Natural Language Processing for Law and Social Science

2. Tokenization

Q&A Page (Moodle)

Homework

- First homework is due Thursday by midnight.
- Submit IPYNB file on EduFlow (reachable from moodle).
- Completion grade – full credit for trying every question and submitting on time (checked programmatically and by random audit)
- Have to submit an assignment (even if late) to see example solution.

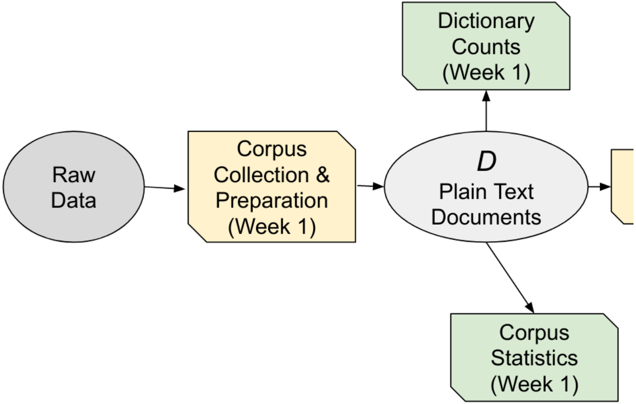
TA Session

- Any feedback on the first TA session?
  - video is linked on syllabus.
- Second TA session is this Friday, 10h-1130h
  - go over week 1 homework
  - go over week 2 notebook
  - can ask questions in advance on moodle.

Final Assignment Info

- There is a final assignment distributed a week or two after class ends.
  - Questions based on the slides and required readings
  - Designed to take 2 hours to complete, but you will have a few days to complete it.

Last Week


Dictionary
Counts
(Week 1)
Raw
Data
Corpus
Collection &
Preparation
(Week 1)
D
Plain Text
Documents
Corpus
Statistics
(Week 1)

Dictionary Methods: Identifying Race-Related Research in Economics (1)

RACE-RELATED RESEARCH IN ECONOMICS AND OTHER SOCIAL SCIENCES*

ARUN ADVANI   ELLIOTT ASH   DAVID CAI   IMRAN RASUL†

DECEMBER 2020

Abstract

How does economics compare to other social sciences in its study of race and ethnicity related issues? We assess this question using a corpus of 500,000 academic publications in economics, political science, and sociology. Using an algorithmic approach to classify race-related publications, we document that economics lags far behind the other disciplines in the volume and share of race-related research. Since 1960, there have been 13,000 race-related

Dictionary Methods: Identifying Race-Related Research in Economics (2)

Corpus. We build a corpus of publications for economics, political science, and sociology. The foundation for this corpus is the *JSTOR* database of academic journals (`jstor.org`). We consider all publications in journals that *JSTOR* characterizes as comprising the disciplines of economics, sociology, and political science. Although publication series are available back to the 1880s, our

this rises steadily over time. Our working sample from 1960 to 2020 covers nearly half a million journal publications: 224, 855 publications from 231 economics journals, 138, 188 publications from 185 sociology journals, and 110, 835 publications from 213 political science journals.

Dictionary Methods: Identifying Race-Related Research in Economics (3)

Identifying Race-Related Research. Given the volume of publications considered, it is infeasible to codify race-related research by hand. We thus take an automated approach and use an algorithm to classify race-related publications. We do so using keywords along two dimensions: (i) the racial or ethnic group being studied; and (ii) the issue being studied. Examples of (case-insensitive) keywords along the group dimension are race, african-american, person of color, and ethnicity. Examples of (case-insensitive) issue keywords include discrimination, prejudice, and stereotype.[2]

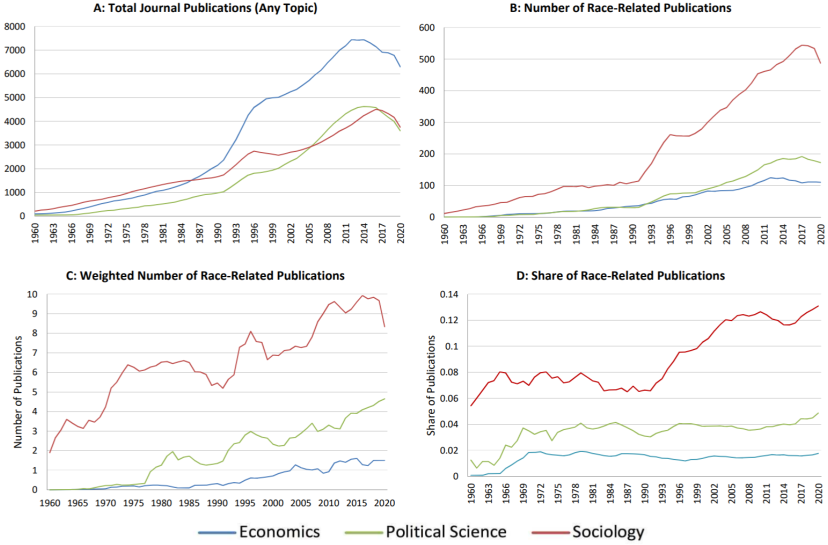
Our algorithm selects a publication as being race-related if: (i) at least one group keyword is in the title; or, (ii) at least one group keyword and at least one issue keyword are mentioned in the title or abstract. For rule (ii) we drop the last sentence of the abstract to avoid false positives from research that only mentions race parenthetically, say because it is part of some robustness check rather than the primary focus of study.

Specifically, we define three bands of group keywords that gradually expand on the racial or ethnic groups being studied. Band 0 consists of only abstract or generic keywords denoting racial and ethnic groups (e.g. race, ethnic, under represented minority). Band 1 adds group keywords relating to the main minority groups in the U.S. (African American, Latinos and Native Americans). Band 2 adds less salient group keywords (e.g. White, South Asian, Indian American, Japanese American) and other minorities based on religious beliefs (e.g. Muslim, Jewish). The full lexicon of group keywords used by Band are shown in Appendix Table A1.

The lexicon of issue keywords, shown in Appendix Table A2, are held constant and not split into bands. These words and phrases are broadly split across five broader topics: discrimination, inequality, diversity, identity, and historical issues. For example, discrimination includes prejudice and stereotypes, while inequality includes disparity and disadvantage.

