Using distributional models

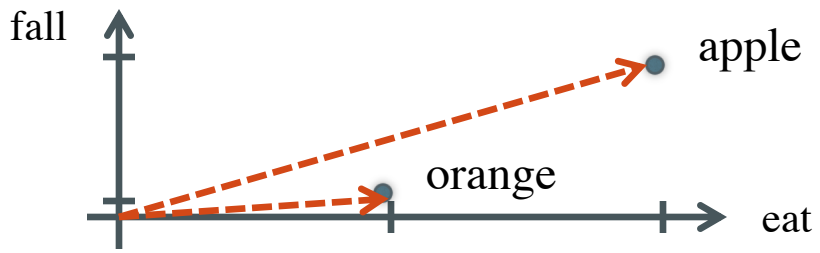
Katrin Erk

Trick 1:
Find the neighbors

Using a distributional model to "predict" similarity in meaning

"Apple" and "orange" as 2 vectors with the same dimension labels

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
794	244	47	221	208	160	145	156	109	104	88
eat	**fall**	**ripe**	**slice**	**peel**	**tree**	**throw**	**fruit**	**pie**	**bite**	**crab**
265	22	25	62	220	64	74	111	4	4	8

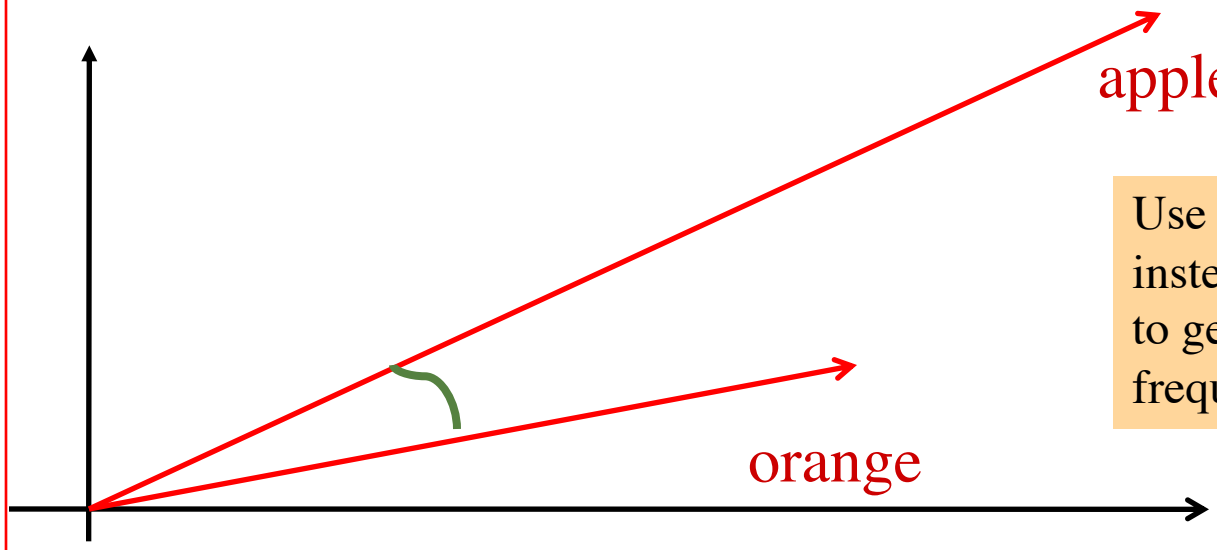


Similarity between two words as proximity in space: Use cosine

Cosine similarity

Cosine similarity:

$$cos(\vec{p}, \vec{q}) = \frac{\sum_{i=1}^{n} p_i \cdot q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}}$$



Use angle between vectors instead of point distance to get around word frequency issues

Interpreting a vector

- What does this vector for “apple” tell us?

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
794	244	47	221	208	160	145	156	109	104	88

- Best way to interpret it: By determining similarity to other vectors

Same corpus (BNC), different contexts (window sizes)

Nearest neighbours of *dog*

2-word window

- cat
- horse
- fox
- pet
- rabbit
- pig
- animal
- mongrel
- sheep
- pigeon

30-word window

- kennel
- puppy
- pet
- bitch
- terrier
- rottweiler
- canine
- cat
- to bark
- Alsatian

Using nearest neighbors: build a thesaurus

- Merriam-Webster:
  - **Synonyms** adversary, antagonist, foe, hostile, opponent
  - **Related Words** archenemy, archfoe, nemesis; ill-wisher; bane, bête noire; assailant, attacker, combatant, invader; competitor, emulator, rival
  - **Near Antonyms** buddy, chum, compadre, crony…
- Automatically generated thesaurus (Dekang Lin):
  - *adversary*: enemy, foe, ally, antagonist, opponent, rival, detractor, neighbor, supporter, competitor, partner, trading partner, accuser, terrorist, critic, Republican, advocate, skeptic, challenger

Using Lin's thesaurus in NLTK

```
import nltk

from nltk.corpus import lin_thesaurus

# The fileid = "simN.lsp" only checks for noun synonyms.

# You can also omit that parameter

 lin_thesaurus.synonyms("adversary", fileid = "simN.lsp")
```

Using nearest neighbors: Predicting human similarity ratings

- Collections of human similarity ratings for word pairs:
  - Rubenstein/Goodenough
  - WordSim353
  - MEN
  - SimLex
- WordSim353 examples:

Word1	Word2	Human_mean
love	sex	6.77
tiger	cat	7.35
tiger	tiger	10.00
book	paper	7.46
computer	keyboard	7.62
computer	internet	7.58
plane	car	5.77

Using nearest neighbors: Predicting human similarity ratings

- We can do pairwise cosine for pairs of words
- Cosine: between 0 and 1
- How to compare that to human ratings on a scale of 0-10, or 0-50?
- Compare rankings:
  - Rank word pairs by human similarity ratings, highest to lowest
  - Rank word pairs by cosine similarity ratings, highest to lowest
  - Compare the two rankings using a correlation measure from statistics, like Spearman's rho or Kendall's tau

