Embeddings II: Document Vectors and Applications

Elliott Ash

Bocconi 2018

Rudolph and Blei (2017)

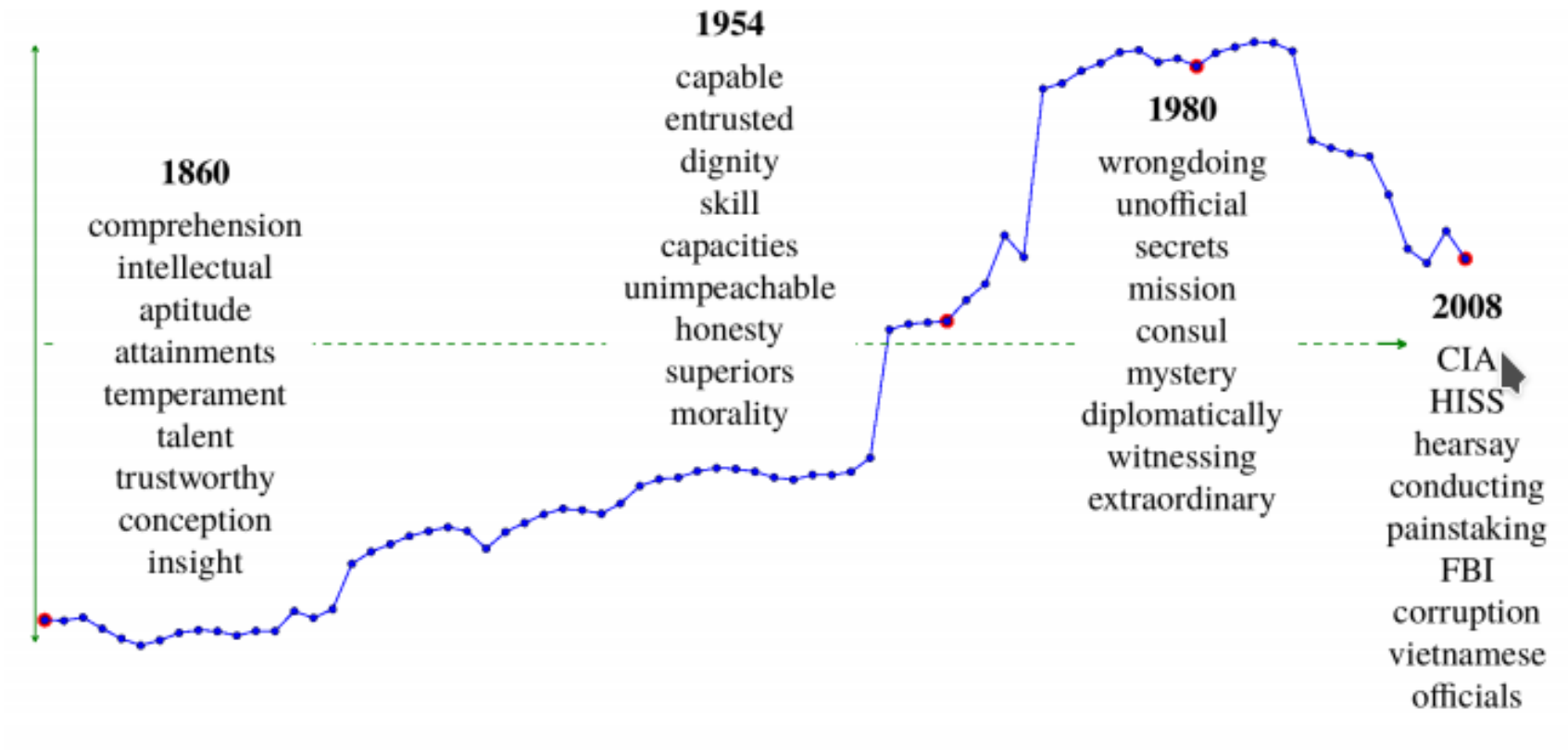
- Train word embeddings on the U.S. Congressional Record, 1858-2009.
- Dynamic word embeddings model:
  - Captures how the meaning of words evolves over time.
  - The innovation is to include "year" in the embedding model, and allow word vectors to drift over time (following a random walk).
    - Doing this with standard word2vec would require you to train a different model in each year, so no information could be shared across years.

Meaning Changes

computer	
1858	**1986**
computer	computer
draftsman	software
draftsmen	computers
copyist	copyright
photographer	technological
computers	innovation
copyists	mechanical
janitor	hardware
accountant	technologies
bookkeeper	vehicles

bush	
1858	**1990**
bush	bush
barberry	cheney
rust	nonsense
bushes	nixon
borer	reagan
eradication	george
grasshoppers	headed
cancer	criticized
tick	clinton
eradicate	blindness

Drift in word “intelligence”



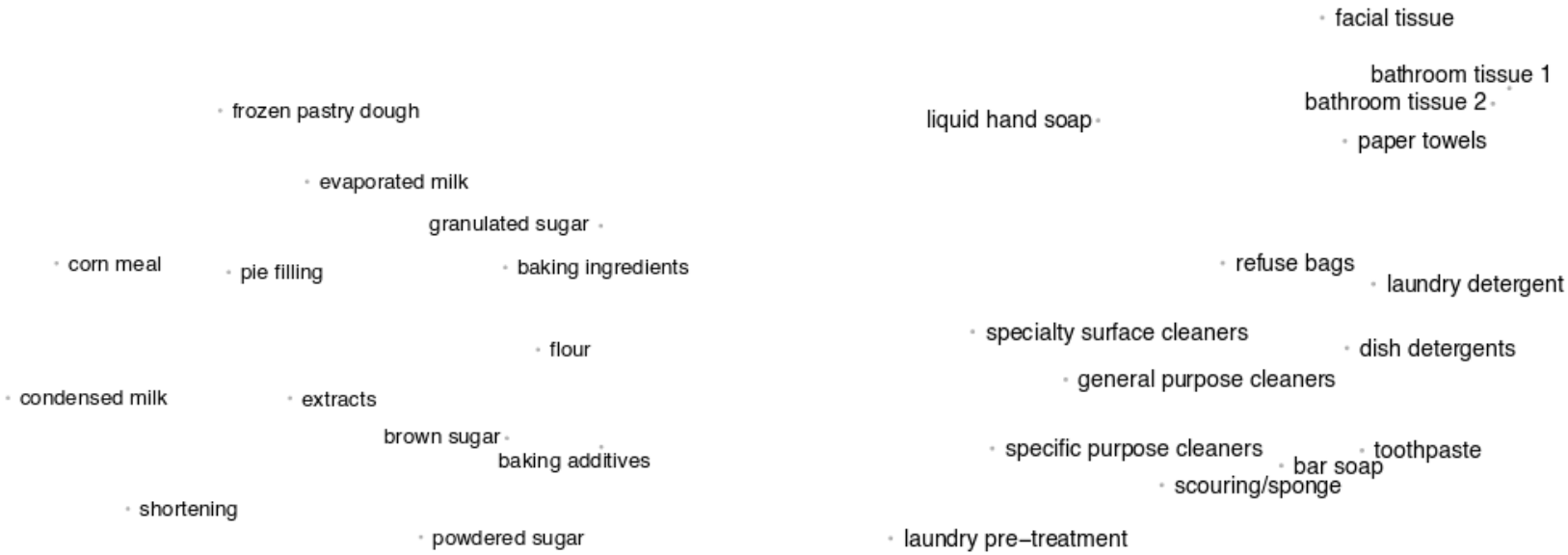
Drift in word “prostitution”

prostitution				
1930	**1945**	**1962**	**1988**	**1990**
prostitution	prostitution	prostitution	harassment	prostitution
punishing	indecent	indecent	intimidation	servitude
immoral	vile	harassment	prostitution	harassment
bootlegging	immoral	intimidation	counterfeit	intimidation
riotous	induces	sexual	illegal	trafficking
forbidden	incite	vile	trafficking	harassing
anarchists	abortion	counterfeit	indecent	apprehended
assemblage	forbid	anarchists	disregard	killings
forbid	harboring	mobs	anarchists	labeled
abet	assemblage	lawbreakers	punishing	naked

Athey, Blei and Ruiz (2018): Shopping cart embeddings

- Use embeddings for goods in a shopping cart, rather than words in a document.
  - predict co-occuring goods in the basket
  - use embedding layer to represent goods in a geometric space.
- Data set: 2 years of shopping data from a large grocery store
  - 570K baskets, 6M purchases, 5.5K unique items