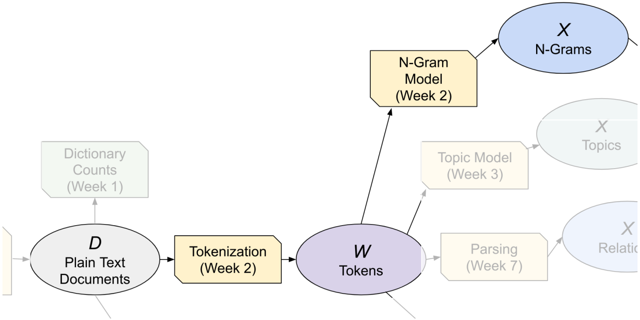
Dictionary Methods: Identifying Race-Related Research in Economics (4)

Figure 1: Race-Related Publications, by Year and Discipline



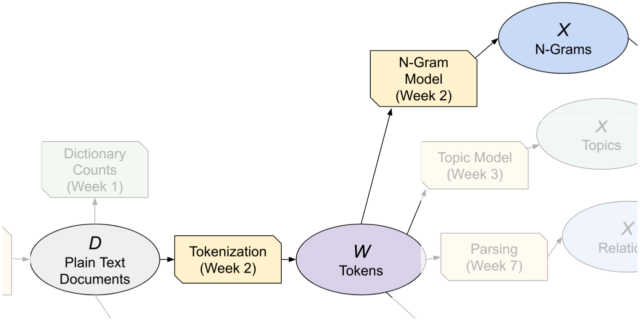
Notes: We use data from JSTOR, Scopus, and the Web of Science to construct the number and shares of race related publications in economics, political science, and sociology. Panel A reports the total number of publications in each discipline. As the publication series start in the 1880s, the publication numbers do not start exactly at zero in 1960, the first year of our working sample. Panel B reports the number of articles that are determined to be race-related by our algorithm. Panel C reports a journal-weighted version of Panel B using the journal quality weights from Angrist et al. [2020]. Panel D reports the share of articles determined to be race-related by our algorithm in each discipline. All series presented are 5-year moving averages.

Today



- Input:
  - A set of documents (e.g. text files), D.

Today



- Input:
  - A set of documents (e.g. text files), D.
- Output (tokens):
  - A sequence, W, containing a list of tokens – words or word pieces for use in natural language processing
- Output (n-grams):
  - A matrix, X, containing statistics about word/phrase frequencies in those documents.

Goals of Tokenization

To summarize: A major goal of tokenization is to produce features that are

- **predictive** in the learning task
- **interpretable** by human investigators
- **tractable** enough to be easy to work with

Goals of Tokenization

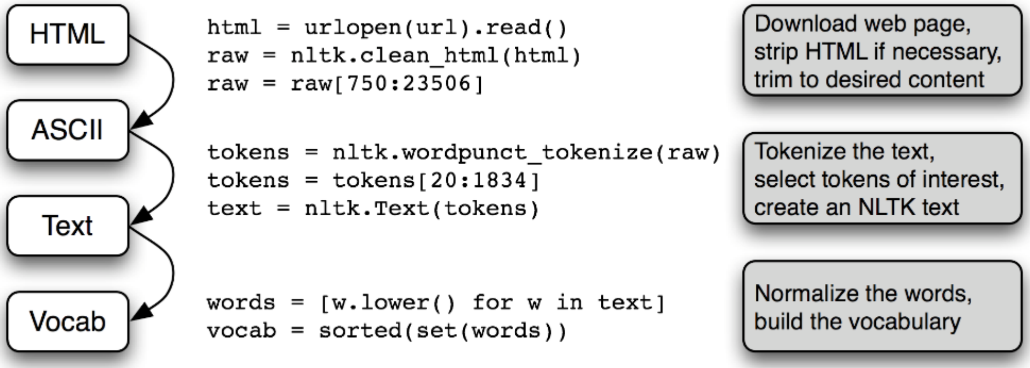
To summarize: A major goal of tokenization is to produce features that are

- **predictive** in the learning task
- **interpretable** by human investigators
- **tractable** enough to be easy to work with

Two broad approaches:

1. convert documents to vectors, usually frequency distributions over pre-processed n-grams.
2. convert documents to sequences of tokens, for inputs to sequential models.

A Standard Tokenization Pipeline



Source: NLTK Book, Chapter 3.

Subword Tokenization for Sequence Models

Modern transformer models (e.g. BERT, GPT) use subword tokenization:

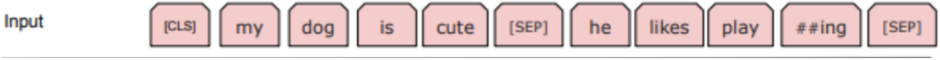
- construct character-level n-grams
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

Subword Tokenization for Sequence Models

Modern transformer models (e.g. BERT, GPT) use subword tokenization:

- construct character-level n-grams
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

e.g., BERT's SentencePiece tokenizer:



- character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
- have to fix a vocabulary size: e.g. BERT uses 30K.

Segmenting paragraphs/sentences

- Many tasks should be done on sentences, rather than corpora as a whole.
  - spaCy is a good (but not perfect) job of splitting sentences, while accounting for periods on abbreviations, etc.
- There isn't a grammar-based paragraph tokenizer.
  - most corpora have new paragraphs annotated.
  - or use line breaks.

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: "judge has" / "has discretion" vs "judge has discretion".

Capitalization

- Removing capitalization is a standard corpus normalization technique
  - usually the capitalized/non-capitalized version of a word are equivalent – e.g. words showing up capitalized at beginning of sentence
  - $\rightarrow$ capitalization not informative.

Capitalization

- Removing capitalization is a standard corpus normalization technique
  - usually the capitalized/non-capitalized version of a word are equivalent – e.g. words showing up capitalized at beginning of sentence
  - $\rightarrow$ capitalization not informative.
- Also: what about "the first amendment" versus "the First Amendment"?
  - Compromise: include capitalized version of words not at beginning of sentence.

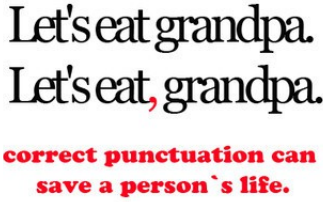
Capitalization

- Removing capitalization is a standard corpus normalization technique
  - usually the capitalized/non-capitalized version of a word are equivalent – e.g. words showing up capitalized at beginning of sentence
  - $\rightarrow$ capitalization not informative.
- Also: what about "the first amendment" versus "the First Amendment"?
  - Compromise: include capitalized version of words not at beginning of sentence.
- For some tasks, capitalization is important
  - needed for sentence splitting, part-of-speech tagging, syntactic parsing, and semantic role labeling.