Using nearest neighbors: Word associations

- Nelson association norms, http://w3.usf.edu/FreeAssociation/
- I give you a word, you write down what you associate with it.
  - "Participants were asked to write the first word that came to mind that was meaningfully related or strongly associated to the presented word on the blank shown next to each item. For example, if given BOOK _____, they might write READ on the blank next to it. "
- Why predict this? To show that a distributional model is a good proxy for the connections that humans perceive between words

Using nearest neighbors: The TOEFL test

- TOEFL: test of English as a foreign language
- Among other things, "find the near-synonym"

1. enormously
a. appropriately
b. uniquely
c. tremendously
d. decidedly

2. provisions
a. stipulations
b. interrelations
c. jurisdictions
d. interpretations

Using nearest neighbors: The TOEFL test

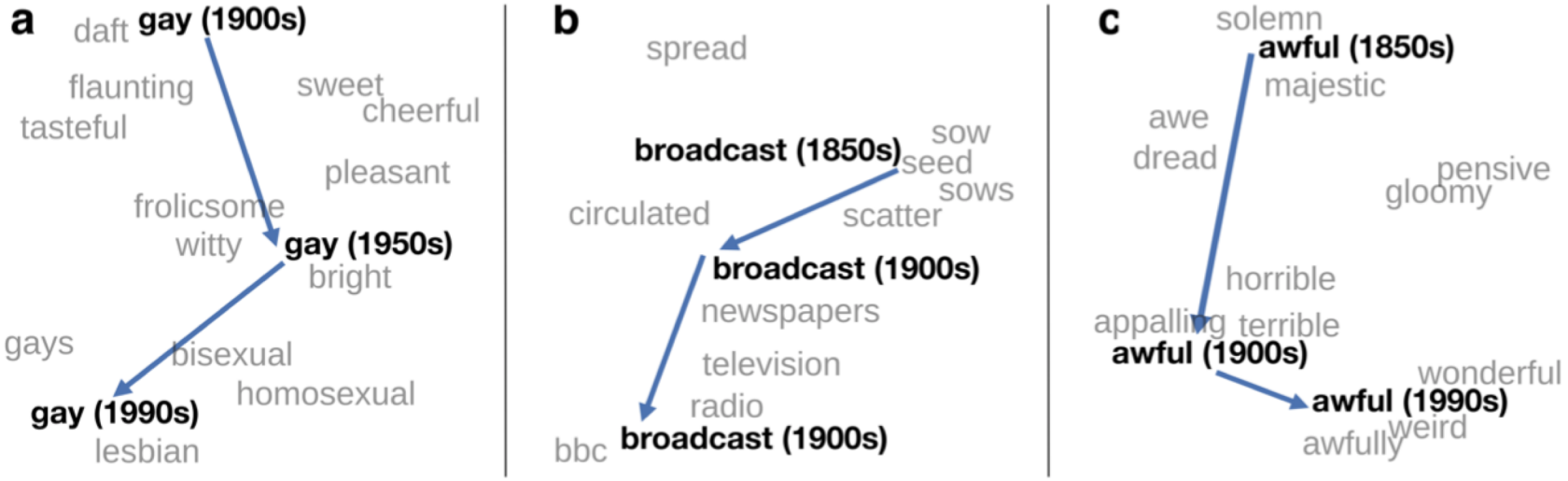
- How do you make your distributional model take the TOEFL test?
- Compute cosine similarity of "enormously" to "appropriately", "uniquely", "tremendously" etc.
  The entry with the highest cosine is the model's answer.

```
1.      enormously
a.      appropriately
b.      uniquely
c.      tremendously
d.      decidedly


2.      provisions
a.      stipulations
b.      interrelations
c.      jurisdictions
d.      interpretations
```

Using nearest neighbors: words over time

- Hamilton, Lescovec, Jurafsky 2016:
  - Compute separate vectors for "gay" for each decade
  - Look at nearest neighbors over time



Using nearest neighbors: words over time

- This is a bit tricky technically:
  - The simplest way to do this is to compute a separate space for every "time slice"
  - But then the dimensions have to be aligned across spaces using a method called "orthogonal Procrustes" (ouch)

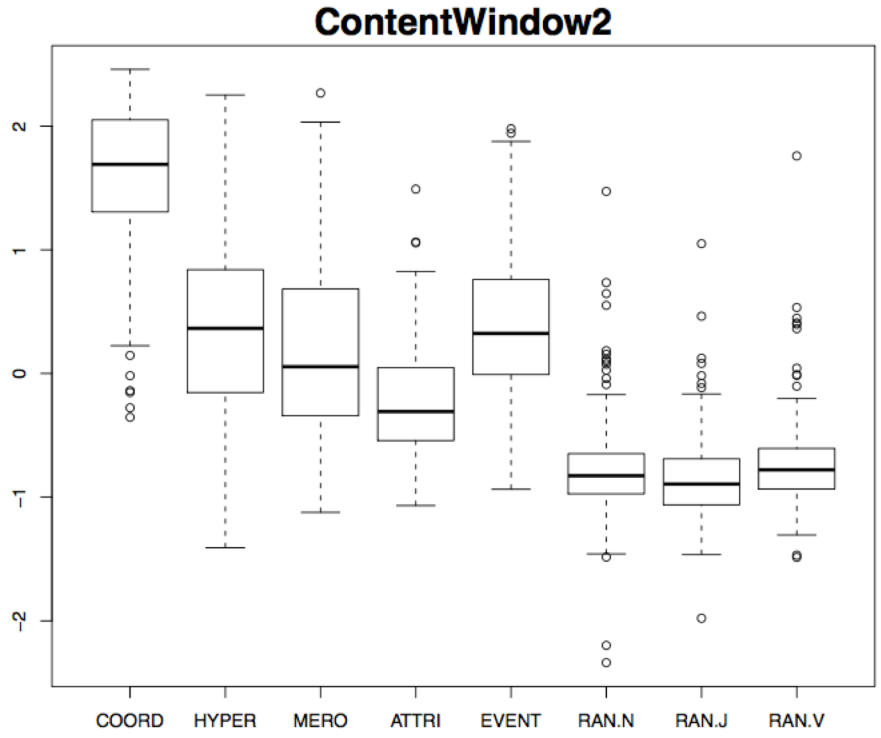
Using nearest neighbors: What semantic relations is this model focusing on?

