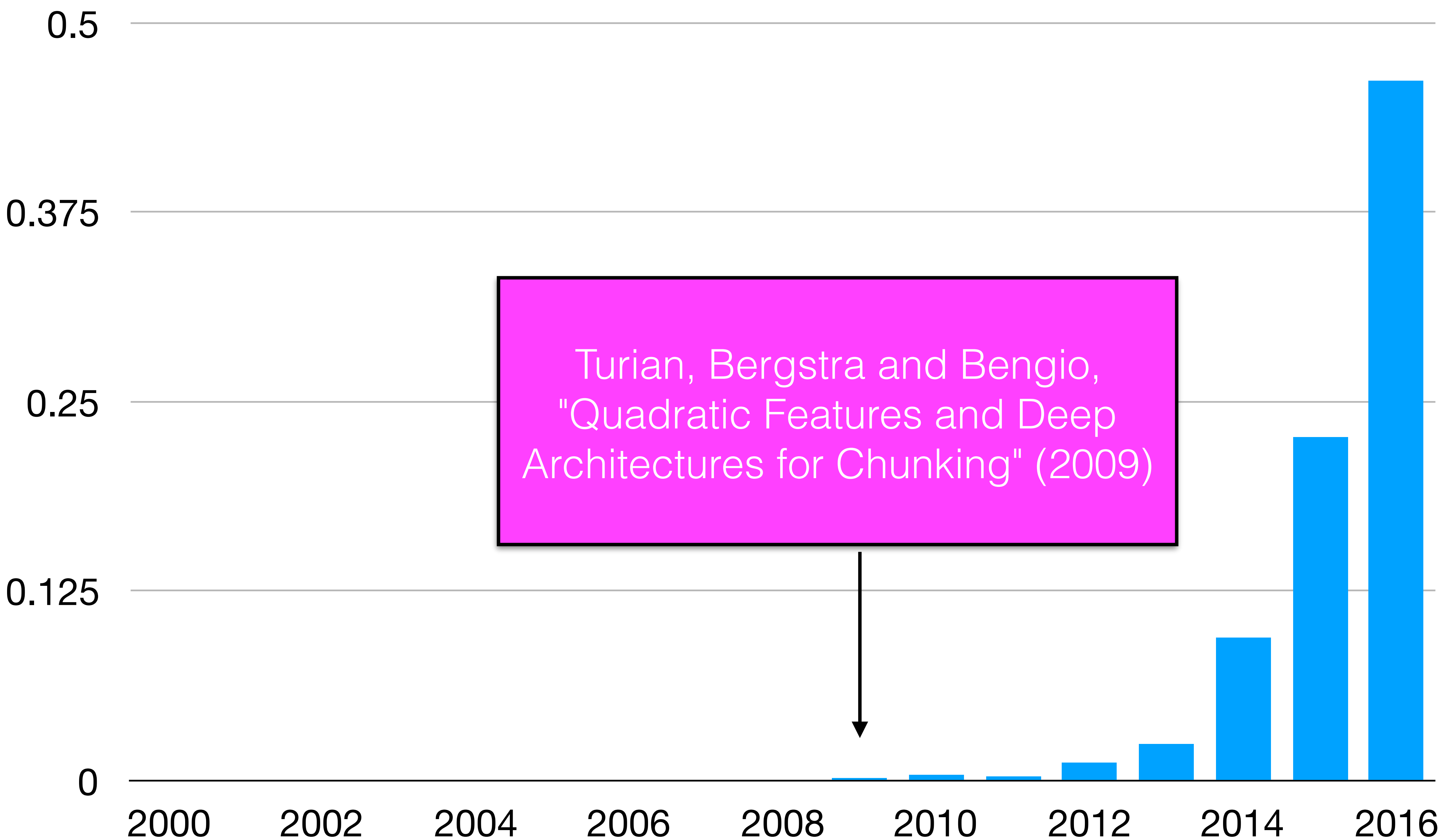
Applied Natural Language Processing

Info 256

Lecture 10: Word embeddings (Feb 21, 2019)