Embeddings capture features of groceries



Embeddings capture complementarity/substitutability

query items	complementarity score		exchangeability score	
mission tortilla soft taco 1	2.40	taco bell taco seasoning mix	0.05	mission fajita size
	2.26	mcrmck seasoning mix taco	0.07	mission tortilla soft taco 2
	2.24	lawrys taco seasoning mix	0.13	mission tortilla fluffy gordita
private brand hot dog buns	2.99	bp franks meat	0.11	ball park buns hot dog
	2.63	bp franks bun size	0.13	private brand hotdog buns potato 1
	2.37	bp franks beed bun length	0.15	private brand hotdog buns potato 2
private brand mustard squeeze bottle	0.50	private brand hot dog buns	0.15	frenchs mustard classic yellow squeeze
	0.41	private brand cutlery full size forks	0.16	frenchs mustard classic yellow squeezed
	0.24	best foods mayonnaise squeeze	0.21	heinz ketchup squeeze bottle
private brand napkins all occasion	0.78	private brand selection plates 6 7/8 in	0.09	vnty fair napkins all occasion 1
	0.50	private brand selection plates 8 3/4 in	0.11	vnty fair napkins all occasion 2
	0.49	private brand cutlery full size forks	0.12	private brand selection premium napkins

Approaches to Vectorizing Documents

- In Lecture 2, we started with the baseline approach to vectorizing documents: sparse vectors of token counts/frequencies.
  - high-dimensional sparse data has pros and cons
  - can use documents of arbitrary length
  - can capture local word order with n-grams, but long-run word order is lost.
- In Lecture 13, we used an embedding layer to take the whole document as input, pad documents to the same length, and represent the document as a flattened series of embedding vectors.
  - potentially captures information on long-range ordering of features in documents
  - DNNs work better with dense vectors
  - computationally demanding, and can only use relatively short documents
- Here, we review some other approaches for document embeddings.

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 (weighted by a_w), of the vectors $\vec{w}$ for each word w in the document.
  - "Document" could be sentence, paragraph, section, etc.
  - vectors trained using Word2Vec or GloVe (pre-trained or trained on the corpus).

Sentence Embeddings (Code)

```
# Continuous bag-of-words representation
from gensim.models import Word2Vec
w2v = Word2Vec.load('w2v-vectors.pkl')

sentvecs = []
for sentence in sentences:
    vecs = [w2v.wv[w] for w in sentence if w in w2v.wv]
    if len(vecs)== 0:
        sentvecs.append(np.nan)
        continue
    sentvec = np.mean(vecs, axis=0)
    sentvecs.append(sentvec.reshape(1,-1))
sentvecs[0]

from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity(sentvecs[0],
                  sentvecs[1])[0][0]
```

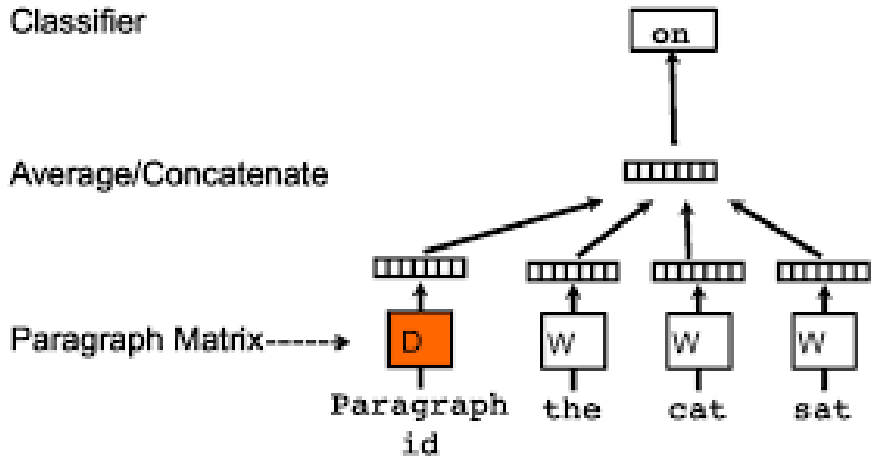
Sentence Embeddings

- Can filter or weight words:
  - drop stopwords
  - filter on parts of speech (e.g., keep only nouns, adjectives, and verbs)
  - weight words by idf (a_w)
- Arora, Liang, and Ma provide a "tough to beat baseline" for sentence embeddings:
  - compute 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.
  - subtract first principal component from matrix of embeddings for the sentence

Doc2Vec (Le and Mikolov)



- Doc2Vec generalizes Word2Vec to whole 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.

Document Vectors

- 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, the components have a topical interpretation; in document embeddings, the components have a geometric interpretation.

Doc2Vec in gensim

```
from gensim.models.doc2vec import Doc2Vec, TaggedDocument
doc_iterator = [TaggedDocument(doc, [i]) for
                i, doc in enumerate(docs)]
model = 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
                max_vocab_size=1000) # max vocab size
```

Using Doc2Vec

```
# matrix of all document vectors:
D = d2v.docvecs.vectors_docs
D.shape

D

# get all pair-wise document similarities
pairwise_sims = cosine_similarity(D)
pairwise_sims.shape

pairwise_sims[:3,:3]
```

Document Clusters

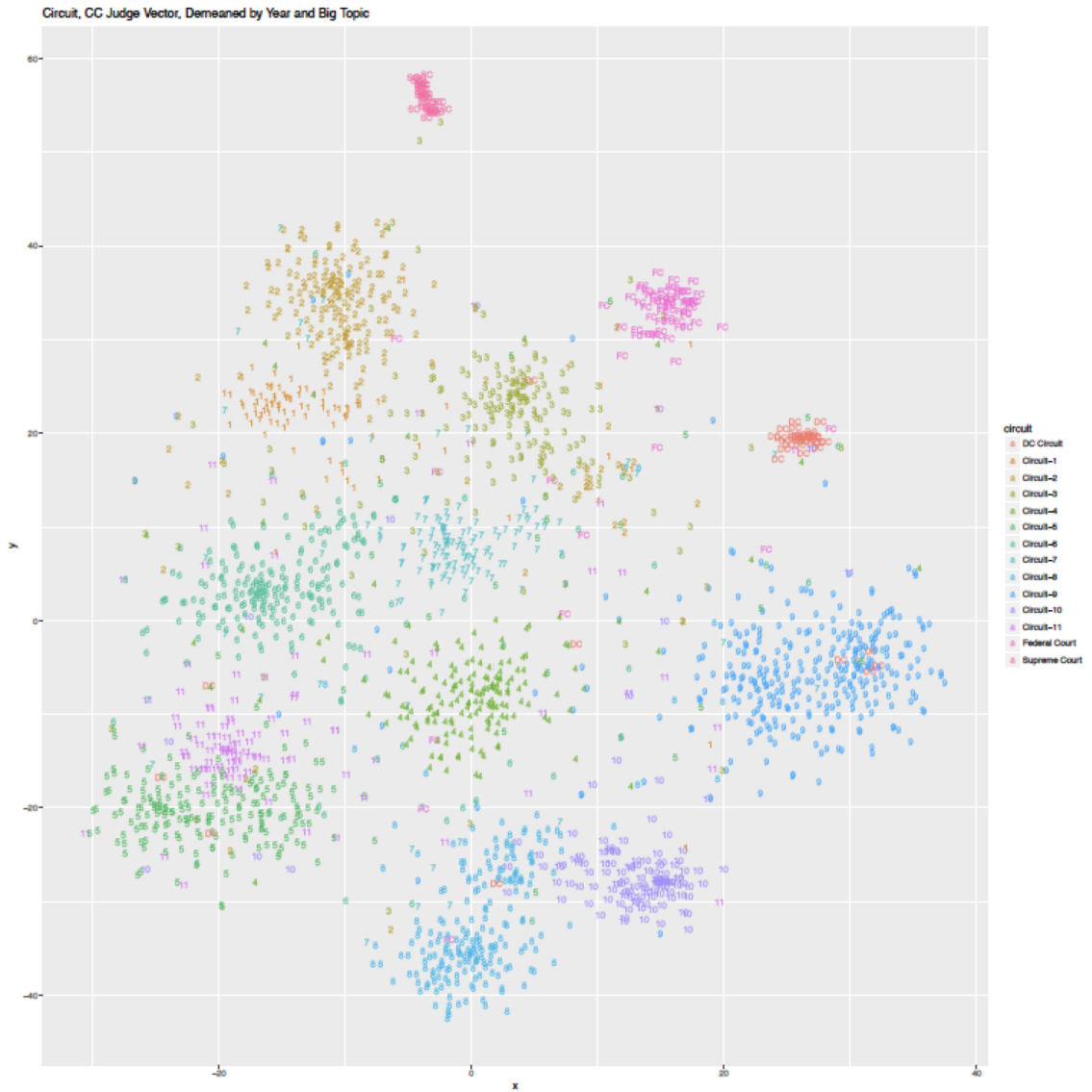
```
num_clusters = 50
kmw = KMeans(n_clusters=num_clusters)
kmw.fit(D)
doc_clusters = kmw.labels_.tolist()
```

Document Vectors for Judicial Opinions

- Ash and Chen (2018) produce document vectors for each case to understand differences between judges and courts.
- We de-mean vectors by group (court, topic, or year) to extract relevant information:
  - de-mean by topic-year to distinguish courts.
  - de-mean by court-topic to distingush years.
  - de-mean by court-year to distinguish topics.

