Natural Language Processing

Representation Learning

Yulia Tsvetkov

yuliats@cs.washington.edu

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Announcements

- Don’t forget to “tag” your HW submission

- HW2 is available for download

What are various ways to represent the meaning of a word?

Lexical Semantics

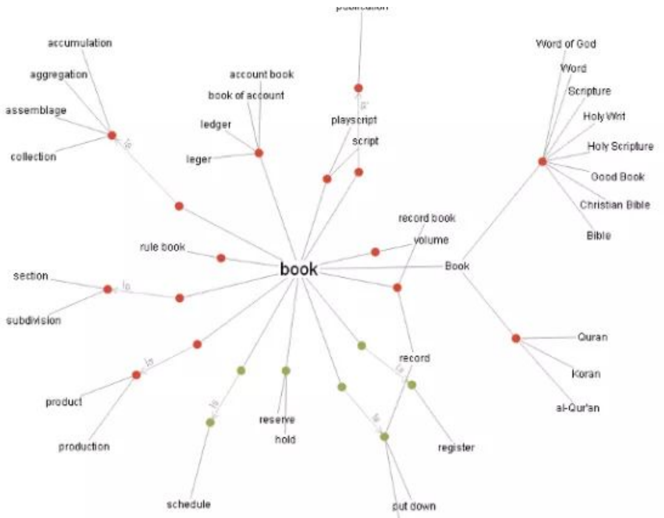
- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
  - Connotation and sentiment
    - *valence*: the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

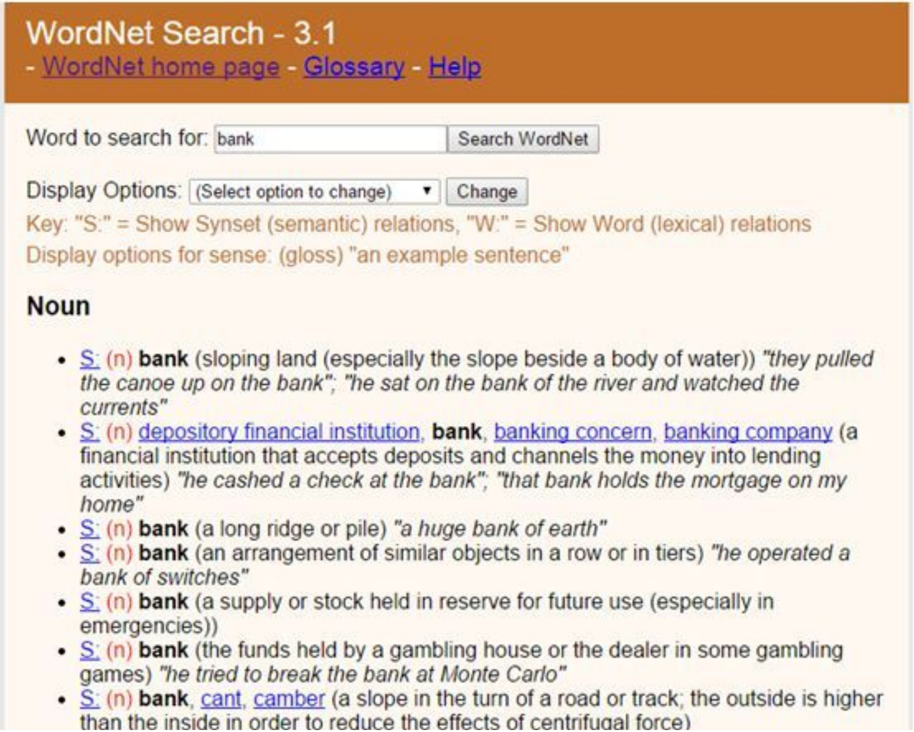
	Valence	Arousal	Dominance
courageous	8.05	5.5	7.38
music	7.67	5.57	6.5
heartbreak	2.45	5.65	3.58
cub	6.71	3.95	4.24
life	6.68	5.59	5.89

Electronic Dictionaries

WordNet

https://wordnet.princeton.edu/





WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: bank [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)

Problems with discrete representations

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

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

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]

- dimensionality: PTB: 50K, Google1T 13M

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?

Example

- 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

Ongchoi: Ipomoea aquatica "Water Spinach"

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

空心菜
kangkong
rau muống
...

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”
  - We build this space automatically by seeing which words are nearby in text

We define meaning of a word as a vector

- Called an "embedding" because it's embedded into a space
- The standard way to represent meaning in NLP

Every modern NLP algorithm uses embeddings as the representation of word meaning

Intuition: why vectors?

Consider sentiment analysis:

- With words, a feature is a word identity
  - Feature 5: 'The previous word was "terrible"'
  - requires exact same word to be in training and test

- With embeddings:
  - Feature is a word vector
  - 'The previous word was vector [35,22,17…]
  - Now in the test set we might see a similar vector [34,21,14]
  - We can generalize to similar but unseen words!!!

There are many kinds of embeddings

- Count-based
  - Words are represented by a simple function of the counts of nearby words
- Class-based
  - Representation is created through hierarchical clustering, Brown clusters
- Distributed prediction-based (type) embeddings
  - Representation is created by training a classifier to distinguish nearby and far-away words: word2vec, fasttext
- Distributed contextual (token) embeddings from language models
  - ELMo, BERT

We'll discuss 2 kinds of embeddings

- **tf-idf**
  - Information Retrieval workhorse!
  - A common baseline model
  - Sparse vectors
  - Words are represented by (a simple function of) the counts of nearby words

- **Word2vec**
  - Dense vectors
  - Representation is created by training a classifier to predict whether a word is likely to appear nearby
  - https://fasttext.cc/docs/en/crawl-vectors.html
  - Later we'll discuss extensions called contextual embeddings

Vector Semantics

Term-document matrix

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	17
soldier	2	80	62	89
fool	36	58	1	4
clown	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	17
soldier	2	80	62	89
fool	36	58	1	4
clown	20	15	2	3

