# Algorithms for NLP

## Vector Semantics

Yulia Tsvetkov – CMU

Slides: Dan Jurafsky – Stanford,
David Bamman – UC Berkeley

# Neural LMs

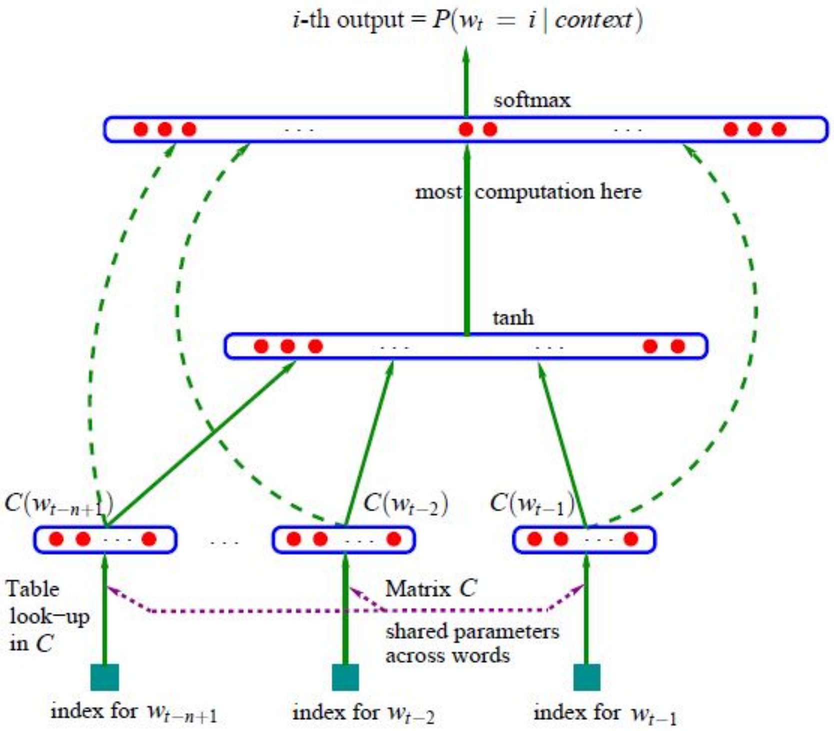
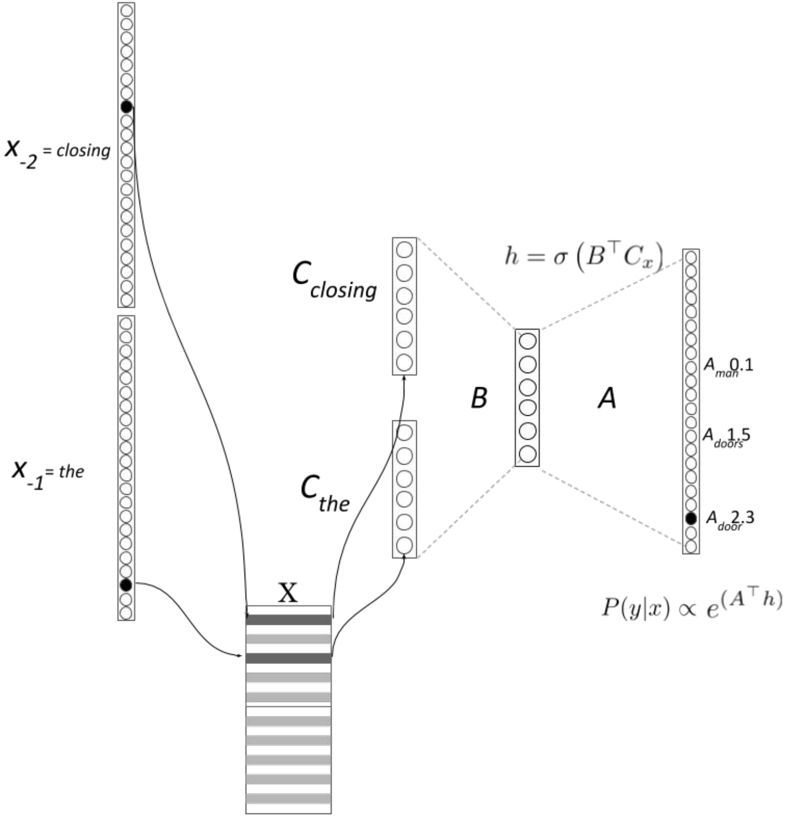
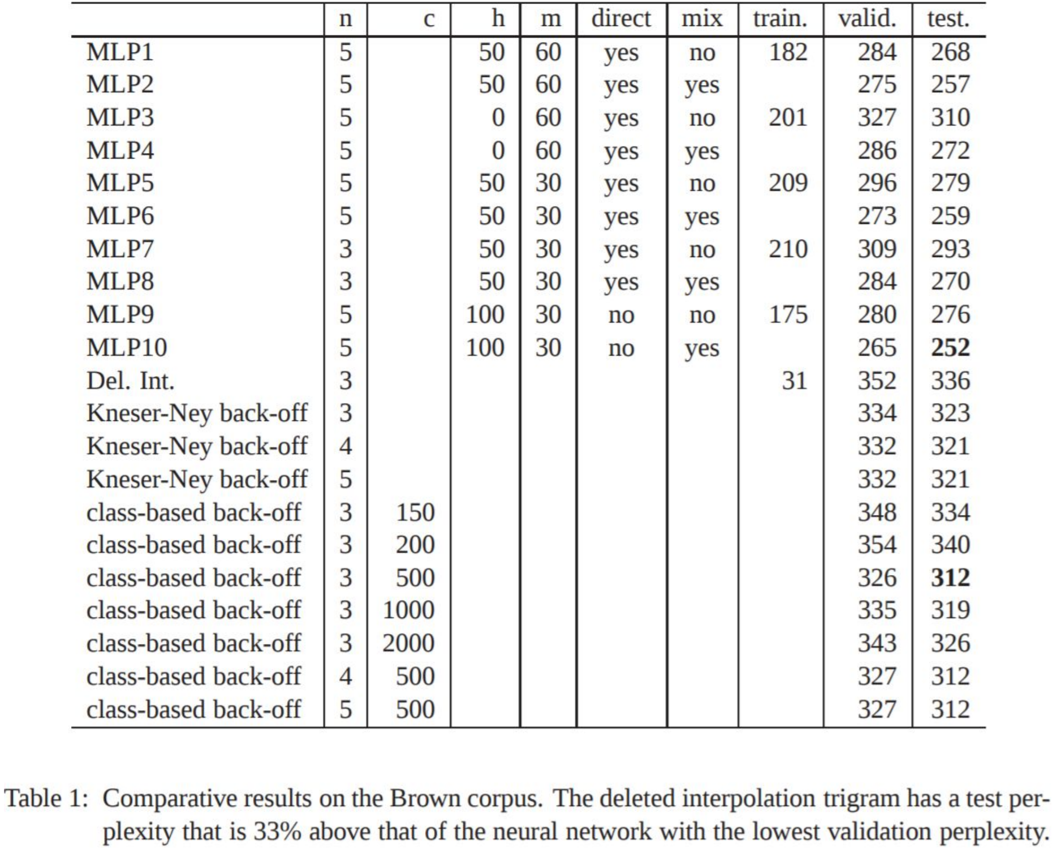


Image: (Bengio et al, 03)

# Neural LM Example



# Neural LMs



| | n | c | h | m | direct | mix | train. | valid. | test. |
|---|---|---|---|---|---|---|---|---|---|
| MLP1 | 5 | | 50 | 60 | yes | no | 182 | 284 | 268 |
| MLP2 | 5 | | 50 | 60 | yes | yes | | 275 | 257 |
| MLP3 | 5 | | 0 | 60 | yes | no | 201 | 327 | 310 |
| MLP4 | 5 | | 0 | 60 | yes | yes | | 286 | 272 |
| MLP5 | 5 | | 50 | 30 | yes | no | 209 | 296 | 279 |
| MLP6 | 5 | | 50 | 30 | yes | yes | | 273 | 259 |
| MLP7 | 3 | | 50 | 30 | yes | no | 210 | 309 | 293 |
| MLP8 | 3 | | 50 | 30 | yes | yes | | 284 | 270 |
| MLP9 | 5 | | 100 | 30 | no | no | 175 | 280 | 276 |
| MLP10 | 5 | | 100 | 30 | no | yes | | 265 | **252** |
| Del. Int. | 3 | | | | | | 31 | 352 | 336 |
| Kneser-Ney back-off | 3 | | | | | | | 334 | 323 |
| Kneser-Ney back-off | 4 | | | | | | | 332 | 321 |
| Kneser-Ney back-off | 5 | | | | | | | 332 | 321 |
| class-based back-off | 3 | 150 | | | | | | 348 | 334 |
| class-based back-off | 3 | 200 | | | | | | 354 | 340 |
| class-based back-off | 3 | 500 | | | | | | 326 | **312** |
| class-based back-off | 3 | 1000 | | | | | | 335 | 319 |
| class-based back-off | 3 | 2000 | | | | | | 343 | 326 |
| class-based back-off | 4 | 500 | | | | | | 327 | 312 |
| class-based back-off | 5 | 500 | | | | | | 327 | 312 |

Table 1: Comparative results on the Brown corpus. The deleted interpolation trigram has a test perplexity that is 33% above that of the neural network with the lowest validation perplexity.

