Verbs (as agent)

$\tau_{\text{MASC-POS}}$		$\tau_{\text{MASC-NEG}}$		$\tau_{\text{MASC-NEU}}$		$\tau_{\text{FEM-POS}}$		$\tau_{\text{FEM-NEG}}$		$\tau_{\text{FEM-NEU}}$	
Verb	Value	Verb	Value	Verb	Value	Verb	Value	Verb	Value	Verb	Value
succeed	1.6	fight	1.2	extend	0.7	celebrate	2.4	persecute	2.1	faint	0.7
protect	1.4	fail	1.0	found	0.8	fascinate	0.8	faint	1.0	be	1.1
favor	1.3	fear	1.0	strike	1.3	facilitate	0.7	fly	1.0	go	0.4
flourish	1.3	murder	1.5	own	1.1	marry	1.8	weep	2.3	find	0.1
prosper	1.7	shock	1.6	collect	1.1	smile	1.8	harm	2.2	fly	0.4
support	1.5	blind	1.6	set	0.8	fan	0.8	wear	2.0	fall	0.1
promise	1.5	forbid	1.5	wag	1.0	kiss	1.8	mourn	1.7	wear	0.9
welcome	1.5	kill	1.3	present	0.9	champion	2.2	gasp	1.1	leave	0.7
favour	1.2	protest	1.3	pretend	1.1	adore	2.0	fatigue	0.7	fell	0.1
clear	1.9	cheat	1.3	prostrate	1.1	dance	1.7	scold	1.8	vanish	1.3
reward	1.8	fake	0.8	want	0.9	laugh	1.6	scream	2.1	come	0.7
appeal	1.6	deprive	1.5	create	0.9	have	1.4	confess	1.7	fertilize	0.6
encourage	1.5	threaten	1.3	pay	1.1	play	1.0	get	0.5	flush	0.5
allow	1.5	frustrate	0.9	prompt	1.0	give	0.8	gossip	2.0	spin	1.6
respect	1.5	fright	0.9	brazen	1.0	like	1.8	worry	1.8	dress	1.4
comfort	1.4	temper	1.4	tarry	0.7	giggle	1.4	be	1.3	fill	0.2
treat	1.3	horrify	1.4	front	0.5	extol	0.6	fail	0.4	fee	0.2
brave	1.7	neglect	1.4	flush	0.3	compassionate	1.9	fight	0.4	extend	0.1
rescue	1.5	argue	1.3	reach	0.9	live	1.4	fake	0.3	sniff	1.6
win	1.5	denounce	1.3	escape	0.8	free	0.9	overrun	2.4	celebrate	1.1
warm	1.5	concern	1.2	gi	0.7	felicitate	0.6	hurt	1.8	clap	1.1
praise	1.4	expel	1.7	rush	0.6	mature	2.2	complain	1.7	appear	0.9
fit	1.4	dispute	1.5	duplicate	0.5	exalt	1.7	lament	1.5	gi	0.8
wish	1.4	obscure	1.4	incarnate	0.5	surpass	1.7	fertilize	0.5	have	0.5
grant	1.3	damn	1.4	freeze	0.5	meet	1.1	feign	0.5	front	0.5

Gendered Verbs (as patient)

$\tau_{\text{MASC-POS}}$		$\tau_{\text{MASC-NEG}}$		$\tau_{\text{MASC-NEU}}$		$\tau_{\text{FEM-POS}}$		$\tau_{\text{FEM-NEG}}$		$\tau_{\text{FEM-NEU}}$	
Verb	Value	Verb	Value	Verb	Value	Verb	Value	Verb	Value	Verb	Value
praise	1.7	fight	1.8	set	1.5	marry	2.3	forbid	1.3	have	1.0
thank	1.7	expel	1.8	pay	1.2	assure	3.4	shame	2.5	expose	0.8
succeed	1.7	fear	1.6	escape	0.4	escort	1.2	escort	1.3	escort	1.4
exalt	1.2	defeat	2.4	use	2.1	exclaim	1.0	exploit	0.9	pour	2.1
reward	1.8	fail	1.3	expel	0.9	play	2.7	drag	2.1	marry	1.3
commend	1.7	bribe	1.8	summon	1.7	pour	2.6	suffer	2.2	take	1.1
fit	1.4	kill	1.6	speak	1.3	create	2.0	shock	2.1	assure	1.6
glorify	2.0	deny	1.5	shop	2.6	have	1.8	fright	2.4	fertilize	1.6
honor	1.6	murder	1.7	excommunicate	1.3	fertilize	1.8	steal	2.0	ask	1.0
welcome	1.9	depose	2.3	direct	1.1	eye	0.9	insult	1.8	exclaim	0.6
gentle	1.8	summon	2.0	await	0.9	woo	3.3	fertilize	1.6	strut	2.3
inspire	1.7	order	1.9	equal	0.4	strut	3.1	violate	2.4	burn	1.7
enrich	1.7	denounce	1.7	appoint	1.7	kiss	2.6	tease	2.3	rear	1.5
uphold	1.5	deprive	1.6	animate	1.1	protect	2.1	terrify	2.1	feature	0.9
appease	1.5	mock	1.6	follow	0.7	win	2.0	persecute	2.1	visit	1.3
join	1.4	destroy	1.5	depose	1.8	excel	1.6	cry	1.8	saw	1.3
congratulate	1.3	deceive	1.7	want	1.1	treat	2.3	expose	1.3	exchange	0.8
extol	1.1	bore	1.6	reach	0.9	like	2.2	burn	2.6	shame	1.6
respect	1.7	bully	1.5	found	0.8	entertain	2.0	scare	2.0	fade	1.2
brave	1.7	enrage	1.4	exempt	0.4	espouse	1.4	frighten	1.8	signal	1.2
greet	1.6	shop	2.7	tip	1.8	feature	1.2	distract	2.3	see	1.2
restore	1.5	elect	2.2	elect	1.7	meet	2.2	weep	2.3	present	1.0
clear	1.5	compel	2.1	unmake	1.5	wish	1.9	scream	2.3	leave	0.8
excite	1.2	offend	1.5	fight	1.2	fondle	1.9	drown	2.1	espouse	1.3
flatter	0.9	scold	1.4	prevent	1.1	saw	1.8	rape	2.0	want	1.1