Capitalization

- Removing capitalization is a standard corpus normalization technique
  - usually the capitalized/non-capitalized version of a word are equivalent – e.g. words showing up capitalized at beginning of sentence
  - $\rightarrow$ capitalization not informative.
- Also: what about "the first amendment" versus "the First Amendment"?
  - Compromise: include capitalized version of words not at beginning of sentence.
- For some tasks, capitalization is important
  - needed for sentence splitting, part-of-speech tagging, syntactic parsing, and semantic role labeling.
  - For sequence data, e.g. language modeling – huggingface tokenizer takes out capitalization but then add a special "capitalized" token before the word.

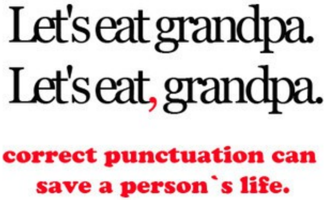
Punctuation



Source: Chris Bail text data slides.

Inclusion of punctuation depends on your task:

Punctuation

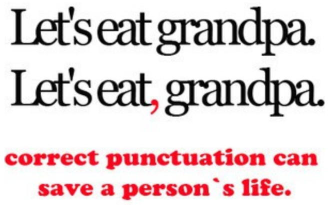


Source: Chris Bail text data slides.

Inclusion of punctuation depends on your task:

- if you are vectorizing the document as a bag of words or bag of n-grams, punctuation won't be needed.

Punctuation



Source: Chris Bail text data slides.

Inclusion of punctuation depends on your task:

- if you are vectorizing the document as a bag of words or bag of n-grams, punctuation won't be needed.
- like capitalization, punctuation is needed for annotations (sentence splitting, parts of speech, syntax, roles, etc)
  - also needed for language models.

Numbers

- for classification using bag of words:
  - can drop numbers, or replace with special characters

Numbers

- for classification using bag of words:
    - can drop numbers, or replace with special characters
- for language models:
    - just treat them like letters.
    - GPT-3 can solve math problems (but not well, this is an area of research)

Drop Stopwords?

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Drop Stopwords?

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

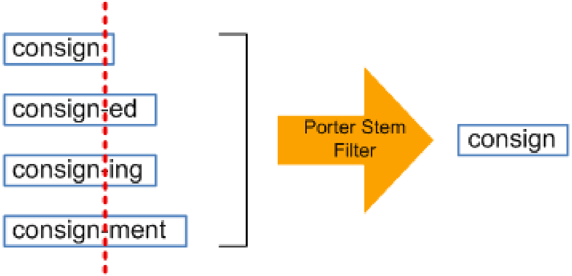
- What about "not guilty"?
- Legal "memes" often contain stopwords:
  - "beyond a reasonable doubt"
  - "with all deliberate speed"

Drop Stopwords?

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

- What about “not guilty”?
- Legal “memes” often contain stopwords:
  - “beyond a reasonable doubt”
  - “with all deliberate speed”
- can drop stopwords by themselves, but keep them as part of phrases.
- can filter out words and phrases using part-of-speech tags (later).

Stemming/lemmatizing



- Effective dimension reduction with little loss of information.
- Lemmatizer produces real words, but N-grams won't make grammatical sense
  - e.g., "judges have been ruling" would become "judge have be rule"

Brainstorming Activity: How to use non-word features

Depending on the first letter of your last name, do one of the following tasks.
Outline a **social-science analysis or dimension of language** that:

- A-F – can be measured by capitalization.
- G-L – can be measured by punctuation.
- M-R – would change depending on the use of stopwords.
- S-Z – would change depending on the use of stemming/lemmatizing.

Think of your answer privately for a moment – we will then type them in the zoom chat.

Tokens

The most basic unit of representation in a text.

- characters: documents as sequence of individual letters {h,e,l,l,o, ,w,o,r,l,d}

Tokens

The most basic unit of representation in a text.

- characters: documents as sequence of individual letters {h,e,l,l,o, ,w,o,r,l,d}
- words: split on white space {hello, world}

Tokens

The most basic unit of representation in a text.

- characters: documents as sequence of individual letters {h,e,l,l,o, ,w,o,r,l,d}
- words: split on white space {hello, world}
- n-grams: learn a vocabulary of phrases and tokenize those: "ETH Zurich → ETH_Zurich"

Tokens

The most basic unit of representation in a text.

- characters: documents as sequence of individual letters {h,e,l,l,o, ,w,o,r,l,d}
- words: split on white space {hello, world}
- n-grams: learn a vocabulary of phrases and tokenize those: "ETH Zurich → ETH_Zurich"
- what else?

Bag-of-words representation

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.

Counts and frequencies

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

$$\text{Term Frequency of } w \text{ in document } k = \frac{\text{Count of } w \text{ in document } k}{\text{Total tokens in document } k}$$

Application: Ranking Partisan language

Monroe et al (2009), "Fightin' Words"

- This paper systematically explores a number of methods for identifying words that are distinctive of groups of speakers
  - in this case, whether U.S. congressmen are Republicans are Democrats.

Application: Ranking Partisan language

Monroe et al (2009), "Fightin' Words"

- This paper systematically explores a number of methods for identifying words that are distinctive of groups of speakers
  - in this case, whether U.S. congressmen are Republicans are Democrats.
- First, they separate speeches by topic using latent dirichlet allocation (next lecture).
  - they then test a number of methods for ranking partisanship of words.