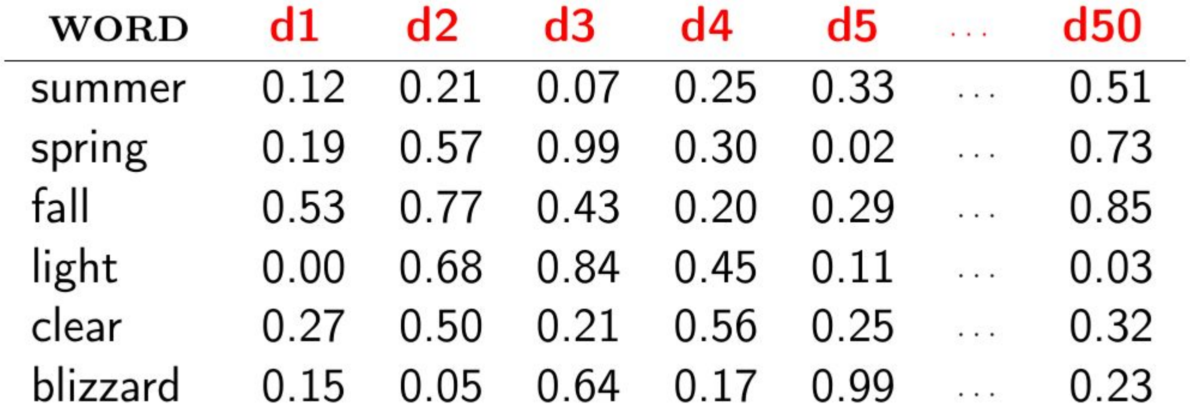
Image: (Bengio et al, 03)

# Low-dimensional Representations

- Learning representations by back-propagating errors
  - Rumelhart, Hinton & Williams, 1986
- A neural probabilistic language model
  - Bengio et al., 2003
- Natural Language Processing (almost) from scratch
  - Collobert & Weston, 2008
- Word representations: A simple and general method for semi-supervised learning
  - Turian et al., 2010
- Distributed Representations of Words and Phrases and their Compositionality
  - Word2Vec; Mikolov et al., 2013

# Today

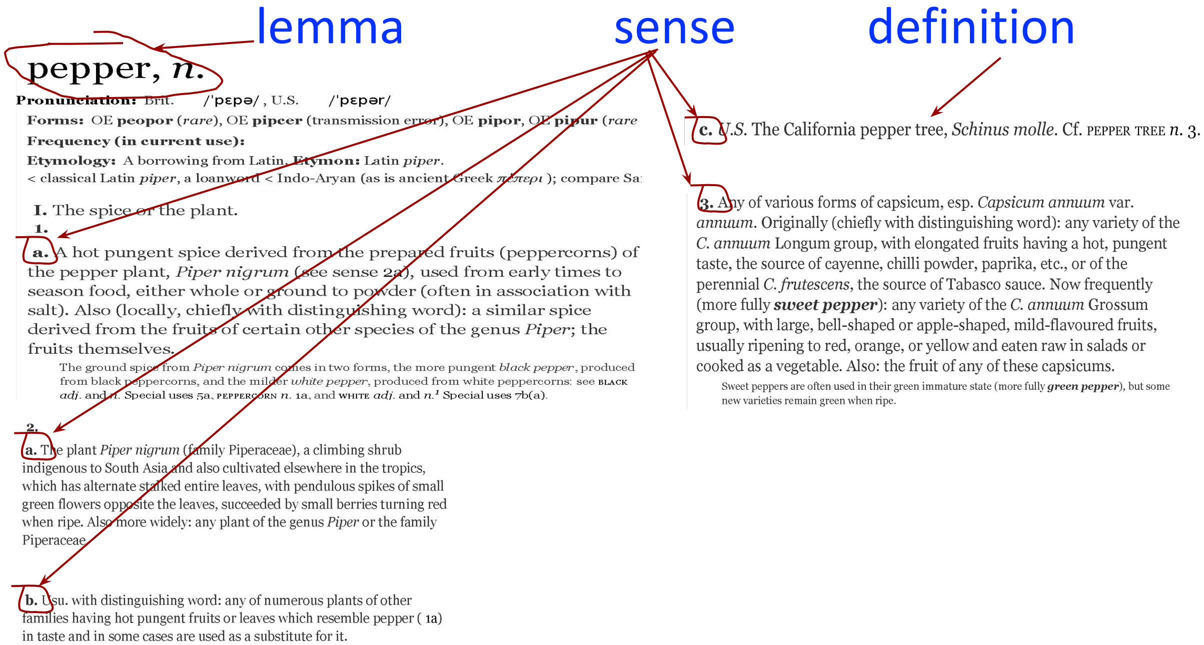
## Word Vectors



| WORD | d1 | d2 | d3 | d4 | d5 | ... | d50 |
|---|---|---|---|---|---|---|---|
| summer | 0.12 | 0.21 | 0.07 | 0.25 | 0.33 | ... | 0.51 |
| spring | 0.19 | 0.57 | 0.99 | 0.30 | 0.02 | ... | 0.73 |
| fall | 0.53 | 0.77 | 0.43 | 0.20 | 0.29 | ... | 0.85 |
| light | 0.00 | 0.68 | 0.84 | 0.45 | 0.11 | ... | 0.03 |
| clear | 0.27 | 0.50 | 0.21 | 0.56 | 0.25 | ... | 0.32 |
| blizzard | 0.15 | 0.05 | 0.64 | 0.17 | 0.99 | ... | 0.23 |

# Lexical Semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions



lemma

sense

definition

**pepper,** ***n.***

**Pronunciation:** Brit. /ˈpɛpə/ , U.S. /ˈpɛpər/

**Forms:** OE **peopor** (*rare*), OE **pipcer** (transmission error), OE **pipor**, OE **pipur** (*rare*

**Frequency (in current use):**

**Etymology:** A borrowing from Latin. **Etymon:** Latin *piper*.

< classical Latin *piper*, a loanword < Indo-Aryan (as is ancient Greek *πέπερι* ); compare Sa

**I.** The spice or the plant.

**1.**

**a.** A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

> The ground spice from *Piper nigrum* comes in two forms, the more pungent *black pepper*, produced from black peppercorns, and the milder *white pepper*, produced from white peppercorns: see BLACK *adj.* and *n.* Special uses 5a, PEPPERCORN *n.* 1a, and WHITE *adj.* and *n.*[1] Special uses 7b(a).

**2.**

**a.** The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate stalked entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

**b.** Usu. with distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper ( 1a) in taste and in some cases are used as a substitute for it.

**c.** *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.* 3.

**3.** Any of various forms of capsicum, esp. *Capsicum annuum* var. *annuum*. Originally (chiefly with distinguishing word): any variety of the *C. annuum* Longum group, with elongated fruits having a hot, pungent taste, the source of cayenne, chilli powder, paprika, etc., or of the perennial *C. frutescens*, the source of Tabasco sauce. Now frequently (more fully ***sweet pepper***): any variety of the *C. annuum* Grossum group, with large, bell-shaped or apple-shaped, mild-flavoured fruits, usually ripening to red, orange, or yellow and eaten raw in salads or cooked as a vegetable. Also: the fruit of any of these capsicums.

> Sweet peppers are often used in their green immature state (more fully ***green pepper***), but some new varieties remain green when ripe.

http://www.oed.com/

# Lemma pepper

- Sense 1:
  - spice from pepper plant
- Sense 2:
  - the pepper plant itself
- Sense 3:
  - another similar plant (Jamaican pepper)
- Sense 4:
  - another plant with peppercorns (California pepper)
- Sense 5:
  - capsicum (i.e. chili, paprika, bell pepper, etc)

A sense or “concept” is the meaning component of a word

# Lexical Semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions
  - Relationships between words or senses

# Relation: Synonymity

- Synonyms have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H20
- Note that there are probably no examples of perfect synonymy
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.

# Relation: Antonymy

Senses that are opposites with respect to one feature of meaning

- Otherwise, they are very similar!
  - dark/light   short/long   fast/slow rise/fall
  - hot/cold   up/down   in/out

More formally: antonyms can

- define a binary opposition or be at opposite ends of a scale
  - long/short, fast/slow
- be reversives:
  - rise/fall, up/down

# Relation: Similarity

Words with similar meanings.

- Not synonyms, but sharing some element of meaning
  - car, bicycle
  - cow, horse

# Ask humans how similar 2 words are

| word1 | word2 | similarity |
|---|---|---|
| vanish | disappear | 9.8 |
| behave | obey | 7.3 |
| belief | impression | 5.95 |
| muscle | bone | 3.65 |
| modest | flexible | 0.98 |
| hole | agreement | 0.3 |

SimLex-999 dataset (Hill et al., 2015)

# Relation: Word relatedness

Also called "word association"

- Words be related in any way, perhaps via a semantic frame or field
  - car, bicycle: **similar**
  - car, gasoline: **related**, not similar

# Semantic field

Words that

- cover a particular semantic domain
- bear structured relations with each other.

**hospitals**

surgeon, scalpel, nurse, anaesthetic, hospital

**restaurants**

waiter, menu, plate, food, menu, chef),

**houses**

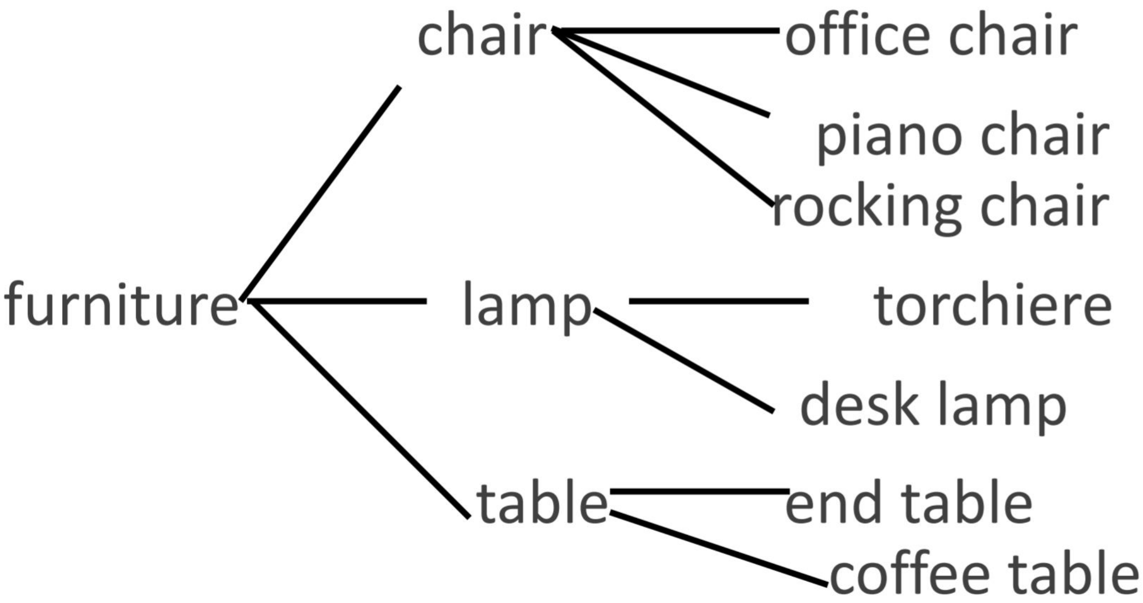
door, roof, kitchen, family, bed

# Relation: Superordinate/ Subordinate

- One sense is a subordinate of another if the first sense is more specific, denoting a subclass of the other
  - car is a subordinate of vehicle
  - mango is a subordinate of fruit
- Conversely superordinate
  - vehicle is a superordinate of car
  - fruit is a subodinate of mango

