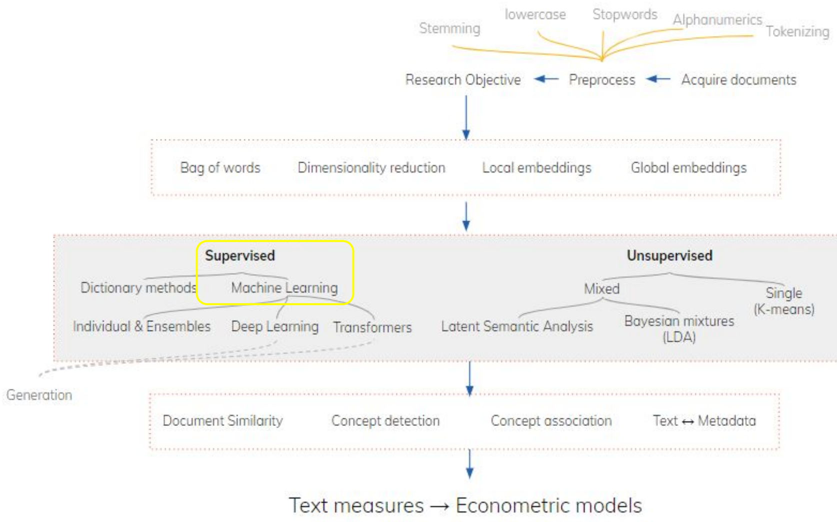
Innovation Growth Lab by nesta

Natural Language Processing in Economics

Supervised Learning & Embeddings

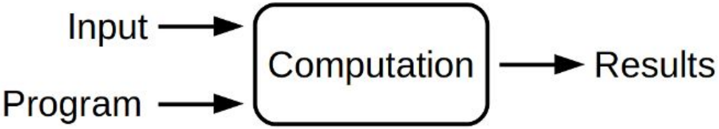


Supervised Machine Learning

What is Machine learning?

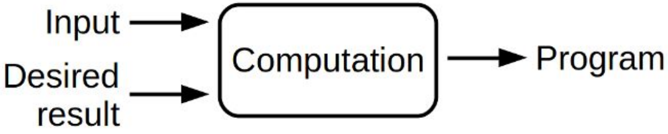
Classical programming

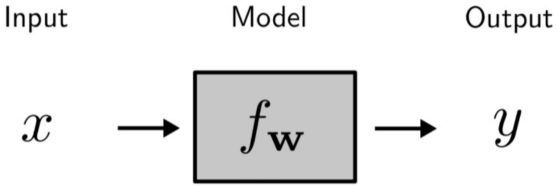
- Humans input the rules and the data, the computer provides answers



Machine learning

- Humans input the data and answers, the computer learns the rules





What is Machine learning?

Minimise a cost function

- A typical cost function is the Mean Squared Error

$$\mathrm{MSE}(\theta) = \frac{1}{n_D}\sum_{i=1}^{n_D}(h(x_i;\theta) - y_i)^2$$

Data

- The data (x, y) is taken as given, and the algorithm searches for parameters θ to minimise the cost function

Example

- OLS assumes the functional form f(x, θ) = $X_i\Theta$ and minimises the MSE

$$\min_{\hat{\theta}} \frac{1}{n_D}\sum_{i=1}^{n_D}(x_i'\hat{\theta} - y_i)^2$$

OLS example

Loss function

- OLS assumes the functional form f(x, θ) = $X_i\Theta$ and minimises the MSE

$$\mathrm{MSE}(\theta) = \frac{1}{n_D}\sum_{i=1}^{n_D}\left(h(\theta;\boldsymbol{x}_i) - y_i\right)^2$$

convex

Partial derivatives

- Estimates how changes in a coefficient would reduce loss across data

$$\frac{\partial \mathrm{MSE}}{\partial \theta_j} = \frac{2}{n_D}\sum_{i=1}^{n_D}(\underbrace{h(\theta;\boldsymbol{x}_i) - y_i}_{\text{error for this obs}})\underbrace{\frac{\partial h(\theta;\boldsymbol{x}_i)}{\partial \theta_j}}_{\text{how } \theta_j \text{ shifts } h(\cdot)}$$

Gradient

- The vector of partial derivatives

$$\nabla_\theta \mathrm{MSE} = \begin{bmatrix} \frac{\partial \mathrm{MSE}}{\partial \theta_1} \\ \frac{\partial \mathrm{MSE}}{\partial \theta_2} \\ \vdots \\ \frac{\partial \mathrm{MSE}}{\partial \theta_{n_x}} \end{bmatrix}$$

Descent

- Nudges Θ against the gradient

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathrm{MSE}$$

A non-convex example

Logistic Regression

- The coefficient of a simple logistic regression seeks to maximise likelihood

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_{model}(y_i|\mathbf{x}_i, \mathbf{w})$$

Bernoulli distributed

- The probability distribution is assumed Bernoulli (in the binary case)

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{N} \log \left[\hat{y}_i^{y_i} \left(1 - \hat{y}_i\right)^{(1-y_i)} \right]$$

Cross Entropy Loss

- We minimise the dissimilarity between the empirical data and model distr.

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} \underbrace{-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)}_{\text{Binary Cross Entropy Loss } \mathcal{L}(\hat{y}_i, y_i)}$$

A non-convex example

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{N} \underbrace{-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)}_{\text{Binary Cross Entropy Loss } \mathcal{L}(\hat{y}_i, y_i)}$$

with $\hat{y} = f_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$ and $\sigma(x) = \frac{1}{1 + e^{-x}}$

Minimiser (derivation)

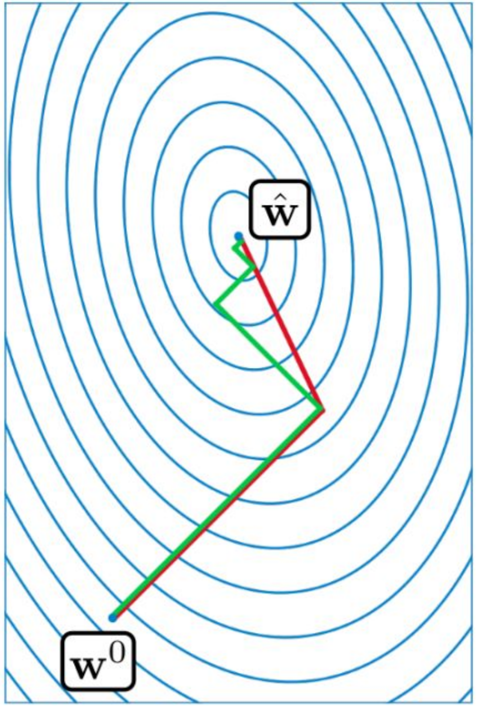
- In contrast to the previous example, the loss in not quadratic in **w**
- We must apply iterative gradient-based optimisation. The gradient is:

$$\nabla_{\mathbf{w}} \mathcal{L}(\hat{y}_i, y_i) = (\hat{y}_i - y_i)\mathbf{x}_i$$

Iteration until convergence

- Given some tolerance ε and step size η, repeat until $\mathbf{v} < \varepsilon$

$$\mathbf{v} = \nabla_{\mathbf{w}} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i=1}^{N} \nabla_{\mathbf{w}} \mathcal{L}(\hat{y}_i, y_i)$$
$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \mathbf{v}$$



Machine learning tips

- **Feature engineering**: Data preprocessing steps are critical for model performance. This includes handling missing values, outlier detection and treatment, and feature scaling.

- **Properly evaluate model performance**: use appropriate evaluation metrics that align with your specific problem, considering factors like class imbalance and cost associations. Use performance metrics: accuracy, precision, recall, F1 scores, AUC curves, etc.

- **Address overfitting and underfitting**: Regularisation techniques, such as L1 or L2 regularisation, can help prevent overfitting by adding penalty terms to the model's objective function. Consider increasing model complexity if underfitting occurs.

- **Cross-validation and hyperparameter tuning**: Use cross-validation techniques (or Bayesian optimisation!) to assess the out-of-sample performance of your model. It involves partitioning the data into training and validation sets, enabling robust evaluation.

- **Understand the problem domain and interpretability**: Gain a deep understanding of the problem domain and the idiosyncrasies of the task at hand.

Machine learning with text data

- We have a corpus **D** of **N** documents i ∈ {1, ..., N}
- Each document **i** has an associated outcome or label **y** with dimensions **M.**
- Some documents are labeled and some are unlabeled
- We seek to learn a mapping **f(i)** based on the labelled data to predict **y** in the unlabeled data

Text features as numeric features

- The methods described in previous tutorials can be used to extract informative numerical information about **i**
  - style features
  - counts over dictionary patterns
  - tokens
  - n-grams
  - principal components
  - topic shares
  - concept associations

Machine learning with text data: models

Generally Regression

- A one-dimensional, continuous, real-valued outcome (sentiment scores, numerical ratings, economic growth)
- **Linear Regression:** Linear models can be applied in text tasks where the objective is continuous.
- **Penalised Regression:** Sparsity-inducing models that reduce high-dimensionality concerns in text data
- **Quantile regressions:** Estimates a quantile of the output conditional on the inputs, rather than the mean.
- **Decision Tree Regression:** Hierarchical algorithms that model a binned relationship with a continuous output
- **Support Vector Regression** (SVR): an extension of the popular vector machines (SVM) for regression tasks

Generally Classification

- **Logistic Regression**: Models the probability of a text belonging to a class given sigmoid transformations.
- **Support Vector Machines:** Search the optimal hyperplane that separates the text data into classes.
- **Decision trees:** A versatile algorithm that uses a hierarchical structure of branching decisions to classify data
- **Random forests:** Build ensembles of decision trees to make more accurate predictions.
- **AdaBoost:** Adaptive boosting algorithm that builds ensembles of weak learners and prioritises poor predicts
- **XGBoost:** State-of-the-art tabular algorithm that utilises ensembles of trees and gradient descent optimis.
- **Neural networks:** Model highly non-linear hierarchical representations of text data

Machine learning with text data: models 🔗

Model	Specificity	Interpretability	Validability
Linear Regression	Moderate	High	High
Penalized Regression	High	Moderate	High
Quantile Regressions	Moderate	High	High
Decision Tree Regression	Moderate	High	Moderate
Support Vector Regression	High	Moderate	High
Logistic Regression	Moderate	High	High
Support Vector Machines	High	Moderate	High
Decision Trees	Moderate	High	Moderate
Random Forests	High	Moderate	High
AdaBoost	High	Low	Moderate
XGBoost	High	Low	High
Neural Networks	High	Low	High

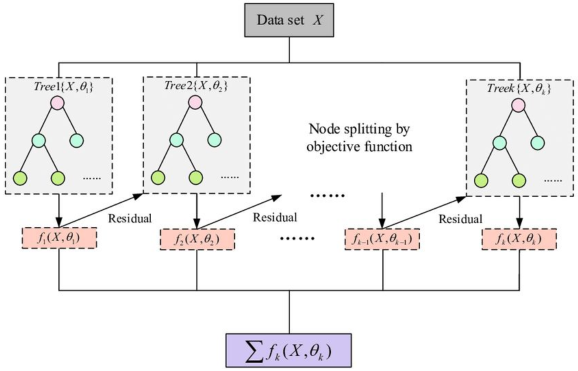
… or just use XGBoost

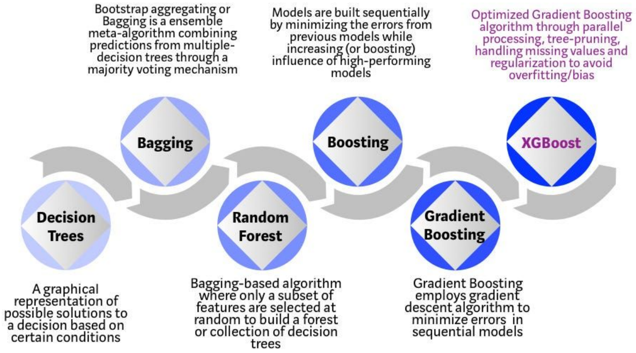
Boosting with Decision Trees

- Use an ensemble of decision trees as weak or base learners
- The boosting process starts with a single decision tree and iteratively adds trees to the ensemble
- Each tree is trained to correct the mistakes made by the previous trees, focusing on high-error samples

Gradient Descent Optimisation

- XGBoost utilises gradient descent optimisation to improve the ensemble's performance
- During each boosting iteration, gradients are computed based on the errors of the ensemble predict.
- The subsequent trees are built to minimise these gradients, reducing overall training loss





A graphical representation of possible solutions to a decision based on certain conditions

Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism

Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees

Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models

Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias

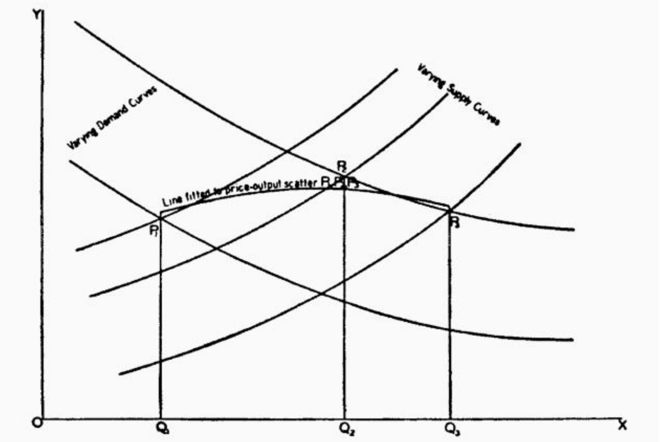
Paper application

Stock, Trebbi (2003) - Who invented IVs?

Context

- First derivation of an IV estimator in Appendix B of *The Tariff of Animal and Vegetable Oils* by Philip G. Wright
    - **First 285 pages**: “a painfully detailed treatise on animal and vegetable oils, their production, uses, markets and tariffs”
    - **Appendix B**: “out of the blue […] a succinct and insightful of why price and quantity data alone are in general inadequate, two separate and correct derivations of IV, and an empirical application to butter and flaxseed.”
- Because Appendix B is so different many people (see Manski 1988) have suggested it might have been written by Philip’s son Sewall Wright, a famous genetic statistician

FIGURE 4. PRICE-OUTPUT DATA FAIL TO REVEAL EITHER SUPPLY OR DEMAND CURVE.



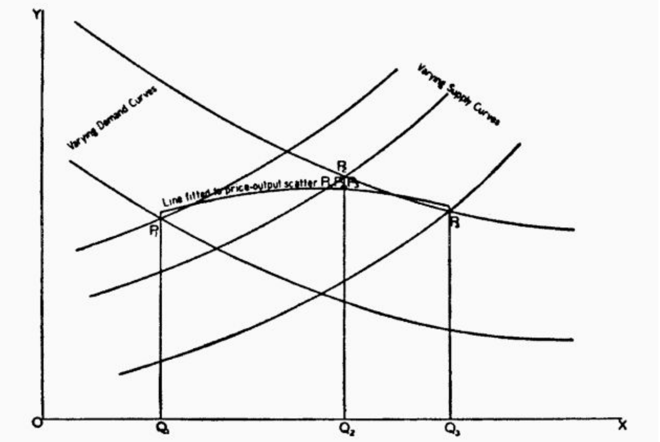
Paper application □

Stock, Trebbi (2003) - Who invented IVs?

A case study

- The case for **Sewall**
  - Appendix uses method of “path coefficients”, which Sewall had recently invented
  - A more eminent statistician
- The case for **Philip**
  - He was an economist while Sewall was not
  - Had written frequently about the identification problem

FIGURE 4. PRICE-OUTPUT DATA FAIL TO REVEAL EITHER SUPPLY OR DEMAND CURVE.