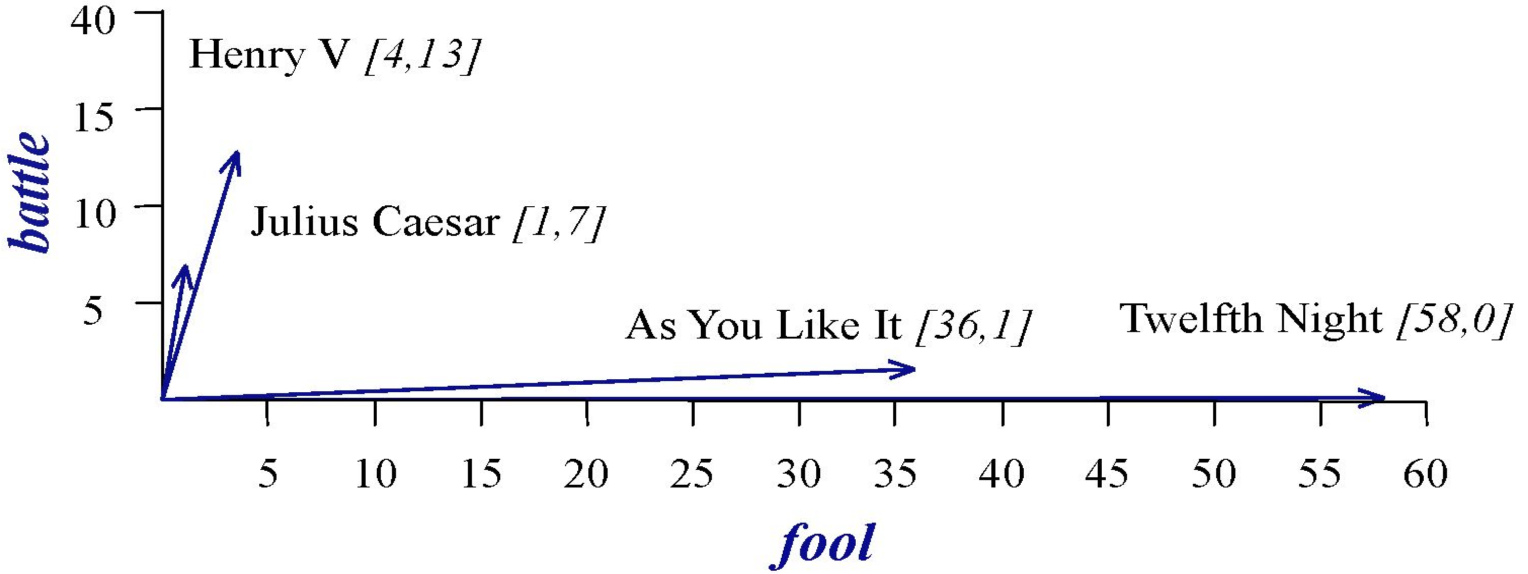
Each document is represented by a vector of words

Vectors are the basis of information retrieval

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
soldier	2	80	62	89
fool	36	58	1	4
clown	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
clown	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"

More common: word-word matrix ("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

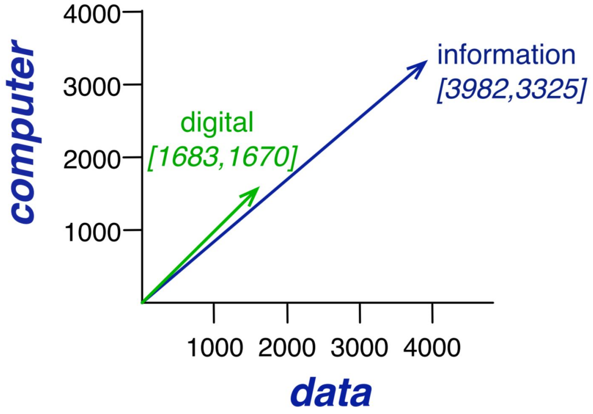
- Two words are "similar" in meaning if their context vectors are similar
  - Similarity == relatedness

Term-context matrix

Two words are similar in meaning if their context vectors are similar

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...



Computing word similarity

The dot product between two vectors is a scalar:

$$\text{dot product}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- The dot product tends to be high when the two vectors have large values in the same dimensions
- Dot product can thus be a useful similarity metric between vectors

Problem with raw dot-product

- Dot product favors long vectors
  - Dot product is higher if a vector is longer (has higher values in many dimension)
    Vector length:

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

- Frequent words (of, the, you) have long vectors (since they occur many times with other words).
  - So dot product overly favors frequent words

Alternative: cosine for computing word similarity

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

Based on the definition of the dot product between two vectors a and b

$$\mathbf{a}\cdot\mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos\theta$$

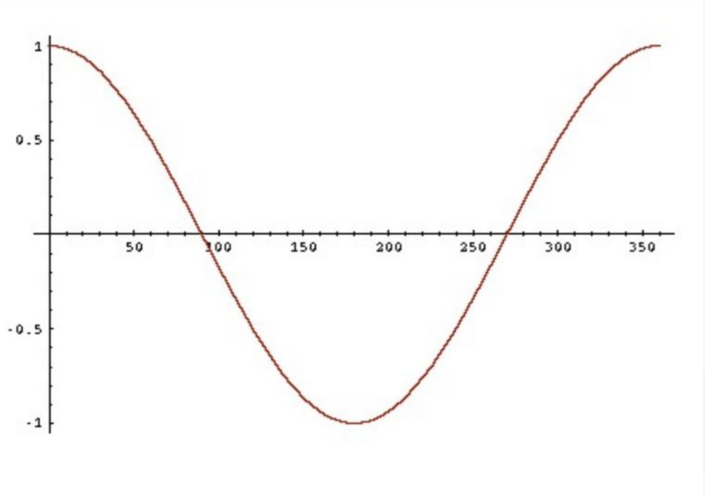
$$\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = \cos\theta$$

Cosine as a similarity metric

-1: vectors point in opposite directions

+1: vectors point in same directions

0: vectors are orthogonal



- But since raw frequency values are non-negative, the cosine for term-term matrix vectors ranges from 0–1