# Taxonomy

**Superordinate** **Basic** **Subordinate**



chair

office chair

piano chair

rocking chair

furniture

lamp

torchiere

desk lamp

table

end table

coffee table

# Lexical Semantics

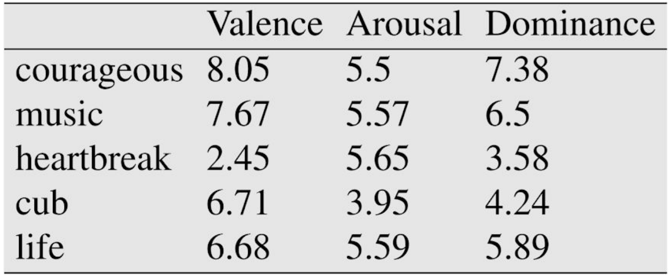
- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness

# Lexical Semantics

- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
    - *John hit Bill*
    - *Bill was hit by John*

# Lexical Semantics

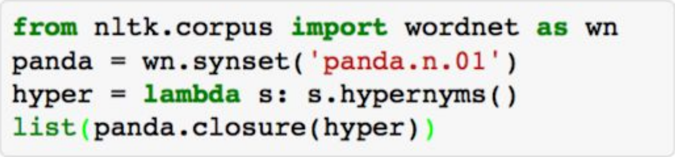
- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
  - Connotation and sentiment
    - *valence*: the pleasantness of the stimulus
    - *arousal*: the intensity of emotion
    - *dominance*: the degree of control exerted by the stimulus



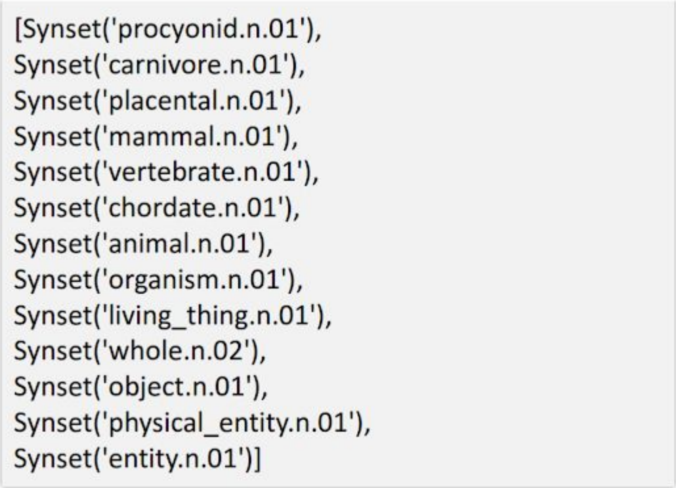
| | Valence | Arousal | Dominance |
|---|---|---|---|
| courageous | 8.05 | 5.5 | 7.38 |
| music | 7.67 | 5.57 | 6.5 |
| heartbreak | 2.45 | 5.65 | 3.58 |
| cub | 6.71 | 3.95 | 4.24 |
| life | 6.68 | 5.59 | 5.89 |

# Electronic Dictionaries

## WordNet

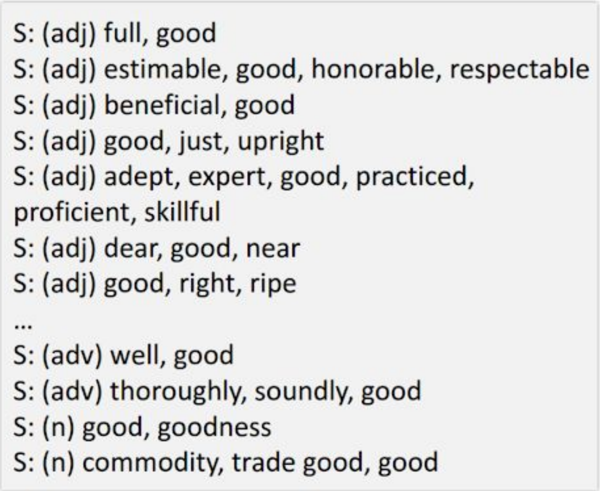


```
from nltk.corpus import wordnet as wn
panda = wn.synset('panda.n.01')
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```



```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

(here, for *good*):



S: (adj) full, good
S: (adj) estimable, good, honorable, respectable
S: (adj) beneficial, good
S: (adj) good, just, upright
S: (adj) adept, expert, good, practiced, proficient, skillful
S: (adj) dear, good, near
S: (adj) good, right, ripe
...
S: (adv) well, good
S: (adv) thoroughly, soundly, good
S: (n) good, goodness
S: (n) commodity, trade good, good

Slide: Chris Manning

# Problems with Discrete Representations

- Too coarse
  - *expert* ↔ *skillful*
- Sparse
  - *wicked, badass, ninja*
- Subjective
- Expensive
- Hard to compute word relationships

*expert* [0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0]

*skillful* [0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0]

# Distributional Hypothesis

“The meaning of a word is its use in the language”

[Wittgenstein PI 43]

“You shall know a word by the company it keeps”

[Firth 1957]

If A and B have almost identical environments we say that they are synonyms.

[Harris 1954]

# Example

What does ongchoi mean?

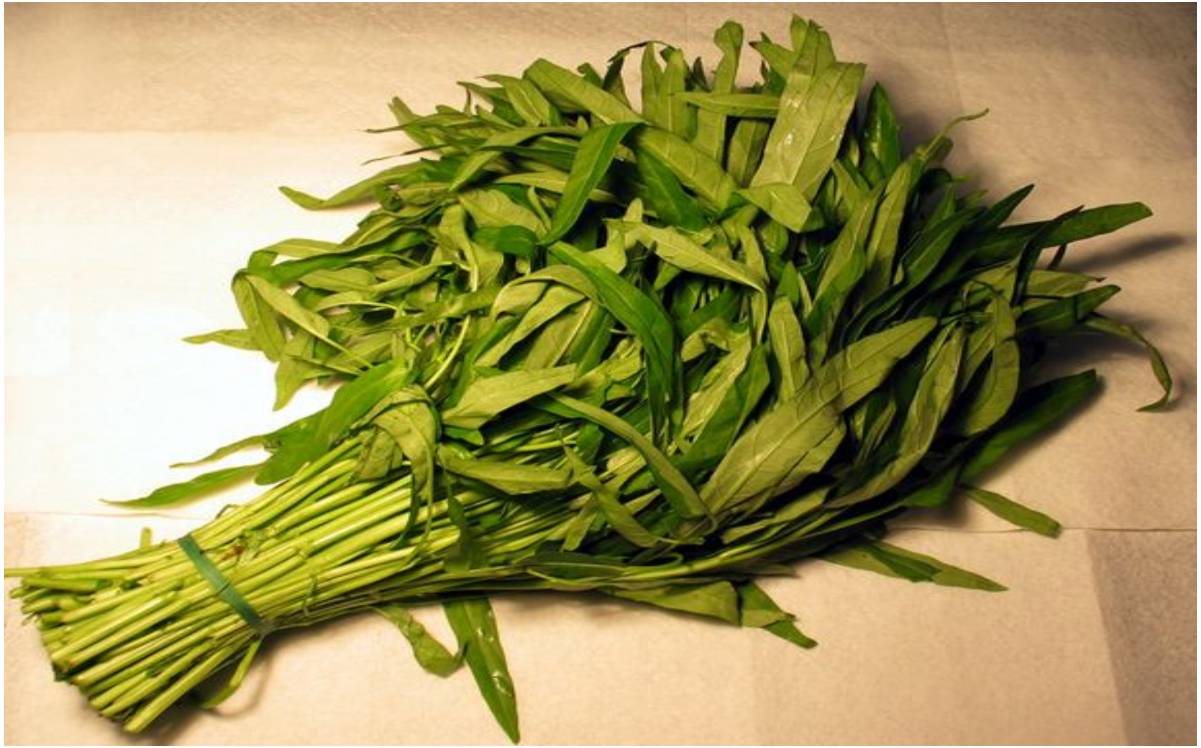
- Suppose you see these sentences:
  - Ongchoi is delicious **sautéed with garlic**.
  - Ongchoi is superb **over rice**
  - Ongchoi **leaves** with salty sauces

- And you've also seen these:
  - …spinach sautéed with garlic over rice
  - Chard stems and leaves are delicious
  - Collard greens and other **salty** leafy greens

Conclusion:

- Ongchoi is a leafy green like spinach, chard, or collard greens

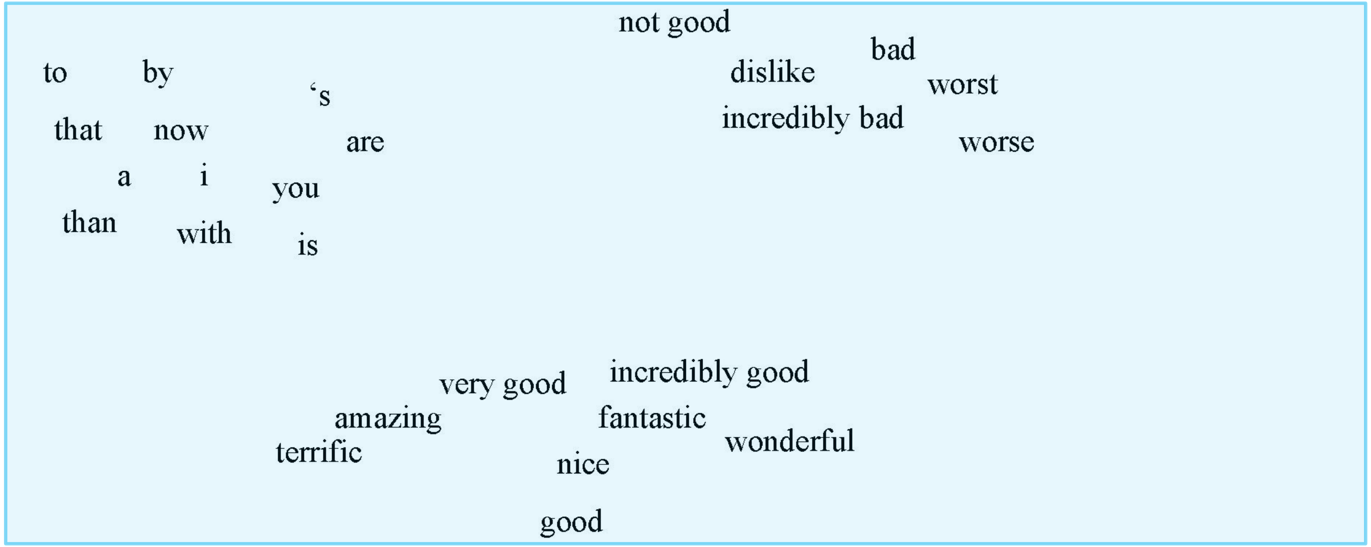
# Ongchoi: Ipomoea aquatica "Water Spinach"



Yamaguchi, Wikimedia Commons, public domain

# Model of Meaning Focusing on Similarity

- Each word = a vector
    - not just “word” or word45.
    - similar words are “nearby in space”
    - the standard way to represent meaning in NLP



# We'll Introduce 3 Kinds of Embeddings

- Count-based
  - Words are represented by a simple function of the counts of nearby words
- Brown clusters
  - Representation is created through hierarchical clustering
- Word2Vec
  - Representation is created by training a classifier to distinguish nearby and far-away words

Next class:

- Fasttext
- ELMO
- Multilingual embeddings

# Term-Document Matrix

| | As You Like It | Twelfth Night | Julius  Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

Context = appearing in the same document.