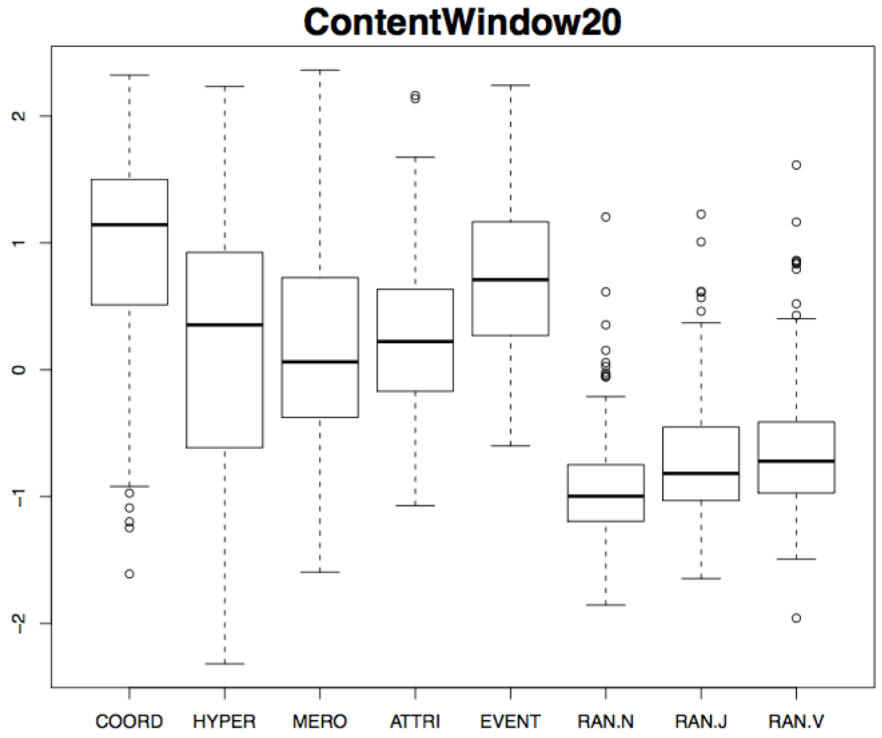
- Baroni and Lenci, "How we BLESSed distributional semantic evaluation", http://www.aclweb.org/anthology/W11-2501
- Data: pairs of words that stand in particular semantic relations

alligator-n	amphibian_reptile	attri scary-j
alligator-n	amphibian_reptile	coord crocodile-n
alligator-n	amphibian_reptile	coord frog-n
alligator-n	amphibian_reptile	event attack-v
alligator-n	amphibian_reptile	event bask-v
alligator-n	amphibian_reptile	hyper animal-n
alligator-n	amphibian_reptile	hyper beast-n
alligator-n	amphibian_reptile	mero eye-n
alligator-n	amphibian_reptile	mero foot-n
alligator-n	amphibian_reptile	random-j cardiac-j
alligator-n	amphibian_reptile	random-n addition-n
alligator-n	amphibian_reptile	random-v admire-v

Using nearest neighbors: What semantic relations is this model focusing on?

- Does this model give higher ratings to word pairs that are co-hyponyms, or event-related, or hyponym-hypernym?





Using nearest neighbors to predict properties

- Property datasets for example from psychology
- McRae et al feature norms:

Moose	is large	27
	has antlers	23
	has legs	14
	has four legs	12
	has fur	7
	has hair	5
	has hooves	5
	is brown	10
	hunted by people	17
	eaten as meat	5
	lives in woods	14
	lives in wilderness	8
	an animal	17

https://utsc.utoronto.ca/~gcree/McRae_etal_%282005%29_Semantic_feature_production_norms_%5BBRM%5D.pdf, https://sites.google.com/site/kenmcraelab/norms-data

Using nearest neighbors to predict properties

- Johns and Jones, 2012
- Say we don't know the properties of alligators
- Then we can simply determine the nearest neighbors of "alligator" in a distributional model
  - Say, "crocodile", "snake"
- How likely is an alligator to be green?
  - Depends on similarity to "crocodile" and whether crocodiles are green, and similarity to "snake" and whether snakes are green

Trick 2: Targets don’t have to be words

Targets can be multi-word expressions

- How similar are “tape” and “red tape”?
  - Treat “red tape” as a word with spaces inside
  - Use it as a target, and compute a vector for it as usual
  - Then we can compare the nearest neighbors for “tape” and “red tape”