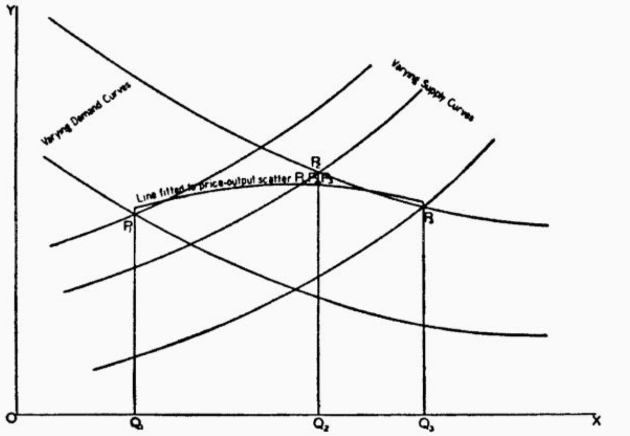
Paper application □

Stock, Trebbi (2003) - Who invented IVs?

Data

- The raw data consists of a sample of texts with sole authorship known to be Philip or Sewall, plus chapter 1 and the famous Appendix B.
- Blocks of 1,000 words are defined as documents, a total of 52 are selected. These include:
  - 20 undisputedly by Sewall
  - 25 undisputedly by Philip
- The prediction set correspond to a remaining
  - 6 blocks from Appendix B
  - 1 block from Chapter 1

FIGURE 4. PRICE-OUTPUT DATA FAIL TO REVEAL EITHER SUPPLY OR DEMAND CURVE.



Paper application □

Stock, Trebbi (2003) - Who invented IVs?

From text to numbers

- Stylometric “function words” from Mosteller & Wallace (1963)
- Grammatical constructions from Mannion & Dixon (1997)
  - 70 function words
  - 18 grammatical constructions
- The result is n = 52, p = 88, and **V** known for 45 blocks

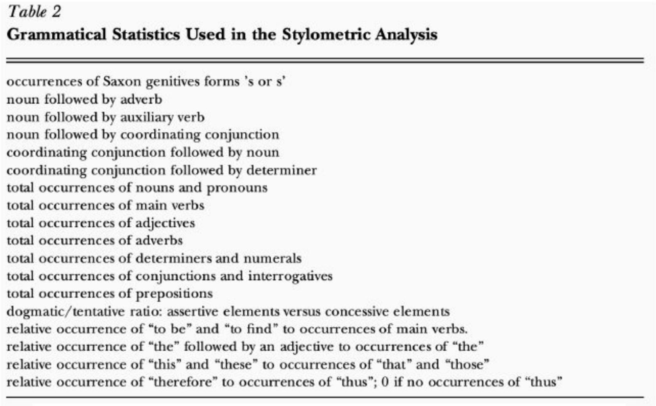


Table 2

Grammatical Statistics Used in the Stylometric Analysis

occurrences of Saxon genitives forms ’s or s’
noun followed by adverb
noun followed by auxiliary verb
noun followed by coordinating conjunction
coordinating conjunction followed by noun
coordinating conjunction followed by determiner
total occurrences of nouns and pronouns
total occurrences of main verbs
total occurrences of adjectives
total occurrences of adverbs
total occurrences of determiners and numerals
total occurrences of conjunctions and interrogatives
total occurrences of prepositions
dogmatic/tentative ratio: assertive elements versus concessive elements
relative occurrence of “to be” and “to find” to occurrences of main verbs.
relative occurrence of “the” followed by an adjective to occurrences of “the”
relative occurrence of “this” and “these” to occurrences of “that” and “those”
relative occurrence of “therefore” to occurrences of “thus”; 0 if no occurrences of “thus”

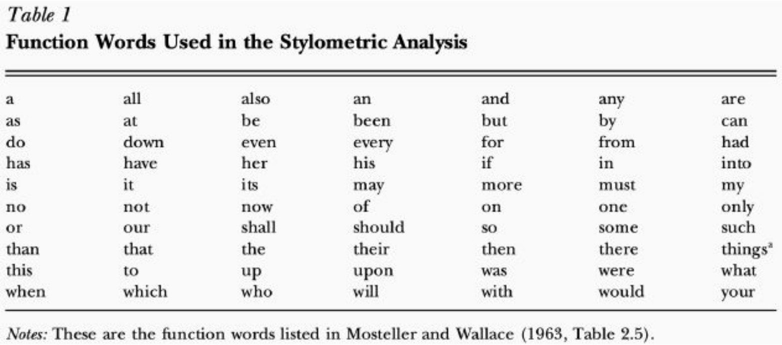
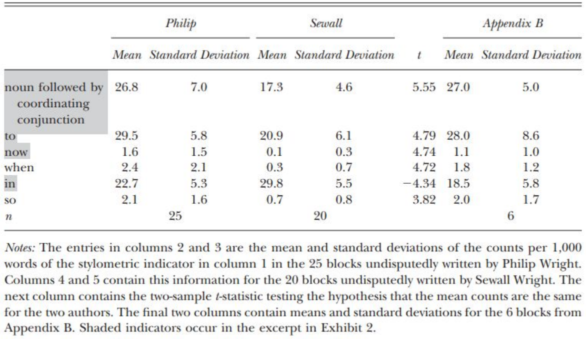


Table 1

Function Words Used in the Stylometric Analysis

a	all	also	an	and	any	are
as	at	be	been	but	by	can
do	down	even	every	for	from	had
has	have	her	his	if	in	into
is	it	its	may	more	must	my
no	not	now	of	on	one	only
or	our	shall	should	so	some	such
than	that	the	their	then	there	things[a]
this	to	up	upon	was	were	what
when	which	who	will	with	would	your

Notes: These are the function words listed in Mosteller and Wallace (1963, Table 2.5).



	Philip		*Sewall*			*Appendix B*	
	Mean	*Standard Deviation*	*Mean*	*Standard Deviation*	*t*	*Mean*	*Standard Deviation*
noun followed by coordinating conjunction	26.8	7.0	17.3	4.6	5.55	27.0	5.0
to	29.5	5.8	20.9	6.1	4.79	28.0	8.6
now	1.6	1.5	0.1	0.3	4.74	1.1	1.0
when	2.4	2.1	0.3	0.7	4.72	1.8	1.2
in	22.7	5.3	29.8	5.5	−4.34	18.5	5.8
so	2.1	1.6	0.7	0.8	3.82	2.0	1.7
n	25		20			6	

Notes: The entries in columns 2 and 3 are the mean and standard deviations of the counts per 1,000 words of the stylometric indicator in column 1 in the 25 blocks undisputedly written by Philip Wright. Columns 4 and 5 contain this information for the 20 blocks undisputedly written by Sewall Wright. The next column contains the two-sample *t*-statistic testing the hypothesis that the mean counts are the same for the two authors. The final two columns contain means and standard deviations for the 6 blocks from Appendix B. Shaded indicators occur in the excerpt in Exhibit 2.

Paper application □

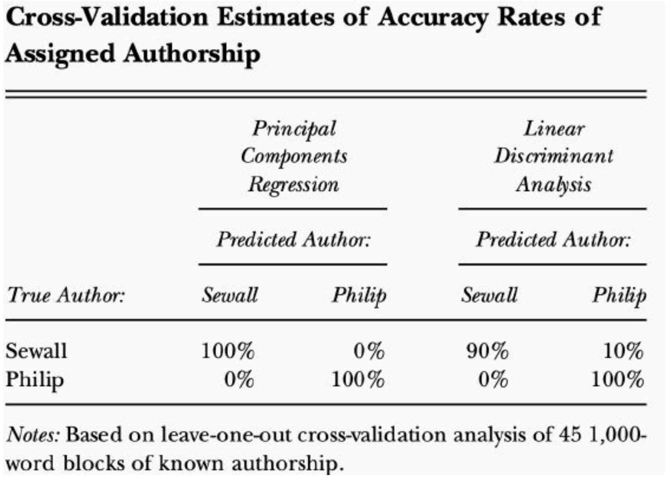
Stock, Trebbi (2003) - Who invented IVs?

Empirical methods

- Principal Components Regression
  - Compute 4 PC for each set of covariates
  - Regress ownership **V** on each separately
- Linear discriminant analysis

$$\hat{V} = \sum_p w_p c_p$$

$$w_p = \frac{\bar{c}_{p:P} - \bar{c}_{p:S}}{s^2_{p:P} + s^2_{p:S}}$$



Cross-Validation Estimates of Accuracy Rates of Assigned Authorship

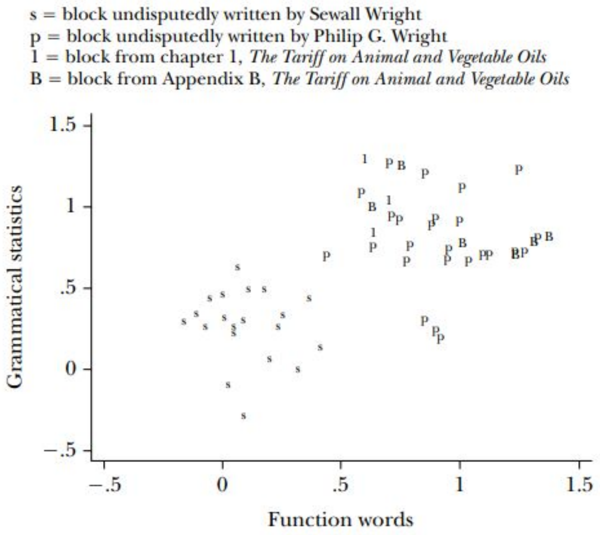
	Principal Components Regression		*Linear Discriminant Analysis*	
	Predicted Author:		*Predicted Author:*	
True Author:	*Sewall*	*Philip*	*Sewall*	*Philip*
Sewall	100%	0%	90%	10%
Philip	0%	100%	0%	100%

Notes: Based on leave-one-out cross-validation analysis of 45 1,000-word blocks of known authorship.

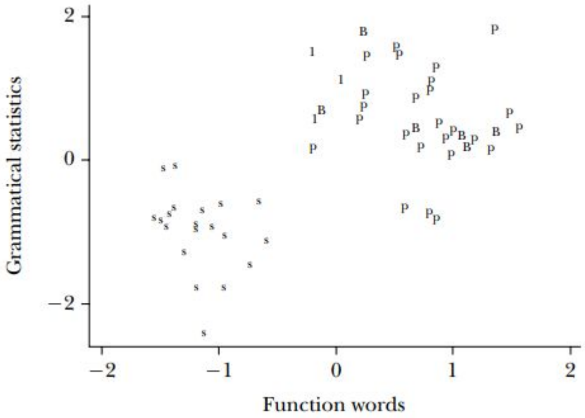
Paper application

Stock, Trebbi (2003) - Who invented IVs?

Scatterplot of Predicted Values from Regression on First Four Principal Components: Grammatical Statistics versus Function Words



Scatterplot of Linear Discriminant Based on Grammatical Statistics versus Linear Discriminant Based on Function Words

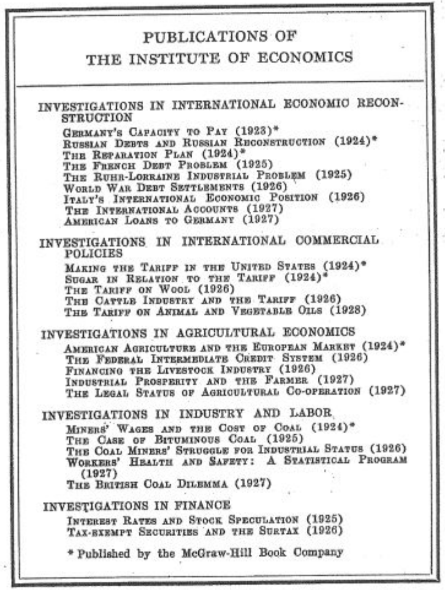


Paper application □

Stock, Trebbi (2003) - Who invented IVs?

Results

- Philip is undoubtedly the author
- This does not mean it was his idea



PUBLICATIONS OF
THE INSTITUTE OF ECONOMICS

INVESTIGATIONS IN INTERNATIONAL ECONOMIC RECONSTRUCTION
Germany's Capacity to Pay (1923)*
Russian Debts and Russian Reconstruction (1924)*
The Reparation Plan (1924)*
The French Debt Problem (1925)
The Ruhr-Lorraine Industrial Problem (1925)
World War Debt Settlements (1926)
Italy's International Economic Position (1926)
The International Accounts (1927)
American Loans to Germany (1927)

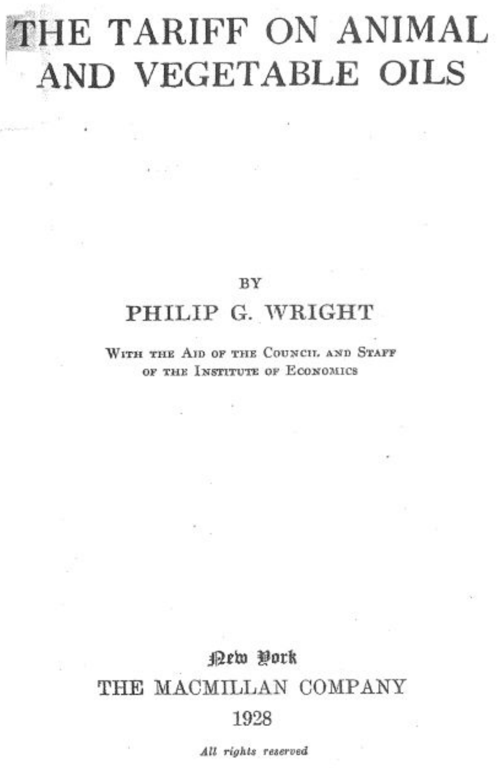
INVESTIGATIONS IN INTERNATIONAL COMMERCIAL POLICIES
Making the Tariff in the United States (1924)*
Sugar in Relation to the Tariff (1924)*
The Tariff on Wool (1926)
The Cattle Industry and the Tariff (1926)
The Tariff on Animal and Vegetable Oils (1928)

INVESTIGATIONS IN AGRICULTURAL ECONOMICS
American Agriculture and the European Market (1924)*
The Federal Intermediate Credit System (1926)
Financing the Livestock Industry (1926)
Industrial Prosperity and the Farmer (1927)
The Legal Status of Agricultural Co-operation (1927)

INVESTIGATIONS IN INDUSTRY AND LABOR
Miners' Wages and the Cost of Coal (1924)*
The Case of Bituminous Coal (1925)
The Coal Miners' Struggle for Industrial Status (1926)
Workers' Health and Safety: A Statistical Program (1927)
The British Coal Dilemma (1927)

INVESTIGATIONS IN FINANCE
Interest Rates and Stock Speculation (1925)
Tax-exempt Securities and the Surtax (1926)

* Published by the McGraw-Hill Book Company



THE TARIFF ON ANIMAL
AND VEGETABLE OILS

BY
PHILIP G. WRIGHT

With the Aid of the Council and Staff
of the Institute of Economics

New York
THE MACMILLAN COMPANY
1928

All rights reserved

Paper application □

Antweiler, Frank (2004) - The Information Content of Internet Boards

Goal

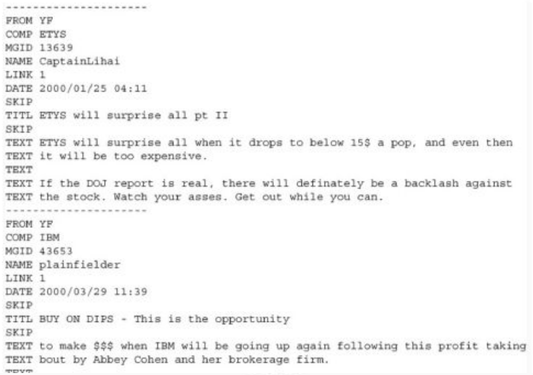
Identify high-frequency correlations (IRF) between stock investor actions and their Yahoo! Posting behavior

Methodology

- **Count words:** document-term frequency
- **Training sample**: manually label 1,000 messages indicating whether writer encourages**:** buy, sell, hold
- **Classification module**: Use Naive Bayes classification to make a prediction on out-of-sample messages

Results

- Small amount of predictability in returns
- Messages do predict volatility
- Disagreement predicts volume



```
---------------------
FROM YF
COMP ETYS
MGID 13639
NAME CaptainLihai
LINK 1
DATE 2000/01/25 04:11
SKIP
TITL ETYS will surprise all pt II
SKIP
TEXT ETYS will surprise all when it drops to below 15$ a pop, and even then
TEXT it will be too expensive.
TEXT
TEXT If the DOJ report is real, there will definately be a backlash against
TEXT the stock. Watch your asses. Get out while you can.
---------------------
FROM YF
COMP IBM
MGID 43653
NAME plainfielder
LINK 1
DATE 2000/03/29 11:39
SKIP
TITL BUY ON DIPS - This is the opportunity
SKIP
TEXT to make $$$ when IBM will be going up again following this profit taking
TEXT bout by Abbey Cohen and her brokerage firm.
```

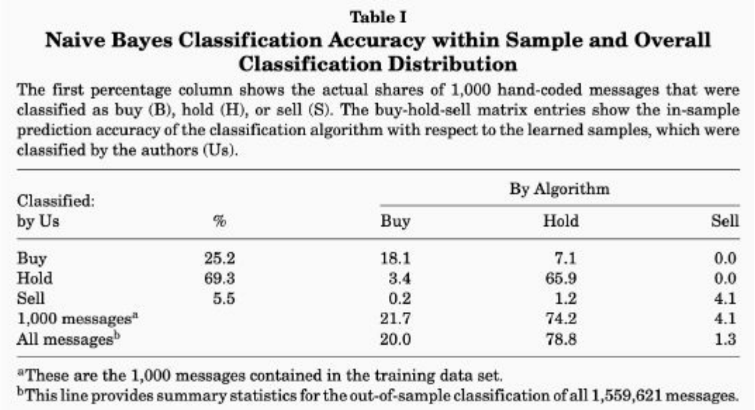


Table I

Naive Bayes Classification Accuracy within Sample and Overall Classification Distribution

The first percentage column shows the actual shares of 1,000 hand-coded messages that were classified as buy (B), hold (H), or sell (S). The buy-hold-sell matrix entries show the in-sample prediction accuracy of the classification algorithm with respect to the learned samples, which were classified by the authors (Us).