Relative Frequency of Words

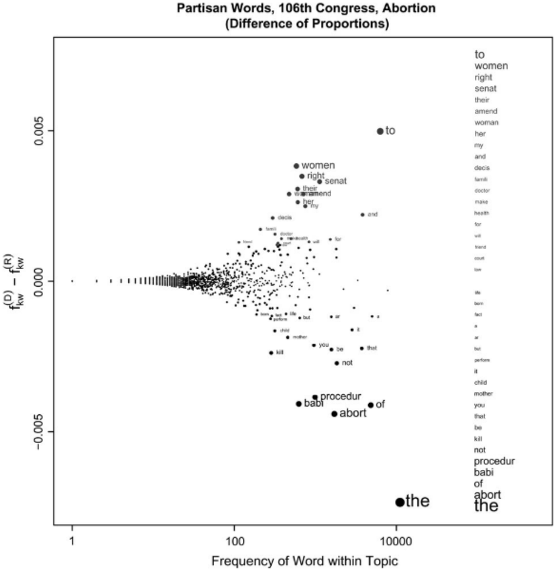


Fig. 1 Feature evaluation and selection using $f_{kw}^{(D)} - f_{kw}^{(R)}$. Plot size is proportional to evaluation weight, $|f_{kw}^{(D)} - f_{kw}^{(R)}|$. The top 20 Democratic and Republican words are labeled and listed in rank order to the right. The results are almost identical for two other measures discussed in the text: unlogged *tf.idf* and frequency-weighted WordScores.

Log Odds Ratio Between Groups

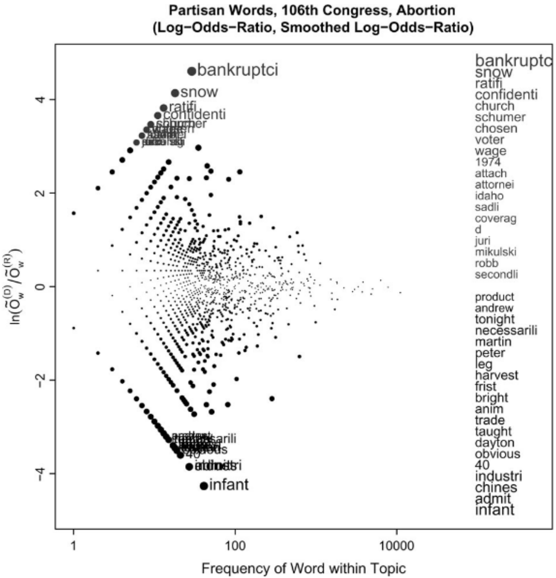


Fig. 2 Feature evaluation and selection using $\hat{\delta}_{kw}^{(D-R)}$. Plot size is proportional to evaluation weight, $\left|\hat{\delta}_{kw}^{(D-R)}\right|$. Top 20 Democratic and Republican words are labeled and listed in rank order. The results are identical to another measure discussed in the text: the log-odds-ratio with uninformative Dirichlet prior.

Bayesian Multinomial Model

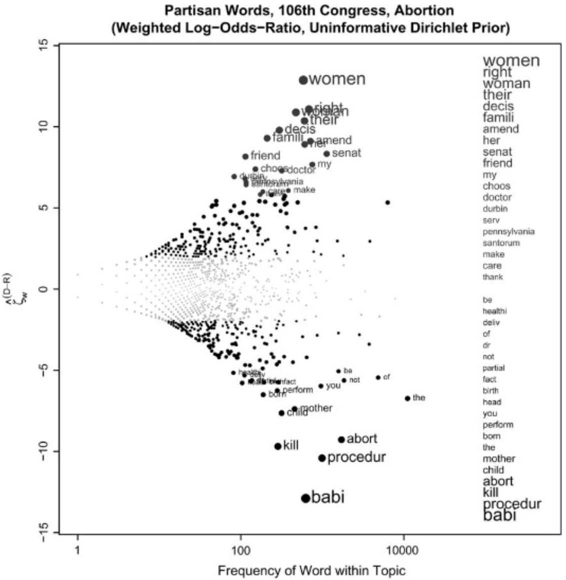


Fig. 4 Feature evaluation and selection using $\hat{\zeta}_{kw}^{(D-R)}$. Plot size is proportional to evaluation weight, $\left|\hat{\zeta}_{kw}^{(D-R)}\right|$; those with $\left|\hat{\zeta}_{kw}^{(D-R)}\right|$<1.96 are gray. The top 20 Democratic and Republican words are labeled and listed in rank order to the right.

Bayesian Multinomial Model, LaPlace Prior

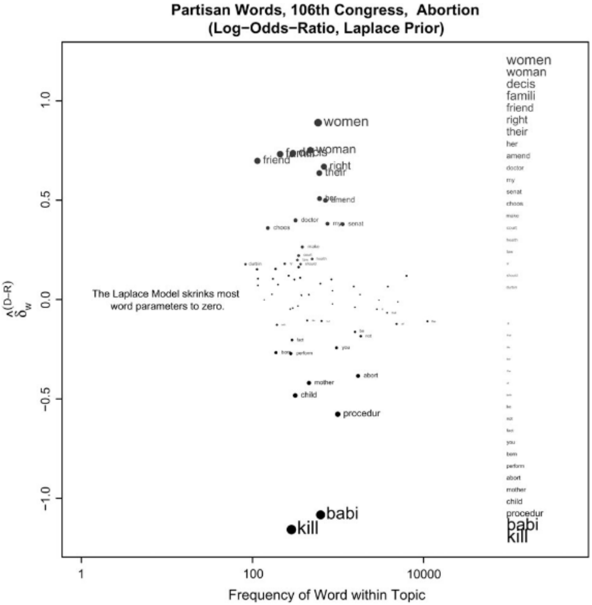


Fig. 6 Feature evaluation and selection using $\hat{\delta}_{kw}^{(D-R)}$. Plot size is proportional to evaluation weight, $\hat{\delta}_{kw}^{(D-R)}$. The top 20 Democratic and Republican words are labeled and listed in rank order to the right.

Questions

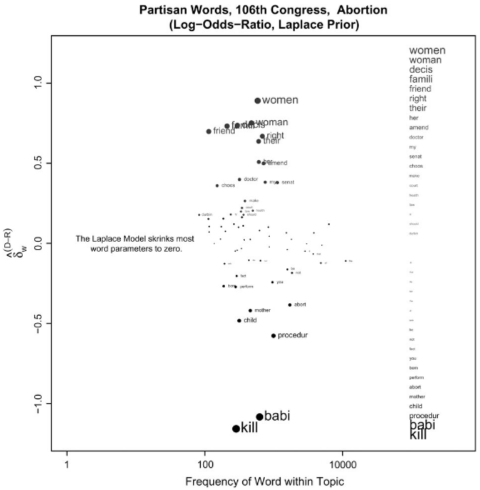


Fig. 6 Feature evaluation and selection using $\hat{\delta}_{kw}^{(D-R)}$. Plot size is proportional to evaluation weight, $\hat{\delta}_{kw}^{(D-R)}$. The top 20 Democratic and Republican words are labeled and listed in rank order to the right.