# Term-Document Matrix

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

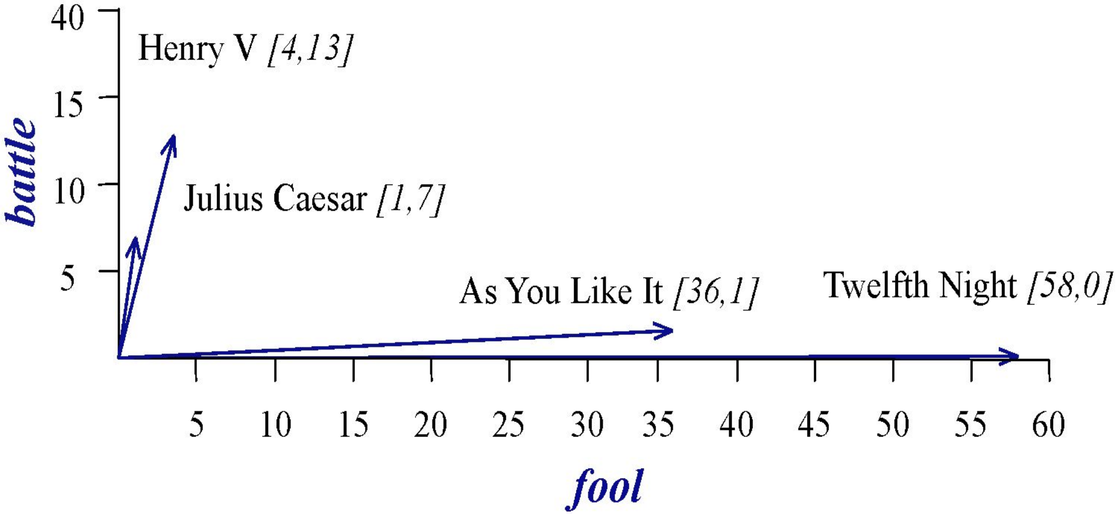
Each document is represented by a vector of words

# Vectors are the Basis of Information Retrieval

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

- Vectors are similar for the two comedies
- Different than the history
- Comedies have more fools and wit and fewer battles.

# Visualizing Document Vectors



# Words Can Be Vectors Too

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

- battle is "the kind of word that occurs in Julius Caesar and Henry V"
- fool is "the kind of word that occurs in comedies, especially Twelfth Night"

# Term-Context Matrix

| | knife | dog | sword | love | like |
|---|---|---|---|---|---|
| knife | 0 | 1 | 6 | 5 | 5 |
| dog | 1 | 0 | 5 | 5 | 5 |
| sword | 6 | 5 | 0 | 5 | 5 |
| love | 5 | 5 | 5 | 0 | 5 |
| like | 5 | 5 | 5 | 5 | 2 |

- Two words are "similar" in meaning if their context vectors are similar
  - Similarity == relatedness

# Count-Based Representations

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

- Counts: term-frequency
  - remove stop words
  - use $\log_{10}$(tf)
  - normalize by document length

# TF-IDF

- What to do with words that are evenly distributed across many documents?

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{idf}_i = \log\left(\frac{N}{\mathrm{df}_i}\right)$$

Total # of  docs in collection

# of  docs that have word i

Words like "the" or "good" have very low idf

$$w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

# Positive Pointwise Mutual Information (PPMI)

- In word--context matrix
- Do words *w* and *c* co-occur more than if they were independent?

$$\mathrm{PPMI}_{\alpha}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_{\alpha}(c)},0)$$

$$P_{\alpha}(c) = \frac{count(c)^{\alpha}}{\sum_{c} count(c)^{\alpha}}$$

- PMI is biased toward infrequent events
  - Very rare words have very high PMI values
  - Give rare words slightly higher probabilities α=0.75

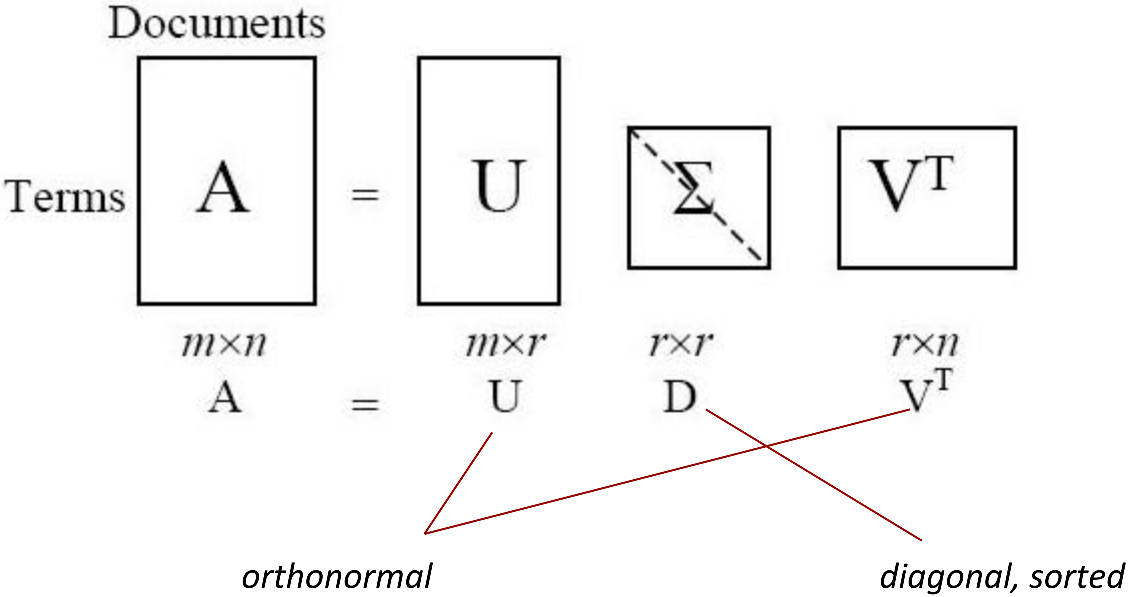
# Dimensionality Reduction

- Wikipedia: ~29 million English documents. Vocab: ~1M words.
  - High dimensionality of word--document matrix
  - Sparsity
  - The order of rows and columns doesn’t matter
- Goal:
  - good similarity measure for words or documents
  - dense representation
- Sparse vs Dense vectors
  - Short vectors may be easier to use as features in machine learning (less weights to tune)
  - Dense vectors may generalize better than storing explicit counts
  - They may do better at capturing synonymy
  - In practice, they work better

| | |
|---|---|
| A | 0 |
| a | 0 |
| aa | 0 |
| aal | 0 |
| aalii | 0 |
| aam | 0 |
| Aani | 0 |
| **aardvark** | 1 |
| aardwolf | 0 |
| ... | 0 |
| zymotoxic | 0 |
| zymurgy | 0 |
| Zyrenian | 0 |
| Zyrian | 0 |
| Zyryan | 0 |
| zythem | 0 |
| Zythia | 0 |
| zythum | 0 |
| Zyzomys | 0 |
| Zyzzogeton | 0 |

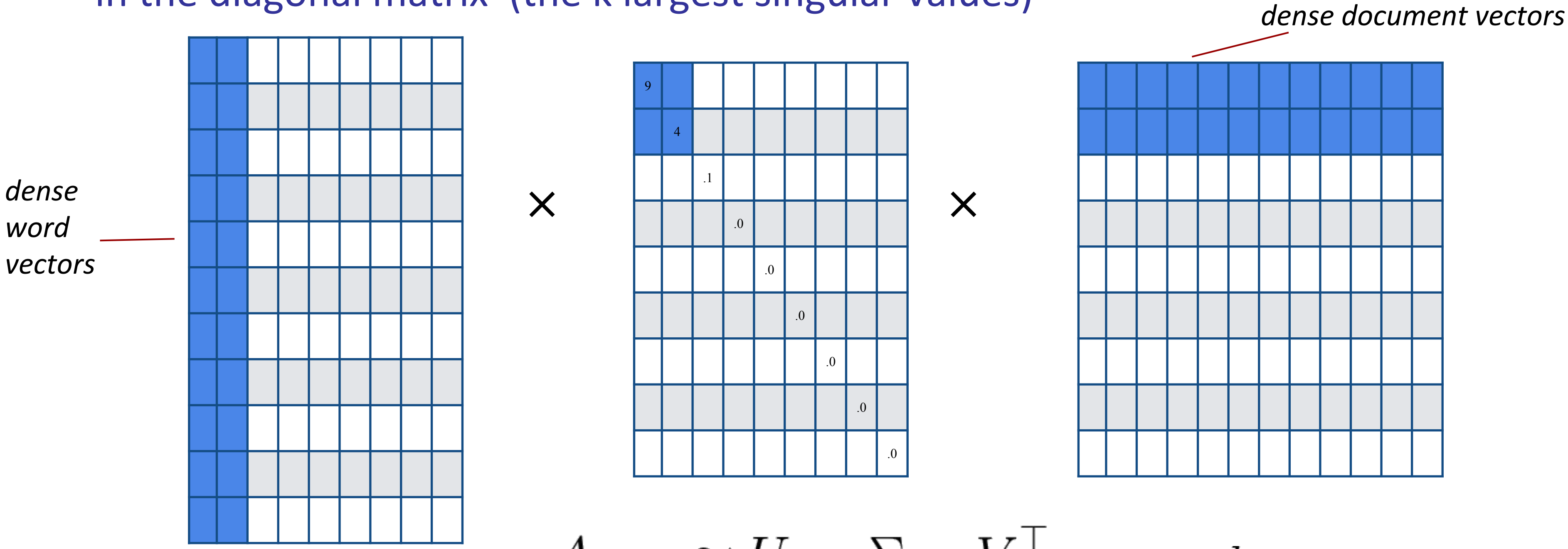
# Singular Value Decomposition (SVD)