Classified:		By Algorithm		
by Us	%	Buy	Hold	Sell
Buy	25.2	18.1	7.1	0.0
Hold	69.3	3.4	65.9	0.0
Sell	5.5	0.2	1.2	4.1
1,000 messages[a]		21.7	74.2	4.1
All messages[b]		20.0	78.8	1.3

[a] These are the 1,000 messages contained in the training data set.

[b] This line provides summary statistics for the out-of-sample classification of all 1,559,621 messages.

Paper application □

Peterson, Spirling (2018) - Measuring Polarisation in Westminster

Goal

Use model accuracy as a predictor to determine the degree of polarisation in political institutions

Results

In years that the classifier is more accurate, speech is more polarised

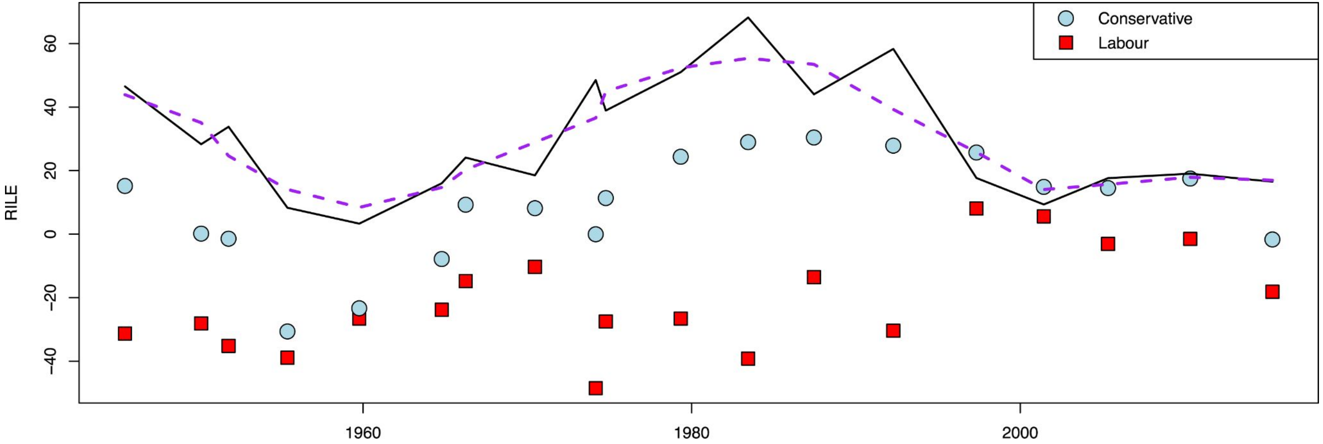
Methodology

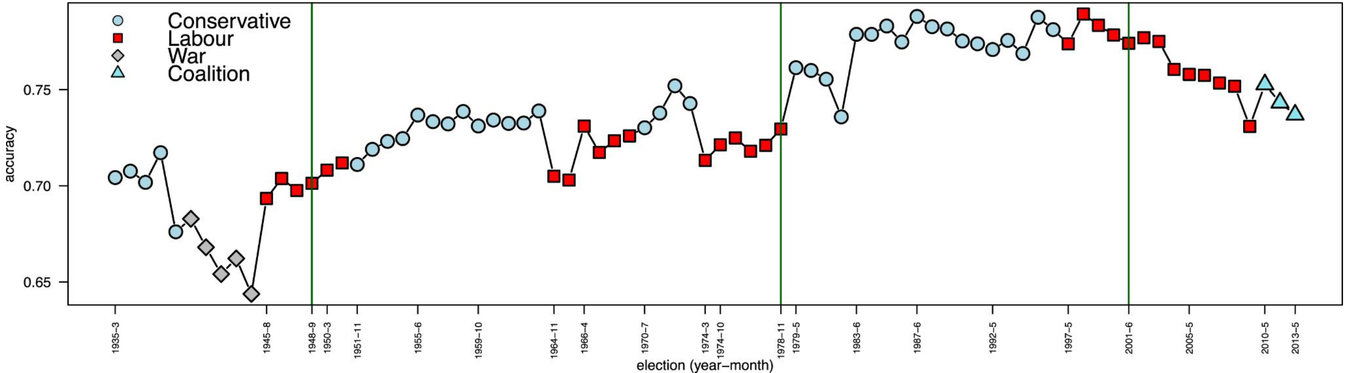
- Collect 3.5M UK parliament speeches between 1935 - 2013 and over 78 sessions
- A standard bag-of-words framework:
  - **Do**: tokenise, normalise, keep tokens that appear in >200 speeches
  - **Do not**: stopwords, stemming
- Label Y = party of speaker (Conservative or Labour)
- Create an ensemble of four classifiers, track accuracy:
  - Perceptron, SGD classifier, hinge-loss classifier, and L2 logistic regression

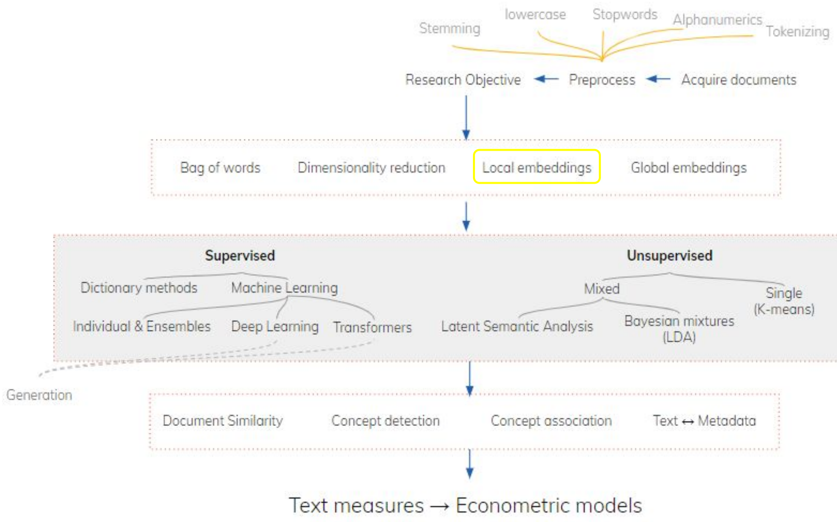
Paper application □

Peterson, Spirling (2018) - Measuring Polarisation in Westminster

"RILE" score from Manifesto Project (up is right wing)







Local Embeddings

What have we been doing?

Learning 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
- **Topic models:** document is a vector of shares over topics
- **Text classifiers:** produces $\hat{\boldsymbol{y}}_i = f(\boldsymbol{x}_i; \hat{\theta})$, a vector of predicted probabilities across classes for each document **i**
    - The vector of class probabilities $\hat{\boldsymbol{y}}_i$ is a **compressed representation** of the predictive text features
    - The vector of features $\boldsymbol{x}_i$ is itself a compressed representation of the unprocessed document **i**
- **Next**: the learned parameters $\hat{\theta}$ can also be interpreted as a learned compressed representation of the data and the relational information between corpus, text features, and outcome variables

Logistic regression example

- The learned matrix of parameters $\hat{\theta}$ relate input words to outcome classes. It contains **n_Y** columns, each a **n_X** vector representing **outcome classes as word distributions**, and vice versa.

Word embeddings as a parameter matrix

Bag of Words → one-hot vectors

- $\hat{\theta}$ is a matrix of parameters learned from the logistical regression, relating features to outcomes
- If **x** is a bag-of-words representation for a document consisting of a list of tokens {w_1, w_2, .., w□}, this representation can be expressed

$$\boldsymbol{x} = \frac{1}{n}\sum_{t=1}^{n} x_t$$

where each x_t is a $\mathbf{n_x}$ dimensional one-hot vector → all entries are zero except for a single one at index **t**

$$\begin{aligned} \text{motel} &= [0000000010000000] \\ \text{hotel} &= [0000100000000000] \end{aligned}$$

Continuous Bag of Words

- Let θ_t be the $\mathbf{n_y}$ dimensional row of $\hat{\theta}$, a **word embedding** for some w_r containing outcome relevant information for that word. The document vector is then

$$\vec{\boldsymbol{d}} = \frac{1}{n}\sum_{t=1}^{n_i} \theta_t$$

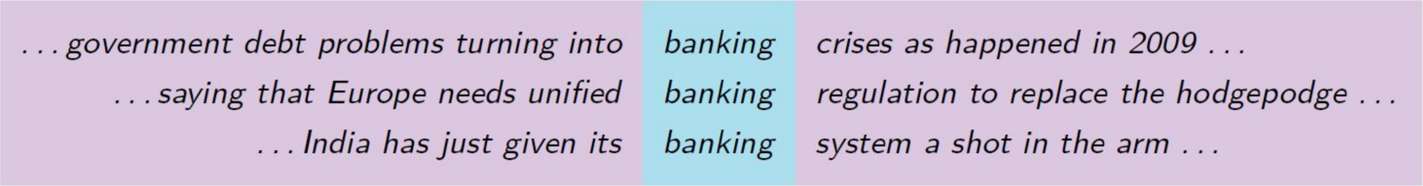
Word embedding matrix

$$\vec{\boldsymbol{d}} = \theta \cdot \boldsymbol{x}$$

Word embeddings with local context

Idea

- A word’s meaning is given by the words that frequently appear close-by - its **context**
    - *“You shall know a word by the company it keeps”* (J.R. Firth 1957)
        - “He filled the wampimuk, passed it around and we all drunk some.”
        - “We found a little, hairy wampimuk sleeping behind the tree”
- When a word **w** appears in a text, its context is the set of words that appear nearby (fixed-size window)
- Use the many contexts of **w** to build up a representation of **w**



. . . government debt problems turning into	*banking*	*crises as happened in 2009 . . .*
. . . saying that Europe needs unified	*banking*	*regulation to replace the hodgepodge . . .*
. . . India has just given its	*banking*	*system a shot in the arm . . .*

Word embeddings with local context

Context in linguistics

- Old NLP research aims to capture words’ distributional properties using a word-context matrix **M**
- Each row **w** in **M** represents a word, and each column **c** represents a linguistic context in which words may occur (ie. banking ↔ unified _ regulation)
- Individual matrix entries quantify the strength between a word and a context, and word rows give distributions of these over contexts
- Context may be more than single words, including sentences, paragraphs, nouns, syntactic links, etc.

Defining association

- **Counts:** The number of time **w** appeared along with context **c**
- **Pointwise mutual information:** The frequency of word **w** and context **c** collocating relative to the frequency of these appearing independently

$$f_M(w,c) = \frac{\Pr(w,c)}{\Pr(w)\Pr(c)} = \frac{\frac{\#(w,c)}{n_D}}{\frac{\#(w)}{n_D}\frac{\#(c)}{n_D}} = \frac{n_D\#(w,c)}{\#(w)\#(c)}$$

Not an embedding!

Note: Matrix **M** is typically too large, practical applications use Singular Value Decomposition to reduce it to some lower dimensional **W** matrix, preserving geometries

An embedding

GloVe embeddings □

Pennington et al. (2014) - Global Vectors for Word Representation

Idea

- Words that co-occur should have a high correlation (an inner product)

Methodology

- **Input**: **C_i□ ,** the local co-occurrence count between words **i** and **j.**
  Co-occurrence window defined as a 10-feature slider
- **Supervised**: Learn word vectors **w** = (**w_1, w_2, ..**) initialised randomly to solve

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

where f(.) is a weighting function used to down weight stopwords

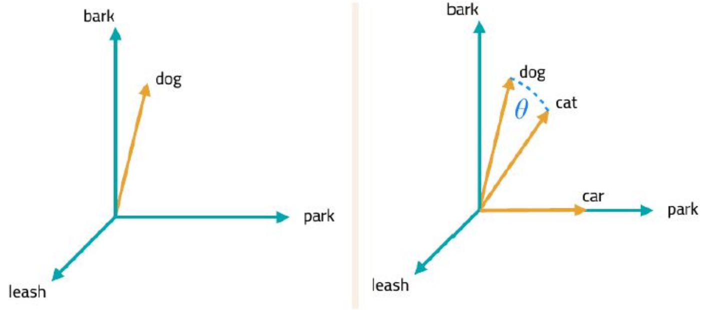
- **Optimisation**: Minimise the square difference between
  - the dot product of word vectors $w_i^T w_j$
  - the empirical co-occurrence of words $\log\left(C_{ij}\right)$

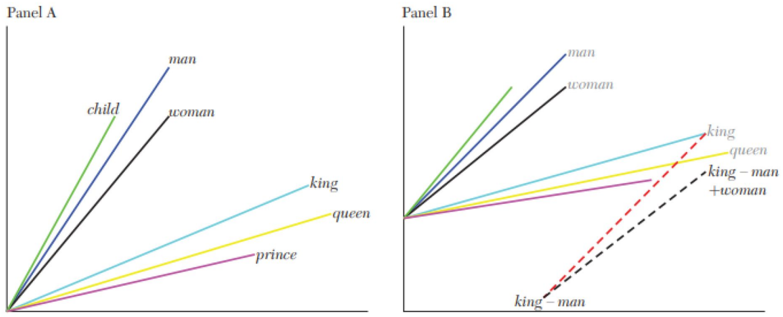
These are initialised in (-1, 1) for some lower dimensional vector

Word similarity

- Once words are represented as vectors, we can use linear algebra to understand the relationship between words given our corpus

- Geometric distance reflects semantic relatedness





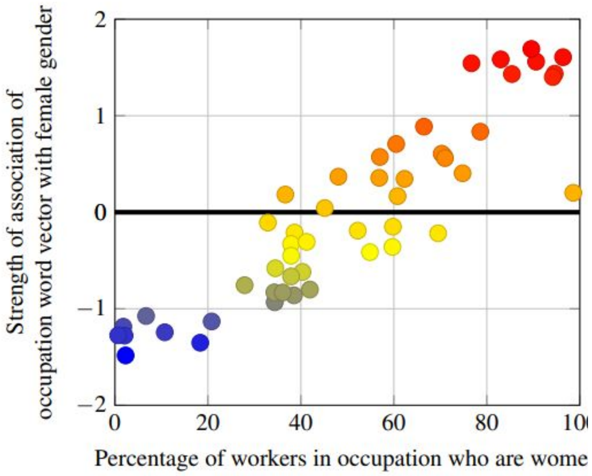
- Familiar metrics for comparing vectors include cosine similarity or Jaccard similarities

$$\cos\theta = \frac{w_1 \cdot w_2}{||w_1||||w_2||}$$

- These models are excellent at dealing with analogies, and due to space linearity one can compute similarities between groups of words by averaging these groups