Cosine examples

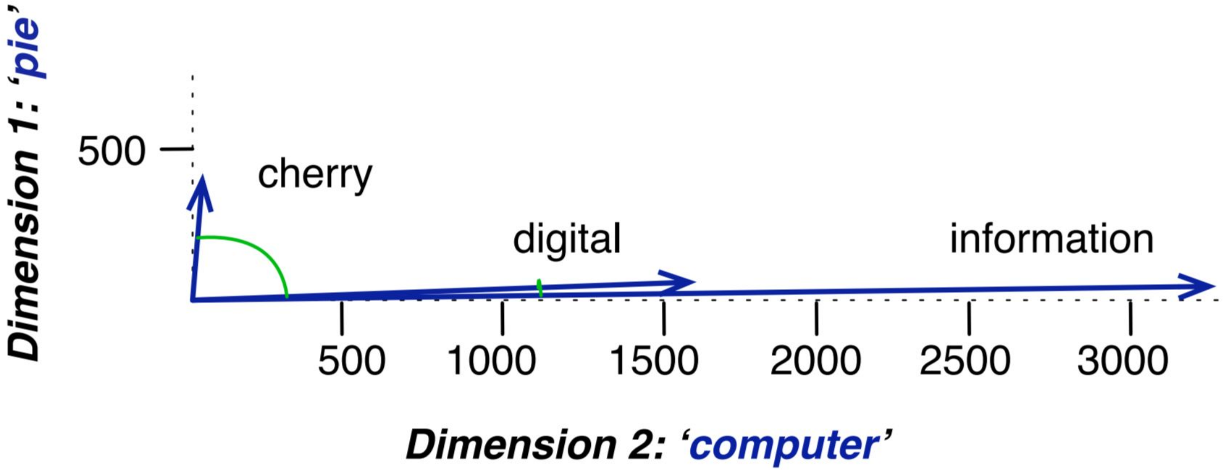
$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}\bullet\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|}\bullet\frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

	pie	data	computer
cherry	442	8	2
digital	114	80	62
information	36	58	1

$$\cos(\text{cherry}, \text{information}) = \frac{442*5+8*3982+2*3325}{\sqrt{442^2+8^2+2^2}\sqrt{5^2+3982^2+3325^2}} = .017$$

$$\cos(\text{digital}, \text{information}) = \frac{5*5+1683*3982+1670*3325}{\sqrt{5^2+1683^2+1670^2}\sqrt{5^2+3982^2+3325^2}} = .996$$

Visualizing angles



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

But raw frequency is a bad representation

- The co-occurrence matrices we have seen represent each cell by word frequencies
- Frequency is clearly useful; if sugar appears a lot near apricot, that's useful information
- But overly frequent words like the, it, or they are not very informative about the context
- It's a paradox! How can we balance these two conflicting constraints?

Two common solutions for word weighting

tf-idf: tf-idf value for word t in document d:

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

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

PMI: Pointwise mutual information

$$\text{PMI}(w_1, w_2) = log \frac{p(w_1,w_2)}{p(w_1)p(w_2)}$$

See if words like "good" appear more often with "great" than we would expect by chance