Female		Male	
Positive	Negative	Positive	Negative
beautiful	battered	just	unsuitable
lovely	untreated	sound	unreliable
chaste	barren	righteous	lawless
gorgeous	shrewish	rational	inseparable
fertile	sheltered	peaceable	brutish
beauteous	heartbroken	prodigious	idle
sexy	unmarried	brave	unarmed
classy	undernourished	paramount	wounded
exquisite	underweight	reliable	bigoted
vivacious	uncomplaining	sinless	unjust
vibrant	nagging	honorable	brutal

BODY	FEELING	MISCELLANEOUS
BEHAVIOR	SPATIAL	TEMPORAL
SUBSTANCE	QUANTITY	SOCIAL

- Female nouns were correlated with adjectives/verbs related to the body and to emotions.

Extracting Modal Verb Structures in Labor Contracts (Ash et al 2020)

- Subject categories:
  - worker, union, owner, and manager.

Extracting Modal Verb Structures in Labor Contracts (Ash et al 2020)

- Subject categories:
  - worker, union, owner, and manager.
- In law, deontic modal verb structures create legal requirements (Kratzer 1991).
  - strict (*shall*, *will*, *must*)
  - permissive (*may*, *can*)

Extracting Modal Verb Structures in Labor Contracts (Ash et al 2020)

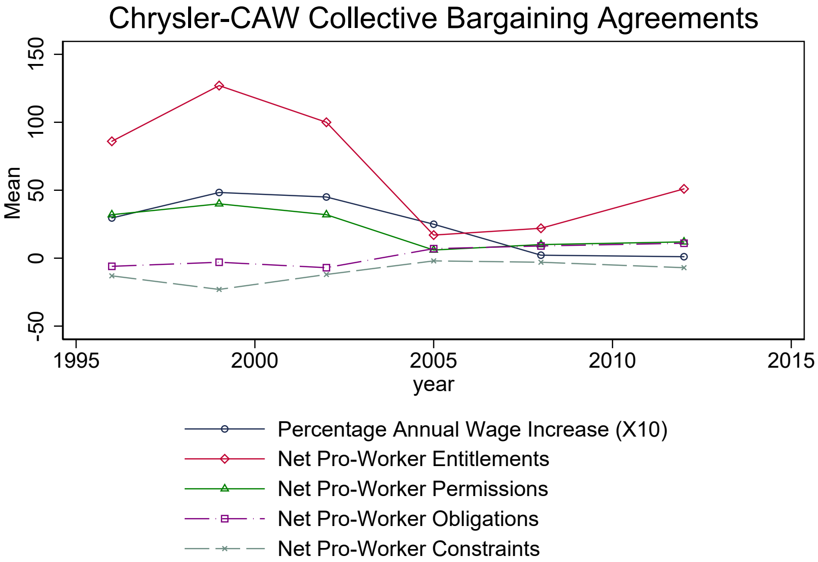
- Subject categories:
  - worker, union, owner, and manager.
- In law, deontic modal verb structures create legal requirements (Kratzer 1991).
  - strict (*shall*, *will*, *must*)
  - permissive (*may*, *can*)
- Statements coded as negative ("shall not" rather than "shall") and active ("shall provide") or passive ("shall be provided").

Most Frequent Subject-Modal-Verb Tuples

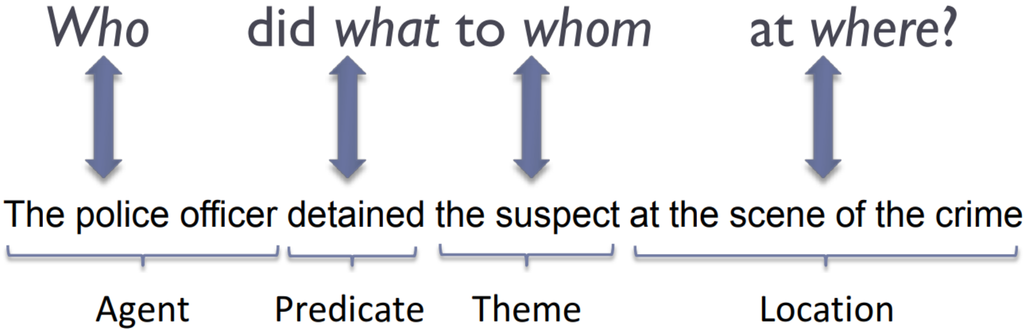
Subject - Modal - Verb	Subject - Modal - Verb	Subject - Modal - Verb
agreement_shall_be	**employee**_shall_be	**employee**_shall_receive
arbitrator_shall_have	**employee**_shall_be_allowed	**employee**_shall_retain
board_shall_have	**employee**_shall_be_considered	**employee**_will_be
case_may_be	**employee**_shall_be_entitled	**employee**_will_be_allowed
committee_shall_meet	**employee**_shall_be_given	**employee**_will_be_entitled
company_shall_pay	**employee**_shall_be_granted	**employee**_will_be_given
company_shall_provide	**employee**_shall_be_laid_off	**employee**_will_be_granted
company_will_pay	**employee**_shall_be_paid	**employee**_will_be_paid
company_will_provide	**employee**_shall_be_required	**employee**_will_be_required
decision_shall_be	**employee**_shall_continue	**employee**_will_have
employee_may_request	**employee**_shall_lose	**employer**_shall_grant