Paper validation □

Caliskan et al. (2017) - Semantics form corpus contain biases



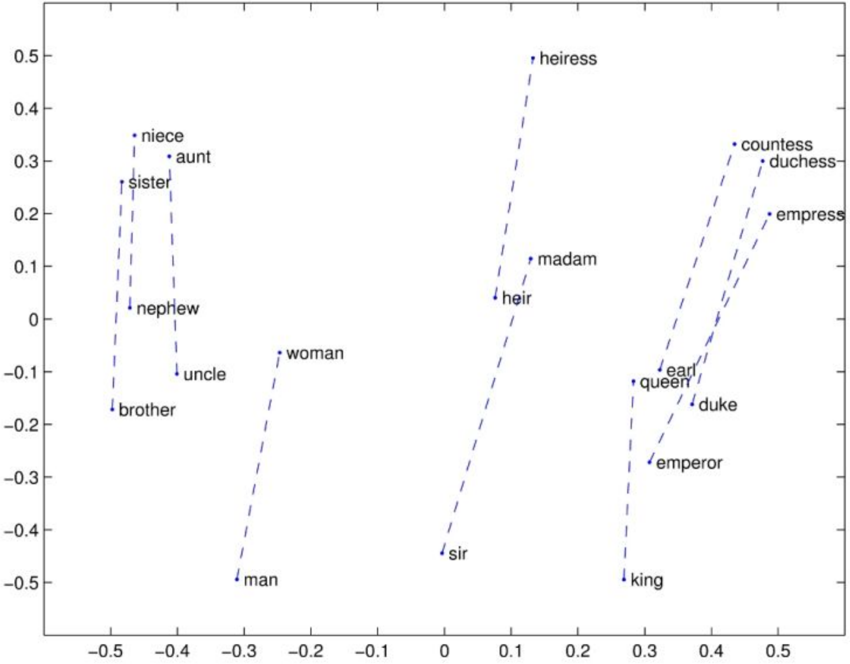


Figure 3. A 2D projection (first two principal components) of the 300-dimensional vector space of the GloVe word embedding (Pennington et al., 2014). The lines illustrate algebraic relationships between related words: pairs of words that differ only by gender map to pairs of vectors whose vector difference is roughly constant. Similar algebraic relationships have been shown for other semantic relationships, such as countries and their capital cities, companies and their CEOs, or simply different forms of the same word.

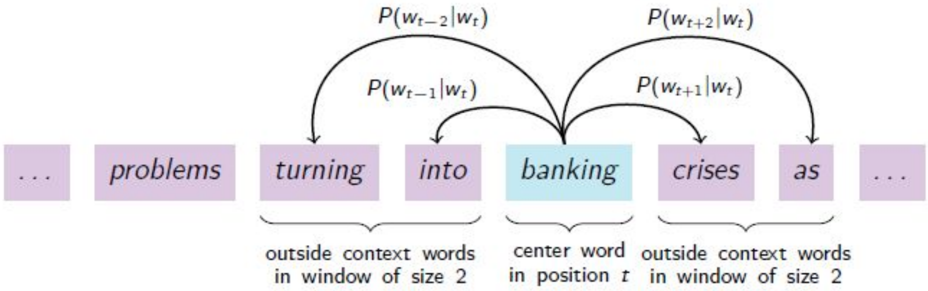
Word Embedding Association Test

word2vec □

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

Idea

A supervised model that seeks to predict context words **o** given center word **c** (and vice versa)


$P(w_{t-2}|w_t)$
$P(w_{t+2}|w_t)$
$P(w_{t-1}|w_t)$
$P(w_{t+1}|w_t)$
...
problems
turning
into
banking
crises
as
...
outside context words in window of size 2
center word in position t
outside context words in window of size 2

Methodology

- For each position **t** in the corpus, predict context words within a window of fixed size **m.** Given **w□=c**

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m} P\left(w_{t+j} \mid w_t; \theta\right)$$

- The **objective fun**

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{m \le j \le m} \log P\left(w_{t+j} \mid w_t; \theta\right)$$

Binary Cross Entropy

word2vec □

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

Objective

- We want to minimise the objective function

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{m\leq j\leq m} \log P\left(w_{t+j} \mid w_t; \theta\right)$$

How

- We will define **two embeddings**, containing the word-to-word mapping for all features, both when a feature is a center word and when it is a context word
  - v_w when **w** is a **center** word
  - u_w when **w** is a **context** word
- The probability of center word **o** given center word **c** is then given by a softmax transform

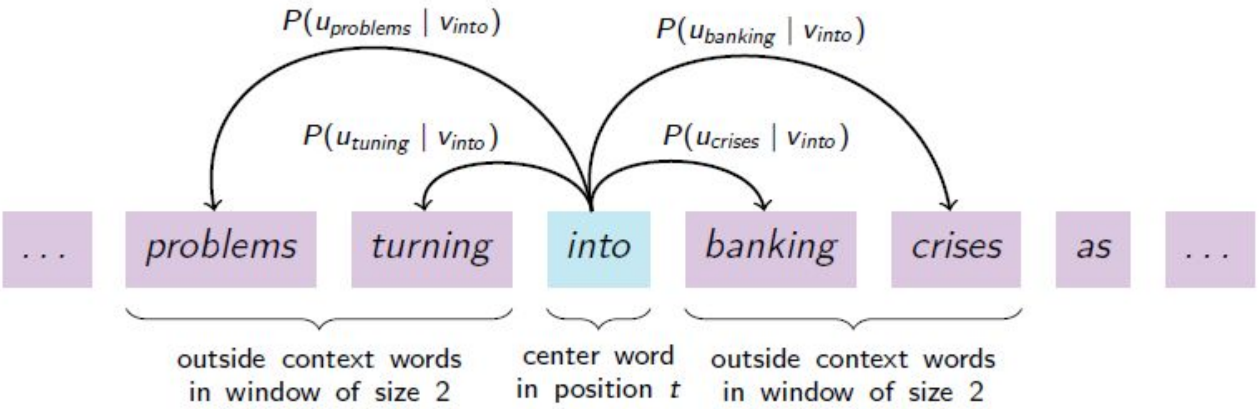
$$P(o \mid c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in V}\exp\left(u_w^T v_c\right)}$$

word2vec

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

How to calculate probabilities

- We will define **two embeddings**, containing the word-to-word mapping for all features, both when a feature is a center word and when it is a context word
    - v_w when **w** is a **center** word
    - u_w when **w** is a **context** word



$P(u_{problems} \mid v_{into})$
$P(u_{banking} \mid v_{into})$
$P(u_{tuning} \mid v_{into})$
$P(u_{crises} \mid v_{into})$
...
problems
turning
into
banking
crises
as
...
outside context words in window of size 2
center word in position t
outside context words in window of size 2

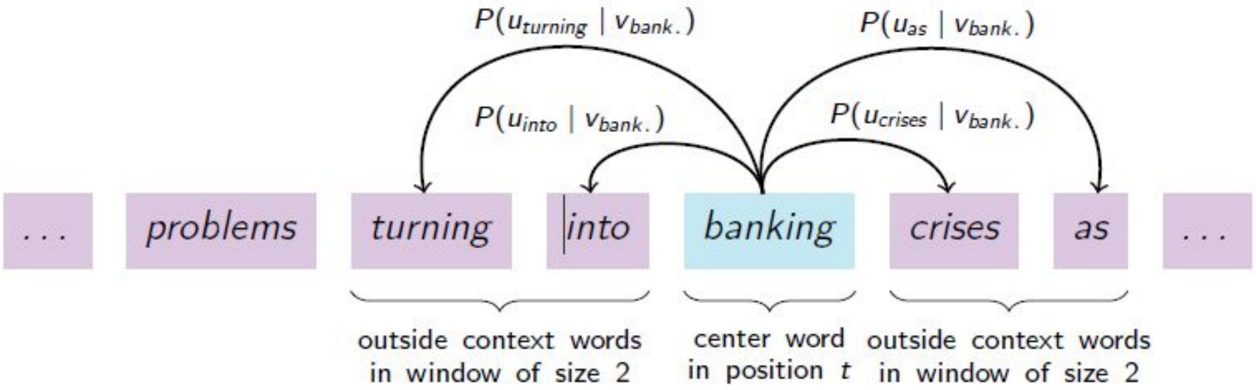
$P(u_{problems} \mid v_{into})$ is short for $P(problems \mid into, u_{problems}, v_{into}, \theta)$

word2vec □

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

How to calculate probabilities

- We will define **two embeddings**, containing the word-to-word mapping for all features, both when a feature is a center word and when it is a context word
  - v_w when **w** is a **center** word
  - u_w when **w** is a **context** word



$P(u_{turning} \mid v_{bank.})$
$P(u_{as} \mid v_{bank.})$
$P(u_{into} \mid v_{bank.})$
$P(u_{crises} \mid v_{bank.})$
. . .
problems
turning
into
banking
crises
as
. . .
outside context words in window of size 2
center word in position t
outside context words in window of size 2

word2vec □

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

Let’s rephrase 🔗

- The objective is to minimise the negative probability of predicting context words, which implies learning the optimal weight matrix **θ** - the concatenation of **c** and **o** matrix. Let’s call these $[W_{input} \quad W_{output}]$
- For a window size **C**, we seek to find **θ** such that
-
$$\underset{\theta}{\text{argmax}}\ p(w_1, w_2, \ldots, w_C | w_{center}; \theta)$$
- The softmax has the following equation

$$p(w_{context}|w_{center}; \theta) = \frac{exp(W_{output_{(context)}} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)}$$

where $W_{output_{(context)}}$ is a row vector for a context word from the output embedding matrix, and **h** is the hidden layer word vector for a center word.

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

$$\underset{\theta}{\text{argmax}}\ log \prod_{c=1}^{C} \frac{exp(W_{output_{(c)}} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)}$$

word2vec

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

- The negative likelihood function for a given center word **w** becomes

$$J(\theta; w^{(t)}) = -log \prod_{c=1}^{C} \frac{exp(W_{output_{(c)}} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)}$$

which after some basic algebraic wrangling, becomes

$$J(\theta; w^{(t)}) = -\sum_{c=1}^{C} (W_{output_{(c)}} \cdot h) + C \cdot log \sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)$$

which is equivalent to our original notation

$$J(\theta; w^{(t)}) = -\sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} \mid w_t; \theta) \longrightarrow \boxed{J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} \mid w_t; \theta)}$$

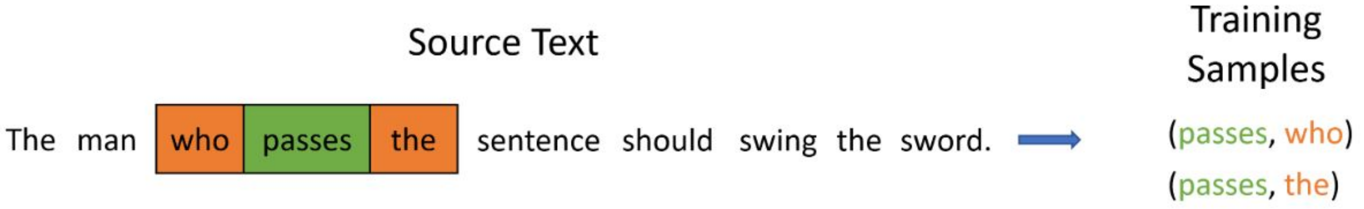
word2vec □

Mikolov et al. (2013) - Efficient Estimation of Word Repr. in Vector Space

- The negative likelihood function for the corpus **w** becomes

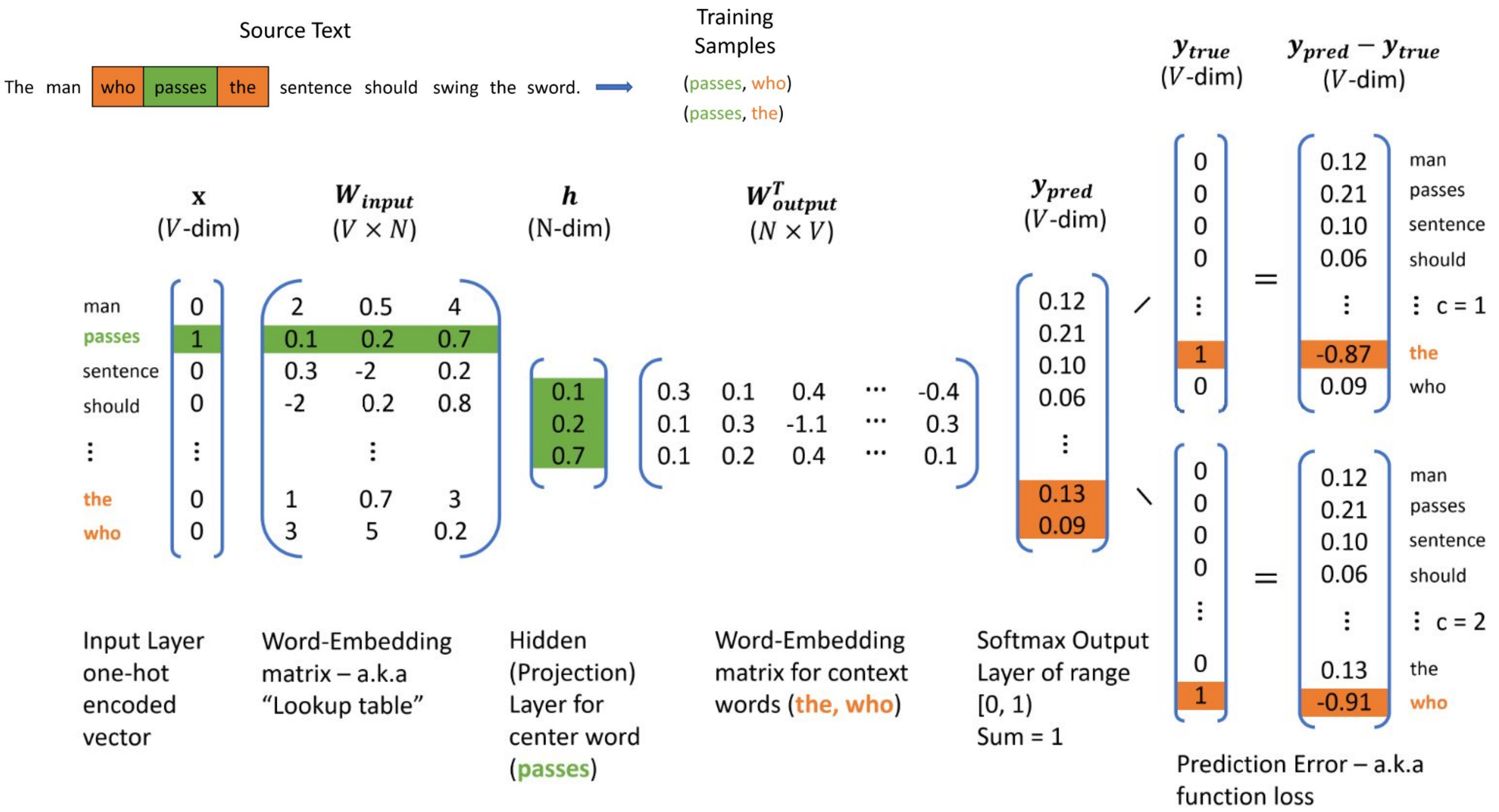
$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\frac{exp(\theta^{(t+j)\top}x^{(t)})}{\sum_{i=1}^{K} exp(\theta^{(i)\top}x^{(t)})}$$

An example



word2vec □

$$\prod_{c=1}^{C} \frac{exp(W_{output_{(c)}} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)}$$