TF-IDF

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

$$\mathrm{tf}_{t,d} = \log_{10}(\mathrm{count}(t,d)+1)$$

Total # of docs in collection

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

# 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?

$$\mathrm{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$$

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

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

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

# name	formula	reference
1. **Joint probability**	$p(xy)$	(Giuliano, 1964)
2. **Conditional probability**	$p(y \mid x)$	(Gregory et al., 1999)
3. **Reverse cond. probability**	$p(x \mid y)$	(Gregory et al., 1999)
4. **Pointwise mutual inf. (*MI*)**	$\log \frac{p(xy)}{p(x*)p(*y)}$	(Church and Hanks, 1990)
5. **Mutual dependency (*MD*)**	$\log \frac{p(xy)^2}{p(x*)p(*y)}$	(Thanopoulos et al., 2002)
6. **Log frequency biased *MD***	$\log \frac{p(xy)^2}{p(x*)p(*y)} + \log p(xy)$	(Thanopoulos et al., 2002)
7. **Normalized expectation**	$\frac{2f(xy)}{f(x*)+f(*y)}$	(Smadja and McKeown, 1990)
8. **Mutual expectation**	$\frac{2f(xy)}{f(x*)+f(*y)} \cdot p(xy)$	(Dias et al., 2000)
9. **Salience**	$\log \frac{p(xy)^2}{p(x*)p(*y)} \cdot \log f(xy)$	(Kilgarriff and Tugwell, 2001)
10. **Pearson's χ^2 test**	$\sum_{i,j} \frac{(f_{ij}-\hat{f}_{ij})^2}{\hat{f}_{ij}}$	(Manning and Schütze, 1999)
11. **Fisher's exact test**	$\frac{f(x*)!f(\bar{x}*)!f(*y)!f(*\bar{y})!}{N!f(xy)!f(x\bar{y})!f(\bar{x}y)!f(\bar{x}\bar{y})!}$	(Pedersen, 1996)
12. **t test**	$\frac{f(xy)-\hat{f}(xy)}{\sqrt{f(xy)(1-(f(xy)/N))}}$	(Church and Hanks, 1990)
13. **z score**	$\frac{f(xy)-\hat{f}(xy)}{\sqrt{\hat{f}(xy)(1-(\hat{f}(xy)/N))}}$	(Berry-Rogghe, 1973)
14. **Poisson significance**	$\frac{\hat{f}(xy)-f(xy)\log \hat{f}(xy)+\log f(xy)!}{\log N}$	(Quasthoff and Wolff, 2002)
15. **Log likelihood ratio**	$-2\sum_{i,j} f_{ij} \log \frac{f_{ij}}{\hat{f}_{ij}}$	(Dunning, 1993)
16. **Squared log likelihood ratio**	$-2\sum_{i,j} \frac{\log f_{ij}^2}{\hat{f}_{ij}}$	(Inkpen and Hirst, 2002)
17. **Russel-Rao**	$\frac{a}{a+b+c+d}$	(Russel and Rao, 1940)
18. **Sokal-Michiner**	$\frac{a+d}{a+b+c+d}$	(Sokal and Michener, 1958)
19. **Rogers-Tanimoto**	$\frac{a+d}{a+2b+2c+d}$	(Rogers and Tanimoto, 1960)
20. **Hamann**	$\frac{(a+d)-(b+c)}{a+b+c+d}$	(Hamann, 1961)
21. **Third Sokal-Sneath**	$\frac{b+c}{a+d}$	(Sokal and Sneath, 1963)
22. **Jaccard**	$\frac{a}{a+b+c}$	(Jaccard, 1912)
23. **First Kulczynsky**	$\frac{a}{b+c}$	(Kulczynski, 1927)
24. **Second Sokal-Sneath**	$\frac{a}{a+2(b+c)}$	(Sokal and Sneath, 1963)
25. **Second Kulczynski**	$\frac{1}{2}(\frac{a}{a+b}+\frac{a}{a+c})$	(Kulczynski, 1927)
26. **Fourth Sokal-Sneath**	$\frac{1}{4}(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c})$	(Kulczynski, 1927)
27. **Odds ratio**	$\frac{ad}{bc}$	(Tan et al., 2002)
28. **Yulle's ω**	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$	(Tan et al., 2002)
29. **Yulle's Q**	$\frac{ad-bc}{ad+bc}$	(Tan et al., 2002)
30. **Driver-Kroeber**	$\frac{a}{\sqrt{(a+b)(a+c)}}$	(Driver and Kroeber, 1932)
31. **Fifth Sokal-Sneath**	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	(Sokal and Sneath, 1963)
32. **Pearson**	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	(Pearson,1950)
33. **Baroni-Urbani**	$\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$	(Baroni-Urbani and Buser, 1976)
34. **Braun-Blanquet**	$\frac{a}{\max(a+b,a+c)}$	(Braun-Blanquet, 1932)
35. **Simpson**	$\frac{a}{\min(a+b,a+c)}$	(Simpson, 1943)
36. **Michael**	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	(Michael, 1920)
37. **Mountford**	$\frac{2a}{2bc+ab+ac}$	(Kaufman and Rousseeuw, 1990)
38. **Fager**	$\frac{a}{\sqrt{(a+b)(a+c)}} - \frac{1}{2}\max(b,c)$	(Kaufman and Rousseeuw, 1990)
39. **Unigram subtuples**	$\log \frac{ad}{bc} - 3.29\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}$	(Blaheta and Johnson, 2001)
40. ***U* cost**	$\log(1+\frac{\min(b,c)+a}{\max(b,c)+a})$	(Tulloss, 1997)
41. ***S* cost**	$\log(1+\frac{\min(b,c)}{a+1})^{-\frac{1}{2}}$	(Tulloss, 1997)
42. ***R* cost**	$\log(1+\frac{a}{a+b}) \cdot \log(1+\frac{a}{a+c})$	(Tulloss, 1997)
43. ***T* combined cost**	$\sqrt{U \times S \times R}$	(Tulloss, 1997)
44. **Phi**	$\frac{p(xy)-p(x*)p(*y)}{\sqrt{p(x*)p(*y)(1-p(x*))(1-p(*y))}}$	(Tan et al., 2002)
45. **Kappa**	$\frac{p(xy)+p(\bar{x}\bar{y})-p(x*)p(*y)-p(\bar{x}*)p(*\bar{y})}{1-p(x*)p(*y)-p(\bar{x}*)p(*\bar{y})}$	(Tan et al., 2002)
46. ***J* measure**	$\max[p(xy)\log\frac{p(y \mid x)}{p(*y)} + p(x\bar{y})\log\frac{p(\bar{y} \mid x)}{p(*\bar{y})},$ $p(xy)\log\frac{p(x \mid y)}{p(x*)} + p(\bar{x}y)\log\frac{p(\bar{x} \mid y)}{p(\bar{x}*)}]$	(Tan et al., 2002)
47. **Gini index**	$\max[p(x*)(p(y \mid x)^2+p(\bar{y} \mid x)^2)-p(*y)^2$ $+p(\bar{x}*)(p(y \mid \bar{x})^2+p(\bar{y} \mid \bar{x})^2)-p(*\bar{y})^2,$ $p(*y)(p(x \mid y)^2+p(\bar{x} \mid y)^2)-p(x*)^2$ $+p(*\bar{y})(p(x \mid \bar{y})^2+p(\bar{x} \mid \bar{y})^2)-p(\bar{x}*)^2]$	(Tan et al., 2002)
48. **Confidence**	$\max[p(y \mid x), p(x \mid y)]$	(Tan et al., 2002)
49. **Laplace**	$\max[\frac{Np(xy)+1}{Np(x*)+2}, \frac{Np(xy)+1}{Np(*y)+2}]$	(Tan et al., 2002)
50. **Conviction**	$\max[\frac{p(x*)p(*y)}{p(x\bar{y})}, \frac{p(\bar{x}*)p(*y)}{p(\bar{x}y)}]$	(Tan et al., 2002)
51. **Piatersky-Shapiro**	$p(xy)-p(x*)p(*y)$	(Tan et al., 2002)
52. **Certainity factor**	$\max[\frac{p(y \mid x)-p(*y)}{1-p(*y)}, \frac{p(x \mid y)-p(x*)}{1-p(x*)}]$	(Tan et al., 2002)
53. **Added value (*AV*)**	$\max[p(y \mid x)-p(*y), p(x \mid y)-p(x*)]$	(Tan et al., 2002)
54. **Collective strength**	$\frac{p(xy)+p(\bar{x}\bar{y})}{p(x*)p(y)+p(\bar{x}*)p(*y)} \cdot \frac{1-p(x*)p(*y)-p(\bar{x}*)p(*y)}{1-p(xy)-p(\bar{x}\bar{y})}$	(Tan et al., 2002)
55. **Klosgen**	$\sqrt{p(xy)} \cdot AV$	(Tan et al., 2002)

(Pecina'09)