Case Study: Canadian Auto Workers Union Contract

Case Study: Canadian Auto Workers Union Contract



Semantic Role Labeling



Source: Jurafsky-Martin slides.

“Higher taxes will hurt the economy.”

“Health insurance saves lives.”

‘Immigrants steal our jobs.’

Our (broad) research agenda: How do narratives influence and/or reflect political and economic outcomes?
A preliminary challenge: How to *identify* and *quantify* narratives.

Raw sentences and their mined narratives

- "President, I think the administration has begun to address the overseas basing issue."
  $\rightarrow$ (administration, address, foreign policy)
- "As always, God bless and protect our troops and their families."
  $\rightarrow$ (god, bless, troop)
  $\rightarrow$ (god, protect, troop)
- "We need to pay attention to agriculture and the survival of the family farm as other countries protect and subsidize their farmers."
  $\rightarrow$ (country, protect, farmer)
  $\rightarrow$ (country, subsidize, farmer)
- show wordviews HTML

Outline

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.
- Context:
  - tweets in response to mass shooting events.

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky

- Research Object:
  - use NLP to understand four dimensions of social media polarization: topic choice, framing, affect, modality.
- Context:
  - tweets in response to mass shooting events.
- Research question:
  - does political partisanship manifest in polarized responses to violent/polarizing events?

Dataset

- 21 mass shooting events, 2015-2018, from Gun Violence Archive

Dataset

- 21 mass shooting events, 2015-2018, from Gun Violence Archive
- tweets about those events, identified by:
  - location keywords (e.g. chattanooga, roseburg, san bernardino, fresno, etc.)
  - event keywords (lemmas): shoot, gun, kill, attack, massacre, victim

Dataset

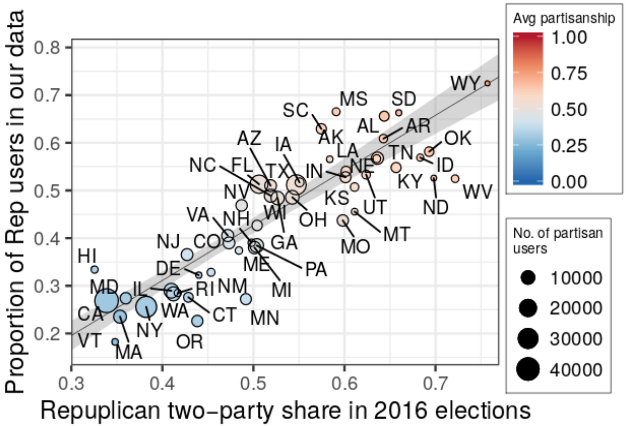
- 21 mass shooting events, 2015-2018, from Gun Violence Archive
- tweets about those events, identified by:
  - location keywords (e.g. chattanooga, roseburg, san bernardino, fresno, etc.)
  - event keywords (lemmas): shoot, gun, kill, attack, massacre, victim
  - filter out retweets and tweets from deactivated accounts
  - $N = 10,000$ (out of 4.4 million tweets from the firehose archive).

Identifying party affiliation of Twitter users

- Party affiliation identified off of whether you follow more Democrats or Republicans, from a list of Twitter accounts associated with legislators, presidential candidates, and party organizations (Volkova et al 2014).
  - at least 51% of tweets for each event can be assigned partisanship this way.

Identifying party affiliation of Twitter users

- Party affiliation identified off of whether you follow more Democrats or Republicans, from a list of Twitter accounts associated with legislators, presidential candidates, and party organizations (Volkova et al 2014).
  - at least 51% of tweets for each event can be assigned partisanship this way.
- For geolocated users this matches up pretty well with party vote shares by state ($R^2 = .82$):



Measuring Partisanship: Pre-processing

- Stemming and stopword removal.
- Event-specific vocabulary:
  - unigrams and bigrams
  - occur in event's tweets at least 50 times
  - must be used by at least two tweeters.