Targets can be pieces of dependency structure

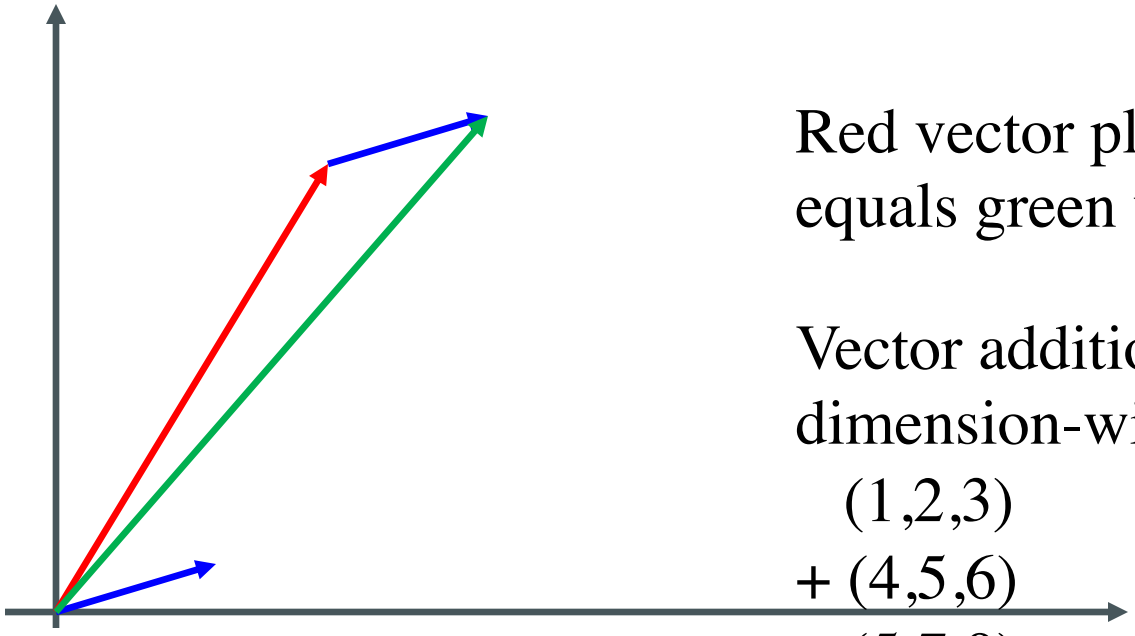
- Lin/Pantel 2001: Similarity not between words but between phrases:
  - X solves Y
  - X finds a solution to Y
- Similarity of X of "X solves Y", X of "X finds a solution to Y": similarity in fillers seen in the two X slots (idiosyncratic similarity measure)
- Similarity of the two phrases: average of their X-slot-similarity and Y-slot-similarity

Targets can be pieces of dependency structure

- Huge collection of more-or-less paraphrases
- **X manufactures Y**: X produces Y; X markets Y; X develops Y; X is supplier of Y; X ships Y; X supplies Y; Y is manufactured by X; X is maker of Y; X introduces Y; X exports Y; X makes Y; X builds Y; X's production of Y; X unveils Y; Y is bought from X; X's line of Y; X assembles Y; X is Y maker; X's Y factory; X's Y production; X is manufacturer of Y; X's Y division; X meets demand for Y; …

Trick 3: Adding and subtracting vectors

Adding vectors

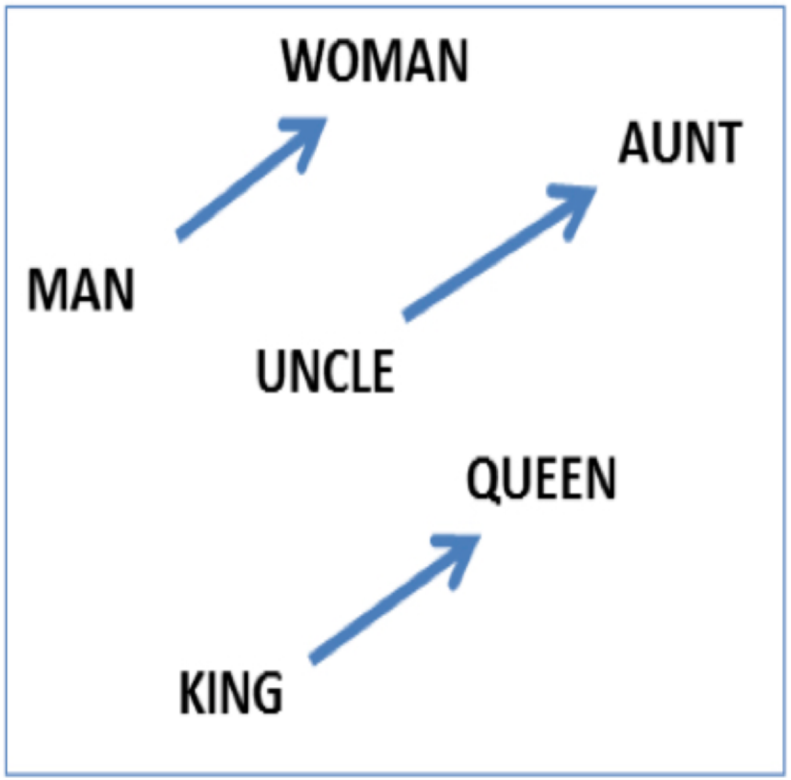


Red vector plus blue vector equals green vector

Vector addition:
dimension-wise addition

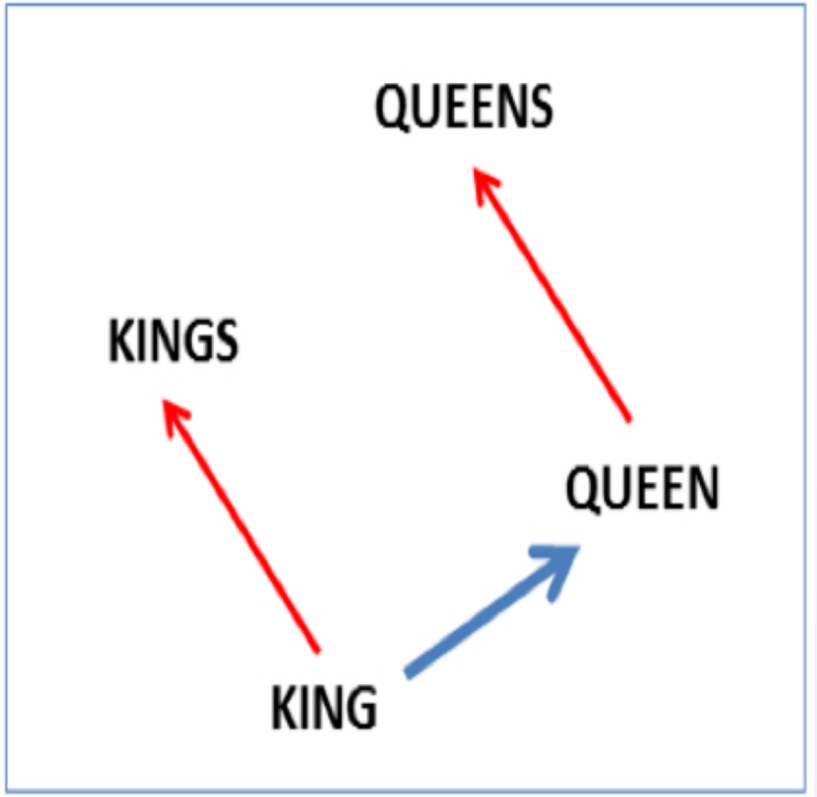
(1,2,3)
+ (4,5,6)
= (5,7,9)