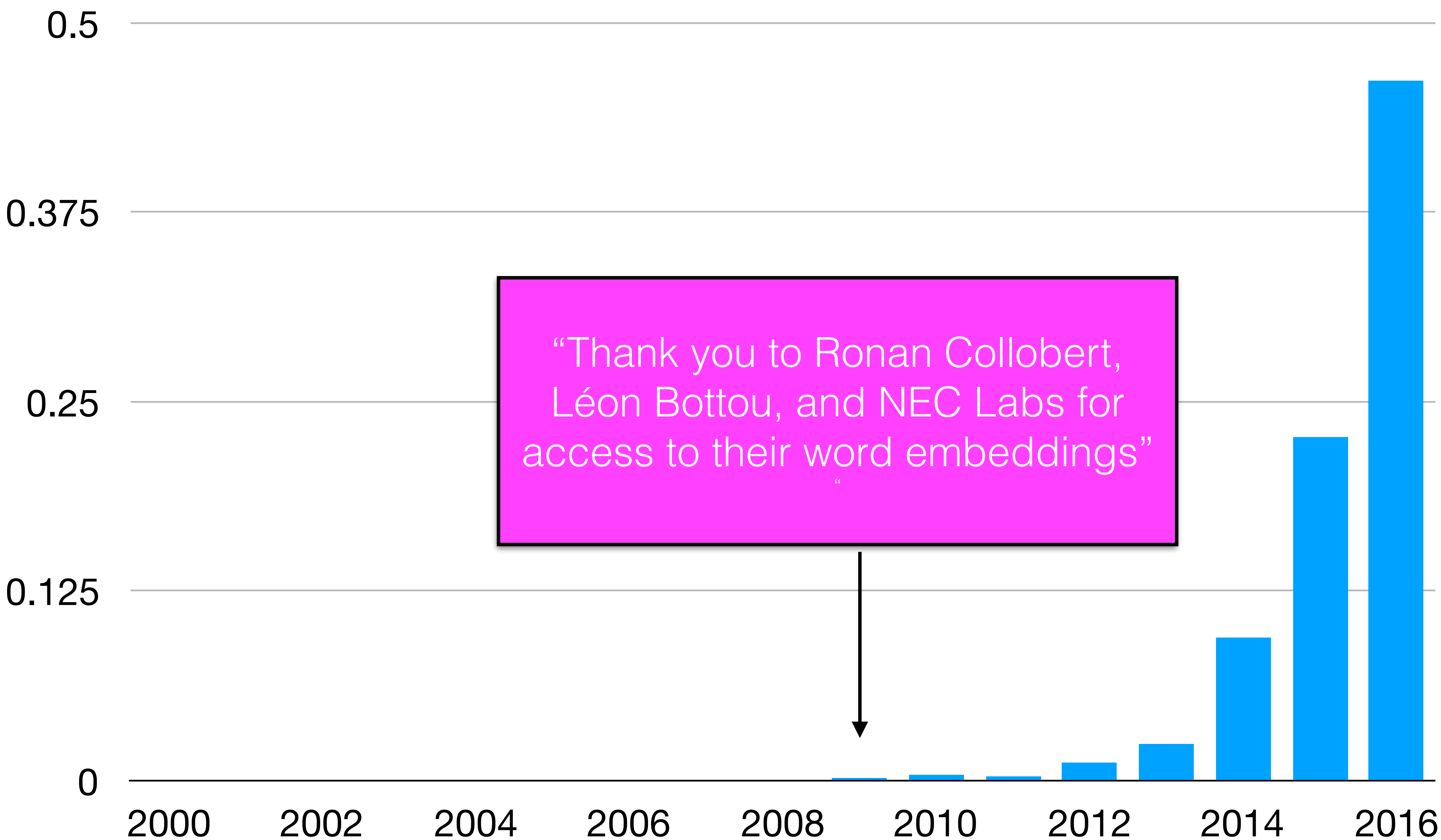
David Bamman, UC Berkeley

“Word embedding” in NLP papers