Visual Structure of Case Vectors by Circuit

Figure 1: Centered by Topic-Year, Averaged by Judge, Labeled by Court



tsne plot

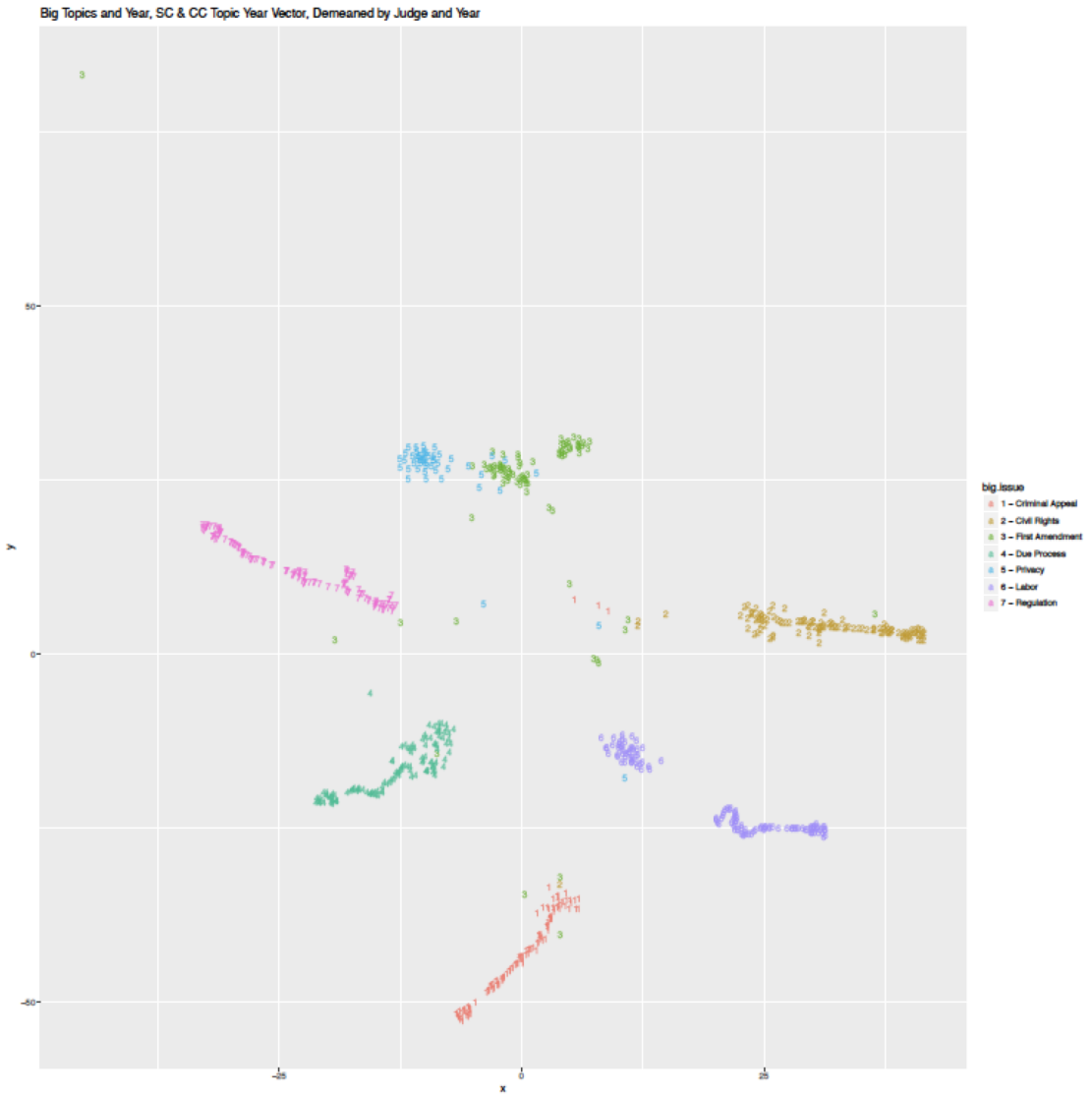
Visual Structure of Case Vectors by Decade

Figure 2: Centered by Court-Topic, Averaged by Court-Year, Labeled by Decade

tsne plot

Visual Structure of Case Vectors by Topic

Figure 3: Centered by Judge-Year, Averaged by Topic-Year, Labeled by Topic



tsne plot

Visual Structure of Case Vectors by Birth Cohort

Figure 5: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Judge Birth Cohort

tsne plot

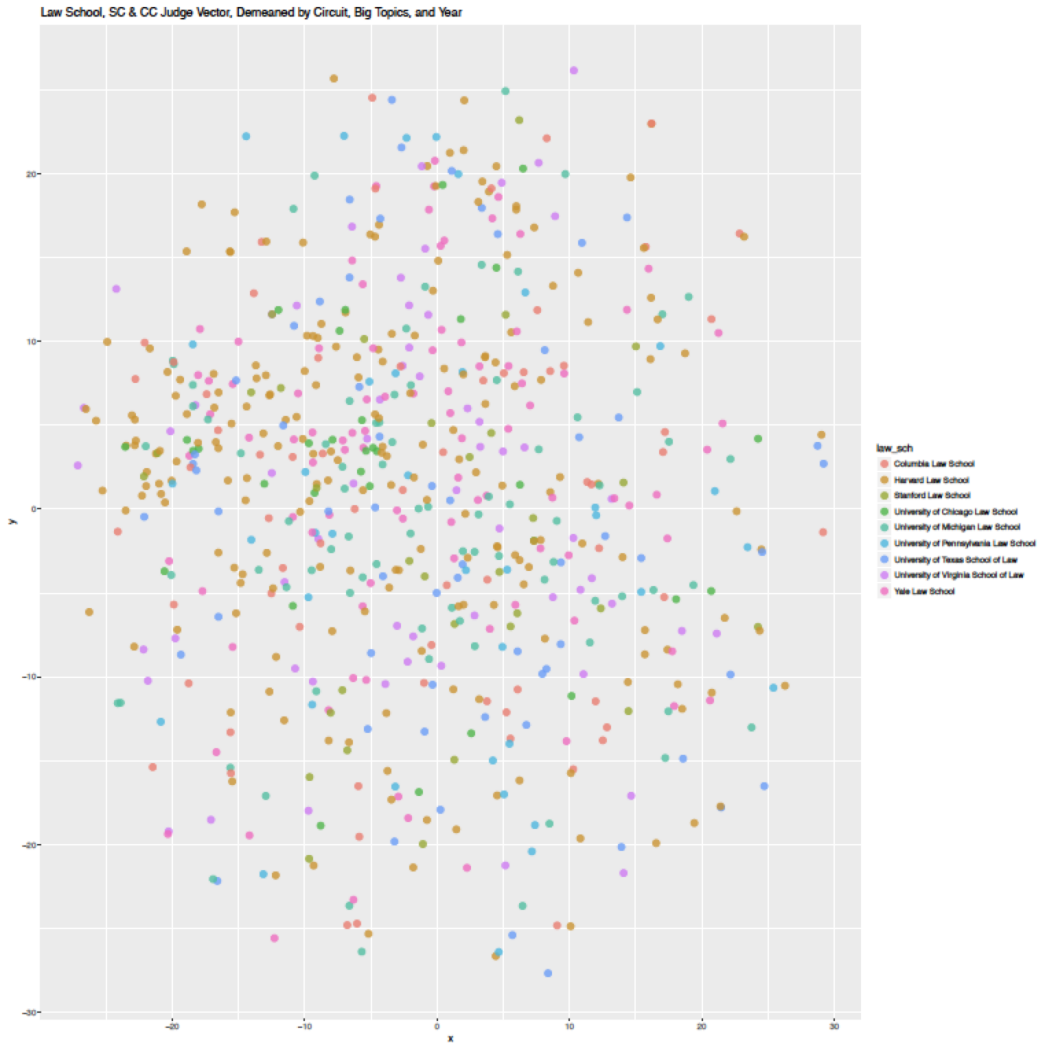
Visual Structure of Case Vectors by Party