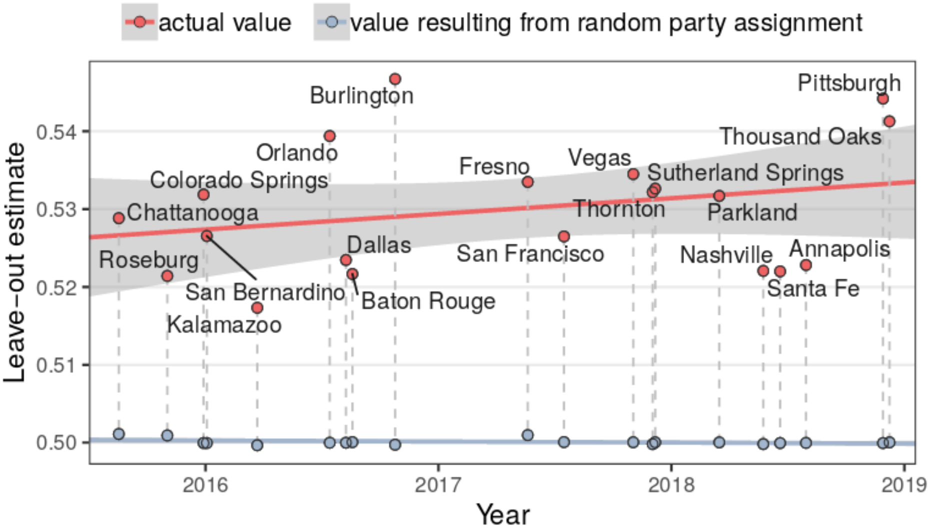
Partisanship metric

- Leave-one-out estimator from Gentzkow et al (2019), applied to each shooting event:
$$\pi = \frac{1}{2}\left(\frac{1}{|D|}\sum_{i\in D}\hat{\boldsymbol{q}}_i \cdot \hat{\boldsymbol{\rho}}_{-i} + \frac{1}{|R|}\sum_{i\in R}\hat{\boldsymbol{q}}_i \cdot (1 - \hat{\boldsymbol{\rho}}_{-i})\right)$$
  - $\hat{\boldsymbol{q}}_i$ = token frequencies for user i, drawn from set of democrats D and set of republicans R
  - $\hat{\boldsymbol{\rho}}_{-i}$ has elements
$$\rho_{-i} = \frac{q_i^D}{q_i^D + q_i^R}$$
    empirical posterior probabilities computed from all other users.

Partisanship metric

- Leave-one-out estimator from Gentzkow et al (2019), applied to each shooting event:
$$\pi = \frac{1}{2}\left(\frac{1}{|D|}\sum_{i\in D}\hat{\boldsymbol{q}}_i\cdot\hat{\boldsymbol{\rho}}_{-i} + \frac{1}{|R|}\sum_{i\in R}\hat{\boldsymbol{q}}_i\cdot(1-\hat{\boldsymbol{\rho}}_{-i})\right)$$
  - $\hat{\boldsymbol{q}}_i$ = token frequencies for user i, drawn from set of democrats D and set of republicans R
  - $\hat{\boldsymbol{\rho}}_{-i}$ has elements
  $$\rho_{-i} = \frac{q_i^D}{q_i^D + q_i^R}$$
  empirical posterior probabilities computed from all other users.
- π is an estimate for expected posterior probability that a Bayesian observer would correctly predict party after observing one randomly sampled token.
  - consistency assumes tokens are drawn from multinomial logit.

Tweet texts about mass shootings are predictive of party



- comparable to $\pi = .53$ in Congressional speeches (GST 2019).
- The increase in polarization over time is not statistically significant.

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?
- Can show polarization separately by party?

Questions/Issues with this Analysis

- How polarized are tweets about other topics (not mass shootings)?
  - why not use a tweeter fixed effect and compare to their other tweets?
  - why not show pre-trends in polarization?
- Can show polarization separately by party?
- Validating π:
  - How accurate is π at the individual level?
  - Where is the binscatter of π versus actual party affiliation?

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)
2. Train GloVe embeddings on random samples of tweets from each event (samples were different sizes, this is not explained)

Sentence Embeddings for Topic Assignment

1. Make a new vocabulary:
   1.1 Sample 10,000 tweets from each event
   1.2 vocabulary of stemmed words occuring at least ten times in at least three events ($N = 2000$)
2. Train GloVe embeddings on random samples of tweets from each event (samples were different sizes, this is not explained)
3. Create Arora et al (2017) embeddings:
   3.1 for each tweet t, compute weighted average vectors v_t for each word, weighted by inverse frequency.
   3.2 take out first principal component of matrix whose rows are v_t

Topics = Embedding Clusters

1. Cluster the embeddings using k-means

Topics = Embedding Clusters

1. Cluster the embeddings using k-means
2. Identify and drop hard-to-classify tweets:
   2.1 compute ratio of distance to closest topic and distance to second-closest topic.
   2.2 drop tweets above the 75th percentile.

Topics = Embedding Clusters

1. Cluster the embeddings using k-means
2. Identify and drop hard-to-classify tweets:
   2.1 compute ratio of distance to closest topic and distance to second-closest topic.
   2.2 drop tweets above the 75th percentile.

- Validation using Amazon Mechanical Turk to choose number of clusters:
  - Identify word intruder: five from one cluster, one from another cluster.
  - Identify tweet intruder: three from one cluster, and one from another cluster.

Topic Content

Topic	10 Nearest Stems
news (19%)	break, custodi, #breakingnew, #updat, confirm, fatal, multipl, updat, unconfirm, sever
investigation (9%)	suspect, arrest, alleg, apprehend, custodi, charg, accus, prosecutor, #break, ap
shooter's identity & ideology (11%)	extremist, radic, racist, ideolog, label, rhetor, wing, blm, islamist, christian
victims & location (4%)	bar, thousand, california, calif, among, los, southern, veteran, angel, via
laws & policy (14%)	sensibl, regul, requir, access, abid, #gunreformnow, legisl, argument, allow, #guncontolnow
solidarity (13%)	affect, senseless, ach, heart, heartbroken, sadden, faculti, pray, #prayer, deepest
remembrance (6%)	honor, memori, tuesday, candlelight, flown, vigil, gather, observ, honour, capitol
other (23%)	dude, yeah, eat, huh, gonna, ain, shit, ass, damn, guess

- The embedding method resulted in more coherent topics (better MTurk validation for words and tweets) than a topic model. $k = 8$ got best coherence.
  - Appendix reports samples of tweets for each topic (but does not say how samples were selected).