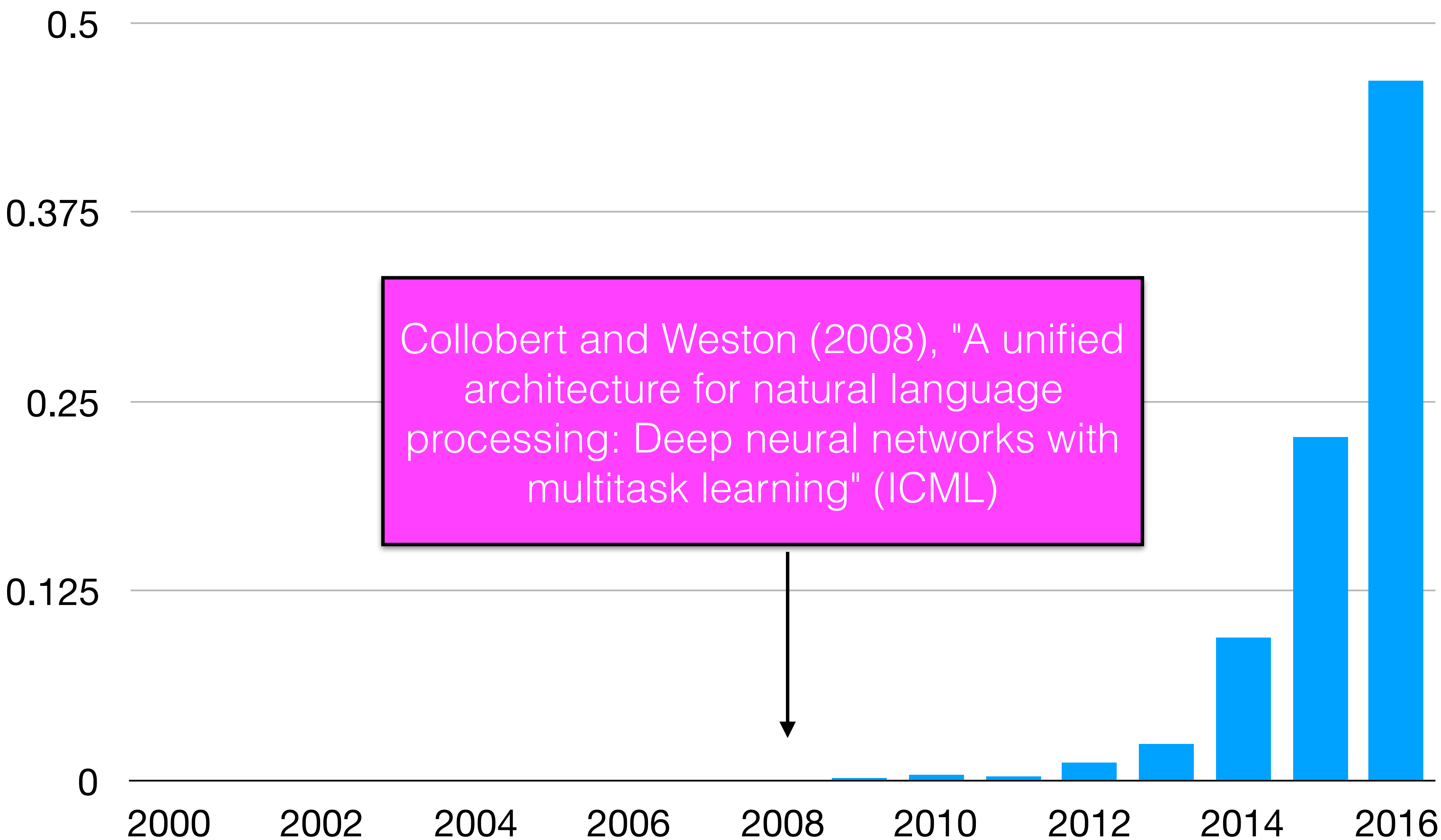
ACL, EMNLP, NAACL, TACL, EACL, CONLL, CL (data from ACL Anthology Network)