Analogy as vector addition/subtraction



Semantic analogy:

King - man + woman ≈ queen



Syntactic analogy: from singular to plural

Pictures: Mikolov et al

Analogy as vector addition/subtraction

- Mikolov, Yih, Zweig: Linguistic regularities in continuous space word representations, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/rvecs.pdf
- Semantic and syntactic analogy
- Clothing is to shirt as dish is to what?
- y = vec(shirt) – vec(clothing) + vec(dish)
- Which word is the nearest neighbor of y?

Making phrase vectors by adding word vectors

- Schütze 1998: one vector per sentence that contains "letter"
  - She wrote an angry letter to her niece.
  - He sprayed the word in big letters.
  - The newspaper gets 100 letters from readers every day.
- Make sentence vector by adding up the vectors of all other (content) words in the sentence:

$$\vec{she} + \vec{wrote} + \vec{angry} + \vec{niece}$$

- Cluster sentence vectors
- Clusters = word senses

Making vectors for "the typical things that get eaten" by adding observed nouns that were direct objects of "eat"

- Selectional preferences:
  - Direct objects of "eat" tend to be foods
  - Subjects of "ponder" tend to be animate
  - Direct objects of "devein" pretty much have to be shrimp

Making vectors for "the typical things that get eaten" by adding observed nouns that were direct objects of "eat"

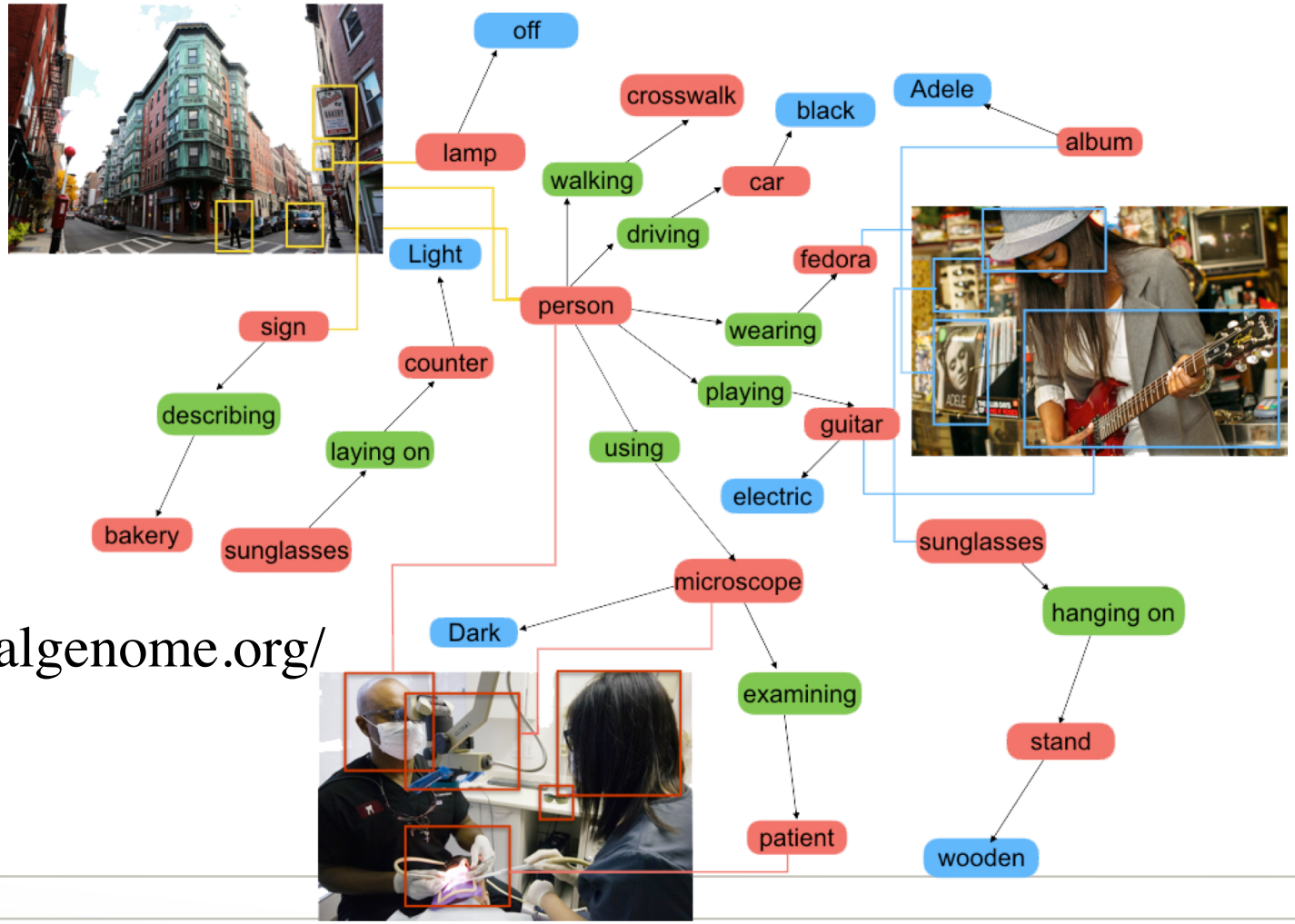
- Learning selectional preference for direct objects of "grate":
  - Make vectors for all words that appear as arguments
  - In a text, collect direct objects of "grate":
    - zest, rind, cheese, carrot, onion, parmesan, nutmeg, …
  - Selectional preference vector: average of "zest", "rind", "cheese", "carrot", …
- Is "car" a good direct object for "grate"?
  - sim(car, direct-obj-preference-of-grate)