Source Text

The man who passes the sentence should swing the sword. →

Training Samples

(passes, who)

(passes, the)

$$\mathbf{x}\ (V\text{-dim}) = \begin{array}{l|c} \text{man} & 0 \\ \text{passes} & 1 \\ \text{sentence} & 0 \\ \text{should} & 0 \\ \vdots & \vdots \\ \text{the} & 0 \\ \text{who} & 0 \end{array} \qquad W_{input}\ (V \times N) = \begin{pmatrix} 2 & 0.5 & 4 \\ 0.1 & 0.2 & 0.7 \\ 0.3 & -2 & 0.2 \\ -2 & 0.2 & 0.8 \\ & \vdots & \\ 1 & 0.7 & 3 \\ 3 & 5 & 0.2 \end{pmatrix} \qquad h\ (\text{N-dim}) = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.7 \end{pmatrix}$$

$$W^{T}_{output}\ (N \times V) = \begin{pmatrix} 0.3 & 0.1 & 0.4 & \cdots & -0.4 \\ 0.1 & 0.3 & -1.1 & \cdots & 0.3 \\ 0.1 & 0.2 & 0.4 & \cdots & 0.1 \end{pmatrix} \qquad y_{pred}\ (V\text{-dim}) = \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ 0.09 \end{pmatrix}$$

$$y_{true}\ (V\text{-dim}) \qquad y_{pred} - y_{true}\ (V\text{-dim})$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} = \begin{array}{c|l} 0.12 & \text{man} \\ 0.21 & \text{passes} \\ 0.10 & \text{sentence} \\ 0.06 & \text{should} \\ \vdots & \vdots\ \ c = 1 \\ -0.87 & \text{the} \\ 0.09 & \text{who} \end{array}$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \begin{array}{c|l} 0.12 & \text{man} \\ 0.21 & \text{passes} \\ 0.10 & \text{sentence} \\ 0.06 & \text{should} \\ \vdots & \vdots\ \ c = 2 \\ 0.13 & \text{the} \\ -0.91 & \text{who} \end{array}$$

Input Layer one-hot encoded vector

Word-Embedding matrix – a.k.a "Lookup table"

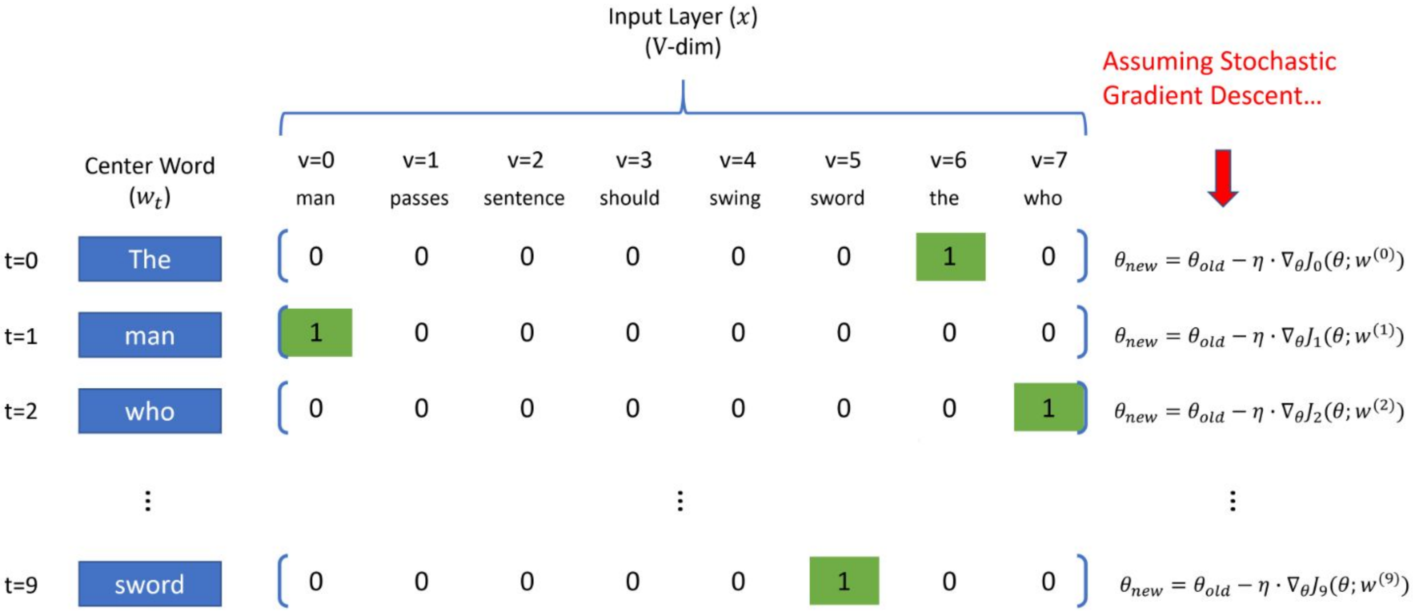
Hidden (Projection) Layer for center word (passes)

Word-Embedding matrix for context words (the, who)

Softmax Output Layer of range [0, 1) Sum = 1

Prediction Error – a.k.a function loss

word2vec



Input Layer (x)
(V-dim)

Assuming Stochastic Gradient Descent...

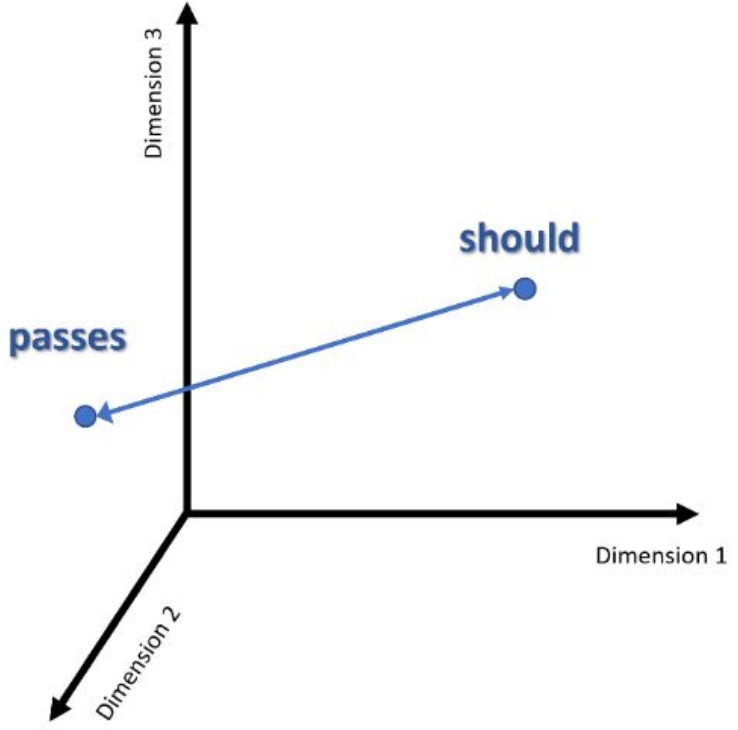
	Center Word (w_t)	v=0 man	v=1 passes	v=2 sentence	v=3 should	v=4 swing	v=5 sword	v=6 the	v=7 who	
t=0	The	0	0	0	0	0	0	1	0	$\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta J_0(\theta; w^{(0)})$
t=1	man	1	0	0	0	0	0	0	0	$\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta J_1(\theta; w^{(1)})$
t=2	who	0	0	0	0	0	0	0	1	$\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta J_2(\theta; w^{(2)})$
	⋮				⋮					⋮
t=9	sword	0	0	0	0	0	1	0	0	$\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta J_9(\theta; w^{(9)})$

Inefficient? Very! See negative sampling objective

word2vec

$$\begin{array}{c} \boldsymbol{W}_{input} \\ (V \times N) \end{array}$$

$$\begin{array}{l} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{array} \begin{pmatrix} 2 & 0.5 & 4 \\ 0.1 & 0.2 & 0.7 \\ 0.3 & -2 & 0.2 \\ -2 & 0.2 & 0.8 \\ & \vdots & \\ 1 & 0.7 & 3 \\ 3 & 5 & 0.2 \end{pmatrix}$$



Optimising the embedding matrices results in representing words in a high quality vector space, capturing word semantics

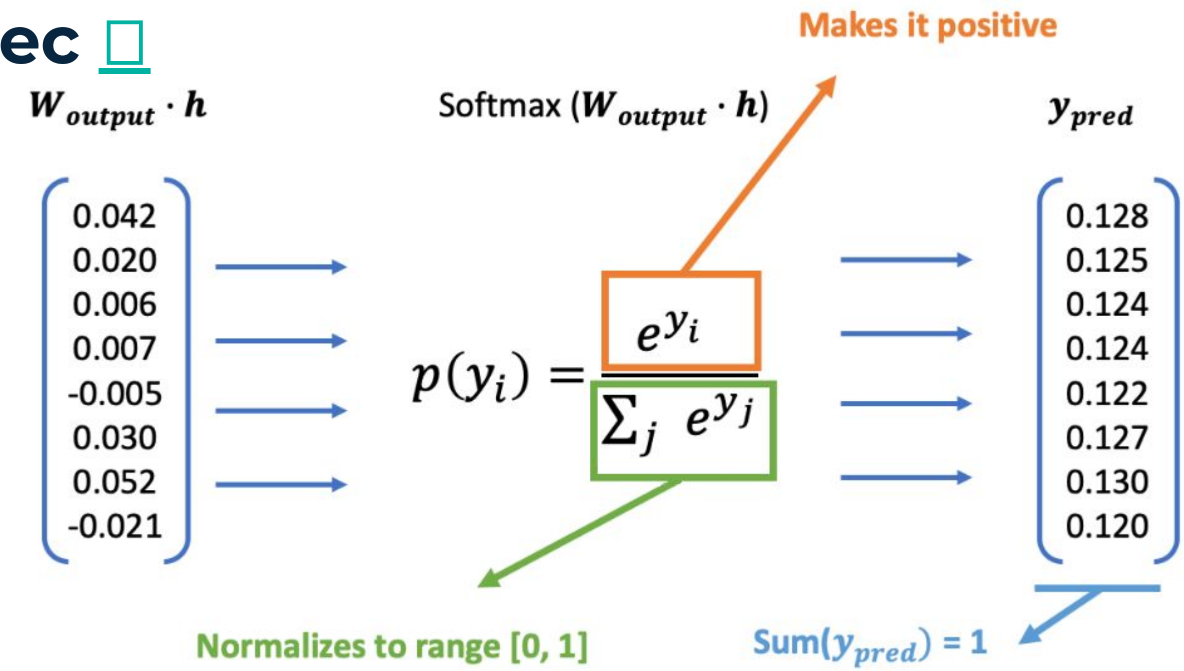
word2vec □

$$\begin{array}{c}\mathbf{x}\\(V\text{-dim})\\ \begin{array}{l}\text{man}\\ \text{passes}\\ \text{sentence}\\ \text{should}\\ \vdots\\ \text{the}\\ \text{who}\end{array}\begin{bmatrix}0\\1\\0\\0\\ \vdots\\0\\0\end{bmatrix}\end{array} \times \begin{array}{c}\boldsymbol{W}_{\boldsymbol{input}}\\(V \times N)\\ \begin{pmatrix}2 & 0.5 & 4\\ 0.1 & 0.2 & 0.7\\ 0.3 & -2 & 0.2\\ -2 & 0.2 & 0.8\\ & \vdots & \\ 1 & 0.7 & 3\\ 3 & 5 & 0.2\end{pmatrix}\end{array} = \begin{array}{c}\boldsymbol{h}\\(N\text{-dim})\\ \begin{bmatrix}0.1\\0.2\\0.7\end{bmatrix}\end{array}$$

Projection of word vector for “passes” from the embedding matrix

$$h = W_{input}^{T} \cdot x \in \mathbb{R}^{N}$$

word2vec □



$W_{output} \cdot h$

$$\begin{pmatrix} 0.042 \\ 0.020 \\ 0.006 \\ 0.007 \\ -0.005 \\ 0.030 \\ 0.052 \\ -0.021 \end{pmatrix}$$

Softmax $(W_{output} \cdot h)$

Makes it positive

$$p(y_i) = \frac{\boxed{e^{y_i}}}{\boxed{\sum_j e^{y_j}}}$$

Normalizes to range [0, 1]

y_{pred}

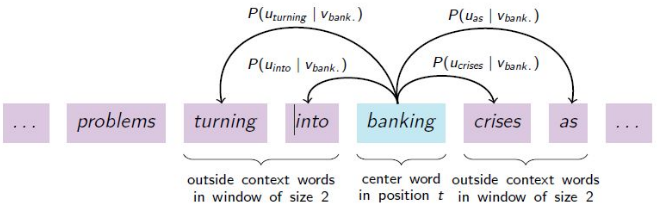
$$\begin{pmatrix} 0.128 \\ 0.125 \\ 0.124 \\ 0.124 \\ 0.122 \\ 0.127 \\ 0.130 \\ 0.120 \end{pmatrix}$$

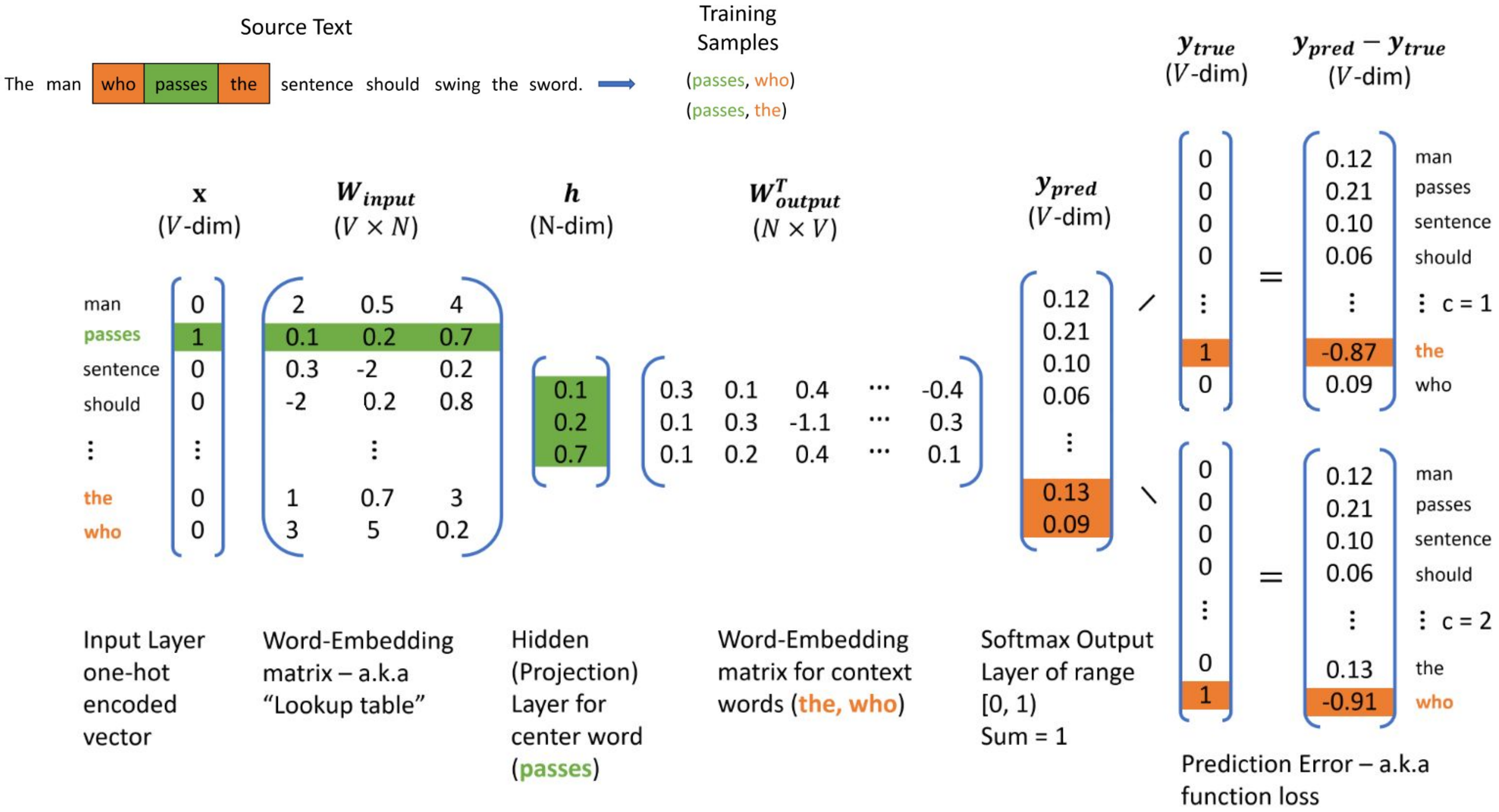
Sum(y_{pred}) = 1

$$\begin{bmatrix} p(w_1|w_{center}) \\ p(w_2|w_{center}) \\ p(w_3|w_{center}) \\ \vdots \\ p(w_V|w_{center}) \end{bmatrix} = \frac{exp(W_{output} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)} \in \mathbb{R}^V$$

$$p(w_{context}|w_{center}) = \frac{exp(W_{output_{(context)}} \cdot h)}{\sum_{i=1}^{V} exp(W_{output_{(i)}} \cdot h)} \in \mathbb{R}^1$$

word2vec □


$P(u_{turning} \mid v_{bank.})$
$P(u_{as} \mid v_{bank.})$
$P(u_{into} \mid v_{bank.})$
$P(u_{crises} \mid v_{bank.})$
. . .
problems
turning
into
banking
crises
as
. . .
outside context words in window of size 2
center word in position t
outside context words in window of size 2