“Word embedding” in NLP papers



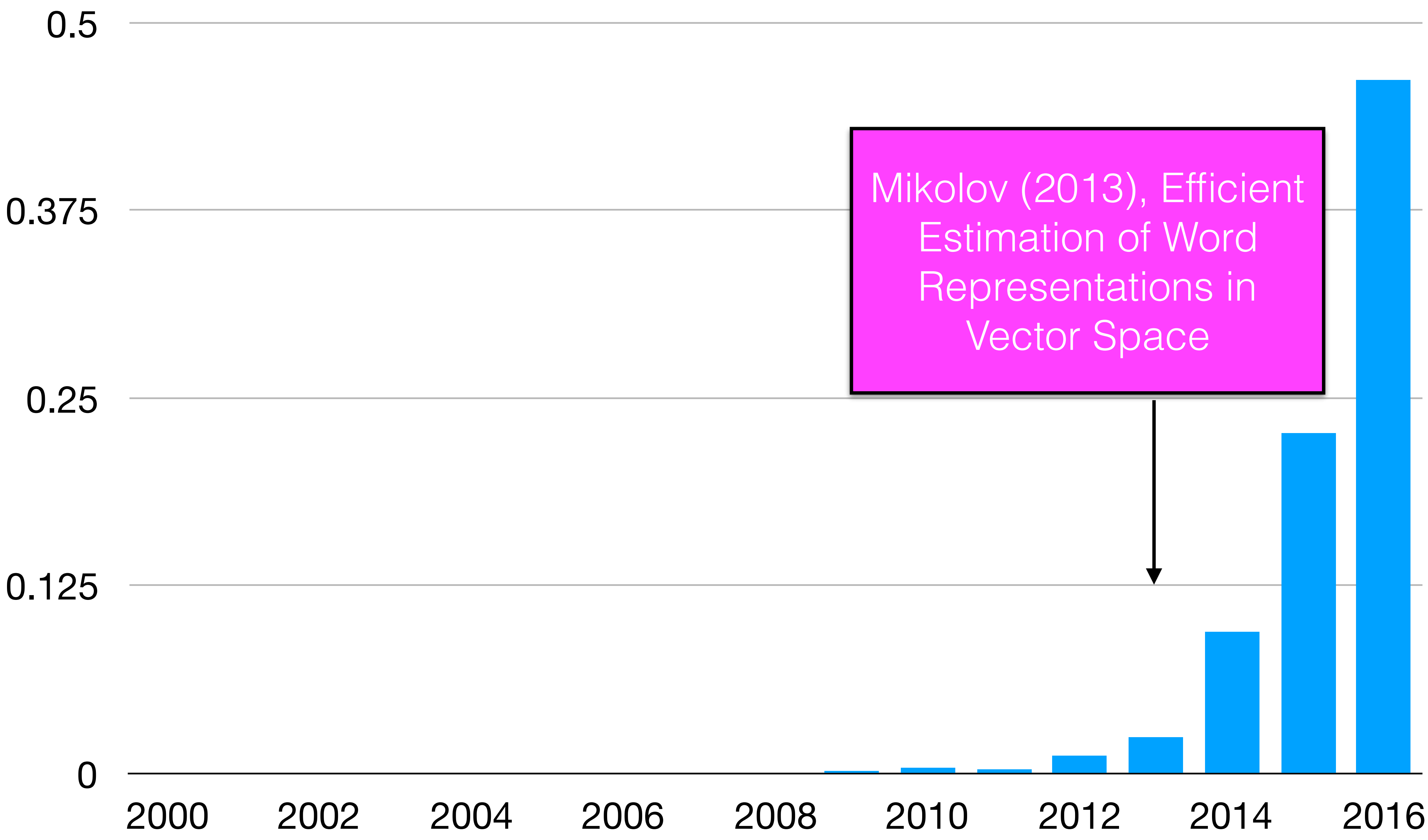
ACL, EMNLP, NAACL, TACL, EACL, CONLL, CL (data from ACL Anthology Network)

“Word embedding” in NLP papers



ACL, EMNLP, NAACL, TACL, EACL, CONLL, CL (data from ACL Anthology Network)