- Solution idea:
  - Find a projection into a low-dimensional space (~300 dim)
  - That gives us a best separation between features



# Truncated SVD

We can approximate the full matrix by only considering the leftmost k terms in the diagonal matrix (the k largest singular values)



$$A_{m \times n} \approx U_{m \times k} \Sigma_{k \times k} V_{k \times n}^{\top}$$

$$k \ll m, n$$

# Latent Semantic Analysis

| #0 | #1 | #2 | #3 | #4 | #5 |
|---|---|---|---|---|---|
| we | music | company | how | program | 10 |
| said | film | mr | what | project | 30 |
| have | theater | its | about | russian | 11 |
| they | mr | inc | their | space | 12 |
| not | this | stock | or | russia | 15 |
| but | who | companies | this | center | 13 |
| be | movie | sales | are | programs | 14 |
| do | which | shares | history | clark | 20 |
| he | show | said | be | aircraft | sept |
| this | about | business | social | ballet | 16 |
| there | dance | share | these | its | 25 |
| you | its | chief | other | projects | 17 |
| are | disney | executive | research | orchestra | 18 |
| what | play | president | writes | development | 19 |
| if | production | group | language | work | 21 |

[Deerwester et al.,  1990]

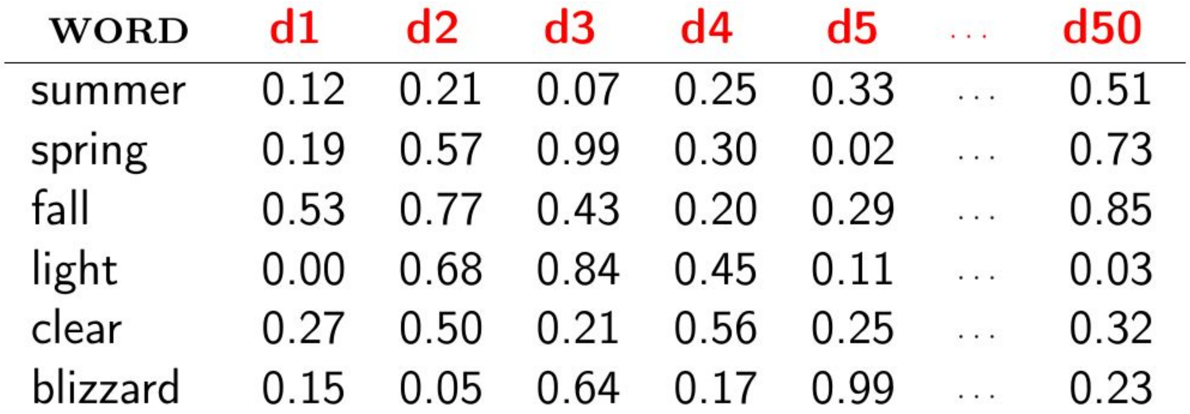
# LSA++

- Probabilistic Latent Semantic Indexing (PLSI)
  - Hofmann, 1999
- Latent Dirichlet Allocation (LDA)
  - Blei et al., 2003
- Nonnegative Matrix Factorization (NMF)
  - Lee & Seung, 1999

# Word Similarity

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

# Evaluation



| WORD | d1 | d2 | d3 | d4 | d5 | ... | d50 |
|---|---|---|---|---|---|---|---|
| summer | 0.12 | 0.21 | 0.07 | 0.25 | 0.33 | ... | 0.51 |
| spring | 0.19 | 0.57 | 0.99 | 0.30 | 0.02 | ... | 0.73 |
| fall | 0.53 | 0.77 | 0.43 | 0.20 | 0.29 | ... | 0.85 |
| light | 0.00 | 0.68 | 0.84 | 0.45 | 0.11 | ... | 0.03 |
| clear | 0.27 | 0.50 | 0.21 | 0.56 | 0.25 | ... | 0.32 |
| blizzard | 0.15 | 0.05 | 0.64 | 0.17 | 0.99 | ... | 0.23 |

- Intrinsic
- Extrinsic
- Qualitative

# Intrinsic Evaluation

| word1 | word2 | similarity (humans) |
|---|---|---|
| vanish | disappear | 9.8 |
| behave | obey | 7.3 |
| belief | impression | 5.95 |
| muscle | bone | 3.65 |
| modest | flexible | 0.98 |
| hole | agreement | 0.3 |

| similarity (embeddings) |
|---|
| 1.1 |
| 0.5 |
| 0.3 |
| 1.7 |
| 0.98 |
| 0.3 |

Spearman's rho (human ranks, model ranks)

- WS-353 (Finkelstein et al. ‘02)
- MEN-3k (Bruni et al. ‘12)
- SimLex-999 dataset (Hill et al., 2015)

# Extrinsic Evaluation

- Chunking
- POS tagging
- Parsing
- MT
- SRL
- Topic categorization
- Sentiment analysis
- Metaphor detection
- etc.
-

# Visualisation

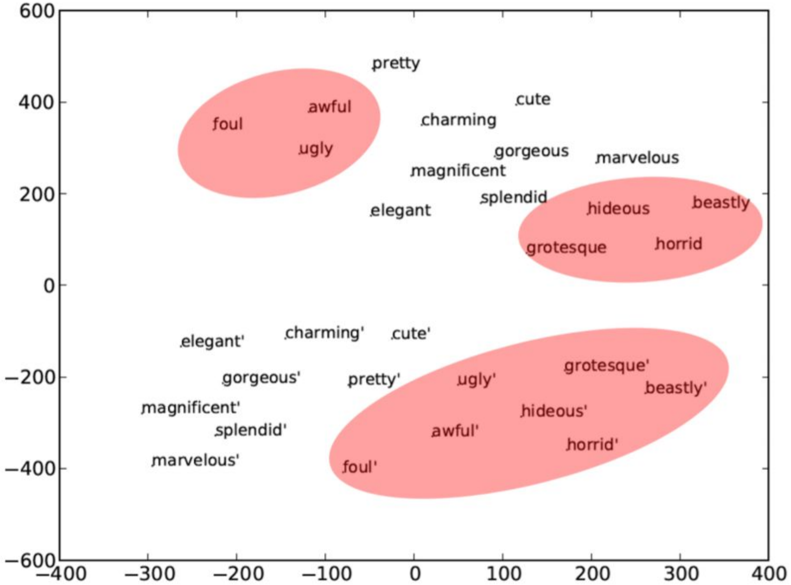


Figure 6.5: Monolingual (top) and multilingual (bottom; marked with apostrophe) word projections of the antonyms (shown in red) and synonyms of "beautiful".

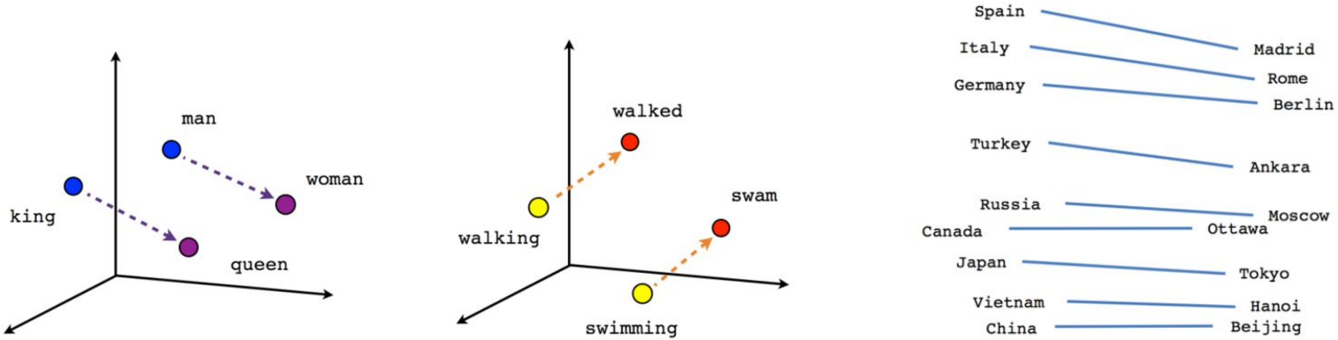
[Faruqui et al., 2015]

- Visualizing Data using t-SNE (van der Maaten & Hinton'08)

# Analogy: Embeddings capture relational meaning!

vector(*‘king’*) - vector(*‘man’*) + vector(*‘woman’*) ≈ vector(‘queen’)

vector(*‘Paris’*) - vector(*‘France’*) + vector(*‘Italy’*) ≈ vector(‘Rome’)