Between-topic vs within-topic polarization

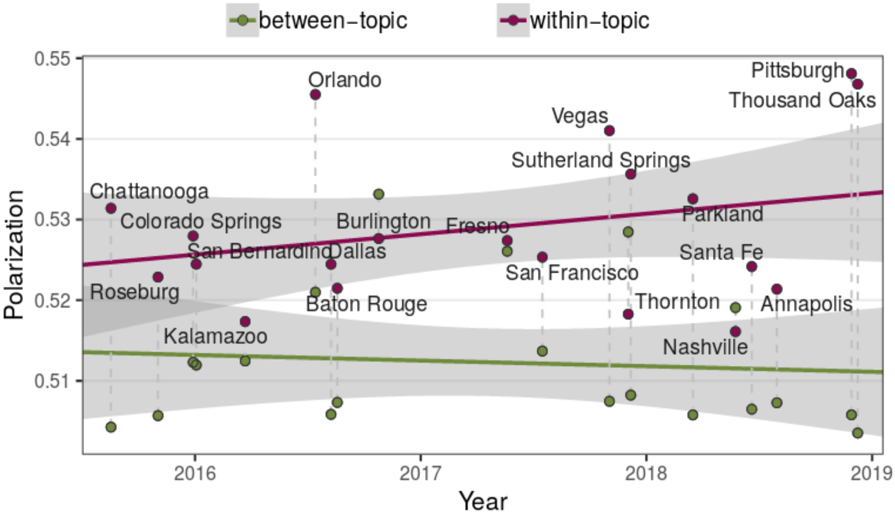
- Within-topic polarization: compute π separately by the tweet clusters.

Between-topic vs within-topic polarization

- Within-topic polarization: compute π separately by the tweet clusters.
- Between-topic polarization: Compute π using cluster counts, rather than token counts.

Between-topic vs within-topic polarization

- Within-topic polarization: compute π separately by the tweet clusters.
- Between-topic polarization: Compute π using cluster counts, rather than token counts.



Trends in within-topic polarization

- Most polarized topics: shooter's identity & ideology (.55), laws & policy (.54)

Figure 6: Las Vegas within-topic polarization in the days after the event. The bar charts show the proportion of each topic in the data at a given time.

- "measuring polarization of topics for other events over time is noisy".

Partisanship of Topics, by Race of Shooter

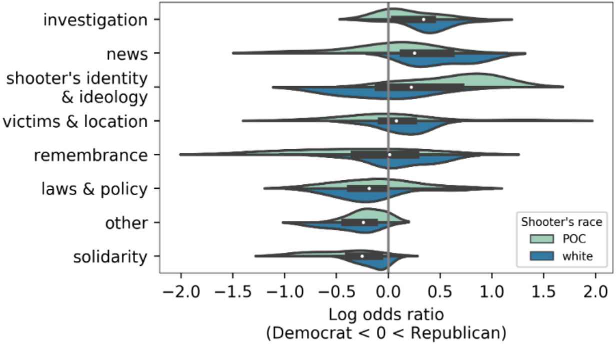


Figure 7: The plot shows the kernel density of the partisan log odds ratios of each topic (one observation per event). The white points show the median and the black rectangles the interquartile range across events.

Partisan Framing Devices: Words

- Partisanship of phrases from supervised model:

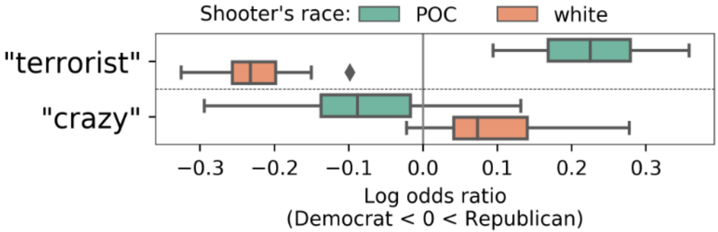
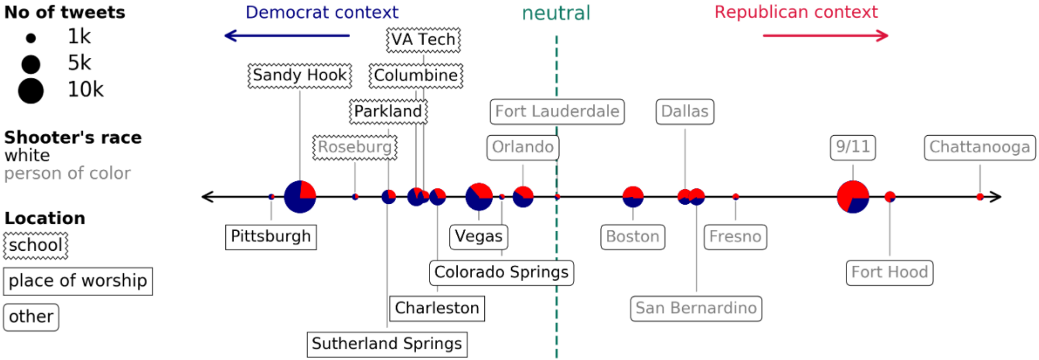


Figure 8: The log odds ratios of "terrorist" and "crazy" across events, grouped by the shooter's race. The boxes show the interquartile range and the diamond an outlier.