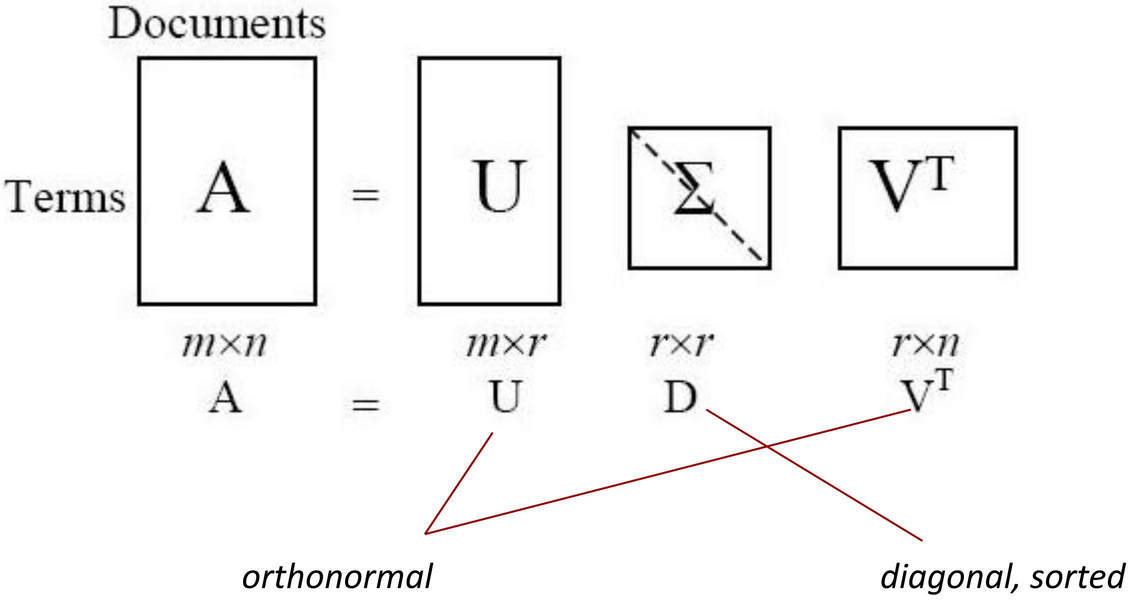
Dimensionality Reduction

- 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

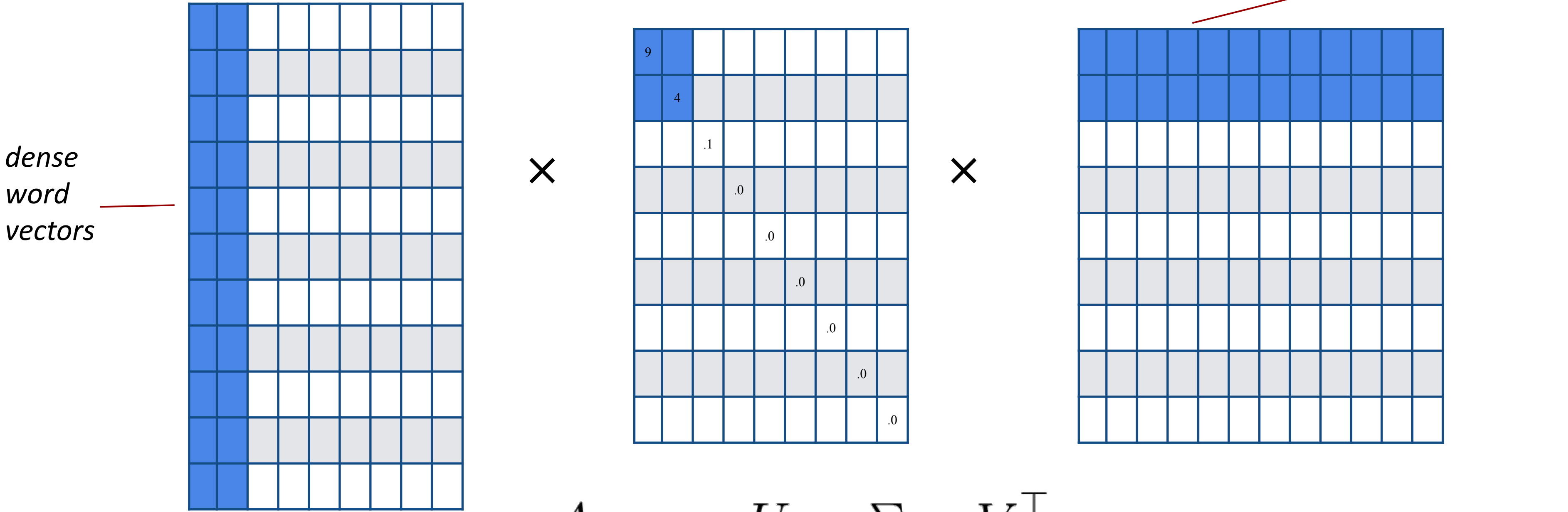
Singular Value Decomposition (SVD)

- 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)



$$A_{m\times n} \approx U_{m\times k}\Sigma_{k\times k}V^{\top}_{k\times n} \qquad k \ll m, n$$

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]

Evaluation

- Intrinsic
- Extrinsic
- Qualitative

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

Extrinsic Evaluation

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

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)

Visualisation

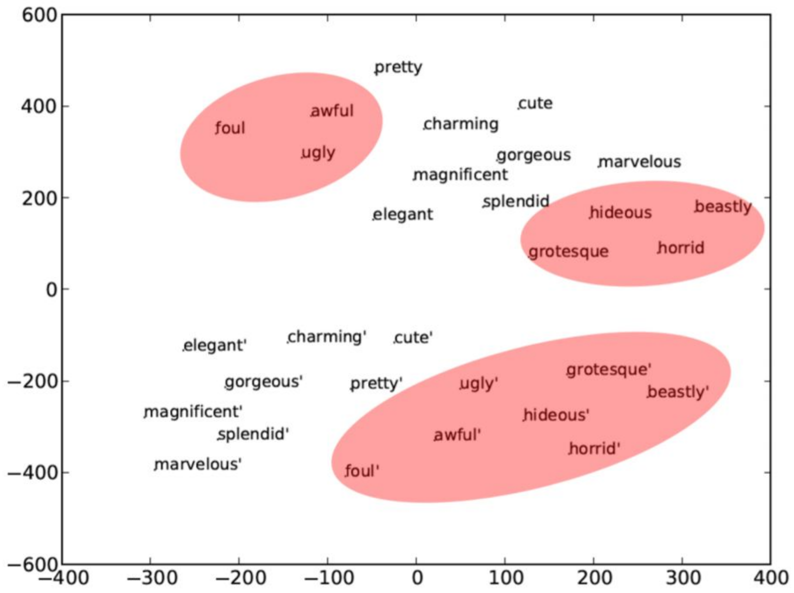


Figure 6.5: Monolingual (top) and multilingual (bottom; marked with apostrophe) word projections of the antonyms (shown in red) and synonyms of "beautiful".