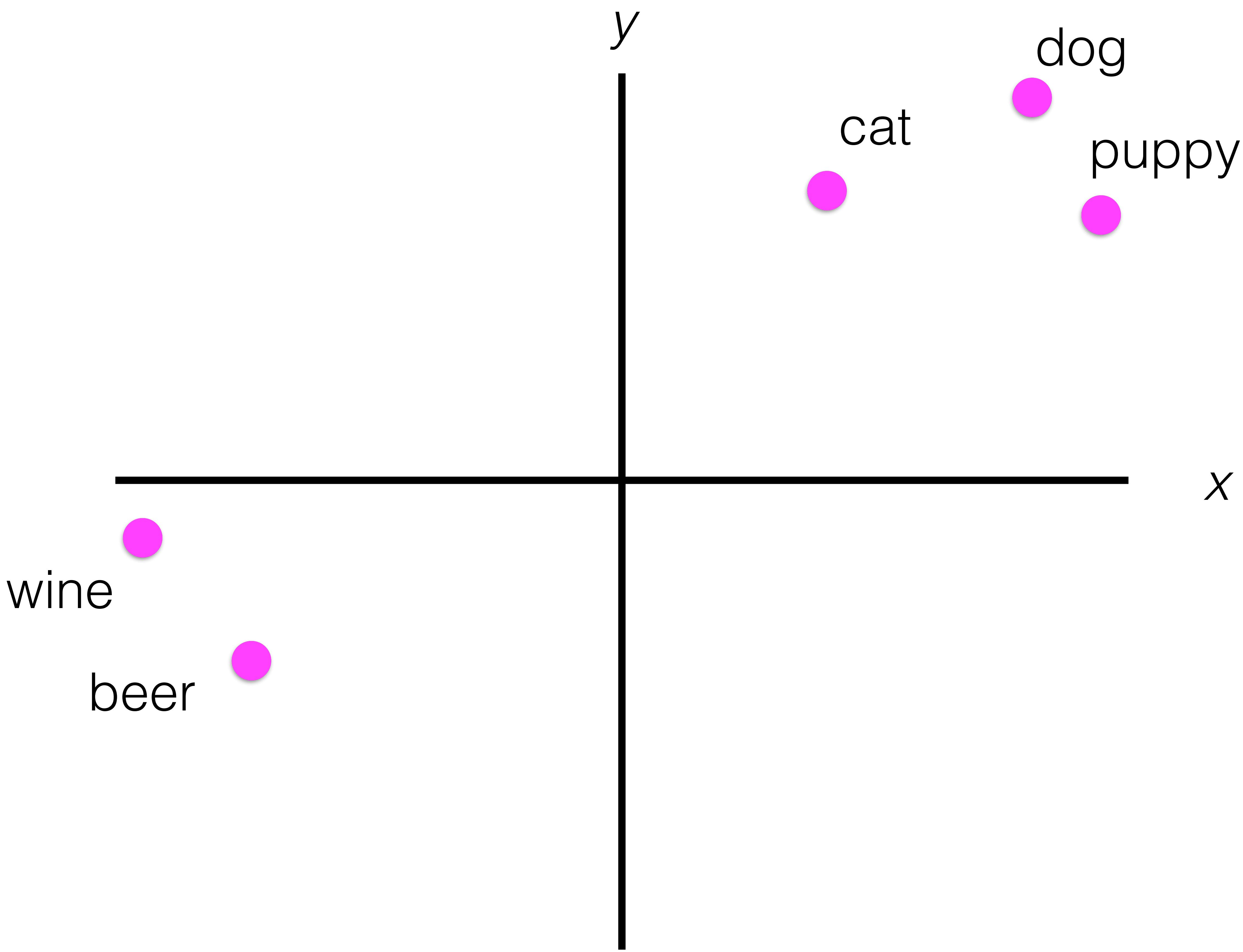
“Word embedding” in NLP papers



ACL, EMNLP, NAACL, TACL, EACL, CONLL, CL (data from ACL Anthology Network)