- Partisan valence of "terrorist" and "crazy" flip depending on race of shooter (these words have the largest racial difference in the joint vocabulary).

Partisan Framing Devices: Events

- Partisanship of keywords for previous events:



- Democrats invoke white shooters, Republicans invoke POC shooters.

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

sadness senseless, loss, tragedi, lost, devast, sad, love, griev, horrif, terribl, pain, violenc, condol, broken, hurt, feel, victim, mourn, horrifi, will, grief, ach, suffer, sick, kill, aw, sicken, evil, massacr, mad

disgust disgust, sick, shame, ignor, wrong, blame, hell, ridicul, idiot, murder, evil, coward, sicken, feel, disgrac, slaughter, action, bad, insan, attack, pathet, outrag, polit, terrorist, mad, damn, lose, shit, lie, asshol

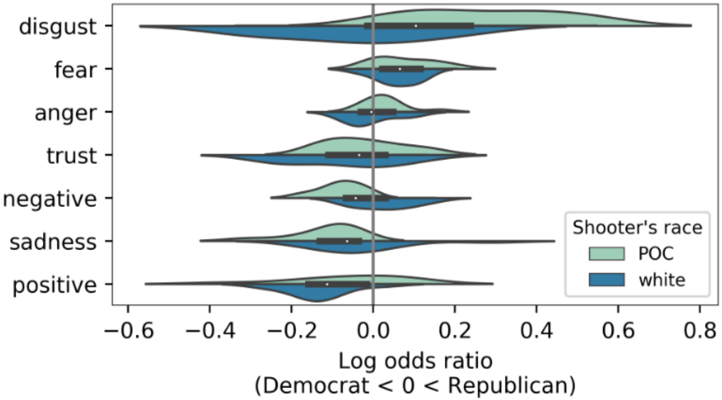
anger gun, will, murder, kill, violenc, wrong, shoot, bad, death, attack, feel, shot, action, arm, idiot, crazi, crimin, terrorist, mad, hell, crime, blame, fight, ridicul, insan, shit, die, threat, terror, hate

fear danger, threat, fear, arm, gun, still, shooter, attack, feel, fight, hide, murder, shot, shoot, bad, kill, chang, serious, violenc, forc, risk, defend, warn, govern, concern, fail, polic, wrong, case, terrorist

trust school, like, good, real, secur, show, nation, don, protect, call, teacher, help, law, great, save, true, wonder, respons, sad, answer, person, feel, safe, thought, continu, love, guard, church, fact, support

Partisanship of Affect Categories

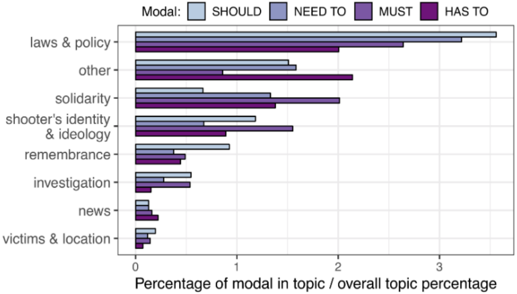
- Compute partisanship scores using affect-category counts:



- Disgust affect flips along partisan lines depending on race of shooter.

Modality

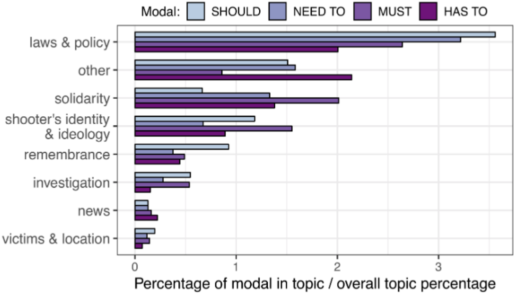
This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.

Modality

This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.
- Democrats use modals more than Republicans; Republicans seem more fatalistic.

Comments

- This is an impressive array of NLP tools aimed at the same research question.
  - could be moving toward a standard for analyzing interpretable dimension in language.

Comments

- This is an impressive array of NLP tools aimed at the same research question.
  - could be moving toward a standard for analyzing interpretable dimension in language.
- For all outcomes, would help to compare to other types of events, and to show pre-trends.
  - there is no baseline for polarization for comparison.
  - they do not distinguish whether outcomes are driven by different people selecting into tweeting, vs within-user changes.

Outline

Causal inference is needed to improve the world

Consider important policy questions like:

- In light of coronavirus, should schools reopen or not for in-person teaching?

Causal inference is needed to improve the world

Consider important policy questions like:

- In light of coronavirus, should schools reopen or not for in-person teaching?
  - No matter how much we know from lab experiments about the biology/epidemiology of the virus, there will be too much uncertainty about costs/benefits to answer this.
  - We need real-world evidence, but we can't experimentally force schools to reopen or not.

Causal inference is needed to improve the world

Consider important policy questions like:

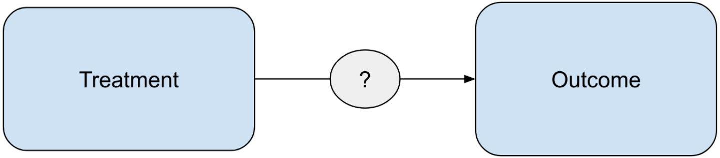
- In light of coronavirus, should schools reopen or not for in-person teaching?
  - No matter how much we know from lab experiments about the biology/epidemiology of the virus, there will be too much uncertainty about costs/benefits to answer this.
  - We need real-world evidence, but we can't experimentally force schools to reopen or not.
- Can use a **natural experiment** to produce causal estimates:
  - e.g., variation in number of coronavirus cases before/after openings, using differences in the timing of openings (differences-in-differences).