- drop stopwords?
- try n-grams?
- How robust across topics?
- Is this useful for anything besides description?

Others?

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.

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>
```

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

Other Transformations?

- e.g., Kelly et al (2019) suggest that including indicators for whether a phrase appears in a document (rather than the count) is often independently predictive.

Other Transformations?

- e.g., Kelly et al (2019) suggest that including indicators for whether a phrase appears in a document (rather than the count) is often independently predictive.
- Could add log counts, quadratics in counts, etc.
- Could also add pairwise interactions between word counts/frequencies.

Other Transformations?

- e.g., Kelly et al (2019) suggest that including indicators for whether a phrase appears in a document (rather than the count) is often independently predictive.
- Could add log counts, quadratics in counts, etc.
- Could also add pairwise interactions between word counts/frequencies.
- These often are not done much because of the dimensionality problem.
  - could use feature selection or principal components to deal with that.
  - for machine learning, could use SVM with a polynomial kernel.

Tokenization: Overview

Pre-Processing Text

Counts and Frequencies

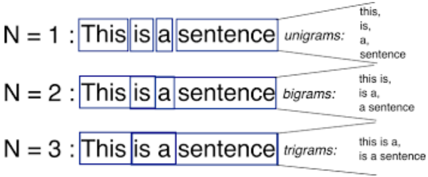
N-Grams

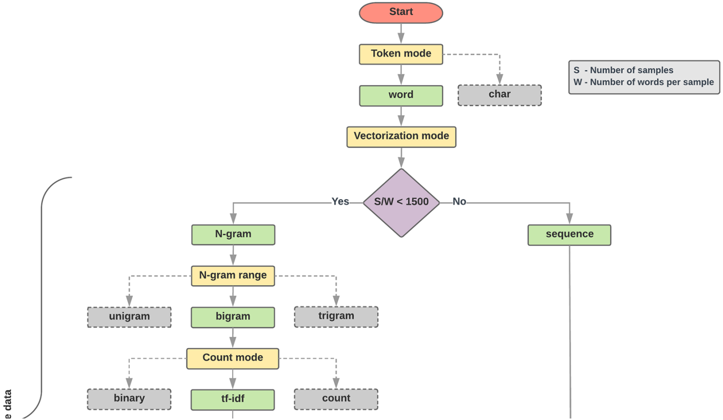
Parts of Speech

Appendix on Course Projects

What are N-grams

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





- Google Developers recommend **tf-idf-weighted bigrams** as a baseline specification for text classification tasks.
  - ideal for fewer, longer documents.
- With more numerous, shorter documents (rows / doclength > 1500), better to use an embedded sequence (starting Week 5).

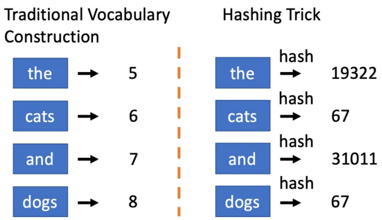
N-grams and high dimensionality

- N-grams will blow up your feature space:
  - filtering out uninformative n-grams is necessary.

N-grams and high dimensionality

- N-grams will blow up your feature space:
  - filtering out uninformative n-grams is necessary.
- Google Developers say that a feature space with $P = 20,000$ will work well for descriptive and prediction tasks.
  - I have gotten good performance with 10K or even 2K features.
  - For supervised learning tasks, a decent baseline is to build a vocabulary of 60K, then use feature selection to get down to 10K.

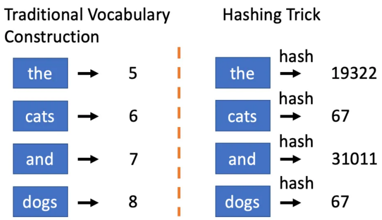
Hashing Vectorizer



- 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



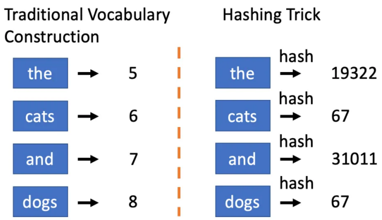
- Rather than make a one-to-one lookup for each n-gram, put n-grams through a hashing function that takes an arbitrary string and outputs an integer in some range (e.g. 1 to 10,000).

```
>>> 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.

Feature selection using univariate comparisions

- χ^2 is a very fast feature selection routine for classification tasks
  - features must be non-negative
  - works on sparse matrices
  - works on multi-class problems
- With negative predictors:
  - use `f_classif.`
- For regression tasks:
  - use `f_regression or OLS coefficients.`

De-Confounded Feature Selection

- What if a feature is important due to a confounding correlation?
  - e.g. in “Fightin Words” paper: say there are more republicans in congress over time, and the word “kill” coincidentally becomes more popular over time.
  - then the republican-”kill” relationship is a spurious correlation and does not say anything about partisan language.

De-Confounded Feature Selection

- What if a feature is important due to a confounding correlation?
  - e.g. in “Fightin Words” paper: say there are more republicans in congress over time, and the word “kill” coincidentally becomes more popular over time.
  - then the republican-”kill” relationship is a spurious correlation and does not say anything about partisan language.
- Solution: de-mean the predictors (word frequencies) by year – that way, partisanship is predicted using only within-year variation.
  - can be done with other groups as well – e.g., compare legislators from the same state.
  - it can also help to de-mean the outcome (partisan label)

De-Confounded Feature Selection

- What if a feature is important due to a confounding correlation?
  - e.g. in "Fightin Words" paper: say there are more republicans in congress over time, and the word "kill" coincidentally becomes more popular over time.
  - then the republican-"kill" relationship is a spurious correlation and does not say anything about partisan language.
- Solution: de-mean the predictors (word frequencies) by year – that way, partisanship is predicted using only within-year variation.
  - can be done with other groups as well – e.g., compare legislators from the same state.
  - it can also help to de-mean the outcome (partisan label)
- What if you want to de-mean by both year and state?
  - $\rightarrow$ take residuals from linear regression of each variable (outcome and predictor) on the category dummies.
  - That is:
    - regress $Y_i = \text{FE}_1 + \text{FE}_2 + \epsilon_i$ and $x_i^w = \text{FE}_1 + \text{FE}_2 + \epsilon_i, \forall w$,
    - take residuals $\tilde{Y}_i = Y_i - \hat{Y}_i$ and $\tilde{x}_i^w = x_i^w - \hat{x}_i^w$
  - Then use residuals as variables, in feature selection step or in machine learning task.