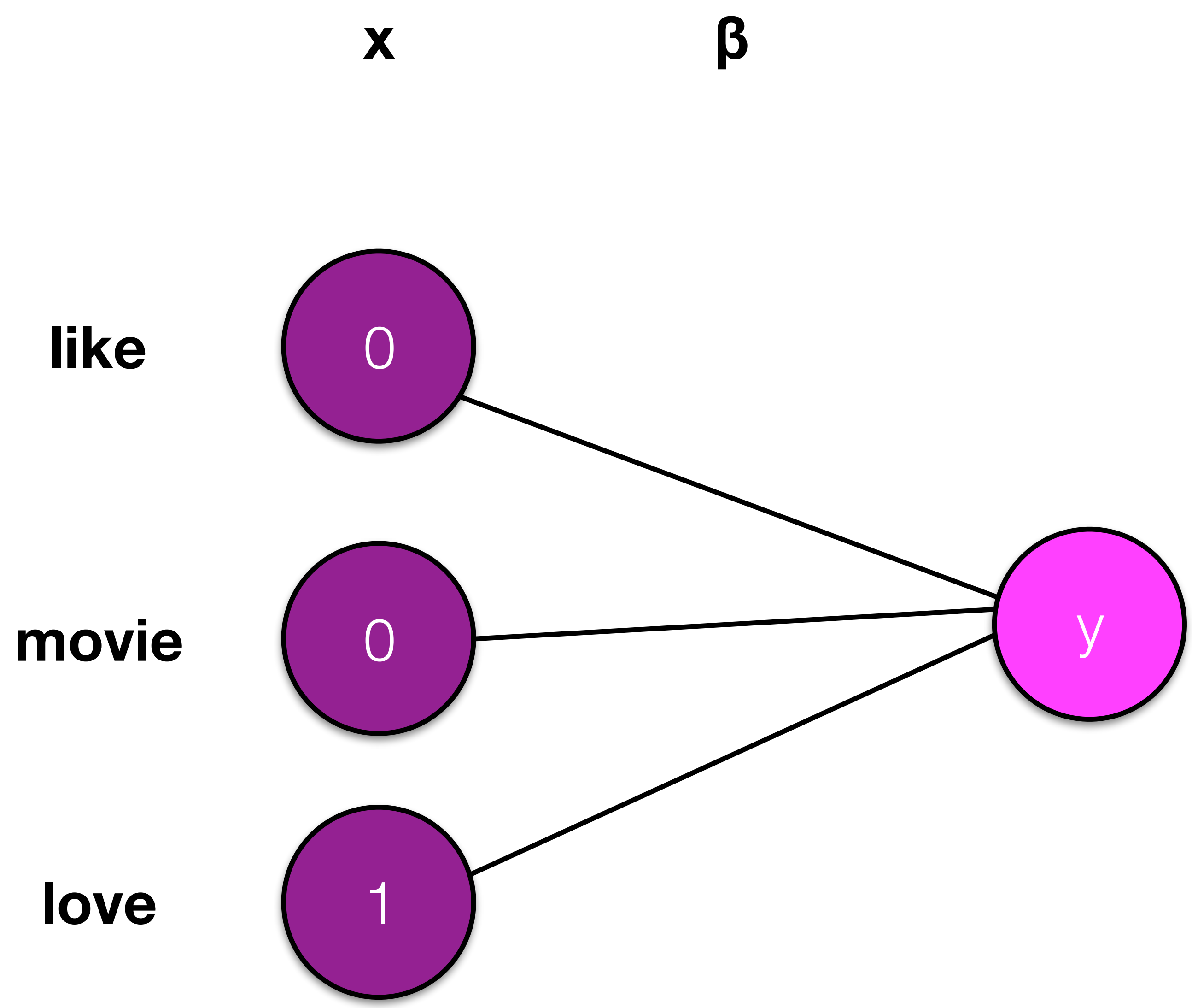
Word embeddings

vocabulary					
…	wine	beer	cat	dog	…
	4.1	4.2	0.1	0.12	
	-0.9	-0.7	0.3	0.28	


y
dog
cat
puppy
x
wine
beer

DISTRIBUTED REPRESENTATIONS [1]

Geoffrey E. Hinton
Computer Science Department
Carnegie-Mellon University
Pittsburgh PA 15213


x
β
like
0
movie
0
love
1
y

x = feature vector

Feature	Value
movie	0
sad	0
funny	0
film	0
love	1
hate	0
it	0
boring	0

β = coefficients

Feature	β
movie	0.1
sad	-6.8
funny	1.4
film	0.3
love	8.7
hate	-7.9
it	0.01
boring	-1.7