Causal inference is needed to improve the world

Consider important policy questions like:

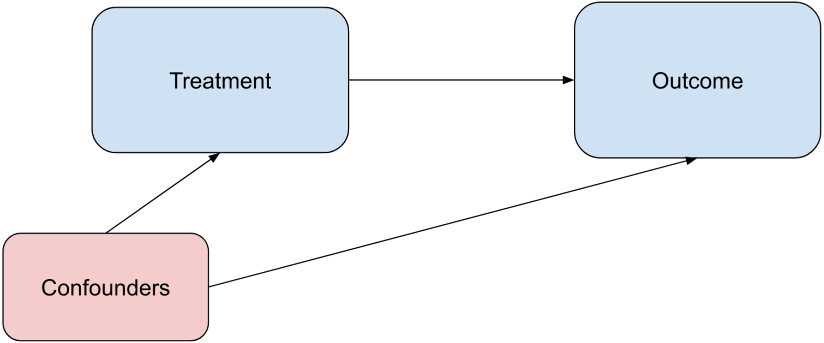
- In light of coronavirus, should schools reopen or not for in-person teaching?
  - No matter how much we know from lab experiments about the biology/epidemiology of the virus, there will be too much uncertainty about costs/benefits to answer this.
  - We need real-world evidence, but we can't experimentally force schools to reopen or not.
- Can use a **natural experiment** to produce causal estimates:
  - e.g., variation in number of coronavirus cases before/after openings, using differences in the timing of openings (differences-in-differences).
- Google/Facebook understand the importance of causal inference with A/B testing; social scientists want to use it to assist public policy.

Causal Graphs



- We are interested in estimating a causal effect (if any) of a “treatment” on an “outcome”.

- ***Unobserved Confounders*** are variables that affect both the treatment and the outcome, which we don't have in our dataset:



- **Observed confounders** are not a problem, because we can adjust (control) for them in causal inference analysis (that is, including them in a regression).

- **Reverse causation**: "the outcome" affects "the "treatment".
  Joint causation: there is bidirectional causation.

- e.g., effect of tax collections on economic growth.

- **Reverse causation**: "the outcome" affects "the "treatment".
  Joint causation: there is bidirectional causation.

- e.g., effect of tax collections on economic growth.
- Resulting estimates are biased (not causal), and cannot be fixed by adjusting for observed confounders.

With joint causality, or with unobserved confounders, it is often impossible to produce statistical estimates with a causal interpretation.

With joint causality, or with unobserved confounders, it is often impossible to produce statistical estimates with a causal interpretation.

- The gold standard: randomized control trials.
  - often not available, e.g. with opening/closing schools under covid-19.

With joint causality, or with unobserved confounders, it is often impossible to produce statistical estimates with a causal interpretation.

- The gold standard: randomized control trials.
    - often not available, e.g. with opening/closing schools under covid-19.
- Second best: natural experiments.
    - differences-in-differences: use longitudinal data and look at groups or places that adopted treatment at different times.

With joint causality, or with unobserved confounders, it is often impossible to produce statistical estimates with a causal interpretation.

- The gold standard: randomized control trials.
  - often not available, e.g. with opening/closing schools under covid-19.
- Second best: natural experiments.
  - differences-in-differences: use longitudinal data and look at groups or places that adopted treatment at different times.
  - regression discontinuity: compare individuals just above or just below some discrete scoring threshold.

With joint causality, or with unobserved confounders, it is often impossible to produce statistical estimates with a causal interpretation.

- The gold standard: randomized control trials.
  - often not available, e.g. with opening/closing schools under covid-19.
- Second best: natural experiments.
  - differences-in-differences: use longitudinal data and look at groups or places that adopted treatment at different times.
  - regression discontinuity: compare individuals just above or just below some discrete scoring threshold.
  - instrumental variables: use a third variable ("instrument") that randomly shifts the probability of treatment.

Fong and Grimmer (2016): Causal effect of political messaging

- What biographical characteristics of politicians influence voter evaluations?

Fong and Grimmer (2016): Causal effect of political messaging

- What biographical characteristics of politicians influence voter evaluations?
- Could run a survey experiment:
  - `Document 1: He earned his Juris Doctor in 1997 from Yale Law School, where he operated free legal clinics for low-income residents of New Haven, Connecticut...`
  - `Document 2: He served in South Vietnam from 1970 to 1971 during the Vietnam War in the Army Rangers' 75th Ranger Regiment, attached to the 173rd Airborne Brigade. He participated in 24 helicopter assaults...`

Fong and Grimmer (2016): Causal effect of political messaging

- What biographical characteristics of politicians influence voter evaluations?
- Could run a survey experiment:
  - `Document 1: He earned his Juris Doctor in 1997 from Yale Law School, where he operated free legal clinics for low-income residents of New Haven, Connecticut...`
  - `Document 2: He served in South Vietnam from 1970 to 1971 during the Vietnam War in the Army Rangers' 75th Ranger Regiment, attached to the 173rd Airborne Brigade. He participated in 24 helicopter assaults...`
- But hard to generalize what features drive differences.