Making vectors for "the typical things that get eaten" by adding observed nouns that were direct objects of "eat"

- Most typical direct objects of…
  - rebut: presumption, allegation, charge, criticism, claim
  - enunciate: principle, word, theory, philosophy, policy
  - break with: tradition, past, precedent, convention, Rome

Trick 4: Context doesn't have to be textual context

The Visual Genome

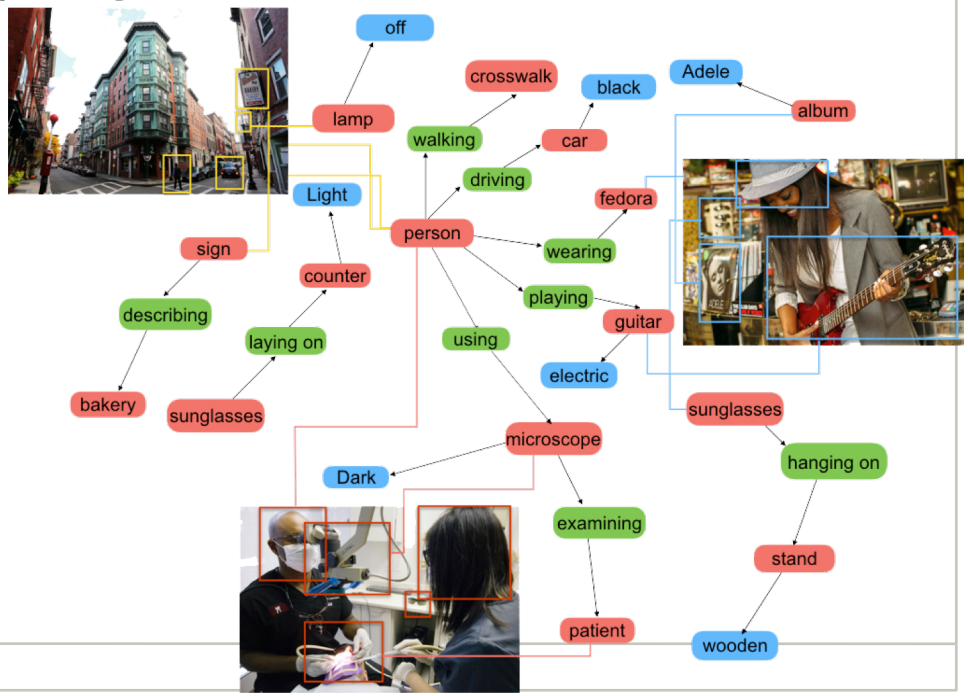


https://visualgenome.org/

The Visual Genome

The Visual Genome is a free dataset consisting of tons of images annotated with:

- Object labels, for example “guitar”
- Attributes, for example “electric” for “guitar”
- Relations, for example “person” “playing” “guitar”



https://visualgenome.org/

Creating a meaning space off the Visual Genome

- Herbelot 2020 “Re-solve it: simulating the acquisition of core semantic competences from small data”
- Targets: object labels in the Visual Genome
- Context items: attributes, relations, other object labels in the same image

Trick 5: Use the vectors as features for a classifier

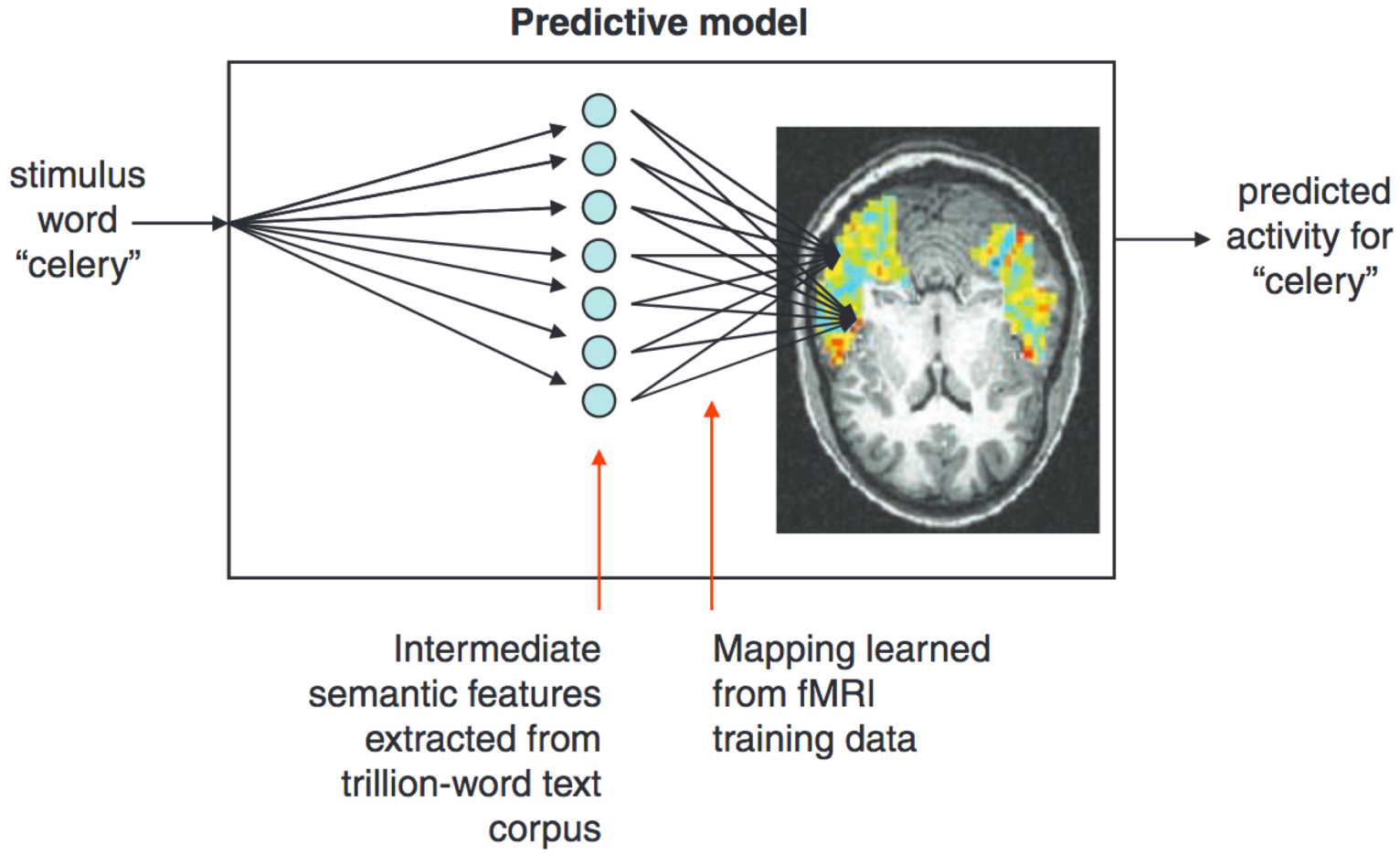
What is a classifier?