Figure 4: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Political Party

tsne plot

Visual Structure of Case Vectors by Law School

Figure 6: Centered by Court-Topic-Year, Averaged by Judge, Labeled by Law School Attended



Relatedness between judges (e.g. Richard Posner)

Circuit Judge Name	Similarity	Rank	Circuit Judge Name	Similarity	Rank
POSNER, RICHARD A.	1.000	1	TONE, PHILIP W.	0.459	16
EASTERBROOK, FRANK H.	0.663	2	SIBLEY, SAMUEL	0.459	17
SUTTON, JEFFREY S.	0.620	3	SCALIA, ANTONIN	0.456	18
NOONAN, JOHN T.	0.596	4	COLLOTON, STEVEN M.	0.445	19
NELSON, DAVID A.	0.592	5	DUNIWAY, BENJAMIN	0.438	20
CARNES, EDWARD E.	0.567	6	GIBBONS, JOHN J.	0.422	21
FRIENDLY, HENRY	0.566	7	BOGGS, DANNY J.	0.420	22
KOZINSKI, ALEX	0.563	8	BREYER, STEPHEN G.	0.414	23
GORSUCH, NEIL M.	0.559	9	GOODRICH, HERBERT	0.412	24
CHAMBERS, RICHARD H.	0.546	10	LOKEN, JAMES B.	0.410	25
FERNANDEZ, FERDINAND F.	0.503	11	WEIS, JOSEPH F.	0.408	26
EDMONDSON, JAMES L.	0.501	12	SCALIA, ANTONIN (SCOTUS)	0.406	27
KLEINFELD, ANDREW J.	0.491	13	BOUDIN, MICHAEL	0.403	28
WILLIAMS, STEPHEN F.	0.481	14	RANDOLPH, A. RAYMOND	0.397	29
KETHLEDGE, RAYMOND M.	0.459	15	MCCONNELL, MICHAEL W.	0.390	30

Document vectors demeaned by court, year, and topic, then aggregated by judge.

Caliskan, Bryson, and Narayanan (*Science* 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. . . "

Word Embedding Association Test

- Target words:
  - programmer, engineer, scientist, ...
  - nurse, teacher, librarian, ...
- Attribute words:
  - man, male, ...
  - woman, female, ...
- WEAT Test:
  - Compute similarities between all target words and all attribute words
  - Compute mean target-attribute clustering

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.

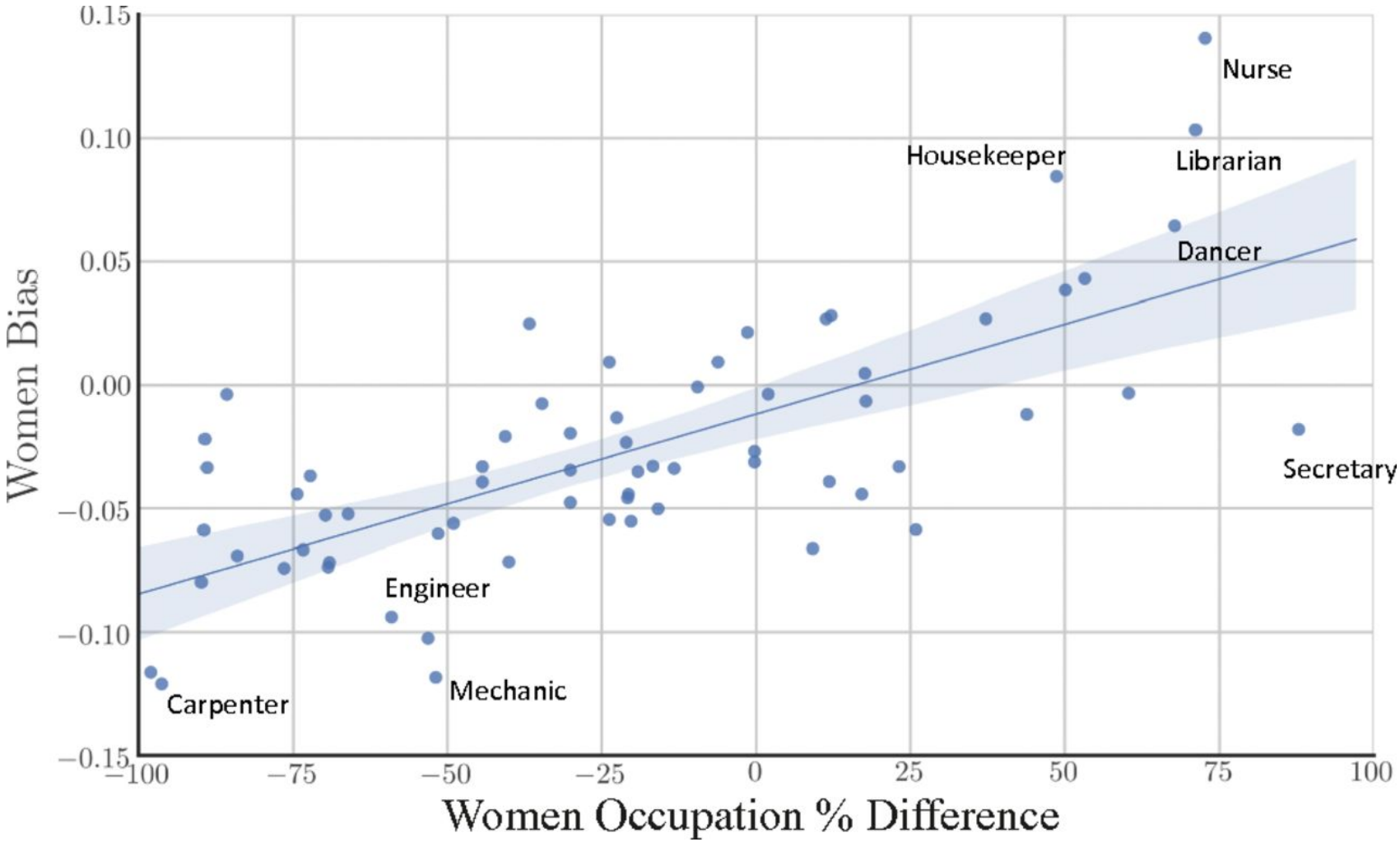
Results

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

Bolukbasi et al (NIPS 2016)