[Faruqui et al., 2015]

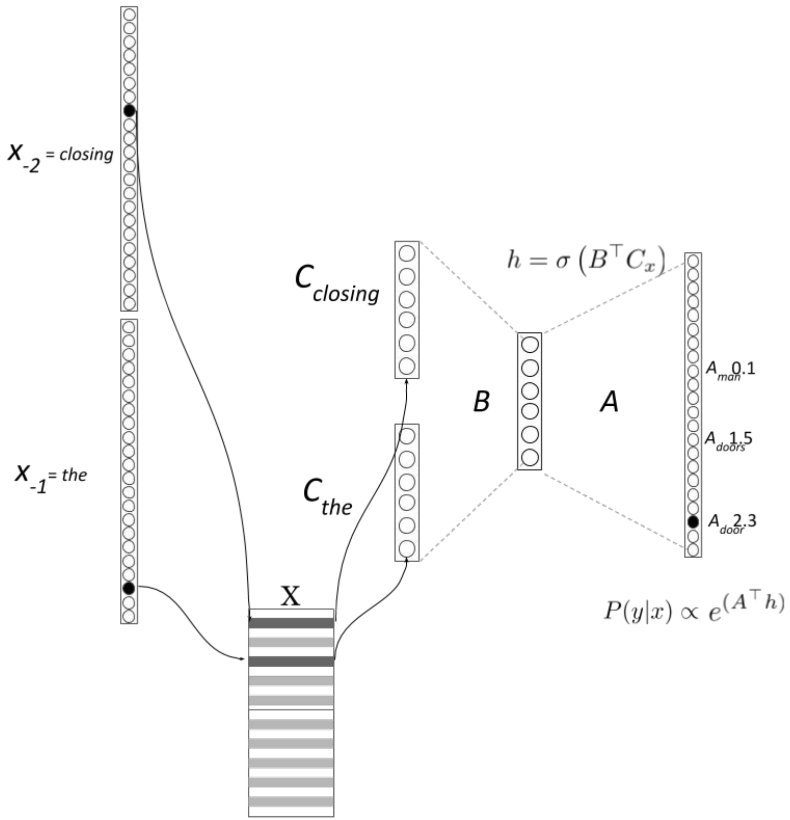
- Visualizing Data using t-SNE (van der Maaten & Hinton'08)

Distributed representations

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

"One hot" vectors and dense word vectors (embeddings)

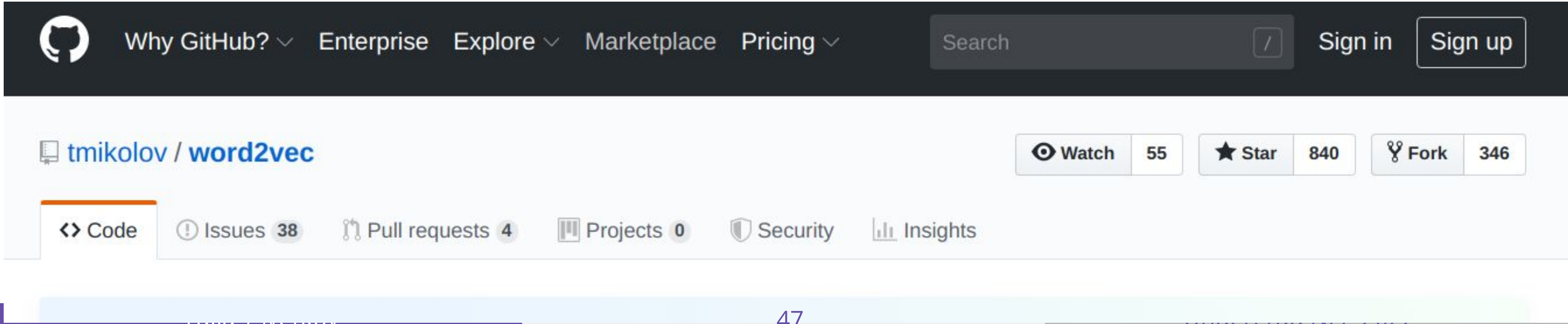


Low-dimensional word representations

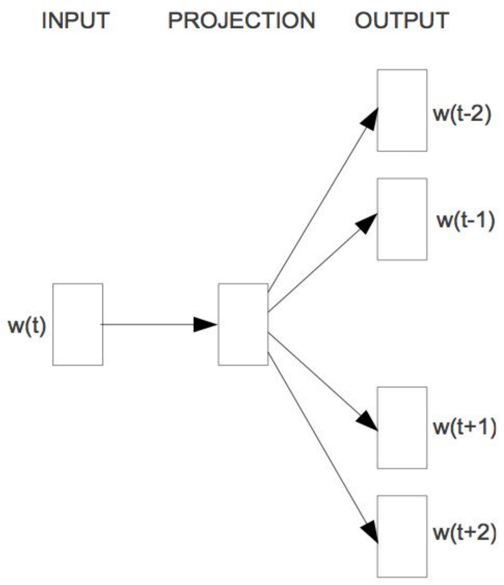
- Learning representations by back-propagating errors
  - Rumelhart, Hinton & Williams, 1986
- A neural probabilistic language model
  - Bengio et al., 2003
- Natural Language Processing (almost) from scratch
  - Collobert & Weston, 2008
- Word representations: A simple and general method for semi-supervised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

Word2Vec

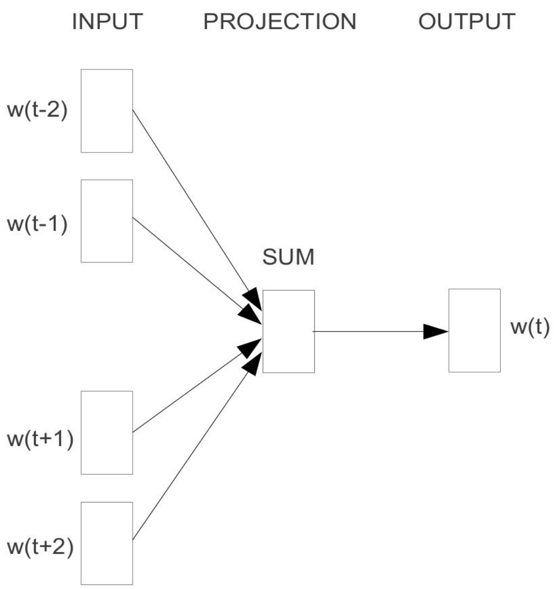
- Popular embedding method
- Very fast to train
- Code available on the web
- Idea: predict rather than count



