Text Data in Business and Economics

Basel University – Autumn 2023

7. Word Embeddings

What have we been doing?
Learning compressed representations of the data

- Dictionary methods: document is represented as a count over the lexicon
- N-grams: document is a count over a vocabulary of phrases
  - the vector of features, x_i, is a representation of the unprocessed document D_i.
- Topic models: document is a vector of shares over topics

What have we been doing?
Learning compressed representations of the data

- Dictionary methods: document is represented as a count over the lexicon
- N-grams: document is a count over a vocabulary of phrases
  - the vector of features, x_i, is a representation of the unprocessed document D_i.
- Topic models: document is a vector of shares over topics
- Text classifier: produces $\hat{y}_i = f(x_i; \hat{\theta})$, a vector of predicted probabilities across classes (say hand-labeled topics) for each document i.
  - this vector of class probabilities is a representation of the outcome-predictive text features x_i

What have we been doing?
Learning compressed representations of the data

- Dictionary methods: document is represented as a count over the lexicon
- N-grams: document is a count over a vocabulary of phrases
  - the vector of features, x_i, is a representation of the unprocessed document D_i.
- Topic models: document is a vector of shares over topics
- Text classifier: produces $\hat{y}_i = f(x_i; \hat{\theta})$, a vector of predicted probabilities across classes (say hand-labeled topics) for each document i.
  - this vector of class probabilities is a representation of the outcome-predictive text features x_i
- For either topic models or classifiers: the learned parameters $\hat{\theta}$ can be understood as a learned compressed representation of the whole corpus:
  - it contains information about the training corpus, the text features, and the outcomes.

Document-Term Matrices are Learned Representations

The **document-term matrix X**:

- A matrix entry $X_{[d,w]}$ quantifies the strength of association between a document d and a word w, generally its count or frequency

Document-Term Matrices are Learned Representations

The **document-term matrix X** :

- A matrix entry $X_{[d,w]}$ quantifies the strength of association between a document d and a word w, generally its count or frequency
- Each row $X_{[d,:]}$ represents a document and is a distribution over terms
- Each column $X_{[:,w]}$ represents a word and is a distribution over documents.
- Both document vectors and word vectors have a spatial interpretation, and can be compared / clustered.

Document-Topic and Topic-Word Matrices are Learned Representations

Topic models (e.g. LDA) transform the document-term matrix X into a document-topic matrix V and a topic-word matrix W.

- Each row of $V_{[d,:]}$ represent documents, give a distribution over topics
- Each column of $W_{[:,w]}$ represent words, also giving a distribution over topics.
- Again, both document vectors and word vectors have a spatial interpretation, and can be compared / clustered.

Document-Topic and Topic-Word Matrices are Learned Representations

Topic models (e.g. LDA) transform the document-term matrix X into a document-topic matrix V and a topic-word matrix W.

- Each row of $V_{[d,:]}$ represent documents, give a distribution over topics
- Each column of $W_{[:,w]}$ represent words, also giving a distribution over topics.
- Again, both document vectors and word vectors have a spatial interpretation, and can be compared / clustered.
- Further, the columns of V and rows of W represent latent topics:
  - $V_{[:,k]}$ summarize topics as distributions over documents
  - $W_{[k,:]}$ summarize topics as distributions over words

ML model coefficients $\hat{\theta}$ are Learned Representations

Say we train a multinomial logistic regression to predict hand-labeled topics with a bag-of-words representation of the documents:

- Let θ be the learned matrix of parameters relating words to topics:
  - It contains n_y rows, which are n_x-vectors representing topics.
  - It contains n_x columns, which are n_y-vectors representing each word in the vocabulary.

ML model coefficients $\hat{\theta}$ are Learned Representations

Say we train a multinomial logistic regression to predict hand-labeled topics with a bag-of-words representation of the documents:

- Let θ be the learned matrix of parameters relating words to topics:
  - It contains n_y rows, which are n_x-vectors representing topics.
  - It contains n_x columns, which are n_y-vectors representing each word in the vocabulary.
- How can we use θ? e.g.:
  - cluster the row vectors $\rightarrow$ which topics are similar / related.
  - cluster the column vectors $\rightarrow$ which words are similar / related.