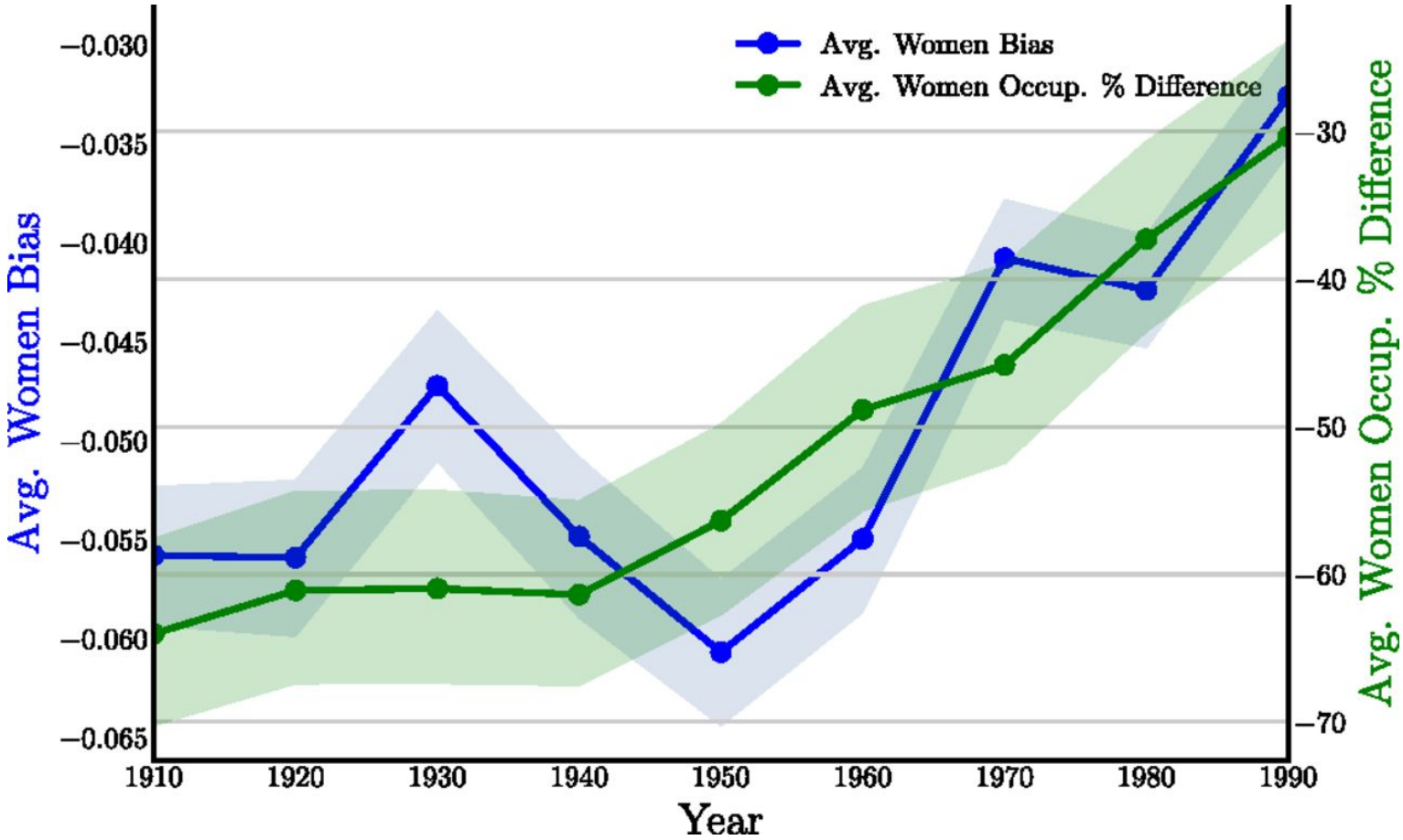
- "Geometrically, **gender bias is first shown to be captured by a direction in the word embedding**."
- "Second, gender neutral words are shown to be linearly separable from gender definition words in the word embedding."
- "Using these properties, we provide a methodology for modifying an embedding to remove gender stereotypes, such as the association between the words receptionist and female, while maintaining desired associations such as between the words queen and female."

Garg, Schiebinger, Jurafsky, and Zou (2018)



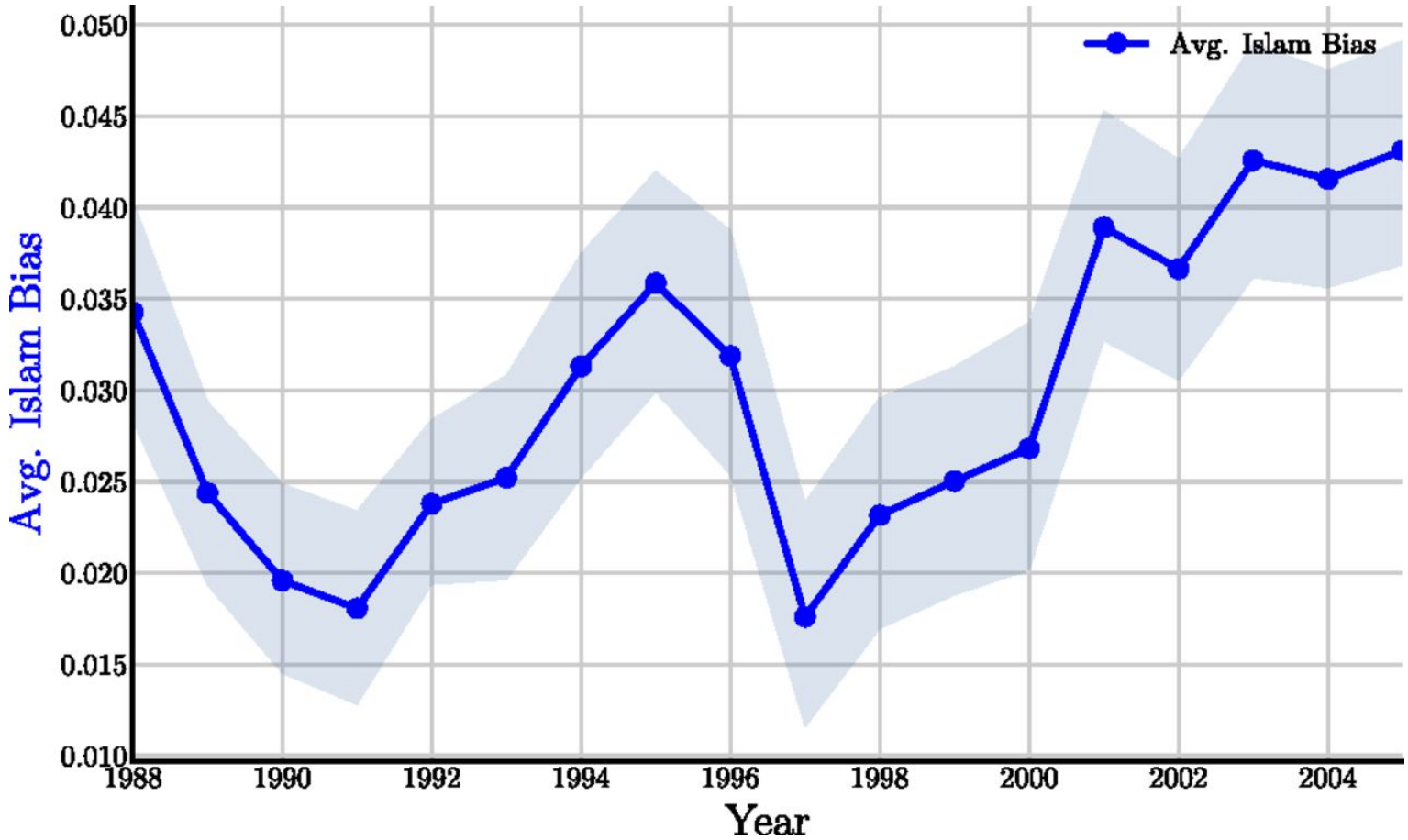
Women's occupation relative percentage vs. embedding bias in Google News vectors.

Garg, Schiebinger, Jurafsky, and Zou (2018)



Average gender bias score over time in COHA embeddings in occupations vs. the average percentage of difference. More positive means a stronger association with women. In blue is relative bias toward women in the embeddings, and in green is the average percentage of difference of women in the same occupations.

Garg et al: Islam↔Terrorism



Religious (Islam vs. Christianity) bias score over time for words related to terrorism in New York Times data.

Garg et al: Ethnic groups $\leftrightarrow$ Occupations

Hispanic	Asian	White
Housekeeper	Professor	Smith
Mason	Official	Blacksmith
Artist	Secretary	Surveyor
Janitor	Conductor	Sheriff
Dancer	Physicist	Weaver
Mechanic	Scientist	Administrator
Photographer	Chemist	Mason
Baker	Tailor	Statistician
Cashier	Accountant	Clergy
Driver	Engineer	Photographer

The top 10 occupations most closely associated with each ethnic group in the Google News embedding.

Garg et al: Female-Associated Words Over Time

1910	1950	1990
Charming	Delicate	Maternal
Placid	Sweet	Morbid
Delicate	Charming	Artificial
Passionate	Transparent	Physical
Sweet	Placid	Caring
Dreamy	Childish	Emotional
Indulgent	Soft	Protective
Playful	Colorless	Attractive
Mellow	Tasteless	Soft
Sentimental	Agreeable	Tidy

Kozlowski, Evans, and Taddy (2018)

- Data-set: Google 5-grams.
  - n-grams of length five for U.S. and U.K. publications.
  - provides counts for each year
- Extract language dimensions:
  - get the complete list of WordNet antonym pairs (e.g, "weak/strong", "tall/short")
    - filter on document frequency to 428 pairs.
  - map the dimensional shifts between the antonyms.
  - compare this vector shift to the one between men and women.