Word2Vec



Skip-gram



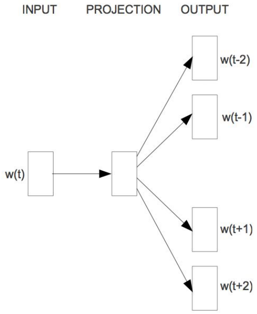
CBOW

- [Mikolov et al.' 13]

Skip-gram Prediction

- Predict vs Count

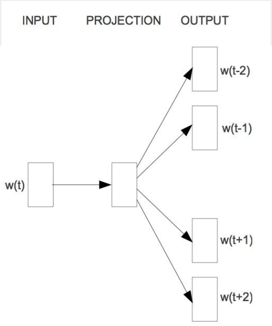
the cat sat on the mat



Skip-gram

Skip-gram Prediction

- Predict vs Count



Skip-gram

the cat sat on the mat

w_t = the → **CLASSIFIER** →

w_{t-2} = <start$_{-2}$>
w_{t-1} = <start$_{-1}$>
w_{t+1} = cat
w_{t+2} = sat

context size = 2

Skip-gram Prediction

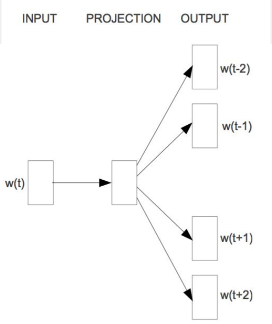
- Predict vs Count

the cat sat on the mat

w_t = cat → **CLASSIFIER** →

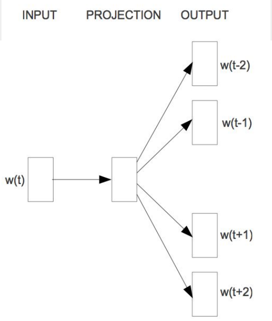
w_{t-2} = <start$_{-1}$>
w_{t-1} = the
w_{t+1} = sat
w_{t+2} = on