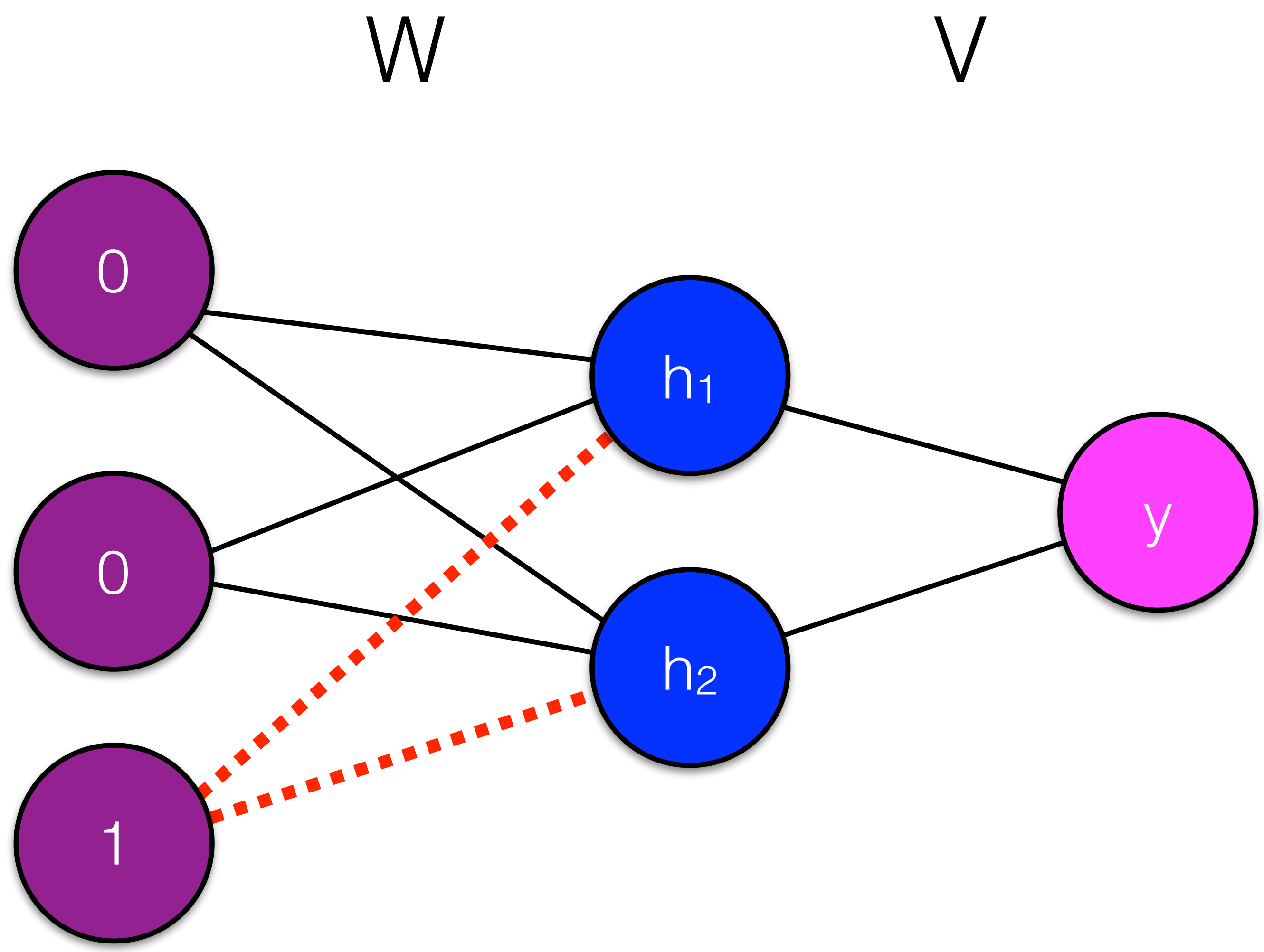
x = feature vector

Feature	Value
movie	0
sad	0
funny	0
film	0
love	1
hate	0
it	0
boring	0

“local representation”

Distributed representation

"Each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities" (Hinton 1984)



W		
like	movie	love
4.1	0.7	0.1
-0.9	1.3	0.3

- Output: low-dimensional representation of words directly read off from the weight matrices.