Mapping gender, class, and race

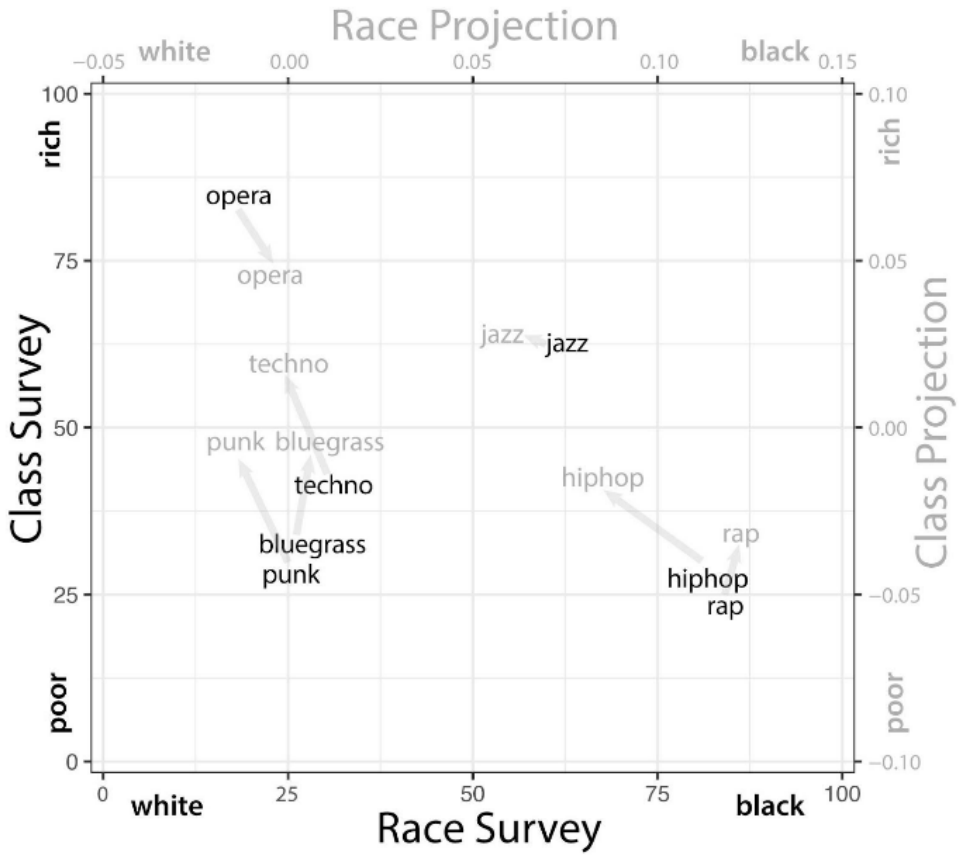
Gender	Class	Race†
man – woman	rich – poor	black – white
men – women	richer – poorer	blacks – whites
he – she	richest – poorest	Blacks – Whites
him – her	affluence – poverty	Black – White
his – her	affluent – impoverished	African – European
his – hers	expensive – inexpensive	African – Caucasian
boy – girl	luxury – cheap	
boys – girls	opulent – needy	
male – female		
masculine – feminine		

Matching antonyms to gender/class

Gender dimension nearest neighbors	
1. rugged–delicate	.219 (.213, .224)
2. soft–loud	-.209 (-.216, -.201)
3. tender–tough	-.202 (-.210, -.197)
4. timid–bold	-.181 (-.186, -.174)
5. soft–hard	-.161 (-.168, -.158)

Class dimension nearest neighbors	
1. weak-strong	-.292 (-.301, -.287)
2. fortunate-unfortunate	.291 (.286, .297)
3. unhappy-happy	-.259 (-.266, -.254)
4. beautiful-ugly	.242 (.238, .245)
5. potent_impotent	.234 (.227, .244)

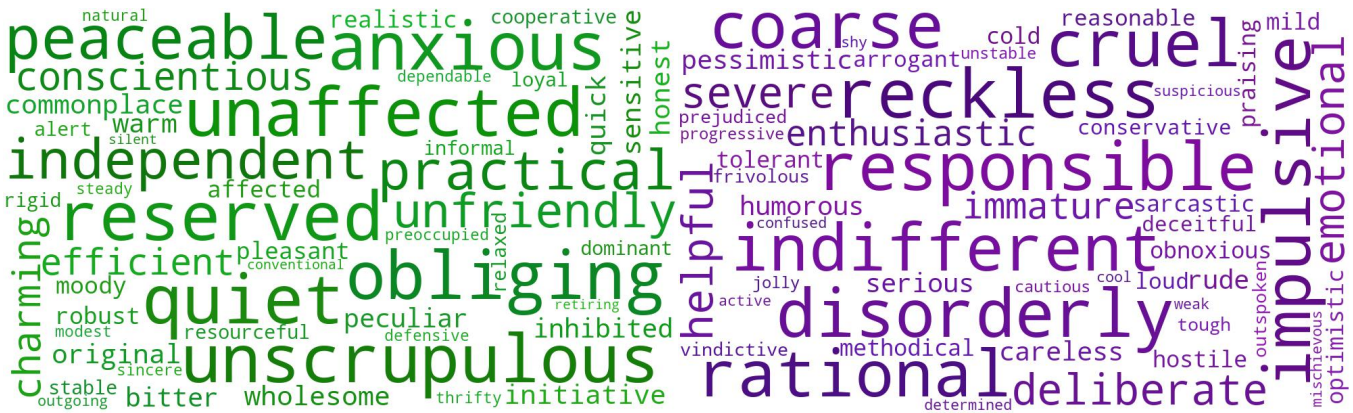
Mapping musical genres to race/class



Word Associations in Judge Language

- Positive (left) and negative (right) language:

- Innocent (left) and guilty (right) language:

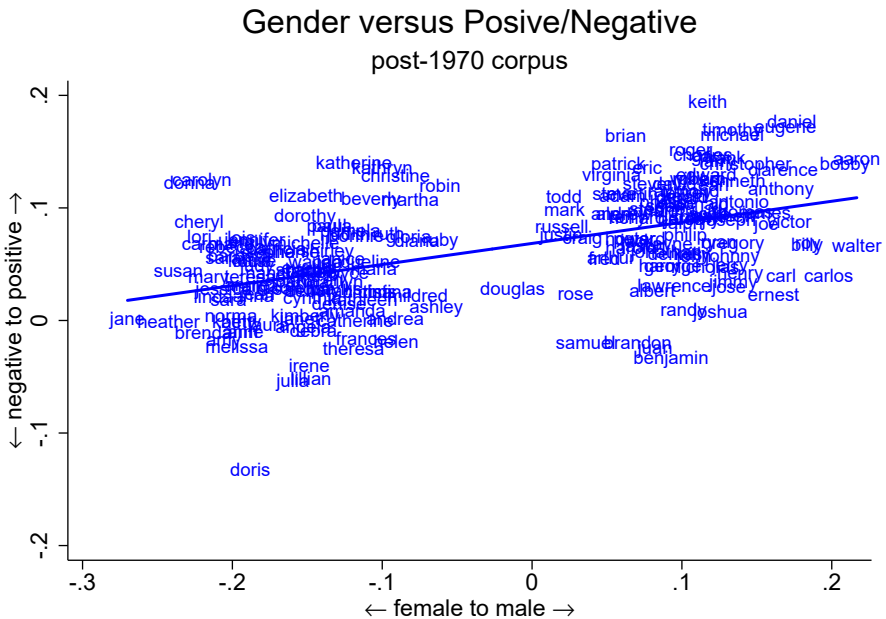


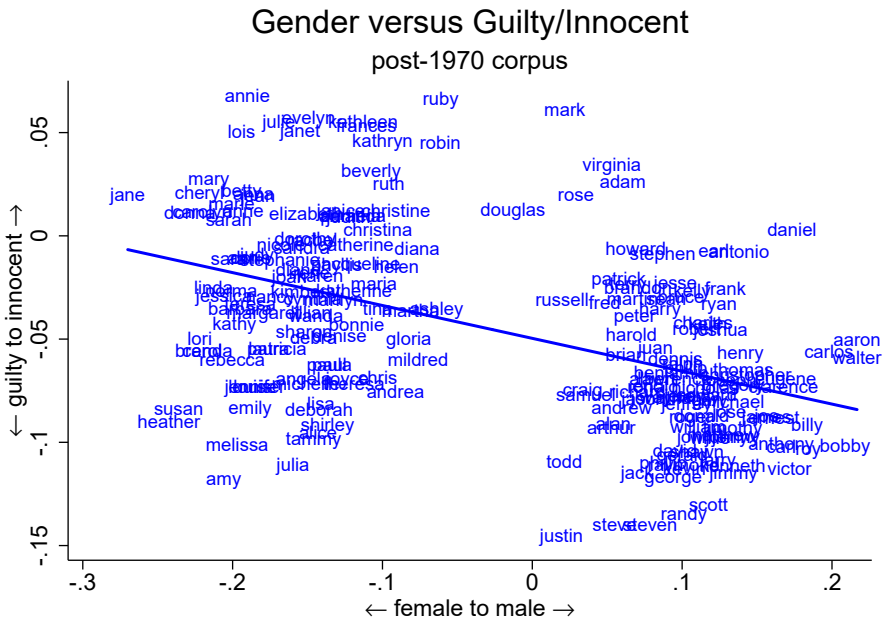
Word Associations in Judge Language

- Male (left) and female (right) adjectives:

- white-race (left) and black-race (right) adjectives:

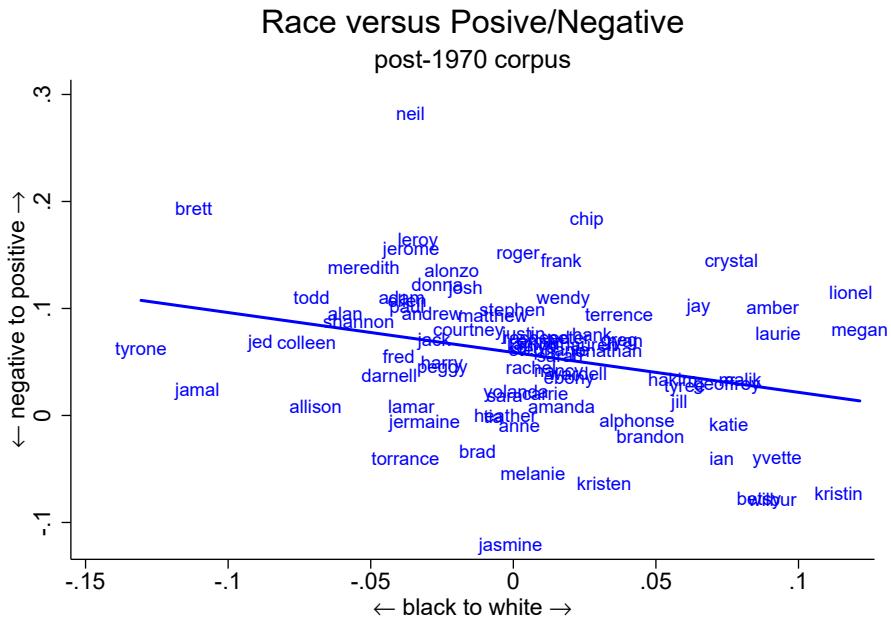
Implicit Gender Bias

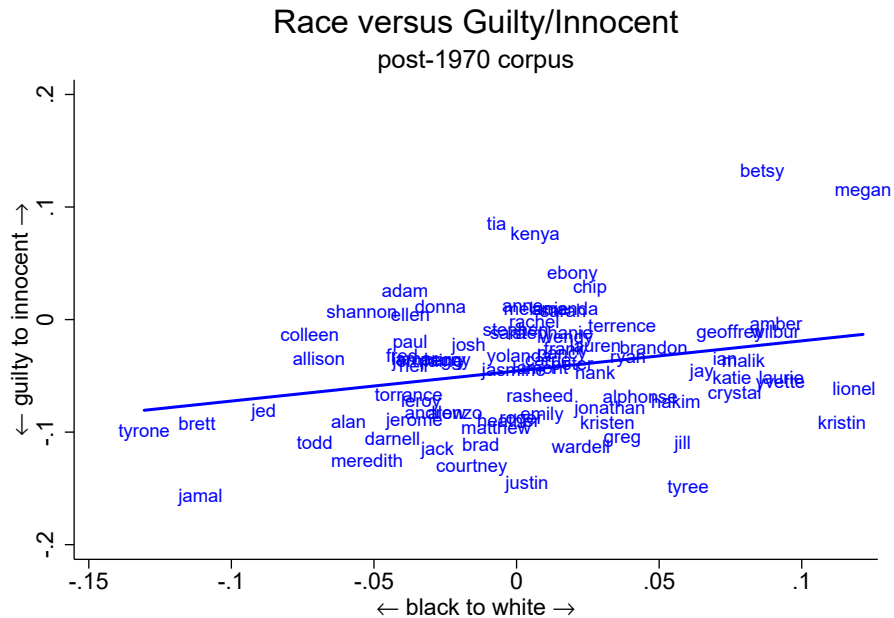




- Male names are more associated to "positive" language and female names are more associated to "innocent" language

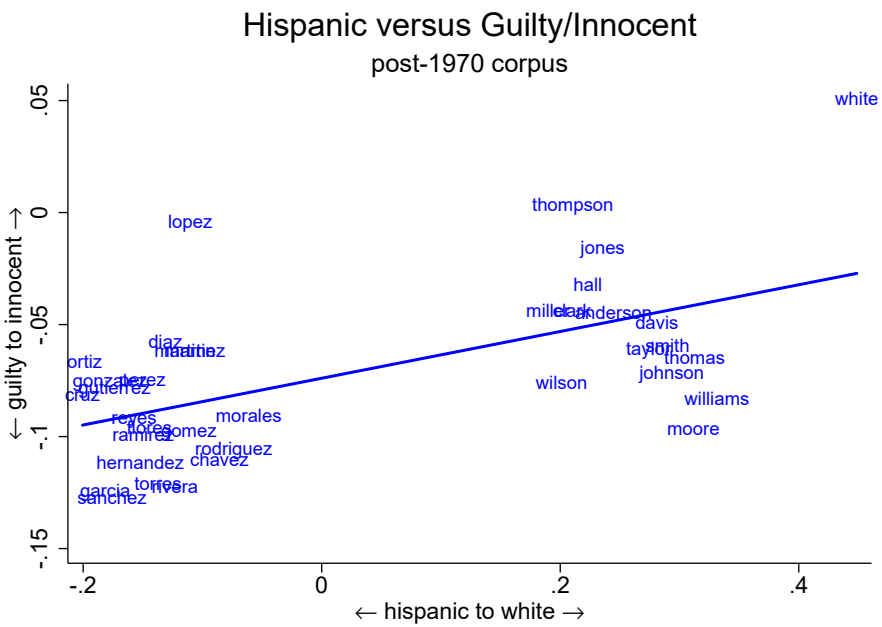
Implicit Racial Bias

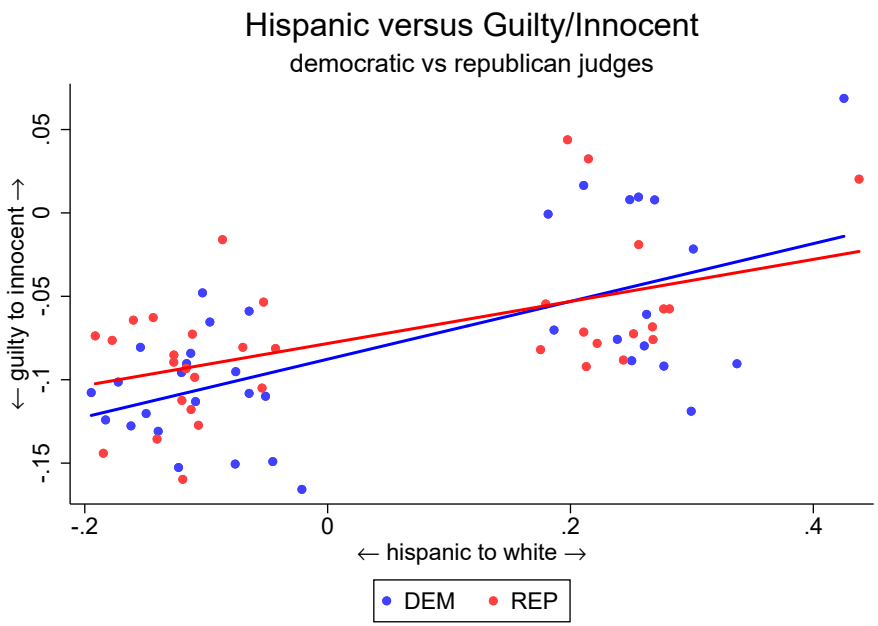




- An apparent "positive" relation for black names, but black-sounding names are more associated to "guilty" language