- We do this on the board…

What is a classifier, and where do vectors come into play?

- A classifier answers a question like “does this datapoint belong to class 1, 2, 3, or 4?”
- Each datapoint is represented to the classifier as a sequence of features
- **A vector can be used as a sequence of features**
  - Either on its own, or together with other features

Predicting human brain activity associated with the meanings of nouns. Mitchell et al, *Science*, 2008.



Predicting human brain activity associated with the meanings of nouns. Mitchell et al, *Science*, 2008.

- Given a word, say "celery"
- Map it to a vector in a distributional model
- Train a classifier that, given a vector, will say whether or not a particular "voxel" (brain pixel) will be active in an fMRI scan
- Assumptions of this model:
  - "… assumes the semantic features that distinguish the meanings of arbitrary concrete nouns are reflected in the statistics of their use within a very large text corpus."
  - "assumes that the brain activity observed when thinking about any concrete noun can be derived as a weighted linear sum of contributions from each of its semantic features."

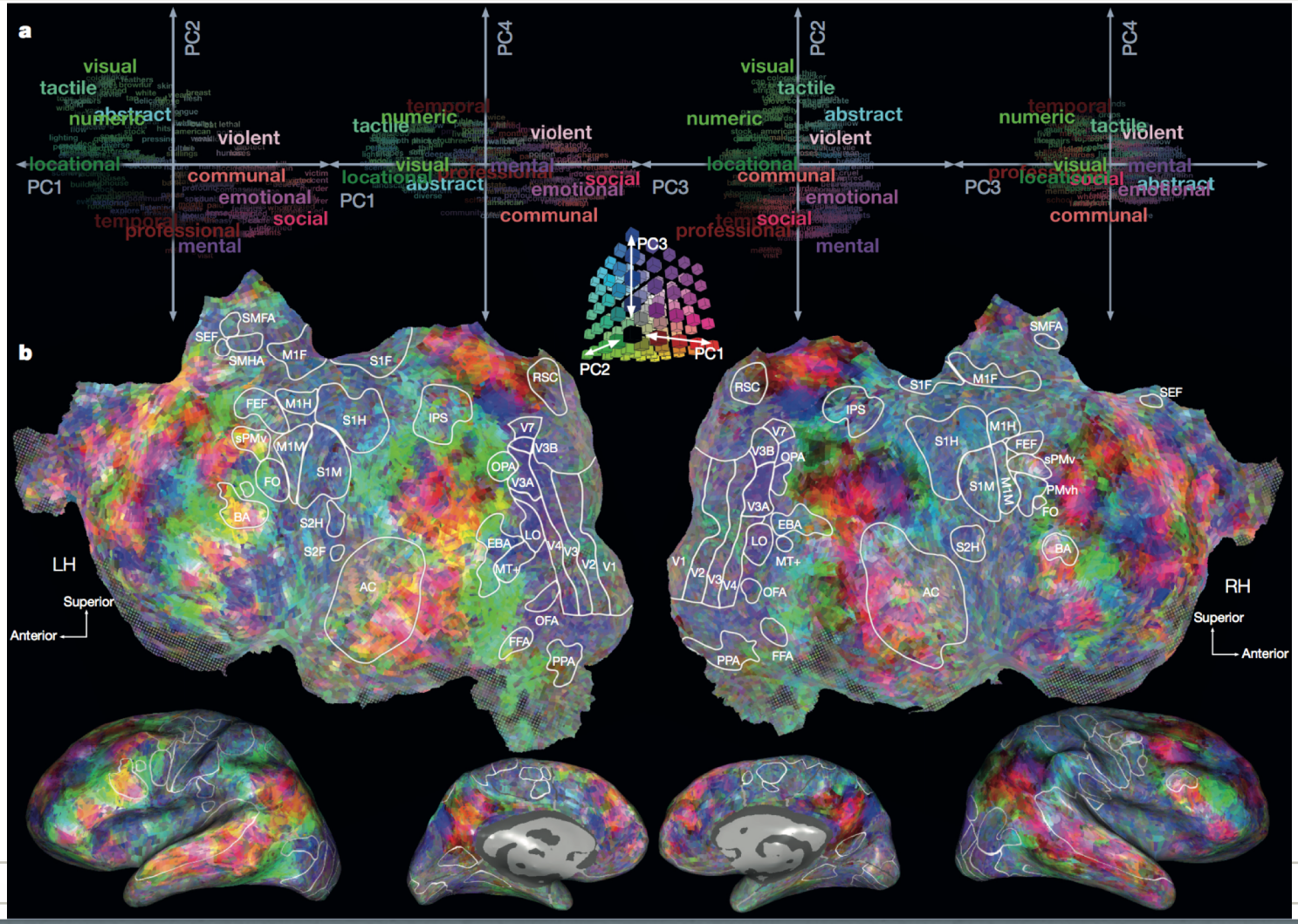
Predicting human brain activity associated with the meanings of nouns. Mitchell et al, *Science*, 2008.