Collocations are Familiar N-grams

- Conceptually, the goal of including n-grams is to featurize **collocations**:
  - Non-compositional: the meaning is not the sum of the parts (kick+the+bucket≠"kick the bucket")

Collocations are Familiar N-grams

- Conceptually, the goal of including n-grams is to featurize **collocations**:
  - Non-compositional: the meaning is not the sum of the parts (kick+the+bucket≠"kick the bucket")
  - Non-substitutable: cannot substitute components with synonyms ("fast food"≠"quick food")

Collocations are Familiar N-grams

- Conceptually, the goal of including n-grams is to featurize **collocations**:
  - Non-compositional: the meaning is not the sum of the parts (kick+the+bucket≠"kick the bucket")
  - Non-substitutable: cannot substitute components with synonyms ("fast food"≠"quick food")
  - Non-modifiable: cannot modify with additional words or grammar: (e.g., "kick around the bucket", "kick the buckets")

Point-wise mutual information

- A metric for identifying collocations is point-wise mutual information:

$$\mathrm{PMI}(w_1, w_2) = \frac{\Pr(w_1\_w_2)}{\Pr(w_1)\Pr(w_2)}$$
$$= \frac{\text{Prob. of collocation, actual}}{\text{Prob. of collocation, if independent}}$$

where w_1 and w_2 are words in the vocabulary, and w_1, w_2 is the N-gram $w_1\_w_2$.

  - ranks words by how often they collocate, relative to how often they occur apart.

Point-wise mutual information

- A metric for identifying collocations is point-wise mutual information:

$$\text{PMI}(w_1, w_2) = \frac{\Pr(w_1\_w_2)}{\Pr(w_1)\Pr(w_2)}$$
$$= \frac{\text{Prob. of collocation, actual}}{\text{Prob. of collocation, if independent}}$$

  where w_1 and w_2 are words in the vocabulary, and w_1, w_2 is the N-gram $w_1\_w_2$.
  - ranks words by how often they collocate, relative to how often they occur apart.
- Generalizes to longer phrases (length N) as the geometric mean of the probabilities:

$$\frac{\Pr(w_1, \ldots, w_N)}{\prod_{i=1}^{n} \sqrt[n]{\Pr(w_i)}}$$

- E.g., for trigrams:

$$\frac{\Pr(w_1, w_2, w_3)}{\sqrt[3]{\Pr(w_1)\Pr(w_2)\Pr(w_3)}}$$

Point-wise mutual information

- A metric for identifying collocations is point-wise mutual information:

$$\mathrm{PMI}(w_1, w_2) = \frac{\Pr(w_1\_w_2)}{\Pr(w_1)\Pr(w_2)} = \frac{\text{Prob. of collocation, actual}}{\text{Prob. of collocation, if independent}}$$

  where w_1 and w_2 are words in the vocabulary, and w_1, w_2 is the N-gram $w_1\_w_2$.
  - ranks words by how often they collocate, relative to how often they occur apart.
- Generalizes to longer phrases (length N) as the geometric mean of the probabilities:

$$\frac{\Pr(w_1, \ldots, w_N)}{\prod_{i=1}^{n} \sqrt[n]{\Pr(w_i)}}$$

- E.g., for trigrams:

$$\frac{\Pr(w_1, w_2, w_3)}{\sqrt[3]{\Pr(w_1)\Pr(w_2)\Pr(w_3)}}$$

- Warning: Rare words that appear together once or twice will have high PMI.
  - Address this with minimum frequency thresholds.

Application: Gentzkow and Shapiro (2010): "What Drives Media Slant?"

Application: Gentzkow and Shapiro (2010): “What Drives Media Slant?”

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

Application: Gentzkow and Shapiro (2010): “What Drives Media Slant?”

- 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: Gentzkow and Shapiro (2010): "What Drives Media Slant?"

- 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. For each phrase w, let D_w be frequency for Democrats, R_w be frequency for Republicans. Let D_w^- and R_w^- be frequencies of *other* phrases.
- Then:

$$\chi_w^2 = \frac{(R_w D_w^- - D_w R_w^-)^2}{(D_w + R_w)(D_w + D_w^-)(R_w + R_w^-)(D_w^- + R_w^-)}$$

  - this is the test statistic for equality between parties of phrase use if they were both drawn from multinomial distributions.
  - in sklearn, it is feature_selection.chi2

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)

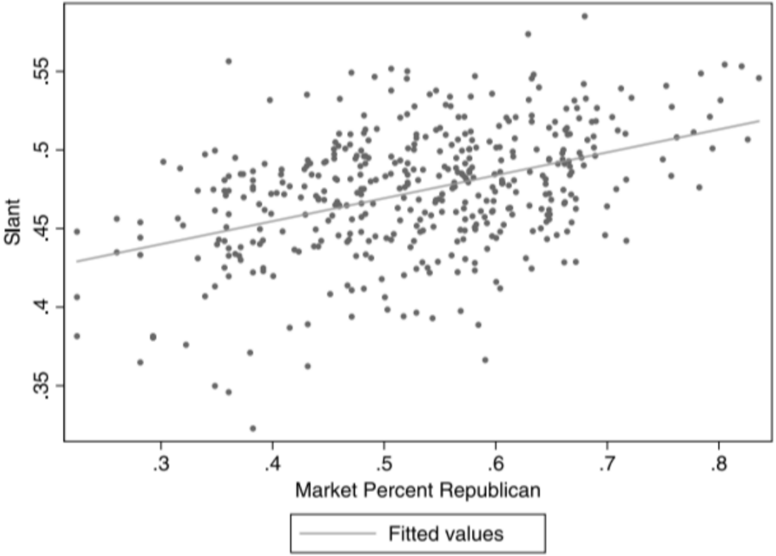


FIGURE 4.—Newspaper slant and consumer ideology. The newspaper slant index against Bush's share of the two-party vote in 2004 in the newspaper's market is shown.