Male-Female

Verb tense

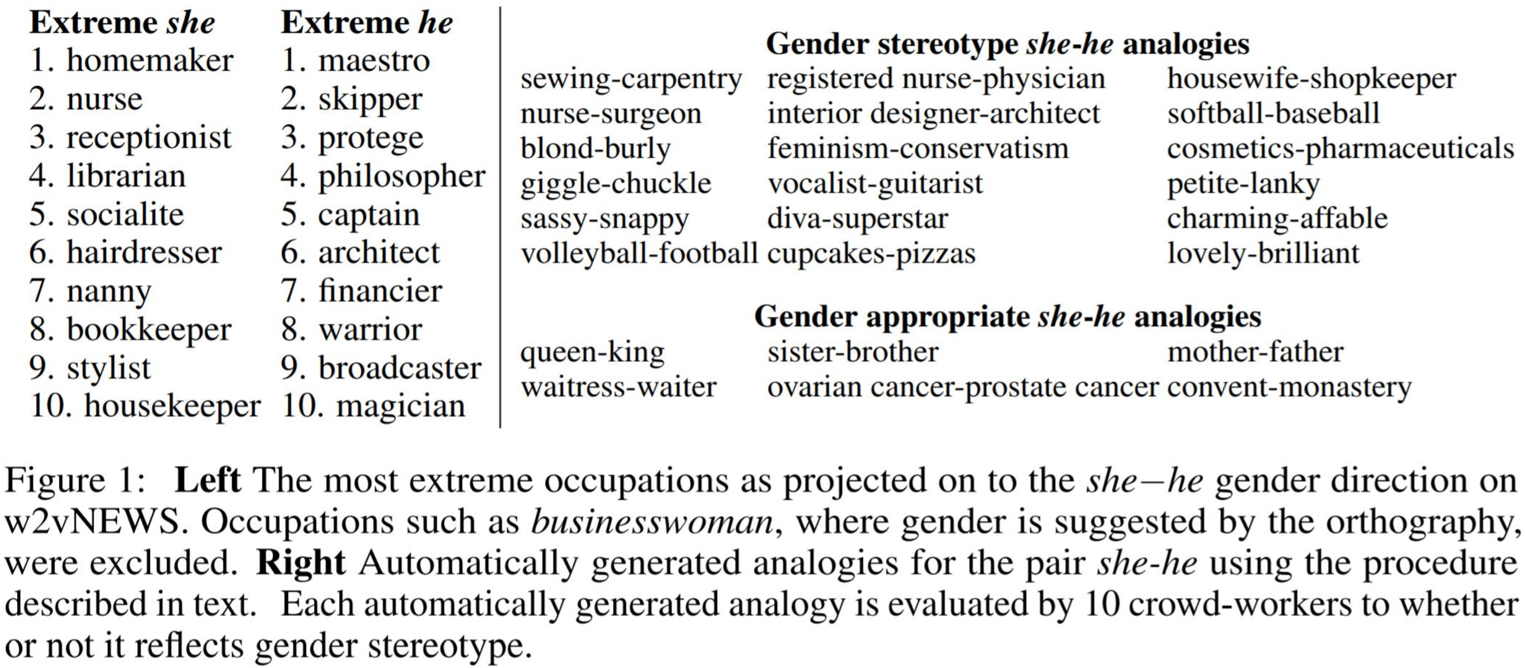
Country-Capital

$$\min cos(man - woman,\ king - x)\ s.t.\ ||king - x||_2 < \delta$$

[Mikolov et al.’ 13]

# and also human biases

$$\min cos(he - she,\ x - y)\ s.t.\ ||x - y||_2 < \delta$$



| **Extreme *she*** | **Extreme *he*** |
|---|---|
| 1. homemaker | 1. maestro |
| 2. nurse | 2. skipper |
| 3. receptionist | 3. protege |
| 4. librarian | 4. philosopher |
| 5. socialite | 5. captain |
| 6. hairdresser | 6. architect |
| 7. nanny | 7. financier |
| 8. bookkeeper | 8. warrior |
| 9. stylist | 9. broadcaster |
| 10. housekeeper | 10. magician |

**Gender stereotype *she-he* analogies**

| | | |
|---|---|---|
| sewing-carpentry | registered nurse-physician | housewife-shopkeeper |
| nurse-surgeon | interior designer-architect | softball-baseball |
| blond-burly | feminism-conservatism | cosmetics-pharmaceuticals |
| giggle-chuckle | vocalist-guitarist | petite-lanky |
| sassy-snappy | diva-superstar | charming-affable |
| volleyball-football | cupcakes-pizzas | lovely-brilliant |

**Gender appropriate *she-he* analogies**

| | | |
|---|---|---|
| queen-king | sister-brother | mother-father |
| waitress-waiter | ovarian cancer-prostate cancer | convent-monastery |

Figure 1: **Left** The most extreme occupations as projected on to the *she*−*he* gender direction on w2vNEWS. Occupations such as *businesswoman*, where gender is suggested by the orthography, were excluded. **Right** Automatically generated analogies for the pair *she-he* using the procedure described in text. Each automatically generated analogy is evaluated by 10 crowd-workers to whether or not it reflects gender stereotype.

[Bolukbasi et al., ‘16]

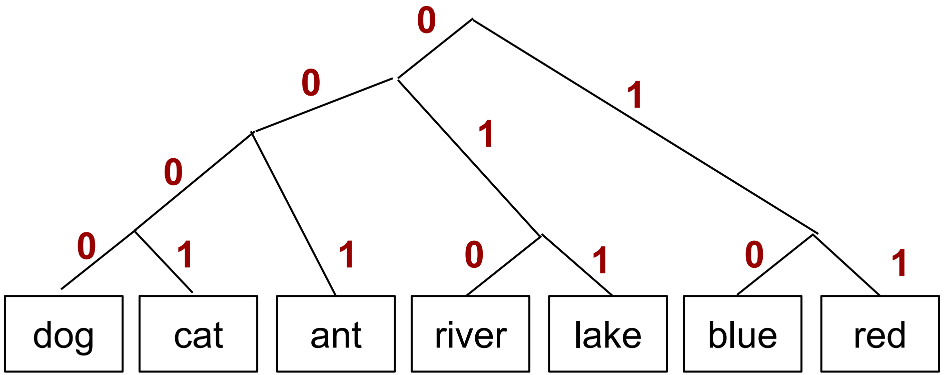
# What we’ve seen by now

- Meaning representation
- Distributional hypothesis
- Count-based vectors
  - term-document matrix
  - word-in-context matrix
  - normalizing counts: tf-idf, PPMI
  - dimensionality reduction
  - measuring similarity
  - evaluation

Next:

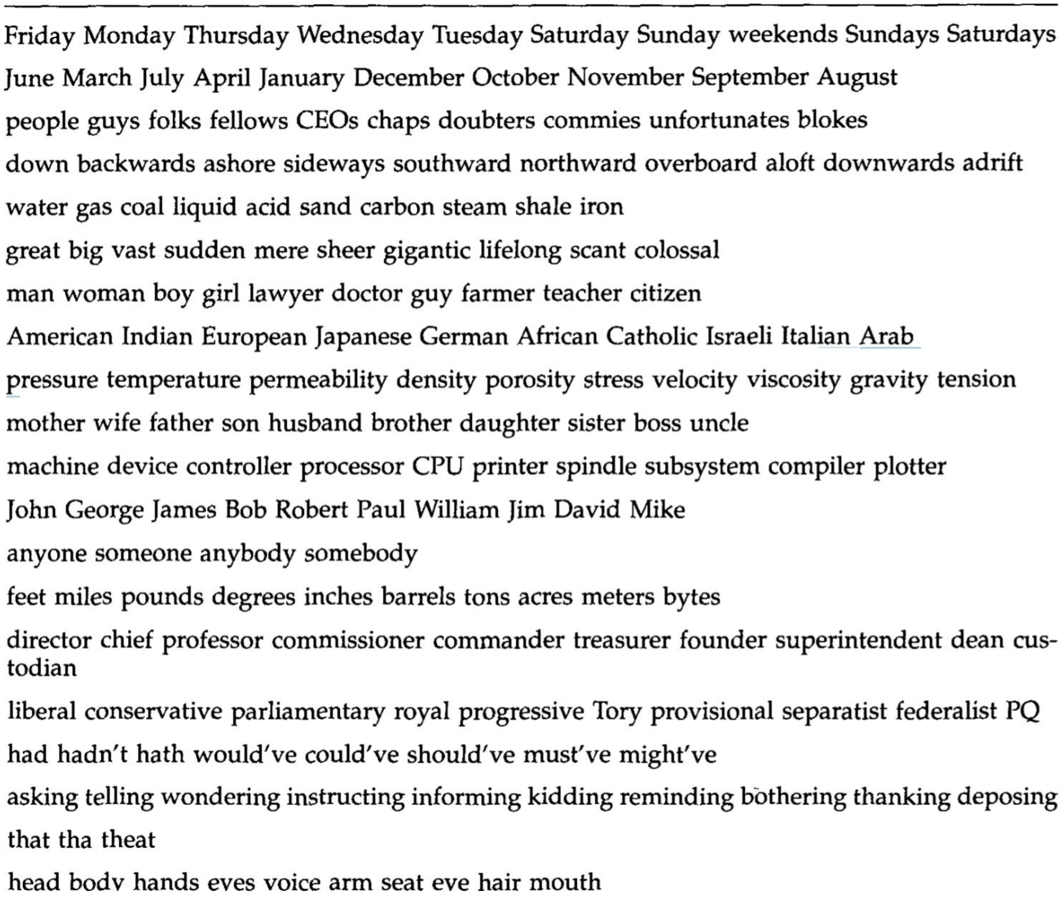
- Brown clusters
  - Representation is created through hierarchical clustering

# Brown Clustering



| | |
|---|---|
| dog | [0000] |
| cat | [0001] |
| ant | [001] |
| river | [010] |
| lake | [011] |
| blue | [10] |
| red | [11] |

# Brown Clustering



Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays

June March July April January December October November September August

people guys folks fellows CEOs chaps doubters commies unfortunates blokes

down backwards ashore sideways southward northward overboard aloft downwards adrift

water gas coal liquid acid sand carbon steam shale iron

great big vast sudden mere sheer gigantic lifelong scant colossal

man woman boy girl lawyer doctor guy farmer teacher citizen

American Indian European Japanese German African Catholic Israeli Italian Arab

pressure temperature permeability density porosity stress velocity viscosity gravity tension

mother wife father son husband brother daughter sister boss uncle

machine device controller processor CPU printer spindle subsystem compiler plotter

John George James Bob Robert Paul William Jim David Mike

anyone someone anybody somebody

feet miles pounds degrees inches barrels tons acres meters bytes

director chief professor commissioner commander treasurer founder superintendent dean custodian

liberal conservative parliamentary royal progressive Tory provisional separatist federalist PQ

had hadn't hath would've could've should've must've might've

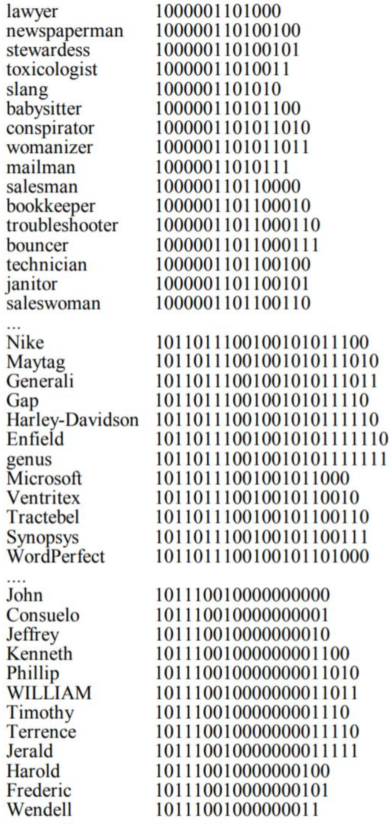
asking telling wondering instructing informing kidding reminding bothering thanking deposing

that tha theat

head body hands eyes voice arm seat eye hair mouth

[Brown et al, 1992]