Source Text

The man who passes the sentence should swing the sword. →

Training Samples

(passes, who)

(passes, the)

$$
\begin{array}{c|c}
 & \mathbf{x} \; (V\text{-dim}) \\
\text{man} & 0 \\
\text{passes} & 1 \\
\text{sentence} & 0 \\
\text{should} & 0 \\
\vdots & \vdots \\
\text{the} & 0 \\
\text{who} & 0
\end{array}
\quad
W_{input} \; (V \times N) = \begin{pmatrix} 2 & 0.5 & 4 \\ 0.1 & 0.2 & 0.7 \\ 0.3 & -2 & 0.2 \\ -2 & 0.2 & 0.8 \\ & \vdots & \\ 1 & 0.7 & 3 \\ 3 & 5 & 0.2 \end{pmatrix}
\quad
\boldsymbol{h} \; (\text{N-dim}) = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.7 \end{pmatrix}
\quad
W^T_{output} \; (N \times V) = \begin{pmatrix} 0.3 & 0.1 & 0.4 & \cdots & -0.4 \\ 0.1 & 0.3 & -1.1 & \cdots & 0.3 \\ 0.1 & 0.2 & 0.4 & \cdots & 0.1 \end{pmatrix}
$$

$$
y_{pred} \; (V\text{-dim}) = \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ 0.09 \end{pmatrix}
$$

$$
\begin{array}{ccccl}
y_{true} \; (V\text{-dim}) & & y_{pred} - y_{true} \; (V\text{-dim}) & & \\
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} & = & \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ -0.87 \\ 0.09 \end{pmatrix} & & \begin{array}{l} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \;\; c = 1 \\ \text{the} \\ \text{who} \end{array} \\
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} & = & \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ -0.91 \end{pmatrix} & & \begin{array}{l} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \;\; c = 2 \\ \text{the} \\ \text{who} \end{array}
\end{array}
$$

Input Layer one-hot encoded vector

Word-Embedding matrix – a.k.a "Lookup table"

Hidden (Projection) Layer for center word (passes)

Word-Embedding matrix for context words (the, who)

Softmax Output Layer of range [0, 1) Sum = 1

Prediction Error – a.k.a function loss

Embeddings: Features

Semantic Similarity

- The embedding space captures words’ common attributes and features
  - **Synonymy**: car ↔ automobile
  - **Hypernymy**: car ↔ vehicle
  - **Co-hyponym**: car ↔ van ↔ truck

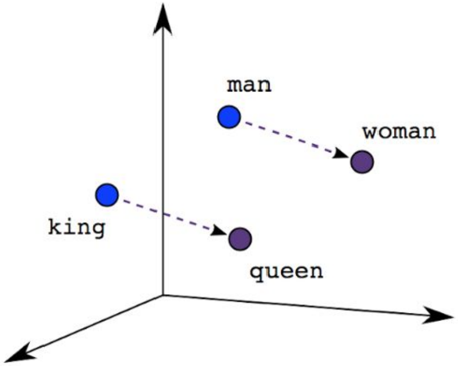
Semantic Similarity

- The embedding space captures words’ semantic association without being similar
  - **Function**: car ↔ drive
  - **Meronymy**: car ↔ trie
  - **Location**: car ↔ road
  - **Attribute**: car ↔ fast

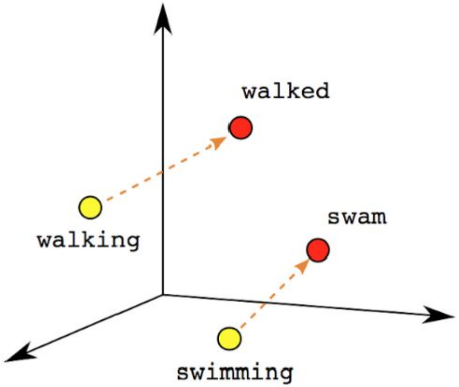
Note the relationships will reflect the choice of window size (small → substitutes, large → topics)

Embeddings: Features

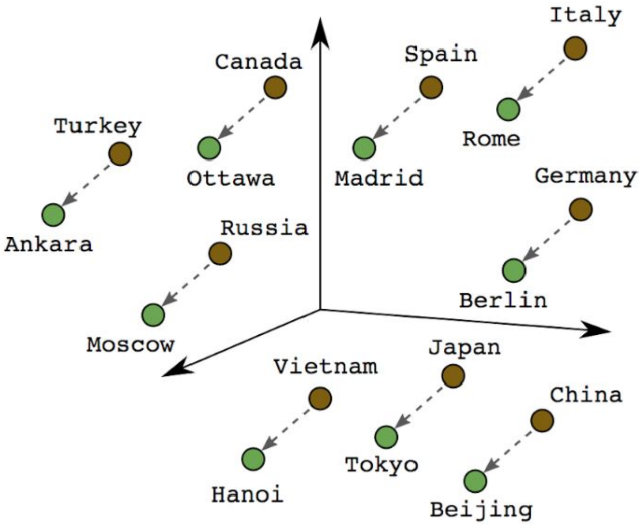
Word2vec algebra can depict conceptual, analogical relationships between words



Male-Female



Verb Tense



Country-Capital

Embeddings: Caveats

- **Polysemy**: Embeddings may struggle with capturing multiple meanings of words. This can be partly addressed by including POS information in the tokens.

- **Out-of-vocabulary words**: These models may encounter out-of-vocabulary words not present in the training data. Large pre-trained models are typically used to minimise risk

- **Contextual variations**: Word embeddings may not fully capture the nuances of contextual variations, including sarcasm, irony, or sentiment.

- **Data bias and representation:** self-supervised models learn all dimensions of word associations, including potentially harmful or biased ones.

Bolukbasi et al (2016)

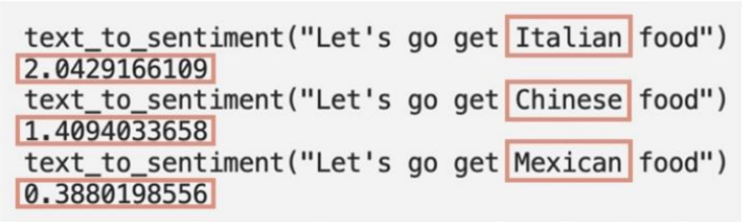
- “Geometrically, gender bias is first shown to be captured by a direction in the word embedding.“
- “Second, gender neutral words are shown to be linearly separable from gender definition words in the word embedding.”
- “Using these properties, we provide a methodology for modifying an embedding to 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.”

Gonen, Goldberg (2019)

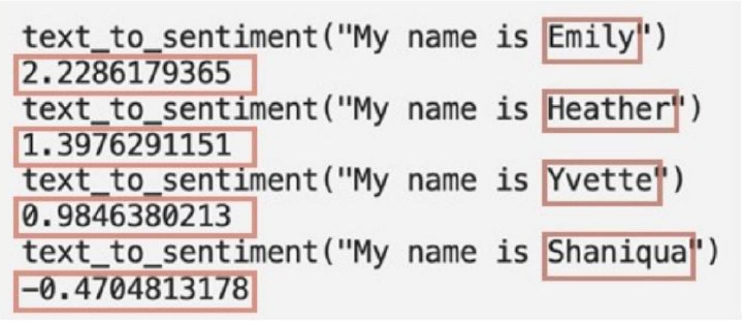
- “We argue that this removal is superficial. While the bias is indeed substantially reduced according to the provided bias definition, the actual effect is mostly hiding the bias, not removing it. The gender bias information is still reflected in the distances between ’gender-neutralized’ words in the debiased embeddings, and can be recovered from them...”

- **The black sheep problem**: Trivial word features are often omitted, be wary of interpretation

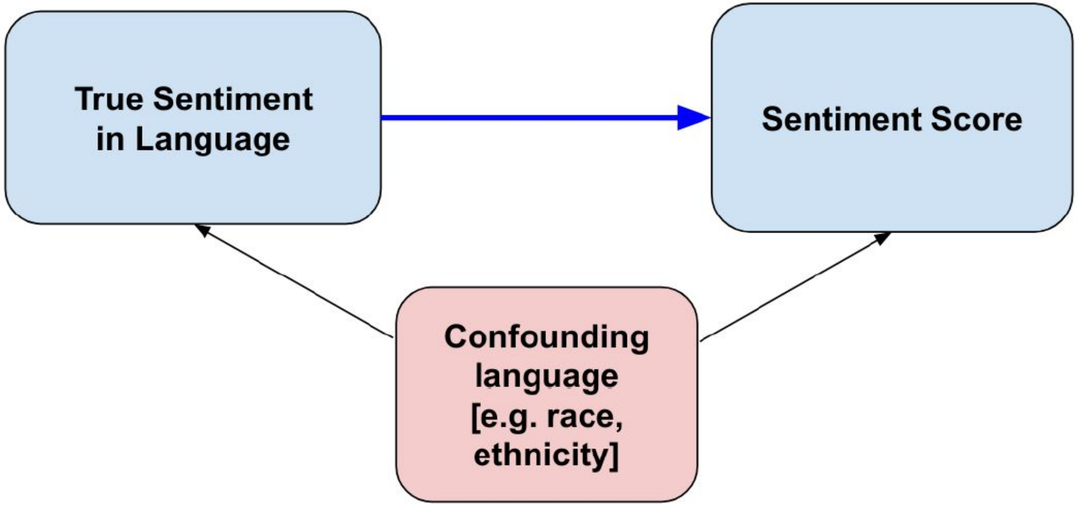
Embeddings: Caveats - Bias



```
text_to_sentiment("Let's go get Italian food")
2.0429166109
text_to_sentiment("Let's go get Chinese food")
1.4094033658
text_to_sentiment("Let's go get Mexican food")
0.3880198556
```



```
text_to_sentiment("My name is Emily")
2.2286179365
text_to_sentiment("My name is Heather")
1.3976291151
text_to_sentiment("My name is Yvette")
0.9846380213
text_to_sentiment("My name is Shaniqua")
-0.4704813178
```



True Sentiment in Language
Sentiment Score
Confounding language [e.g. race, ethnicity]

A great paper on the effects of slanted language on bias:

Djourelova, Milena. 2023. "Persuasion through Slanted Language: Evidence from the Media Coverage of Immigration."

Paper application

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes

Goal

Develop a systematic framework to analyse word embeddings trained over 100 of text data to identify historical patterns of bias and stereotype changes in the US

Motivation

In word-embedding models, words are assigned to a high-dimensional vector in a way that they capture relationships not found through simple co-occurrence analysis

Idea

Exploit differences in Euclidean distance between ethnic-gender terms and professions-stereotypes words to quantify historical trends

Findings

The embedding captures societal shifts and sheds light on how specific adjectives and occupations became more closely associated with certain populations over time

Paper application □

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes

Word embeddings

- word2vec embeddings trained on the Google News dataset
- Nine decade-specific embeddings trained on text from the Corpus of Historical American English

Word lists

- **Gender**: he, she, son, daughter, male, female, boy, girl, etc.
- **Ethnicity**: harris, ruiz, cho, thompson, gomez, lin, etc.
- **Occupations**: janitor, teacher, shoemaker, scientist, carpenter, etc.
- **Adjectives**: headstrong, inventive, enterprising, poised, moody, etc.

Paper application □

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes

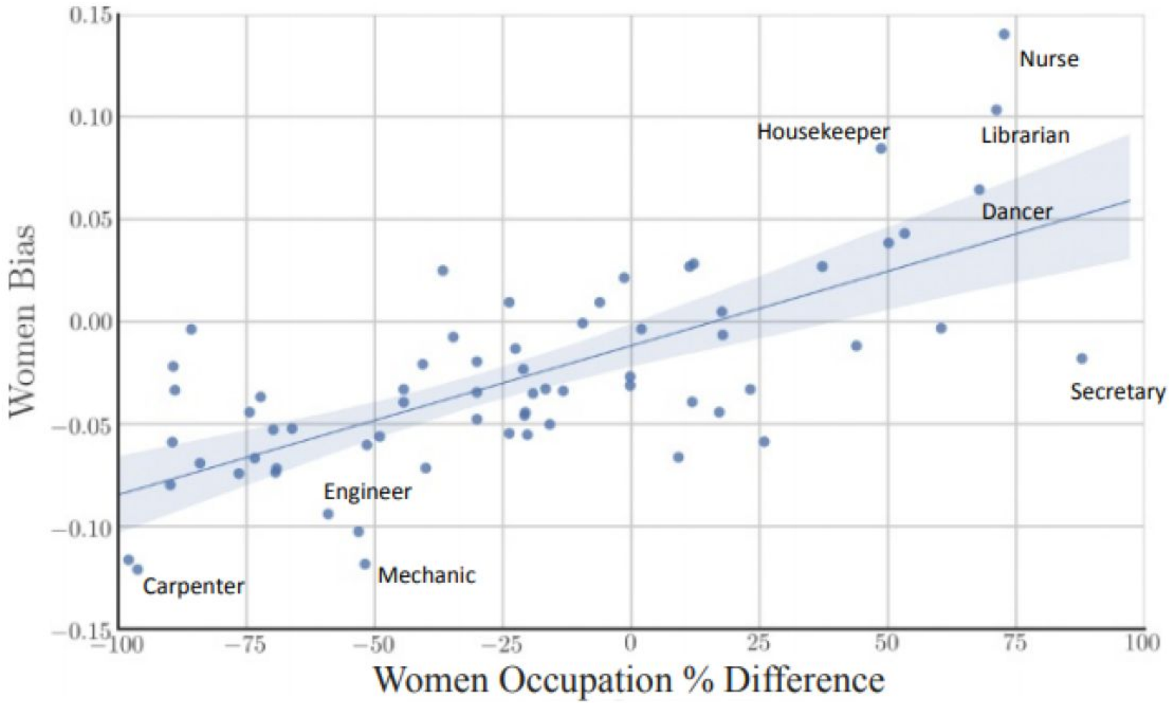
Methodology

- Measure the strength of association between **occupations** or **adjectives** AND **gender** or **ethnicity**
  - Compute the average vector representation of a gender or ethnic group
  - Calculate the average Euclidean distance between the representative vector and each vector in a list of neutral words
  - Use the difference of the average distance between gender or ethnicity pairs as a measure of embedding bias
- ie. the occupational embedding bias for women
  - Compute average embedding distance between words *she*, *female* and occupational words *teacher*, *lawyer*. Repeat the same process for words *he*, *male*
  - Compute the average distances between group pair

$$\text{relative norm distance} = \sum_{v_m \in M} \|v_m - v_1\|_2 - \|v_m - v_2\|_2$$