context size = 2



Skip-gram

Skip-gram Prediction



Skip-gram

- Predict vs Count

the cat <u>sat</u> on the mat

w_t = sat → **CLASSIFIER** →

w_{t-2} = the
w_{t-1} = cat
w_{t+1} = on
w_{t+2} = the

context size = 2

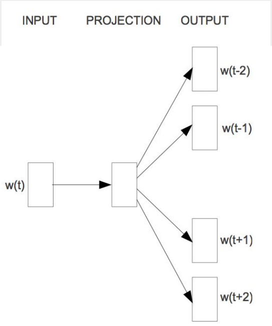
Skip-gram Prediction

- Predict vs Count

the cat sat <u>on</u> the mat

w_t = on → **CLASSIFIER** →

w_{t-2} = cat
w_{t-1} = sat
w_{t+1} = the
w_{t+2} = mat



Skip-gram

context size = 2

Skip-gram Prediction

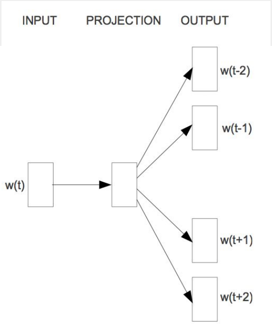
- Predict vs Count

the cat sat on <u>the</u> mat

w_t = the → CLASSIFIER →

w_{t-2} = sat
w_{t-1} = on
w_{t+1} = mat
w_{t+2} = <end$_{+1}$>

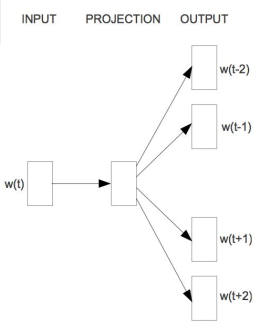
context size = 2



Skip-gram

Skip-gram Prediction

- Predict vs Count



Skip-gram

the cat sat on the mat

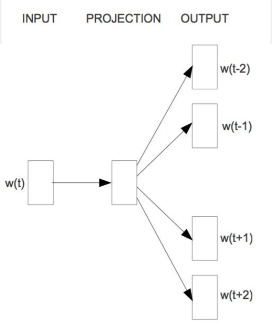
w_t = mat → CLASSIFIER →

w_{t-2} = on
w_{t-1} = the
w_{t+1} = <end_{+1}>
w_{t+2} = <end_{+2}>

context size = 2

Skip-gram Prediction

- Predict vs Count



Skip-gram

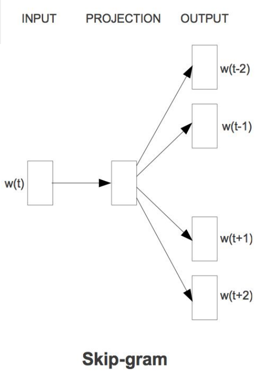
w_t = the → **CLASSIFIER** →

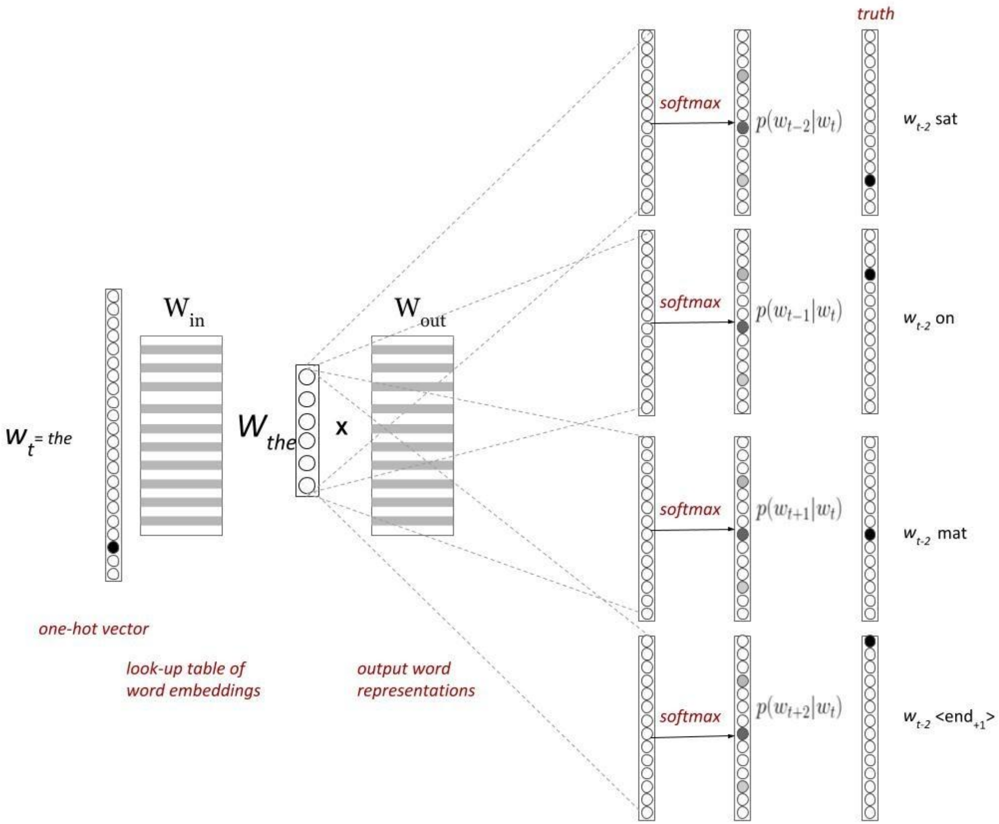
w_{t-2} = sat
w_{t-1} = on
w_{t+1} = mat
w_{t+2} = <end$_{+1}$>

w_t = the → **CLASSIFIER** →

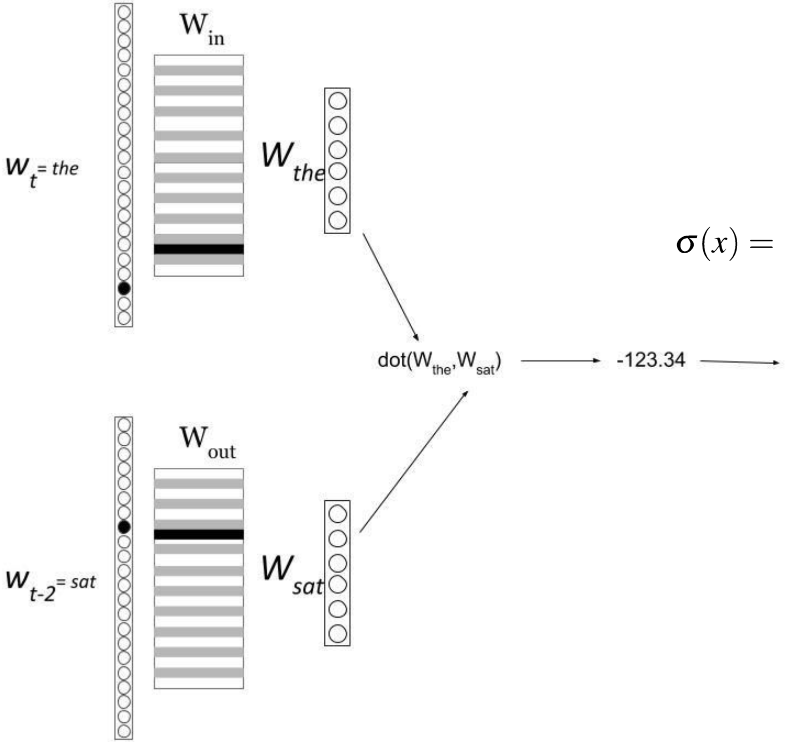
w_{t-2} = <start$_{-2}$>
w_{t-1} = <start$_{-1}$>
w_{t+1} = cat
w_{t+2} = sat

Skip-gram Prediction

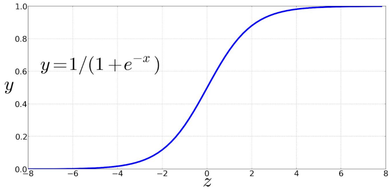




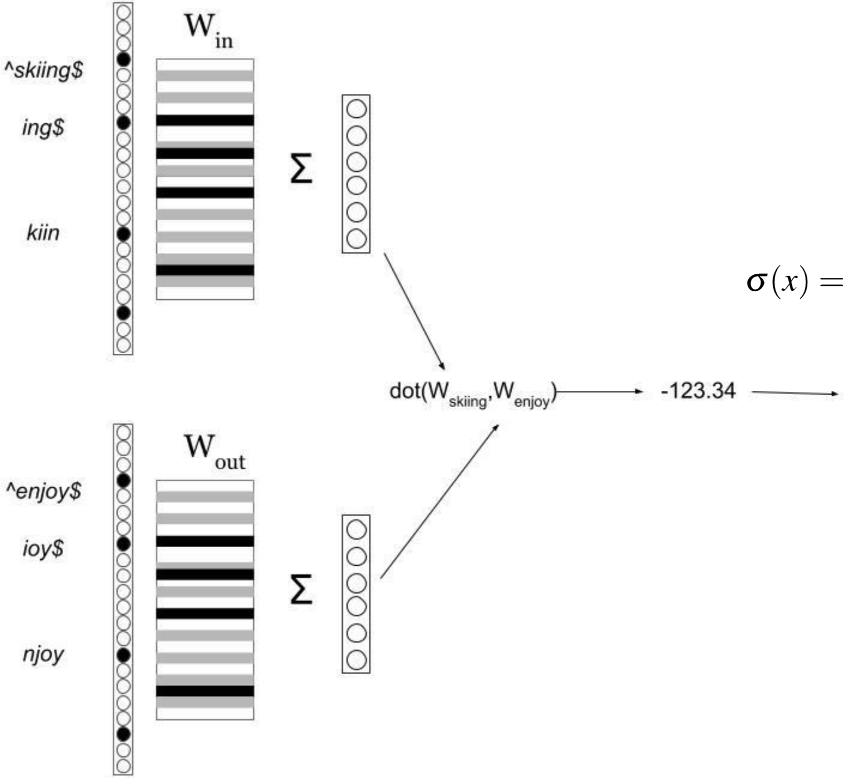
How to compute p(+|t,c)?



$$\sigma(x) = \frac{1}{1+e^{-x}}$$



FastText



$$\sigma(x) = \frac{1}{1+e^{-x}}$$