# Brown Clustering



| | |
|---|---|
| lawyer | 1000001101000 |
| newspaperman | 100000110100100 |
| stewardess | 100000110100101 |
| toxicologist | 10000011010011 |
| slang | 1000001101010 |
| babysitter | 100000110101100 |
| conspirator | 1000001101011010 |
| womanizer | 1000001101011011 |
| mailman | 10000011010111 |
| salesman | 100000110110000 |
| bookkeeper | 1000001101100010 |
| troubleshooter | 10000011011000110 |
| bouncer | 10000011011000111 |
| technician | 1000001101100100 |
| janitor | 1000001101100101 |
| saleswoman | 1000001101100110 |
| ... | |
| Nike | 1011011100100101011100 |
| Maytag | 10110111001001010111010 |
| Generali | 10110111001001010111011 |
| Gap | 1011011100100101011110 |
| Harley-Davidson | 10110111001001010111110 |
| Enfield | 101101110010010101111110 |
| genus | 101101110010010101111111 |
| Microsoft | 10110111001001011000 |
| Ventritex | 101101110010010110010 |
| Tractebel | 1011011100100101100110 |
| Synopsys | 1011011100100101100111 |
| WordPerfect | 1011011100100101101000 |
| .... | |
| John | 101110010000000000 |
| Consuelo | 101110010000000001 |
| Jeffrey | 101110010000000010 |
| Kenneth | 10111001000000001100 |
| Phillip | 101110010000000011010 |
| WILLIAM | 101110010000000011011 |
| Timothy | 10111001000000001110 |
| Terrence | 101110010000000011110 |
| Jerald | 101110010000000011111 |
| Harold | 101110010000000100 |
| Frederic | 101110010000000101 |
| Wendell | 10111001000000011 |

**Table 1: Sample bit strings**

[ Miller et al., 2004]

# Brown Clustering

- $\mathcal{V}$ is a vocabulary
- $C : \mathcal{V} \rightarrow \{1, 2, \ldots k\}$ is a partition of the vocabulary into *k* clusters
- $p(C(w_i)|C(w_{i-1}))$ is a probability of cluster of $w_i$ to follow the cluster of $w_{i-1}$
- $p(w_i|C(w_i)) = \dfrac{count(w_i)}{\Sigma_{x \in C(w_i)} count(x)}$

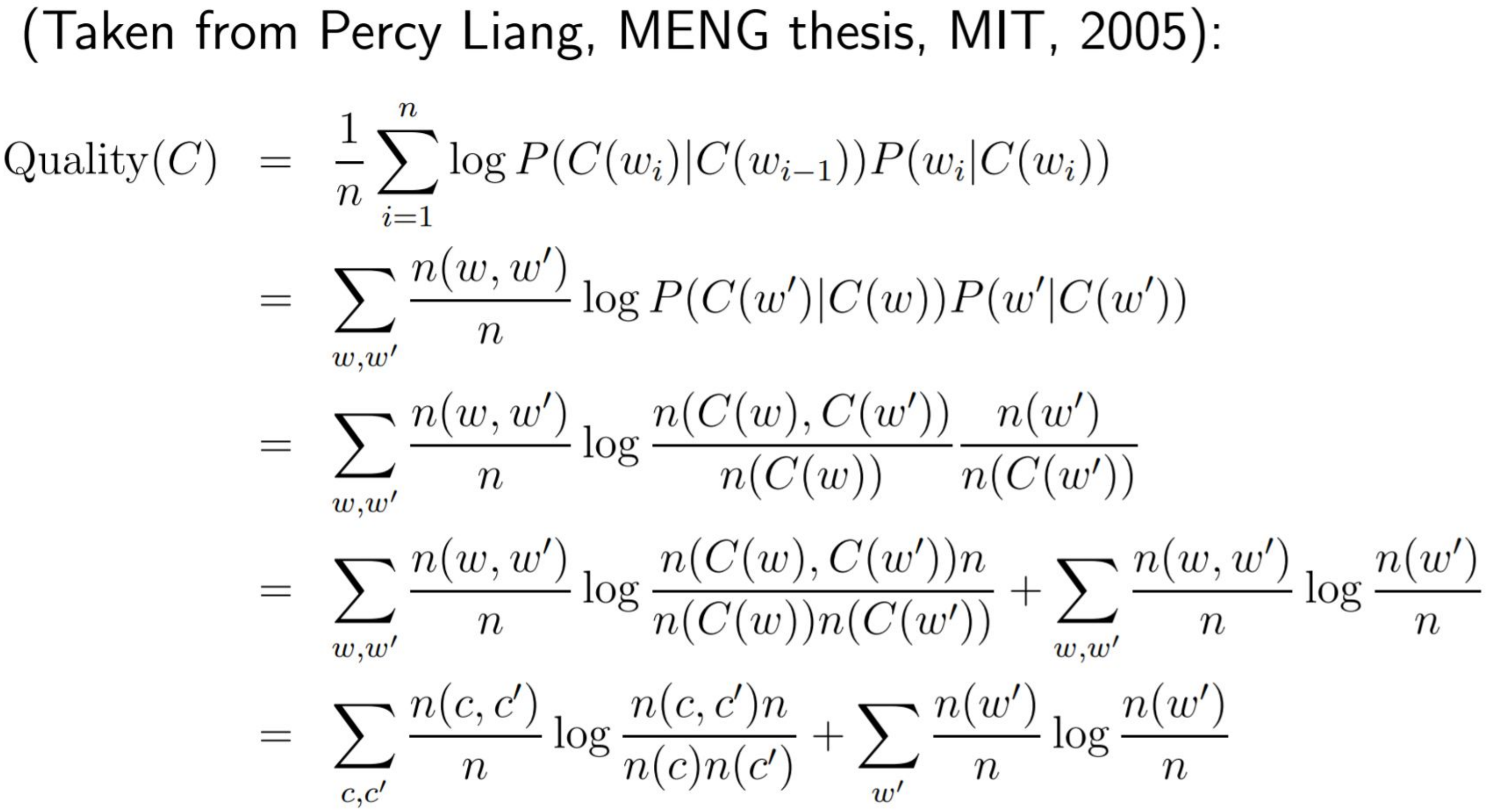
The model:

$$\text{Quality}(C) = \prod_{i=1}^{n} p(w_i|C(w_i))p(C(w_i)|C(w_{i-1}))$$

# Next

- How do we measure the quality of a partition Quality(C)?
- How to cluster?

# Quality(C)



(Taken from Percy Liang, MENG thesis, MIT, 2005):

$$
\begin{aligned}
\text{Quality}(C) &= \frac{1}{n}\sum_{i=1}^{n} \log P(C(w_i)|C(w_{i-1}))P(w_i|C(w_i)) \\
&= \sum_{w,w'} \frac{n(w,w')}{n} \log P(C(w')|C(w))P(w'|C(w')) \\
&= \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{n(C(w),C(w'))}{n(C(w))}\frac{n(w')}{n(C(w'))} \\
&= \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{n(C(w),C(w'))n}{n(C(w))n(C(w'))} + \sum_{w,w'} \frac{n(w,w')}{n} \log \frac{n(w')}{n} \\
&= \sum_{c,c'} \frac{n(c,c')}{n} \log \frac{n(c,c')n}{n(c)n(c')} + \sum_{w'} \frac{n(w')}{n} \log \frac{n(w')}{n}
\end{aligned}
$$

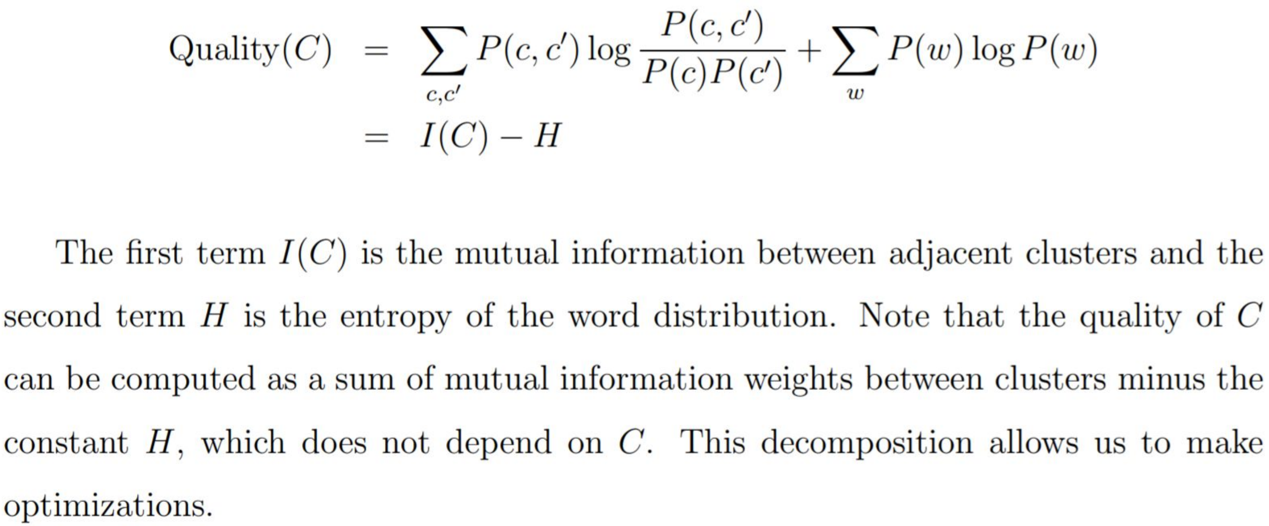
Slide by Michael Collins

# Quality(C)

- Define

$$P(c,c') = \frac{n(c,c')}{n} \quad P(w) = \frac{n(w)}{n} \quad P(c) = \frac{n(c)}{n}$$

- Then (again from Percy Liang, 2005):



$$\begin{aligned}\text{Quality}(C) &= \sum_{c,c'} P(c,c') \log \frac{P(c,c')}{P(c)P(c')} + \sum_{w} P(w) \log P(w) \\ &= I(C) - H\end{aligned}$$

The first term $I(C)$ is the mutual information between adjacent clusters and the second term $H$ is the entropy of the word distribution. Note that the quality of $C$ can be computed as a sum of mutual information weights between clusters minus the constant $H$, which does not depend on $C$. This decomposition allows us to make optimizations.