Paper application □

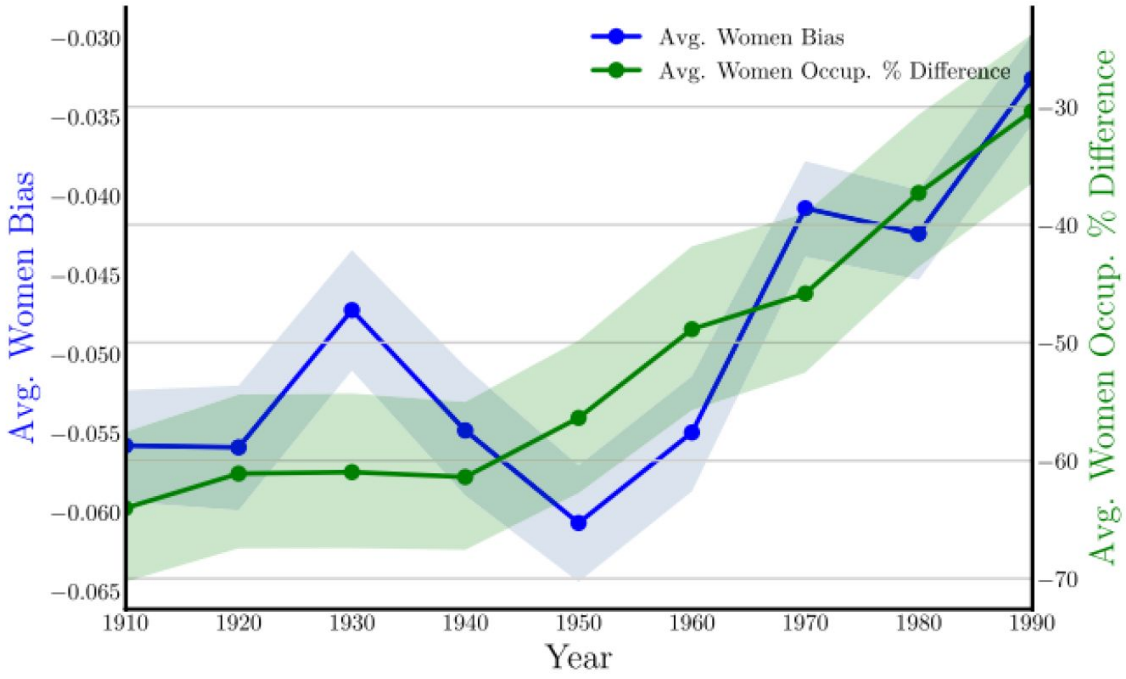
Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes



Occupation difference as the relative percentage of women in each occupation using data from the Integrated Public Use Microdata Series

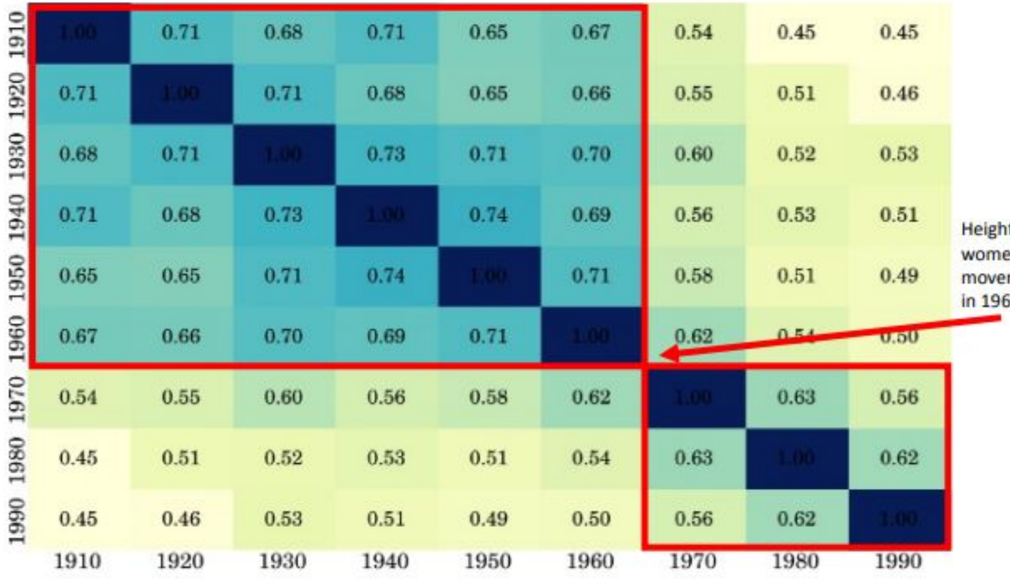
Paper application □

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes



Paper application

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes



	1910	1920	1930	1940	1950	1960	1970	1980	1990
1910	1.00	0.71	0.68	0.71	0.65	0.67	0.54	0.45	0.45
1920	0.71	1.00	0.71	0.68	0.65	0.66	0.55	0.51	0.46
1930	0.68	0.71	1.00	0.73	0.71	0.70	0.60	0.52	0.53
1940	0.71	0.68	0.73	1.00	0.74	0.69	0.56	0.53	0.51
1950	0.65	0.65	0.71	0.74	1.00	0.71	0.58	0.51	0.49
1960	0.67	0.66	0.70	0.69	0.71	1.00	0.62	0.54	0.50
1970	0.54	0.55	0.60	0.56	0.58	0.62	1.00	0.63	0.56
1980	0.45	0.51	0.52	0.53	0.51	0.54	0.63	1.00	0.62
1990	0.45	0.46	0.53	0.51	0.49	0.50	0.56	0.62	1.00

Height of women's movements in 1960s-70s

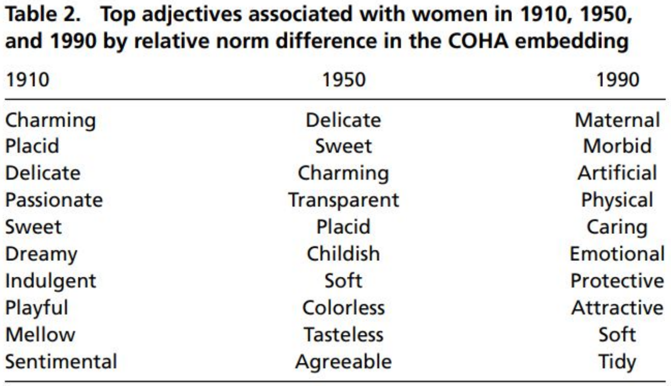


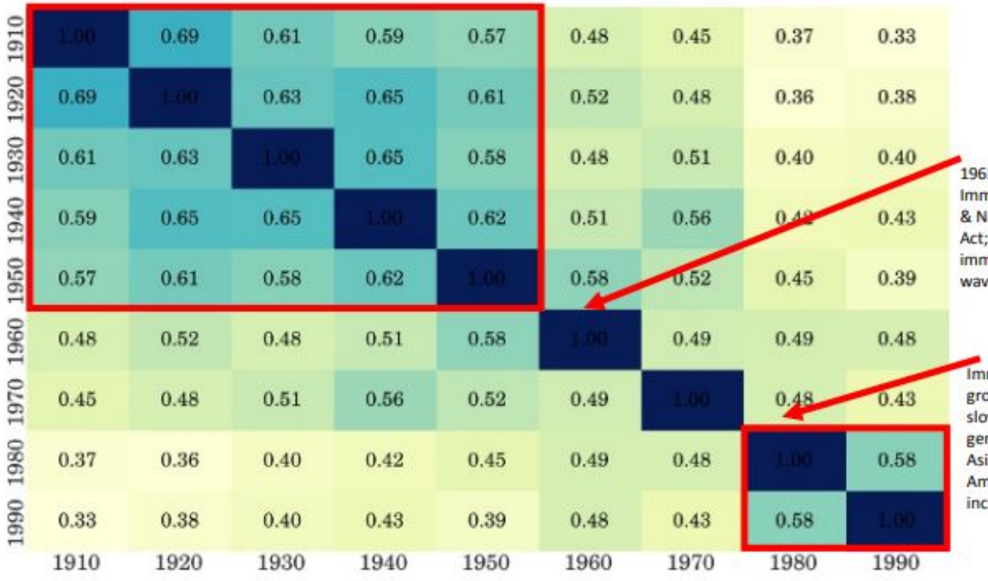
Table 2. Top adjectives associated with women in 1910, 1950, and 1990 by relative norm difference in the COHA embedding

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

Pearson correlation in embedding female bias scores for adjectives over time

Paper application

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes



	1910	1920	1930	1940	1950	1960	1970	1980	1990
1910	1.00	0.69	0.61	0.59	0.57	0.48	0.45	0.37	0.33
1920	0.69	1.00	0.63	0.65	0.61	0.52	0.48	0.36	0.38
1930	0.61	0.63	1.00	0.65	0.58	0.48	0.51	0.40	0.40
1940	0.59	0.65	0.65	1.00	0.62	0.51	0.56	0.42	0.43
1950	0.57	0.61	0.58	0.62	1.00	0.58	0.52	0.45	0.39
1960	0.48	0.52	0.48	0.51	0.58	1.00	0.49	0.49	0.48
1970	0.45	0.48	0.51	0.56	0.52	0.49	1.00	0.48	0.43
1980	0.37	0.36	0.40	0.42	0.45	0.49	0.48	1.00	0.58
1990	0.33	0.38	0.40	0.43	0.39	0.48	0.43	0.58	1.00

1965 Immigration & Nationality Act; Asian immigration wave

Immigration growth slows; 2nd generation Asian Americans increase

Pearson correlation in embedding Asian bias scores for adjectives over time

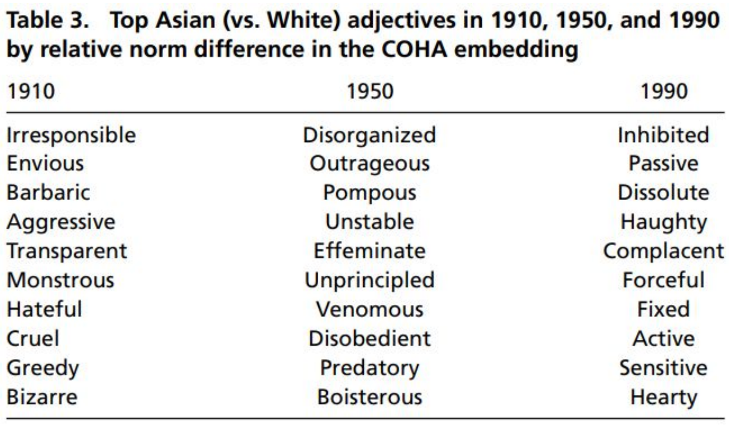
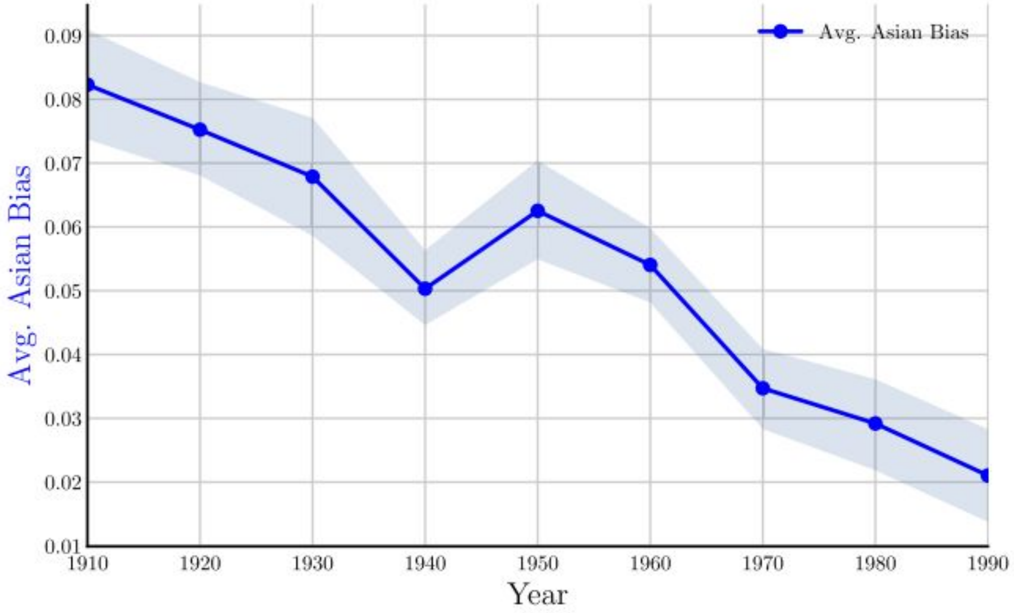


Table 3. Top Asian (vs. White) adjectives in 1910, 1950, and 1990 by relative norm difference in the COHA embedding

1910	1950	1990
Irresponsible	Disorganized	Inhibited
Envious	Outrageous	Passive
Barbaric	Pompous	Dissolute
Aggressive	Unstable	Haughty
Transparent	Effeminate	Complacent
Monstrous	Unprincipled	Forceful
Hateful	Venomous	Fixed
Cruel	Disobedient	Active
Greedy	Predatory	Sensitive
Bizarre	Boisterous	Hearty

Paper application □

Garg et al. (2018) - WE quantify 100 years of ethnic and gender stereotypes



Asian bias score over time for words related to outsiders in COHA data

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Motivation

Under the premise that text captures culture, construct **cultural dimensions** of class from the numerical representation of word embeddings. Identify the evolution of class relationships over the XXth century.

Idea

Class as the systematic and hierarchical distinction of people and groups in social standing. Dimensions:

- **Money**: easy to convert into various forms of power → **affluence**
- **Education**: determines the labour market position → **education**
- **Status:** based on authority and social position → **status**
- **Cultivated taste**: based on the culture consumed → **cultivation**
- **Gender**: misogynistic or patriarchal hierarchies → **gender**
- **Race**: reflected in post-colonial, structural racism → **race**

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Models

- Three pre-trained word embedding models: (i) Google n-grams US, (ii) Google news embeddings, (iii) GloVe

Dimensions

- **Affluence**: rich vs. poor, wealthy vs. impoverished, luxury vs. cheap
- **Education**: educated vs. uneducated, knowledgeable vs. ignorant
- **Status**: acclaimed vs. modest, eminent vs. mundane
- **Cultivation**: civil vs. uncivil, cultured vs. uncultured
- **Gender**: masculine vs. feminine, he vs. she, male vs. female
- **Race**: black vs. white, African vs. European

Words that are opposites semantically will display systematic differences in their vector representation

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Method example

- Solving the analogy is equivalent to projecting a word vector knot a specific dimension

$$\overrightarrow{\text{king}} + \overrightarrow{\text{woman}} - \overrightarrow{\text{man}} \approx \overrightarrow{\text{queen}}$$

- The projection of the word vector for king onto a gender dimension captured by **woman - man** yields **queen**
- Collate lists of antonyms similar to woman - man for the different dimensions of class, ie. **rich - poor**
- Project words onto dimension-specific antonym lists to identify the cultural associations embedded in **w**

$$\overrightarrow{\text{hockey}} + \overrightarrow{\text{rich}} - \overrightarrow{\text{poor}} \approx \overrightarrow{\text{lacrosse}}$$

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Method example

- Solving the analogy is equivalent to projecting a word vector knot a specific dimension