Phrase Dictionaries

- WordNet has some phrases as single entities.
- The Paraphrase Database 2.0 (PPDB, `paraphrase.org/#/download`) has a large database of equivalent/related words/phrases.
- Could take wikipedia article names as lists of multi-word expressions.
- In law, could use legal dictionaries (e.g., "first amendment", "beyond a reasonable doubt").

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 .

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 .

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	The **Mt. Sanitas** loop is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states, provinces	**Palo Alto** is raising the fees for parking.
Facility	FAC	bridges, buildings, airports	Consider the **Golden Gate Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.

Figure 18.1 A list of generic named entity types with the kinds of entities they refer to.

- Blackstone has a trained legal NER system in spaCy (for UK law).

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`
- 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.
- In particular, noun phrases are often informative features – spaCy can do fast noun phrase chunking.

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`
- 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.
- In particular, noun phrases are often informative features – spaCy can do fast noun phrase chunking.
- Can count parts of speech tags as features – e.g., using more adjectives, or using more passive verbs.

Parts of speech tags

- Parts of speech (POS) tags provide useful word categories corresponding to their functions in sentences:
  - Eight main parts of speech: verb (VB), noun (NN), pronoun (PR), adjective (JJ), adverb (RB), determinant (DT), preposition (IN), conjunction (CC).
  - The Penn TreeBank POS tag set (used in many applications) has 36 tags: `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`
- 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.
- In particular, noun phrases are often informative features – spaCy can do fast noun phrase chunking.
- Can count parts of speech tags as features – e.g., using more adjectives, or using more passive verbs.
- POS n-gam frequencies (e.g. NN, NV, VN, ...), like function words, are good stylistic features for authorship detection.
  - not biased by topics/content

What do do with out-of-vocab words

- unless using a hashing vectorizer, have to choose a vocabulary for featurizing a document.
  - e.g., top 10K words by frequency
- what to do with the words that get dropped out?

What do do with out-of-vocab words

- unless using a hashing vectorizer, have to choose a vocabulary for featurizing a document.
  - e.g., top 10K words by frequency
- what to do with the words that get dropped out?
  - drop them
  - replace with "unknown" token
  - replace with part-of-speech tag
  - run (auxiliary) hashing vectorizer on them
  - replace with in-vocab hypernym (from WordNet)
  - others?

Parts of Speech Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

Parts of Speech Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

- Imagine you consider lending $2,000 to one of two borrowers on a crowdfunding website. The borrowers are identical in terms of demographic and financial characteristics. However, the text they provided when applying for a loan differs:
  - Borrower #1: "*I am a hard working person, married for 25 years, and have two wonderful boys. Please let me explain why I need help. I would use the $2,000 loan to fix our roof. Thank you, god bless you, and I promise to pay you back.*"
  - Borrower #2: "*While the past year in our new place has been more than great, the roof is now leaking and I need to borrow $2,000 to cover the cost of the repair. I pay all bills (e.g., car loans, cable, utilities) on time.*"
- Which borrower is more likely to default?

Parts of Speech Predict Loan Repayment

Netzer, Lemaire, and Herzenstein (2019), "When Words Sweat"

- Imagine you consider lending $2,000 to one of two borrowers on a crowdfunding website. The borrowers are identical in terms of demographic and financial characteristics. However, the text they provided when applying for a loan differs:
  - Borrower #1: "*I am a hard working person, married for 25 years, and have two wonderful boys. Please let me explain why I need help. I would use the $2,000 loan to fix our roof. Thank you, god bless you, and I promise to pay you back.*"
  - Borrower #2: "*While the past year in our new place has been more than great, the roof is now leaking and I need to borrow $2,000 to cover the cost of the repair. I pay all bills (e.g., car loans, cable, utilities) on time.*"
- Which borrower is more likely to default?
- "Loan requests written by defaulting borrowers are more likely to include words (or themes) related to the borrower's family, financial and general hardship, mentions of god, and the near future, as well as pleading lenders for help, and using verbs in present and future tenses."

Loan Application Words Predicting Repayment (Netzer, Lemaire, and Herzenstein 2019)

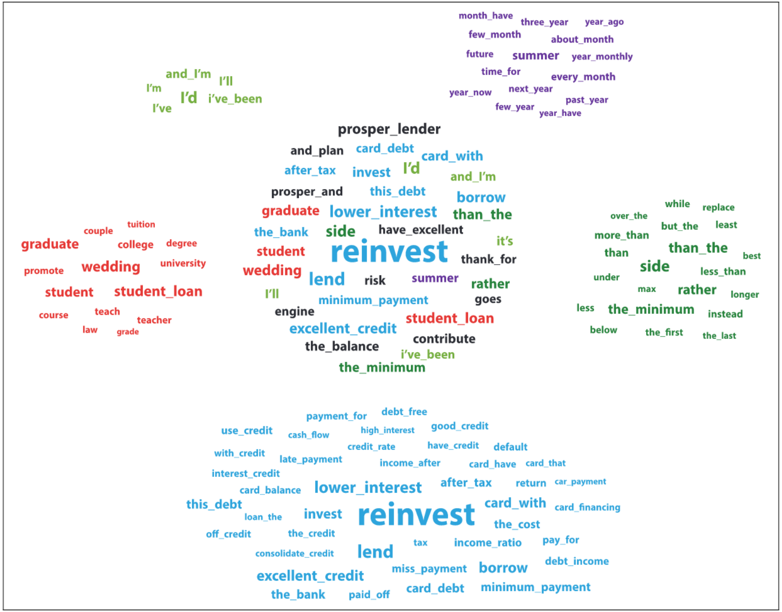


Figure 2. Words indicative of loan repayment.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the right and moving clockwise: relative words, financial literacy words, words related to a brighter financial future, "I" words, and time-related words.