Fong and Grimmer (2016): Approach

- Lab experiment: 1,886 participants, 5,303 responses

1. Randomly assign texts, X_i , to respondents i
   - Sees up to 3 texts from the corpus of $>$ 2200 Wikipedia biographies
2. Obtain responses Y_i for each respondent
   - Feeling thermometer rating: 0-100

Fong and Grimmer (2016): Approach

- Lab experiment: 1,886 participants, 5,303 responses

1. Randomly assign texts, X_i , to respondents i
   - Sees up to 3 texts from the corpus of $>$ 2200 Wikipedia biographies
2. Obtain responses Y_i for each respondent
   - Feeling thermometer rating: 0-100
3. Structural topic model variant ("supervised indian buffet process"):
   - Discover mapping from texts X to latent topic treatments $\vec{D}$ based on their effect on Y.

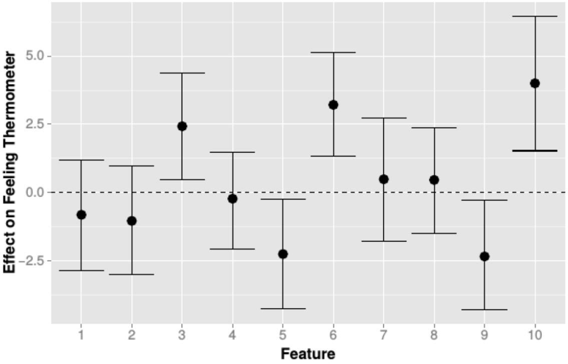
Fong and Grimmer (2016): Approach

- Lab experiment: 1,886 participants, 5,303 responses

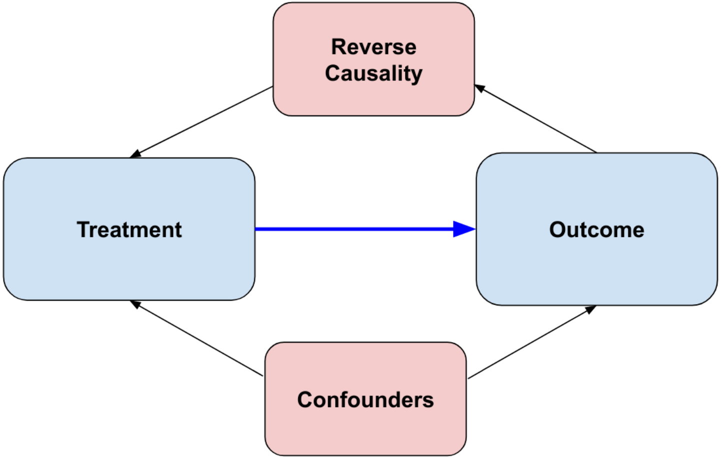
1. Randomly assign texts, X_i , to respondents i
   - Sees up to 3 texts from the corpus of $>$ 2200 Wikipedia biographies
2. Obtain responses Y_i for each respondent
   - Feeling thermometer rating: 0-100
3. Structural topic model variant ("supervised indian buffet process"):
   - Discover mapping from texts X to latent topic treatments $\vec{D}$ based on their effect on Y.
4. Measure causal effects of these treatments on Y_i

Fong and Grimmer (2016): Results

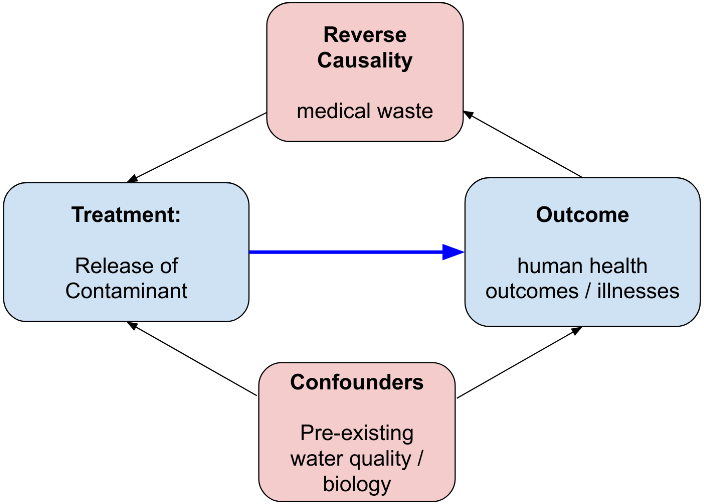
Treatment	Keywords
3	director, university, received, president, phd, policy
5	elected, house, democratic, seat
6	united_states, military, combat, rank
9	law, school_law, law_school, juris_doctor, student
10	war, enlisted, united_states, assigned, army



Causal Graphs



Causal Graph Example: Pollution of a River



Activity: Practice with Causal Graphs

- Think of two example causal inference questions:
  1. where you have **language as an outcome**
  2. where you have **language as a treatment**
- Try to personalize it:
  - a research question from your field
  - a policy you are interested in
  - a mystery you are fascinated by

Activity: Practice with Causal Graphs

- Think of two example causal inference questions:
  1. where you have **language as an outcome**
  2. where you have **language as a treatment**
- Try to personalize it:
  - a research question from your field
  - a policy you are interested in
  - a mystery you are fascinated by
- Link to causal graph template posted in zoom chat:
  - make a copy, fill it in
  - make your doc viewable and paste link into padlet (also in zoom chat).
  - will review these at beginning of next lecture.