Outline

- Word Embedding with Local Context
  - Properties of Word Embeddings
  - Using Word Embeddings
- Bias in Language

- An influential line of work in NLP, known as “word embedding”, reframes text analysis:
  - previously: focus on global document counts
  - now: represent the meaning of words by neighboring words – their local contexts.

- An influential line of work in NLP, known as "word embedding", reframes text analysis:
  - previously: focus on global document counts
  - now: represent the meaning of words by neighboring words – their **local contexts.**
- Move from high-dimensional sparse representations to low-dimensional dense representations

- An influential line of work in NLP, known as "word embedding", reframes text analysis:
  - previously: focus on global document counts
  - now: represent the meaning of words by neighboring words – their **local contexts.**
- Move from high-dimensional sparse representations to low-dimensional dense representations
- "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."

GloVe Embeddings (Pennington et al 2014)

- Define a co-occurrence matrix W, with W_{ij} = local co-occurrence counts between words i, j
  - that is, within some co-occurence window, typically 10 words.

GloVe Embeddings (Pennington et al 2014)

- Define a co-occurrence matrix W, with W_{ij} = local co-occurrence counts between words i, j
  - that is, within some co-occurence window, typically 10 words.
- Define word vectors $v = (v_1, ..., v_i, ..., v_{n_w})$, where $v_i \in (-1, 1)^{n_E}$,
  - initialized randomly
  - n_E typically ≈ 200

GloVe Embeddings (Pennington et al 2014)

- Define a co-occurrence matrix W, with W_{ij} = local co-occurrence counts between words i, j
  - that is, within some co-occurence window, typically 10 words.
- Define word vectors $v = (v_1, ..., v_i, ..., v_{n_W})$, where $v_i \in (-1, 1)^{n_E}$,
  - initialized randomly
  - n_E typically ≈ 200
- then use gradient descent to solve

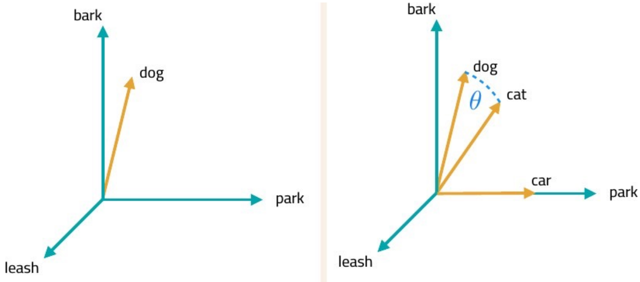
$$\min_{v} \sum_{ij} f(W_{ij})\left(v_i^T v_j - \log(W_{ij})\right)$$

  - $f(\cdot)$ is a non-negative, increasing, concave weighting function
- Minimizes **squared difference** between:
  - **dot product of word vectors,** $v_i^T v_j$
  - **empirical co-occurrence,** $\log(W_{ij})$
- Intuitively: words that co-occur should have high correlation (dot product)

Outline

Word Similarity

- Once words are represented as vectors $\{v1, v2, ...\}$, 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":



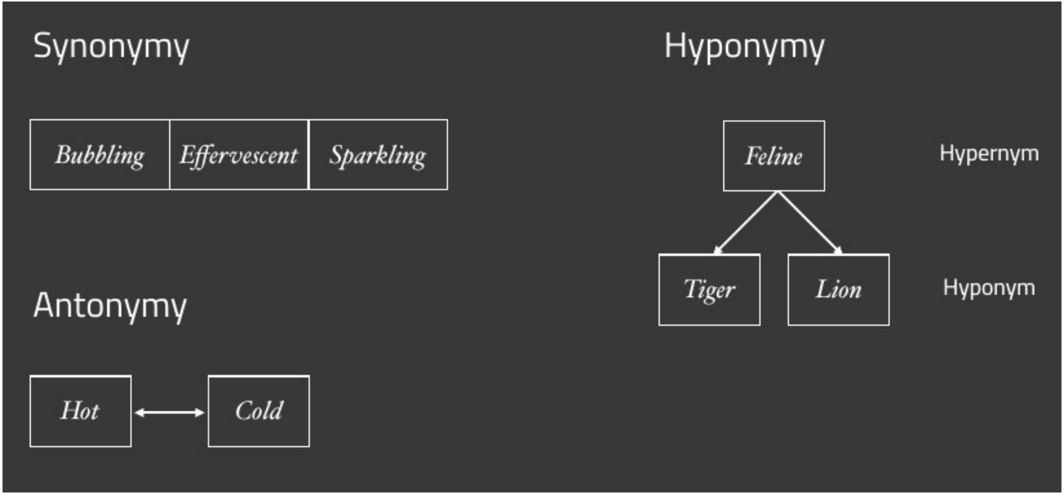
- The standard metric for comparing vectors is cosine similarity:

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

- Thanks to linearity, can compute similarities between groups of words by averaging the groups.

Word Embeddings Encode Linguistic Relations

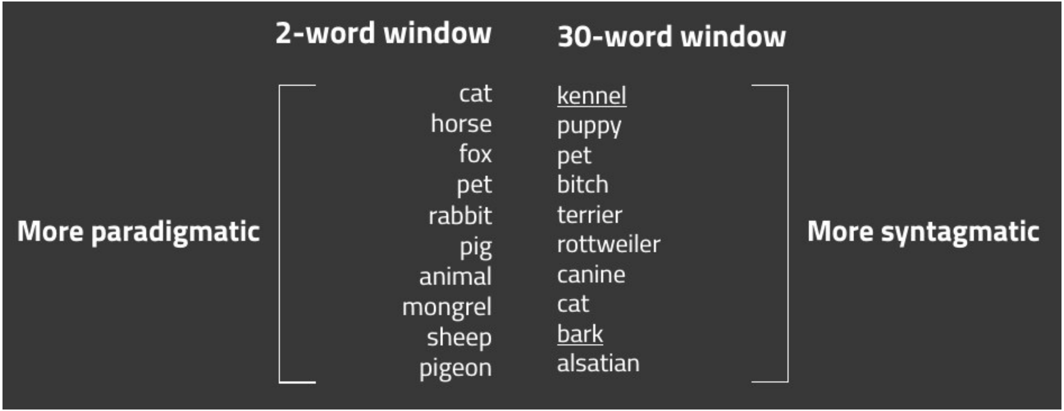
Word Embeddings Encode Linguistic Relations



What do word embeddings capture (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.

Parts of Speech and Phrases

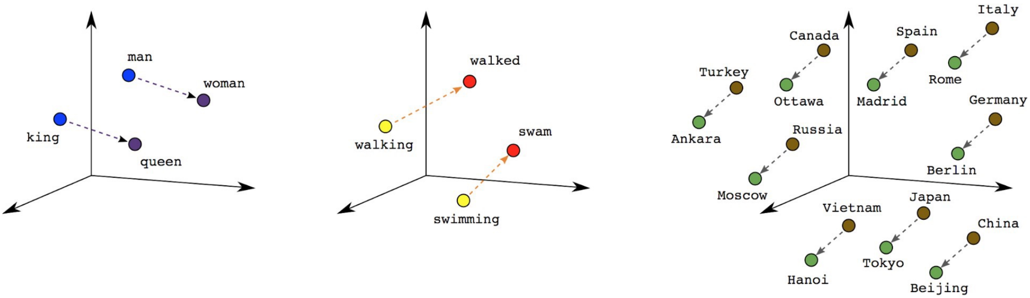
- In the default model multiple senses of a word are merged.
  - e.g. "I like a bird" (verb) and "I am like a bird" (preposition).
- Can improve the quality of embeddings in these cases by attaching the POS to the word (e.g. "like:verb", "like:prep") before training.

Parts of Speech and Phrases

- In the default model multiple senses of a word are merged.
  - e.g. "I like a bird" (verb) and "I am like a bird" (preposition).
- Can improve the quality of embeddings in these cases by attaching the POS to the word (e.g. "like:verb", "like:prep") before training.
- The default model only works by word, but "new york $\neq$ "new" + "york"
  - can tokenize phrases together (see Week 2 lecture) before training.