Slide by Michael Collins

# A Naive Algorithm

- We start with $|\mathcal{V}|$ clusters: each word gets its own cluster
- Our aim is to find *k* final clusters
- We run $|\mathcal{V}| - k$ merge steps:
  - At each merge step we pick two clusters $c_i$ and $c_j$ , and merge them into a single cluster
  - We greedily pick merges such that Quality(C) for the clustering C after the merge step is maximized at each stage
- Cost? Naive = $O(|\mathcal{V}|^5)$. Improved algorithm gives $O(|\mathcal{V}|^3)$: still too slow for realistic values of $|\mathcal{V}|$

Slide by Michael Collins

# Brown Clustering Algorithm

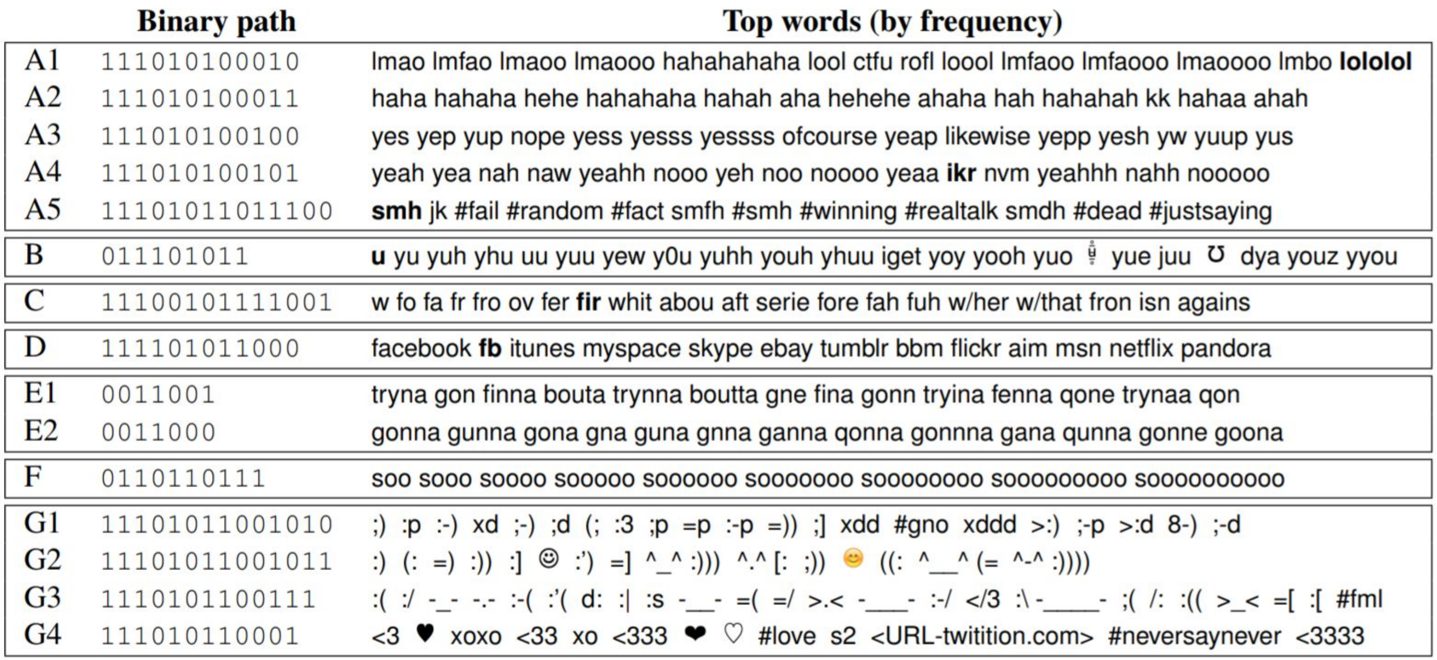
- Parameter of the approach is *m* (e.g., *m = 1000*)
- Take the top *m* most frequent words,
  put each into its own cluster, $c_1, c_2, \ldots c_m$
- For *i = (m + 1)* … $|\mathcal{V}|$
  - Create a new cluster, $c_{m+1}$, for the *i*'th most frequent word.
    We now have *m + 1* clusters
  - Choose two clusters from $c_1 \ldots c_{m+1}$ to be merged: pick the merge that gives a maximum value for Quality(C).
    We're now back to *m* clusters

- Carry out *(m − 1)* final merges, to create a full hierarchy

- Running time: $O(|\mathcal{V}|m^2 + n)$ where *n* is corpus length

Slide by Michael Collins

# Part-of-Speech Tagging for Twitter



| | Binary path | Top words (by frequency) |
|---|---|---|
| A1 | 111010100010 | lmao lmfao lmaoo lmaooo hahahahaha lool ctfu rofl loool lmfaoo lmfaooo lmaoooo lmbo **lololol** |
| A2 | 111010100011 | haha hahaha hehe hahahaha hahah aha hehehe ahaha hah hahahah kk hahaa ahah |
| A3 | 111010100100 | yes yep yup nope yess yesss yessss ofcourse yeap likewise yepp yesh yw yuup yus |
| A4 | 111010100101 | yeah yea nah naw yeahh nooo yeh noo noooo yeaa **ikr** nvm yeahhh nahh nooooo |
| A5 | 11101011011100 | **smh** jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying |
| B | 011101011 | **u** yu yuh yhu uu yuu yew y0u yuhh youh yhuu iget yoy yooh yuo ü yue juu Ʊ dya youz yyou |
| C | 11100101111001 | w fo fa fr fro ov fer **fir** whit abou aft serie fore fah fuh w/her w/that fron isn agains |
| D | 111101011000 | facebook **fb** itunes myspace skype ebay tumblr bbm flickr aim msn netflix pandora |
| E1 | 0011001 | tryna gon finna bouta trynna boutta gne fina gonn tryina fenna qone trynaa qon |
| E2 | 0011000 | gonna gunna gona gna guna gnna ganna qonna gonnna gana qunna gonne goona |
| F | 0110110111 | soo sooo soooo sooooo soooooo sooooooo soooooooo sooooooooo soooooooooo |
| G1 | 11101011001010 | ;) :p :-) xd ;-) ;d (; :3 ;p =p :-p =)) ;] xdd #gno xddd >:) ;-p >:d 8-) ;-d |
| G2 | 11101011001011 | :) (: =) :)) :] ☺ :') =] ^_^ :))) ^.^ [: ;)) 😊 ((: ^__^ (= ^-^ :)))) |
| G3 | 1110101100111 | :( :/ -_- -.- :-( :'( d: :\| :s -__- =( =/ >.< -___- :-/ </3 :\ -____- ;( /: :(( >_< =[ :[ #fml |
| G4 | 111010110001 | <3 ♥ xoxo <33 xo <333 ❤ ♡ #love s2 <URL-twitition.com> #neversaynever <3333 |

Figure 2: Example word clusters (HMM classes): we list the most probable words, starting with the most probable, in descending order. Boldfaced words appear in the example tweet (Figure 1). The binary strings are root-to-leaf paths through the binary cluster tree. For example usage, see e.g. search.twitter.com, bing.com/social and urbandictionary.com.

[ Owoputi et al., 2013]

# Plan for Today

- Count-based
  - Words are represented by a simple function of the counts of nearby words
- Brown clusters
  - Representation is created through hierarchical clustering
- Word2Vec
  - Representation is created by training a classifier to distinguish nearby and far-away words

Next class:

- Fasttext
- ELMO
- Multilingual embeddings

# Dense Embeddings You Can Download

**Word2vec** (Mikolov et al.' 13)

https://code.google.com/archive/p/word2vec/

**Fasttext** (Bojanowski et al.' 17)

http://www.fasttext.cc/

**Glove** (Pennington et al., 14)

http://nlp.stanford.edu/projects/glove/

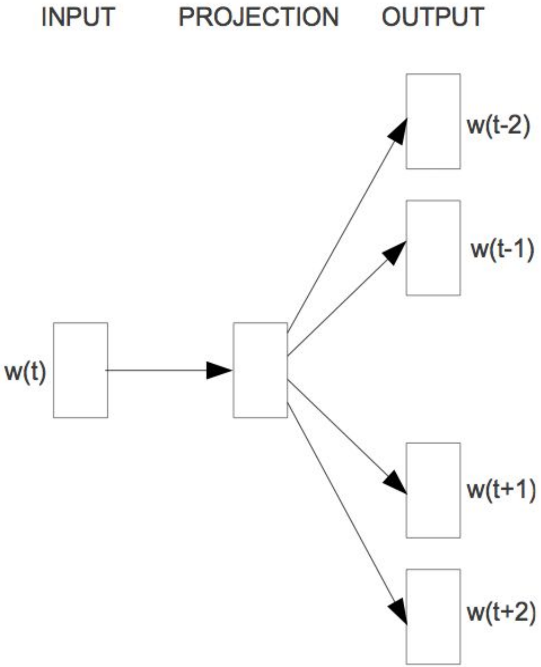
# Word2Vec

- Popular embedding method
- Very fast to train
- Code available on the web
- Idea: predict rather than count

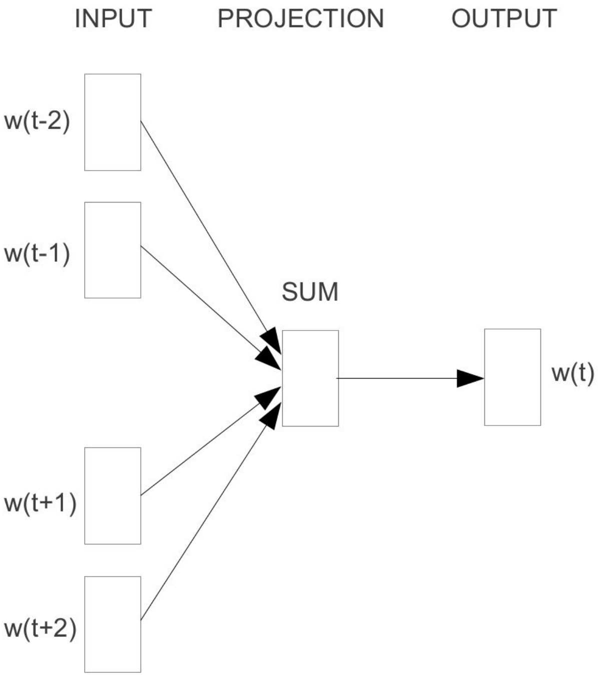
# Word2Vec

- Instead of counting how often each word w occurs near "apricot"
  - Train a classifier on a binary prediction task:
  - Is *w* likely to show up near "apricot"?

# Word2Vec



**Skip-gram**



**CBOW**

[Mikolov et al.’ 13]

# Next Class

- Word2Vec
- Fasttext
- ELMo
- Multilingual embeddings