Loan Application Words Predicting Default (Netzer, Lemaire, and Herzenstein 2019)

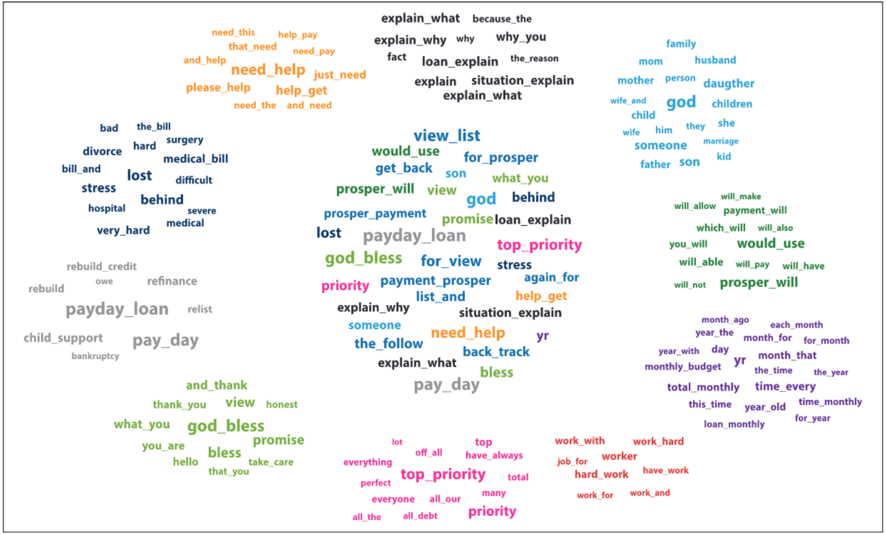
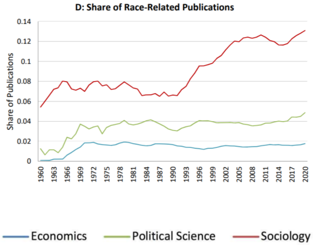


Figure 3. Words indicative of loan default.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the top and moving clockwise: words related to explanations, external influence words and others, future-tense words, time-related words, work-related words, extremity words, words appealing to lenders, words relating to financial hardship, words relating to general hardship, and desperation/plea words.

Advani et al 2021, "Race-Related Research"



Monroe et al 2009, "Fightin Words"

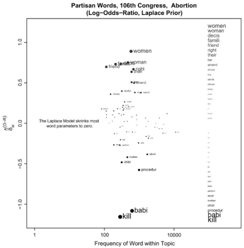


Fig. 6 Feature evaluation and selection using $\hat{\delta}_{kw}^{(D-R)}$. Plot size is proportional to evaluation weight, $|\hat{\delta}_{kw}^{(D-R)}|$. The top 20 Democratic and Republican words are labeled and listed in rank order to the right.

Gentzkow and Shapiro 2010, "What Drives Media Slant"

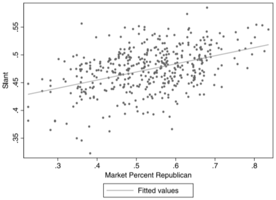


FIGURE 4.—Newspaper slant and consumer ideology. The newspaper slant index against Bush's share of the two-party vote in 2004 in the newspaper's market is shown.

Netzmer et al 2019, "When Words Sweat"

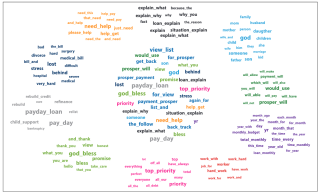


Figure 3. Words indicative of loan default.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the top and moving clockwise: words related to explanations, external influence words and others, future-tense words, time-related words, work-related words, extremity words, words appealing to lenders, words relating to financial hardship, words relating to general hardship, and desperation/plea words.

Social Science Applications: Questions for Understanding

- What is the research question?
- What dataset is being used? Why this dataset?
- What is the paper trying to measure using the dataset? Why?
- What NLP method is being used for the measurement?
  - How was the method validated? What other method could they have tried?
- What were the main results from a substantive social-science standpoint?
  - Why are they important? What results seemed incomplete or non-robust?
- What are the limitations and open questions?

Course Project Logistics

`https://bit.ly/NLP-proj`

- If you are signed up for the credits, the focus of your work in this course should be on the project.
  - Can be done individually or in small groups (up to 4 students).
  - Do an original analysis using methods learned in the course, and write a paper about it.

Previous Year's Projects (1)

Previous Year's Projects (1)

- One of the groups began building a legal research application for Swiss lawyers:
  - see `https://deepjudge.ai/`
  - feature-rich legal search engine, won some VC funding and now part of ETH AI Center

Previous Year's Projects (1)

- One of the groups began building a legal research application for Swiss lawyers:
  - see `https://deepjudge.ai/`
  - feature-rich legal search engine, won some VC funding and now part of ETH AI Center
- Another group partnered with a local company to build out environmental-regulation analytics
  - won an Innosuisse grant.

Previous Year's Projects (2)

Five projects have been published:

1. "Legal language modeling with transformers" (Lazar Peric, Stefan Mijic, Dominik Stammbach, Elliott Ash), *Proceedings of ASAIL* (2020).
2. "Entropy in Legal Language" (Roland Friedrich, Mauro Luzzatto, Elliott Ash), *NLLP @ KDD* (2020).
3. "Towards Automated Anamnesis Summarization: BERT-based Models for Symptom Extraction" (Anton Schäfer, Nils Blach, Oliver Rausch, Maximilian Warm, Nils Krüger), *Machine Learning for Health at NeurIPS* (2020).
4. "Kwame: A Bilingual AI Teaching Assistant for Online SuaCode Courses" (George Boateng), *International Conference on AI in Education* (2021)
5. "MemSum: Extractive Summarization of Long Documents using Multi-Step Episodic Markov Decision Processes" (Nianlong Gu, Elliott Ash, Richard Hahnloser), forthcoming ACL Main Conference (2022)

Previous Year's Projects (3)

A number of other projects that are likely to get published, e.g.:

1. partisan tweet generator that responds in the style of a Republican or Democrat.
2. analysis of bias towards immigrants in the early 1900s using old newspapers.
3. causal analysis using deep instrumental variables of what arguments in judicial opinions increase citations
4. partisan question answering system that answers questions with a partisan slant.
5. an audio/text analysis of central bank speeches and inflation beliefs.
6. system for predicting judicial decisions based on the submitted briefs

Project Topics and First Steps

- Picking a topic:
  - You are welcome to come up with your own topic. We will provide feedback on that.
  - We have a list of suggested topics with project advisors.
  - I can also provide advice about which of these topics is a good fit based on team interests and skills.
- First steps:
  - once you have formed a group, send to Afra a list of team members with their resumes, research experience, and interests.
    - if you are interested in one or more of the suggested topics, include that in the email
  - we will then match project advisors and set up meeting

Questions / comments?

- As suggested, we will set up a meet-and-greet for those doing projects.