Vector Directions ↔ Meaning

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



Male-Female

Verb Tense

Country-Capital

Word Embeddings for Analogies

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

- 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).

Outline

Tokenizing for Word Embeddings

- Have to think about what pre-processing will do:
  - capitalization
  - punctuation
  - stopwords/function-words
  - 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

“Enriching word vectors with subword information” (Bojanowski et al 2017)

- These are known as “Fasttext” embeddings
- Each word is augmented by a bag of (hashed) character n-grams. (e.g., spicy = (spicy, spi, pic, icy)).
- Learn embeddings for the whole word as well as character segments, and construct word embedding by summing over the components
- Competitive with word2vec in standard tasks; better in some languages.
- Produces good embeddings for unseen words.

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.
  - And I don't see a solution to it.
- Relatedly, antonyms are often rated similarly, have to be careful with that.

Word Vectors can produce Document Vectors

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

- The "continuous bag of words" representation for document D is the sum, or the average (potentially weighted by a_w), of the vectors $\underline{v}_w$ for each word w in the document.
- Word vectors v constructed using GloVe or Word2Vec (pre-trained or trained on the corpus)
- "Document" could be sentence, paragraph, section, article, etc.
- Arora, Liang, and Ma (2017) 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.

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.
- 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.

Standard word embeddings (e.g. word2vec/glove) have a number of limitations:

- **polysemy:** you get one vector for multiple senses of a word (e.g. “**glass** of water” vs “window **glass**”)

Standard word embeddings (e.g. word2vec/glove) have a number of limitations:

- **polysemy:** you get one vector for multiple senses of a word (e.g. "**glass** of water" vs "window **glass**")
- **rare words:** a word that shows up just once or twice won't be well-defined
- **n-grams:** does not produce embeddings for multi-word phrases

Scientists attending ACL work on **cutting edge** research in NLP

Petrichor: the earthy scent produce when rain falls on dry soil

Roger Federer won the first **set**^{NN} of the match

Standard word embeddings (e.g. word2vec/glove) have a number of limitations:

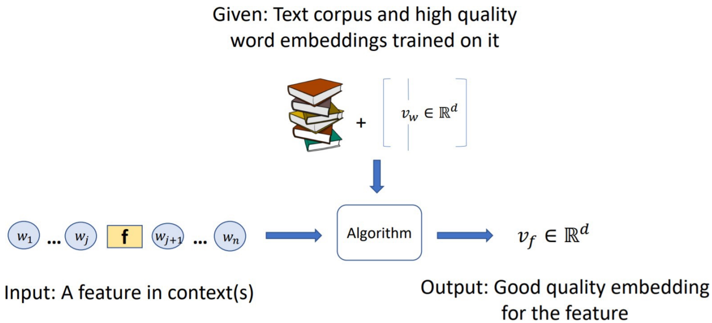
- **polysemy:** you get one vector for multiple senses of a word (e.g. “**glass** of water” vs “window **glass**”)
- **rare words:** a word that shows up just once or twice won’t be well-defined
- **n-grams:** does not produce embeddings for multi-word phrases

Scientists attending ACL work on **cutting edge** research in NLP

Petrichor: the earthy scent produce when rain falls on dry soil

Roger Federer won the first **set**^{NN} of the match

- Goal of Khodak et al (2018): produce embeddings “a la carte” given a context:



A la carte embeddings

- Given a target word f and its context c, define

$$v_j^{avg} = \frac{1}{|c|} \sum_{w \in c} v_w$$

the average vector for the words in the context.

- Arora et al (2018) prove that for vectors produced by a generative language model, there exists a matrix A such that

$$v_f \approx A v_f^{avg}$$

- The "induction matrix" A can be learned with a least-squares (linear regression) objective

$$A^* = \underset{A}{\operatorname{argmin}} \sum_{w} |v_w - A v_w^{avg}|_2^2$$

where w indexes over all the tokens in the corpus.