- Here is the surprising bit:
  - "we designed a set of 25 semantic features defined by 25 verbs: "see," "hear," "listen," "taste," "smell," "eat," "touch," "rub," "lift," "manipulate," "run," "push," "fill," "move," "ride," "say," "fear," "open," "ap- proach," "near," "enter," "drive," "wear," "break," and "clean." These verbs generally correspond to basic sensory and motor activities, actions per- formed on objects, and actions involving changes to spatial relationships."
  - This worked relatively well: 0.77 accuracy on words unseen in training, within-participant prediction
  - This worked better than any other random set of 25 features

Natural speech reveals the semantic maps that tile the human cerebral cortex. Huth et al, *Nature*. 2016

- Language in context: People listened to the Moth Radio Hour
- Again, fMRI: which voxel is active at which point?
- Again, to predict whether a specific voxel will be active for, say, "celery", use distributional vector of "celery" as features
- Dimensionality reduction: which voxels tend to react to the same words?
- Then, represent words through "voxel group dimensions" instead of distributional dimensions
- Automatically group words by closeness in that space
- Inspect the word groups: What do they do?
- This yields a map of which voxels react to what kinds of words

Natural speech reveals the semantic maps that tile the human cerebral cortex.
Huth et al, *Nature*. 2016



Classifying word pairs: Are they hyponym/hypernym?

- Is “dog” a hyponym of “animal”? Is “dog” a hyponym of “furniture”?
- Train a classifier:
  - Input: distributional vectors of “dog”, “animal”
  - Output: yes/no
  - Also input other features depending on “dog”, “animal” vectors, for example vec(animal) – vec(dog)

Classify a word: Does an alligator have antlers?

- Concept properties, for example McRae et al feature norms
- Given the vector of a word, for example ”alligator”
- Train classifier: does this concept have the property “has_antlers”? Is it “eaten as meat”?

Moose	is large	27
	has antlers	23
	has legs	14
	has four legs	12
	has fur	7
	has hair	5
	has hooves	5
	is brown	10
	hunted by people	17
	eaten as meat	5
	lives in woods	14
	lives in wilderness	8
	an animal	17

Other tricks

Phrase vectors and component-wise multiplication (a.k.a the Hadamard product)

- Mitchell and Lapata, http://www.aclweb.org/anthology/P08-1028 and https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1551-6709.2010.01106.x
- Can we make a vector for a phrase from the vectors of its component words?
- They collected human ratings for phrase pairs

Use knowledge – exercise influence : 5
Develop technique – use power : 2
Support offer – provide help : 7
Left arm – elderly woman : 1
Industrial area – whole country : 4
Assistant manager – board member : 6
Office worker – kitchen door : 2

Phrase vectors and component-wise multiplication (a.k.a the Hadamard product)

- Can we make a vector for a phrase from the vectors of its component words?
- You already know: vector addition
  - Averaging over the contexts that are indicative of each component word
- They propose: componentwise vector multiplication

$$(1, 0, 2) \otimes (10, 20, 30) = (10, 0, 60)$$

  - What does this do?

Nouns are vectors, adjectives are matrices

- Baroni and Zamparelli, https://www.aclweb.org/anthology/D/D10/D10-1115.pdf
- “red” as a function that maps ”apple” to “red apple”: Given the set of apples, it picks out the red ones
- A matrix as a function that maps vectors to new vectors

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} = \begin{bmatrix} 1 \cdot 10 + 2 \cdot 20 + 3 \cdot 30 \\ 4 \cdot 10 + 5 \cdot 20 + 6 \cdot 30 \\ 7 \cdot 10 + 8 \cdot 20 + 9 \cdot 30 \end{bmatrix} = \begin{bmatrix} 140 \\ 320 \\ 500 \end{bmatrix}$$

Nouns are vectors, adjectives are matrices

- How to learn the adjective matrices?
- Compute a vector for "red apple" as a multi-word, as we discussed above (targets don't have to be words)
- Train the "red" matrix to map X vectors to "red X" vectors as closely as possible

- For more work on matrices and other higher-order tensors for phrase representations, see http://www.eecs.qmul.ac.uk/profiles/sadrzadehmehrnoosh.html

Vector projection, vector rejection

- Projecting a vector on an axis

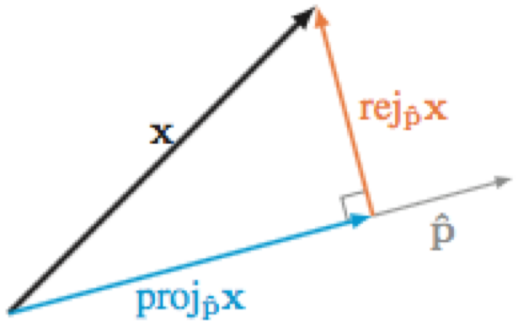


Figure 4.2: A vector $\hat{\mathbf{p}}$ is used to break $\mathbf{x}$ into two orthogonal components, its projection and the rejection over $\hat{\mathbf{p}}$.

Figure: Stephen Roller

Vector projection, vector rejection

- The "axis" can be an arbitrary other vector.
- Projection of vector x on axis p:

$$proj_p(x) = \frac{x \cdot p}{veclen(p)} \cdot p$$

where vector length is again $veclen(p) = \sqrt{\sum_i p_i \cdot p_i}$

- Rejection of x over p (removing direction p from x):

$$rej_p(x) = x - proj_p(x)$$

Vector projection, vector rejection

- Andrew Ng, "debiasing" a distributional model
  - Removing the gender/race/… component from a word
- Why?
  - People express their biases in their writing
  - Distributional models learn from people's writings
  - So distributional models pick up the biases too
- Check different professions: how close are their vectors to the vectors of man, of woman?

Vector projection, vector rejection

- "Debiasing" word embeddings:
- Bias vector estimated as b = vec(woman) – vec(man)
  - What does that mean? Compare to king-man+woman
- What are the nearest neighbors of b?
- How similar is b to the embeddings of typically "male-associated", "female-associated" terms?
- Then, debiasing the vector for a word w: use $\text{rej}_b(\text{vec}(w))$