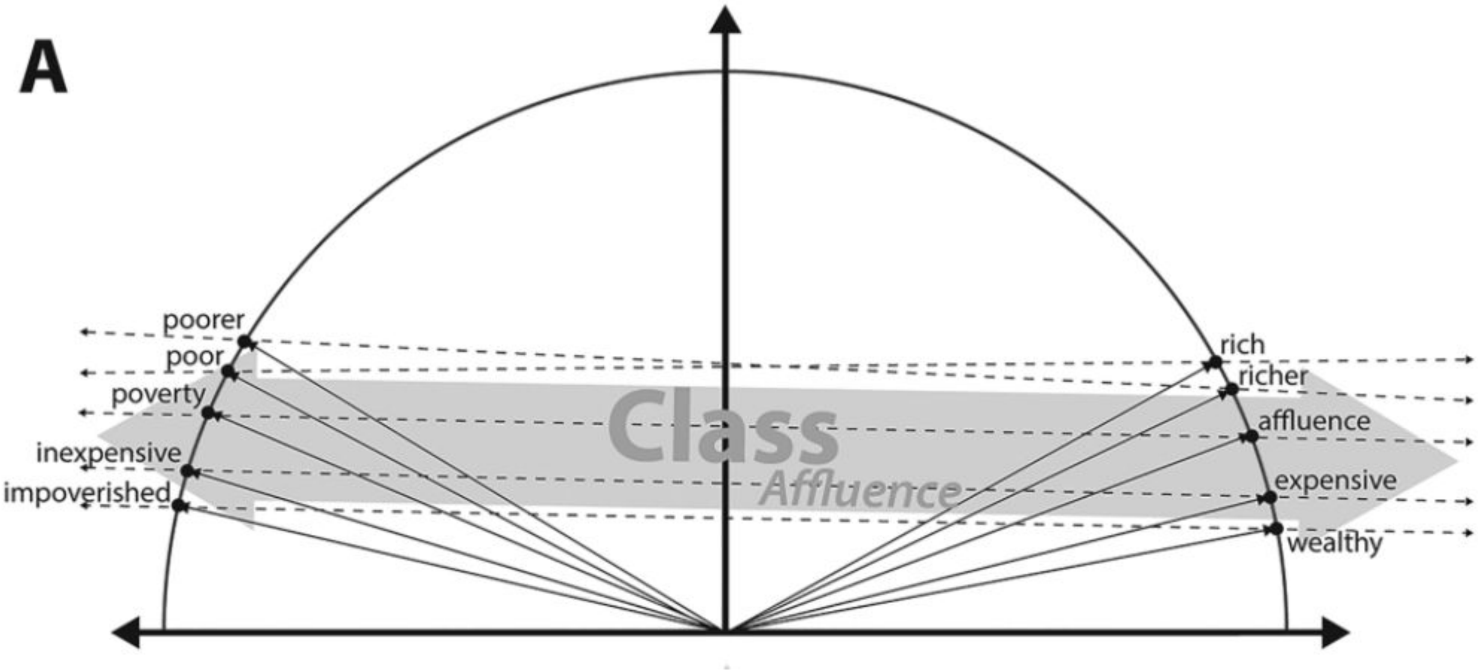
$$\overrightarrow{\text{king}} + \overrightarrow{\text{woman}} - \overrightarrow{\text{man}} \approx \overrightarrow{\text{queen}}$$

- The projection of the word vector for king onto a gender dimension captured by **woman - man** yields **queen**
- Collate lists of antonyms similar to woman - man for the different dimensions of class, ie. **rich - poor**
- Project words onto dimension-specific antonym lists to identify the cultural associations embedded in **w**

$$\overrightarrow{\text{hockey}} + \overrightarrow{\text{rich}} - \overrightarrow{\text{poor}} \approx \overrightarrow{\text{lacrosse}}$$

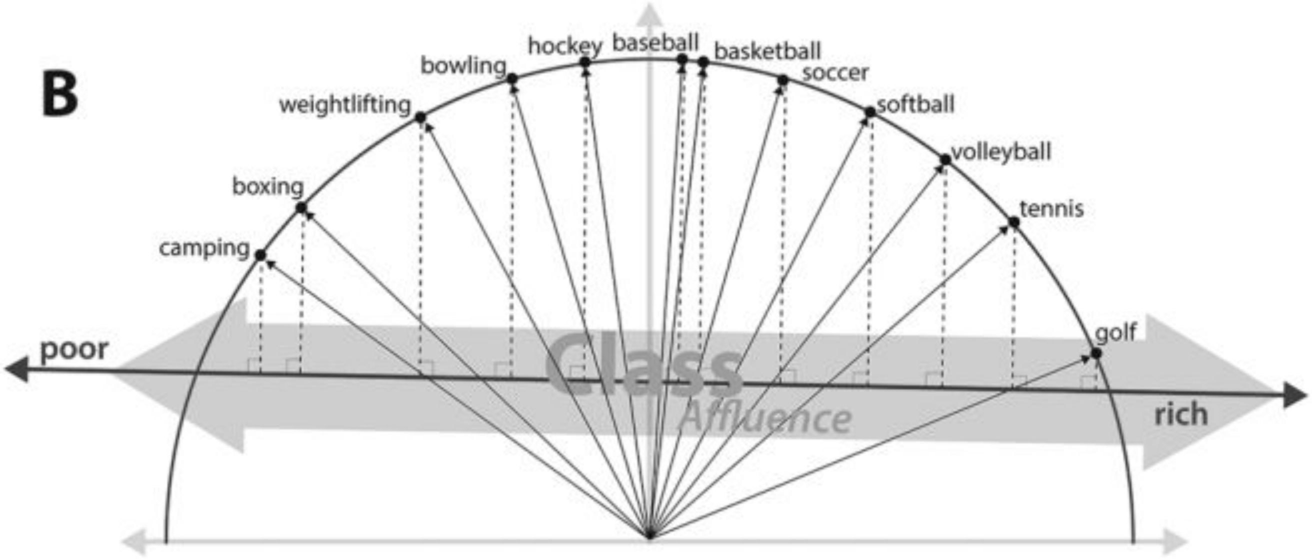
Paper application

Kozlowski et al. (2019) - The Geometry of Culture (and Class)



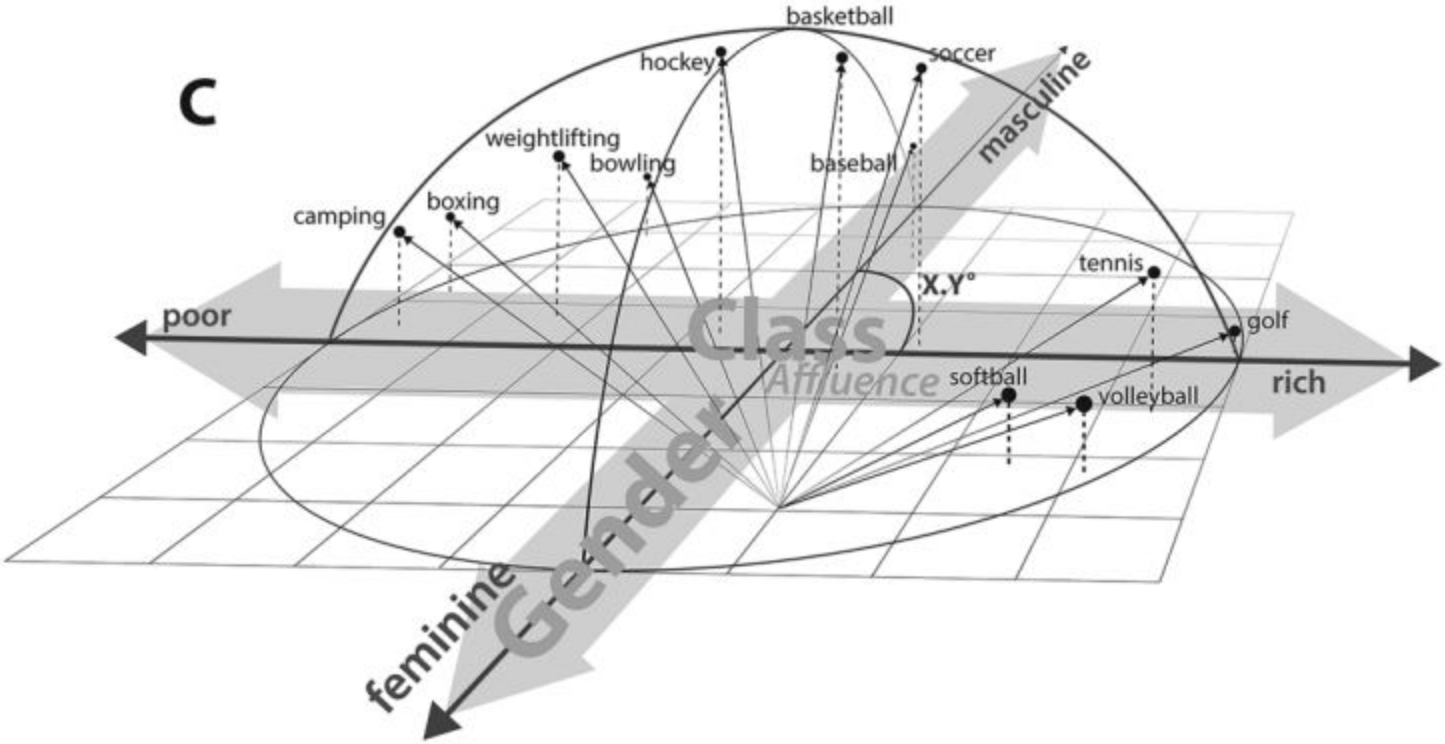
Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)



Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)



Paper application

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Validation

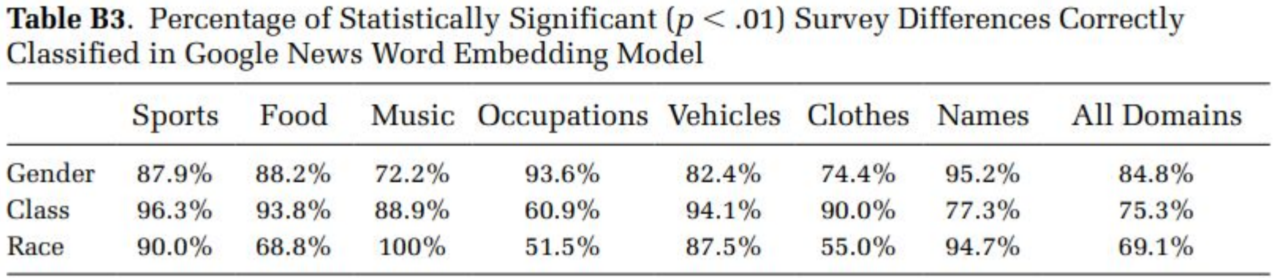


Table B3. Percentage of Statistically Significant ($p < .01$) Survey Differences Correctly Classified in Google News Word Embedding Model

	Sports	Food	Music	Occupations	Vehicles	Clothes	Names	All Domains
Gender	87.9%	88.2%	72.2%	93.6%	82.4%	74.4%	95.2%	84.8%
Class	96.3%	93.8%	88.9%	60.9%	94.1%	90.0%	77.3%	75.3%
Race	90.0%	68.8%	100%	51.5%	87.5%	55.0%	94.7%	69.1%

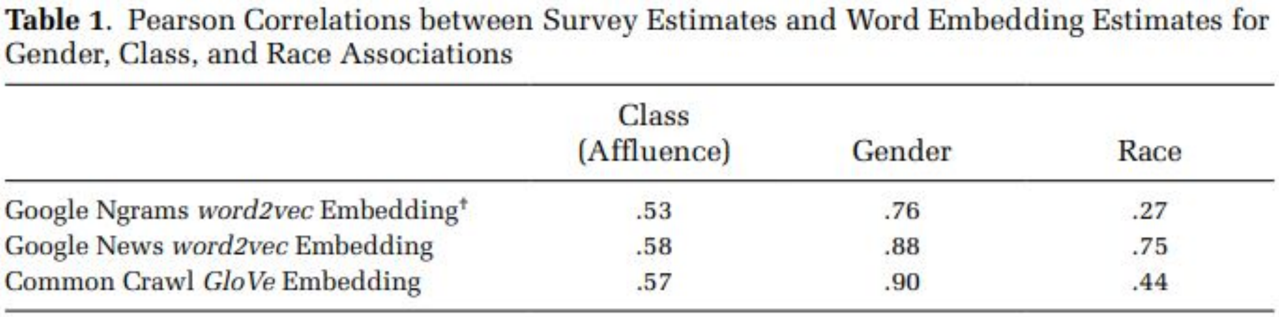


Table 1. Pearson Correlations between Survey Estimates and Word Embedding Estimates for Gender, Class, and Race Associations

	Class (Affluence)	Gender	Race
Google Ngrams *word2vec* Embedding†	.53	.76	.27
Google News *word2vec* Embedding	.58	.88	.75
Common Crawl *GloVe* Embedding	.57	.90	.44

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

Validation

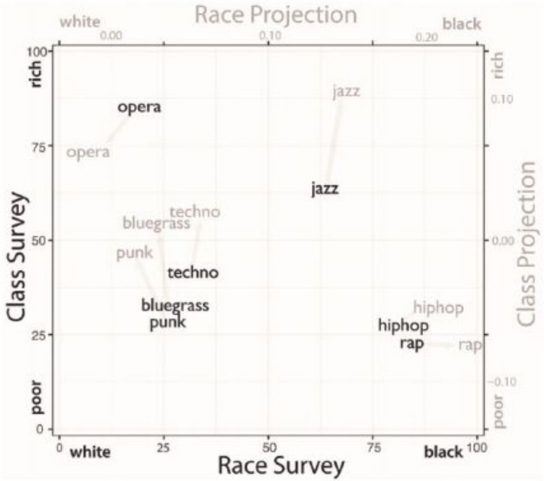


Figure 3. Projection of Music Genres onto Race and Class Dimensions of the Google News Word Embedding (Gray) and Average Survey Ratings for Race and Class Associations (Black)

Paper application □

Kozlowski et al. (2019) - The Geometry of Culture (and Class)

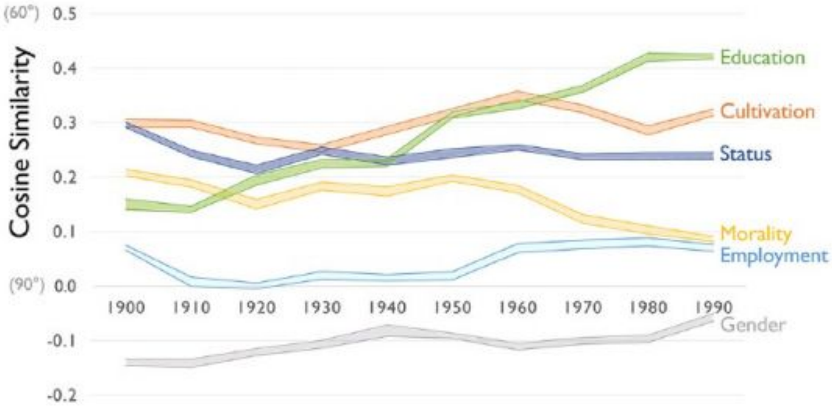


Figure 5. Cosine Similarity between the Affluence Dimension and Six Other Cultural Dimensions of Class by Decade; 1900 to 1999 Google Ngrams Corpus

Note: Bands represent 90 percent bootstrapped confidence intervals produced by subsampling.

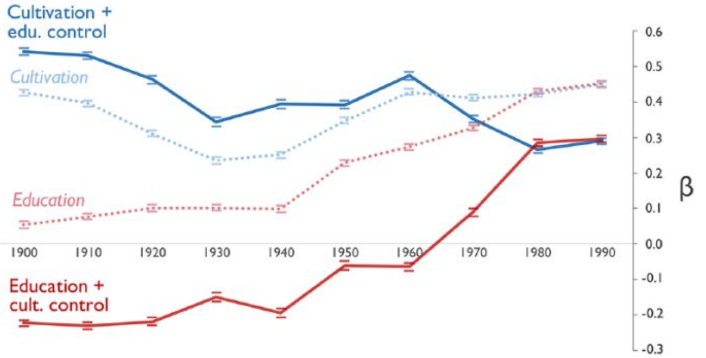


Figure 6. Standardized Coefficients from OLS Regression Models in Which Word Projections on Cultivation and Education Dimensions Predict Projection on the Affluence Dimension; 1900 to 1999 Google Ngrams Corpus

Note: A separate OLS regression model is fit for each decade; $N = 50{,}000$ most common words in each decade.

A paper not discussed, but a worth read: Demsky et al. (2019)