- empirically:

$$\text{cosine}(v_f, A^* v_f^{avg}) \geq 0.9$$

Outline

Implicit attitudes

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

Implicit attitudes

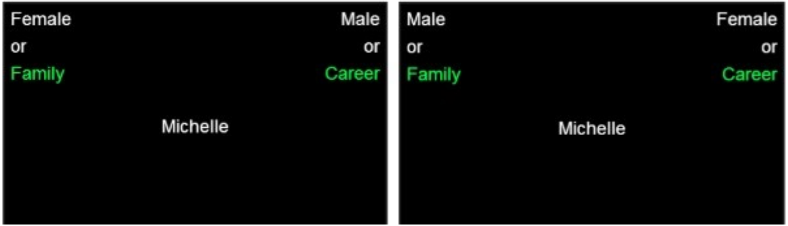
"Attitudes that affect our understanding, actions, and decisions in an unconscious manner" (Kirnan institute, OSU)

- Generally measured using Implicit Association Tests (IATs)

Implicit attitudes

"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)

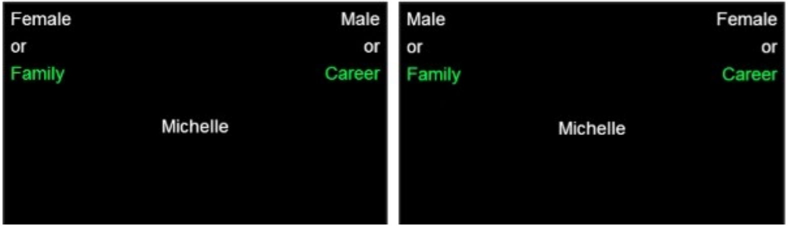


- Comparing reaction times across trials with different word pairs:
  - subjects tend to be slower and more error-prone in assignments against stereotype (e.g. "Michelle" goes to "Female or Career").

Implicit attitudes

"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)



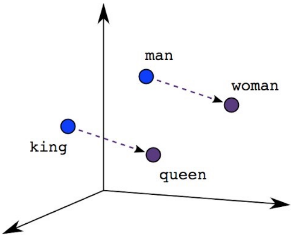
- Comparing reaction times across trials with different word pairs:
  - subjects tend to be slower and more error-prone in assignments against stereotype (e.g. "Michelle" goes to "Female or Career").
  - IAT score = difference in reaction time between stereotype-consistent and stereotype-inconsistent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

Caliskan, Bryson, and Narayanan (*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. . . “

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. . . "



Analogies

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

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.

Result
s

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

Result
s

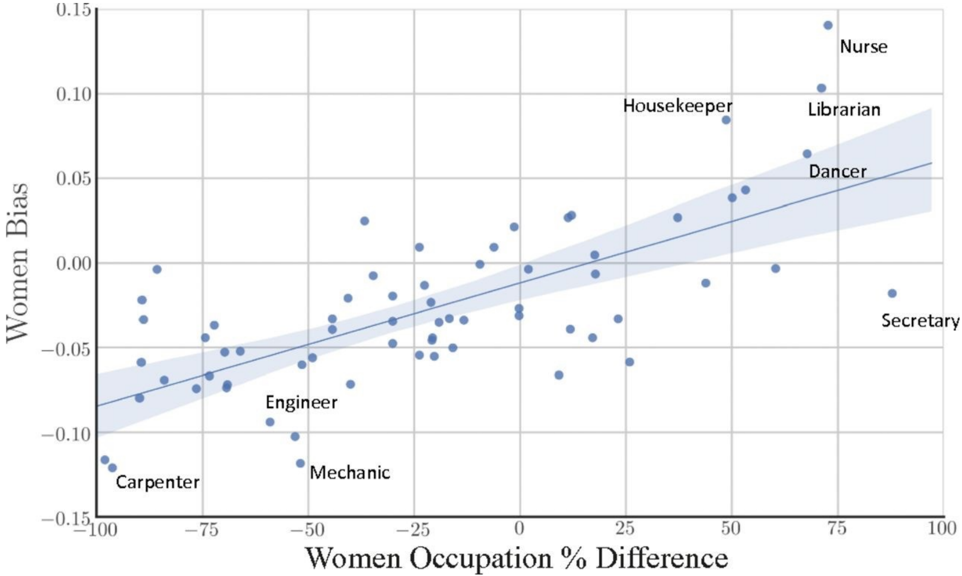
- Pleasant vs. Unpleasant?
  - Flowers vs. Insects
  - Musical instruments vs. weapons.
  - European-American names vs. African-American names
- Male names vs. Female names:

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

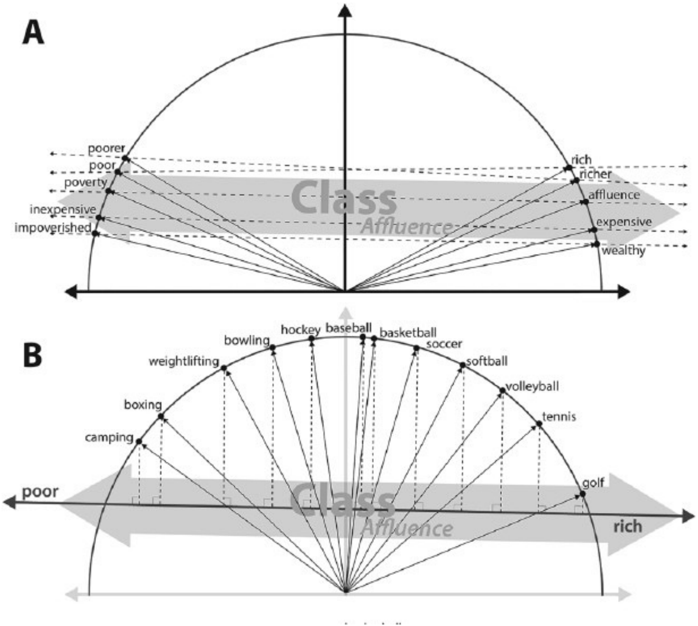
What do we learn from this?

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)



Time Series Analysis of Affluence

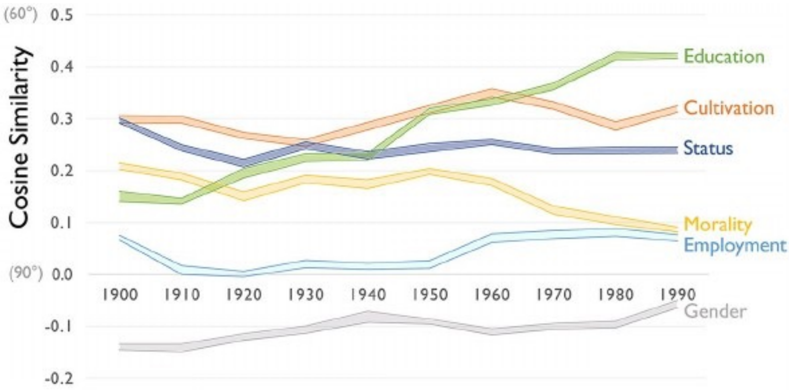


Figure 5. Cosine Similarity between the Affluence Dimension and Six Other Cultural Dimensions of Class by Decade; 1900 to 1999 Google Ngrams Corpus
Note: Bands represent 90 percent bootstrapped confidence intervals produced by subsampling.

"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," …"

Discussion

Discussion

- What are we measuring?
  - e.g., how do measurements by person or by group correlate with actual attitudes, for example as measured by IAT scores?

Discussion

- What are we measuring?
  - e.g., how do measurements by person or by group correlate with actual attitudes, for example as measured by IAT scores?
- What attitudes can be reliably measured?
  - e.g. racial sentiment, especially in light of black sheep problem.

Discussion

- What are we measuring?
  - e.g., how do measurements by person or by group correlate with actual attitudes, for example as measured by IAT scores?
- What attitudes can be reliably measured?
  - e.g. racial sentiment, especially in light of black sheep problem.
- In what domains is this relevant?
  - social media, news media, politics, legal, scientific, ...

Discussion

- What are we measuring?
  - e.g., how do measurements by person or by group correlate with actual attitudes, for example as measured by IAT scores?
- What attitudes can be reliably measured?
  - e.g. racial sentiment, especially in light of black sheep problem.
- In what domains is this relevant?
  - social media, news media, politics, legal, scientific, ...
- Does language matter?
  - Djourelova (2020): style change from "illegal" to "undocumented" immigrant softened attitudes toward immigration.