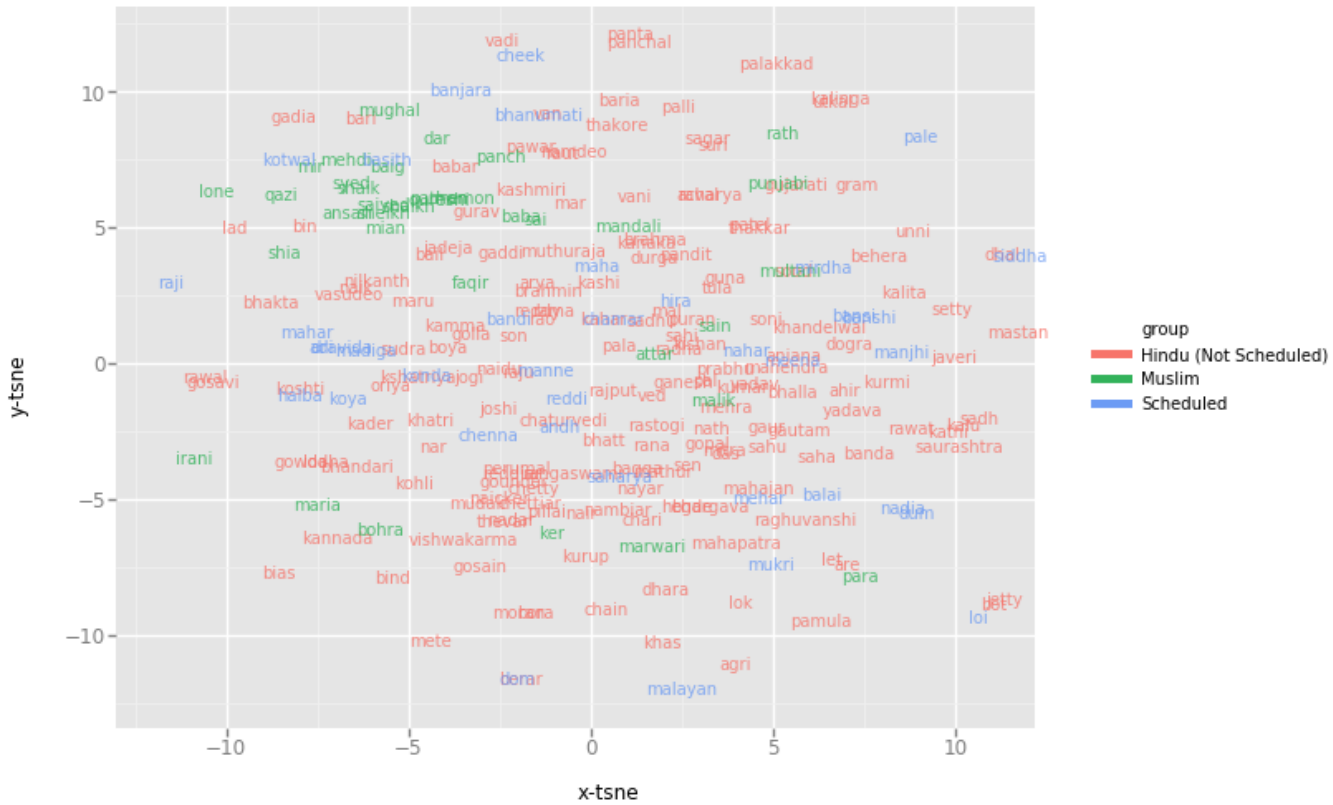
Implicit Hispanic Bias





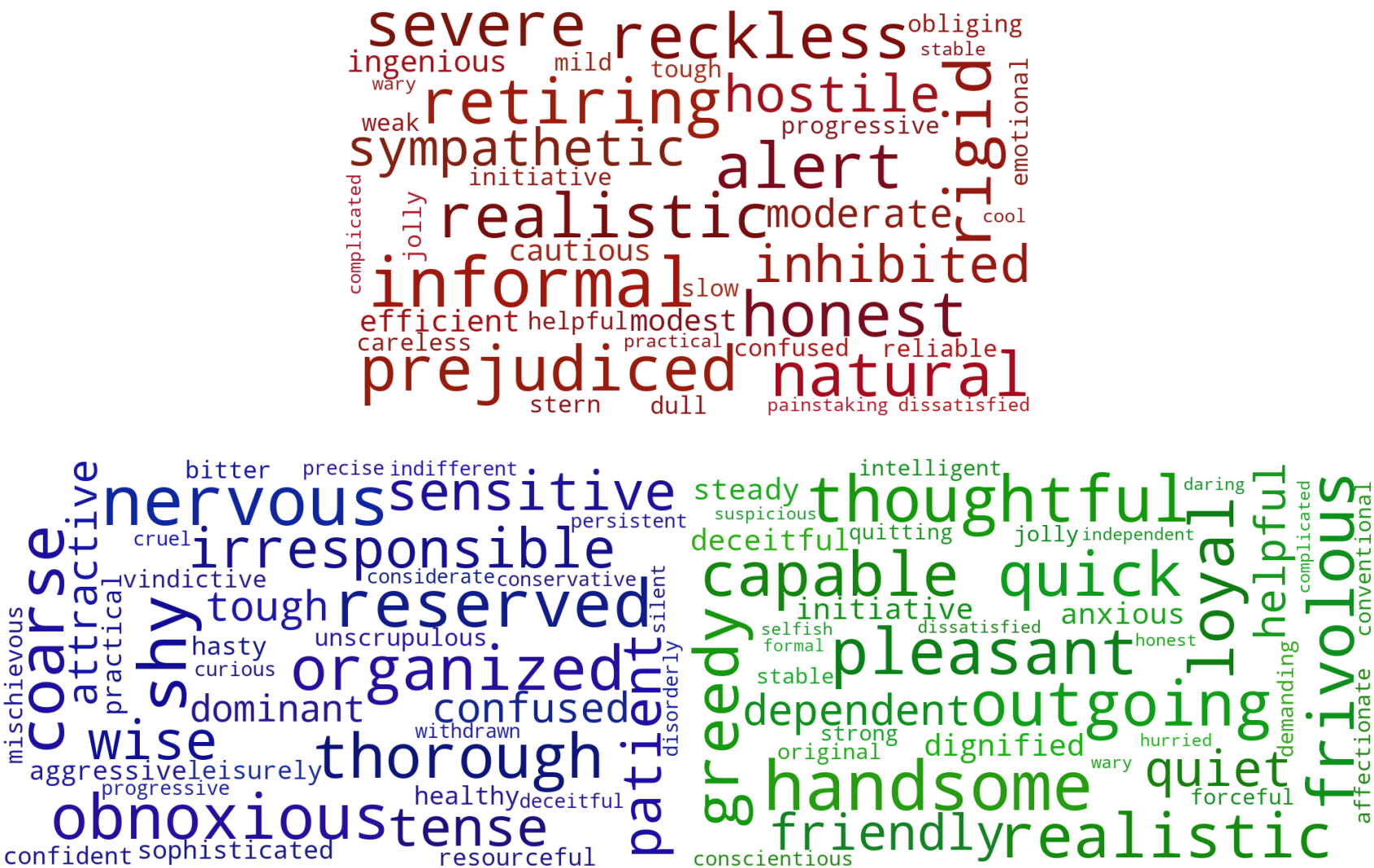
- Hispanic surnames more associated to "guilty", and stronger association for Democrat-appointed judges.

Caste Associations in Indian Courts



- t-SNE plot of the word vectors for each name, labeled by caste.

Caste Associations in Indian Courts



- Implicitly associated adjectives for Hindu, Scheduled, and Muslim litigants, respectively.