Dimensionality reduction

…	…
the	1
a	0
an	0
for	0
in	0
on	0
dog	0
cat	0
…	…

the is a point in V-dimensional space

the

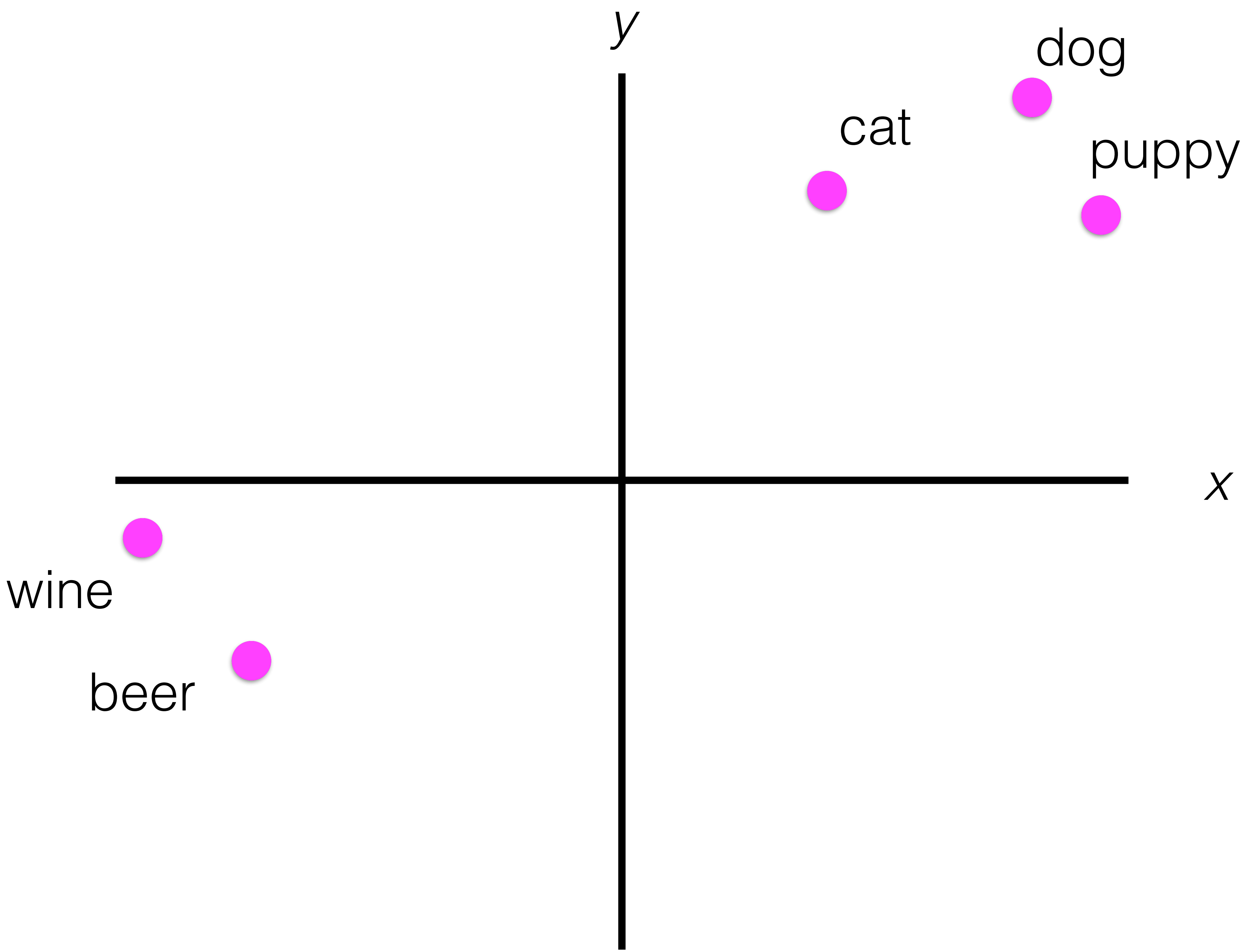
4.1
-0.9

the is a point in 2-dimensional space

Similarity

People are good at generalizing newly acquired knowledge. If you learn a new fact about an object, your expectations about other similar objects tend to change. If, for example, you learn that chimpanzees like onions you will probably raise your estimate of the probability that gorillas like onions. In a network that uses distributed representations, this kind of generalization is automatic. The new knowledge about chimpanzees is incorporated by modifying some of the connection strengths so as to alter the causal effects of the distributed pattern of activity that represents chimpanzees.[2] The modifications automatically change the causal effects of all similar activity patterns. So if the representation of gorillas is a similar activity pattern over the same set of units, its causal effects will be changed in a similar way.

Hinton 1984


y
dog
cat
puppy
x
wine
beer

Distributed Representations

- Not unique to language; any feature can have a distributed representation in this context.
- Inputs that have similar relationships to their outputs will have similar representations (shared strength in learning, generalizability)

Lexical semantics

"You shall know a word by the company it keeps"

[Firth 1957]

Dense vectors from prediction

- Learning low-dimensional representations of words by framing a predicting task: using context to predict words in a surrounding window

- Transform this into a supervised prediction problem

Classification

A mapping *h* from input data x (drawn from instance space $\mathcal{X}$) to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{X}$ = set of all documents
$\mathcal{Y}$ = {english, mandarin, greek, …}

x = a single document
y = ancient greek

Classification

A mapping h from input data x (drawn from instance space $\mathcal{X}$) to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{Y}$ = {the, of, a, dog, iphone, …}

x = (context)
y = word

Dense vectors from prediction

Skipgram model (Mikolov et al. 2013): given a single word in a sentence, predict the words in a context window around it.

a cocktail with gin and seltzer

x	y
gin	a
gin	cocktail
gin	with
gin	and
gin	seltzer

Window size = 3

Logistic regression

$$P(y = 1 \mid x, \beta) = \frac{1}{1 + \exp\left(-\sum_{i=1}^{F} x_i \beta_i\right)}$$

output space $\mathcal{Y} = \{0, 1\}$

x = feature vector

Feature	Value
the	0
and	0
bravest	0
love	0
loved	0
genius	0
not	0
fruit	1
BIAS	1

β = coefficients

Feature	β
the	0.01
and	0.03
bravest	1.4
love	3.1
loved	1.2
genius	0.5
not	-3.0
fruit	-0.8
BIAS	-0.1

Multiclass logistic regression

$$P(Y = y \mid X = x; \beta) = \frac{\exp(x^\top \beta_y)}{\sum_{y' \in \mathcal{Y}} \exp(x^\top \beta_{y'})}$$

output space $\qquad \mathcal{Y} = \{1, \ldots, K\}$

One set of β for each class.

x = feature vector

β = coefficients

Feature	Value
the	0
and	0
bravest	0
love	0
loved	0
genius	0
not	0
fruit	1
BIAS	1

Feature	β_1 k="a"	β_2 k="an"	β_3 k="and"	β_4 k="ant"	β_5 k="anti"
the	1.33	-0.80	-0.54	0.87	0
and	1.21	-1.73	-1.57	-0.13	0
bravest	0.96	-0.05	0.24	0.81	0
love	1.49	0.53	1.01	0.64	0
loved	-0.52	-0.02	2.21	-2.53	0
genius	0.98	0.77	1.53	-0.95	0
not	-0.96	2.14	-0.71	0.43	0
fruit	0.59	-0.76	0.93	0.03	0
BIAS	-1.92	-0.70	0.94	-0.63	0

Language Model

- We can use multi class logistic regression for predicting words in context by treating the vocabulary as the output space

$$\mathcal{Y} = \mathcal{V}$$

Dimensionality reduction

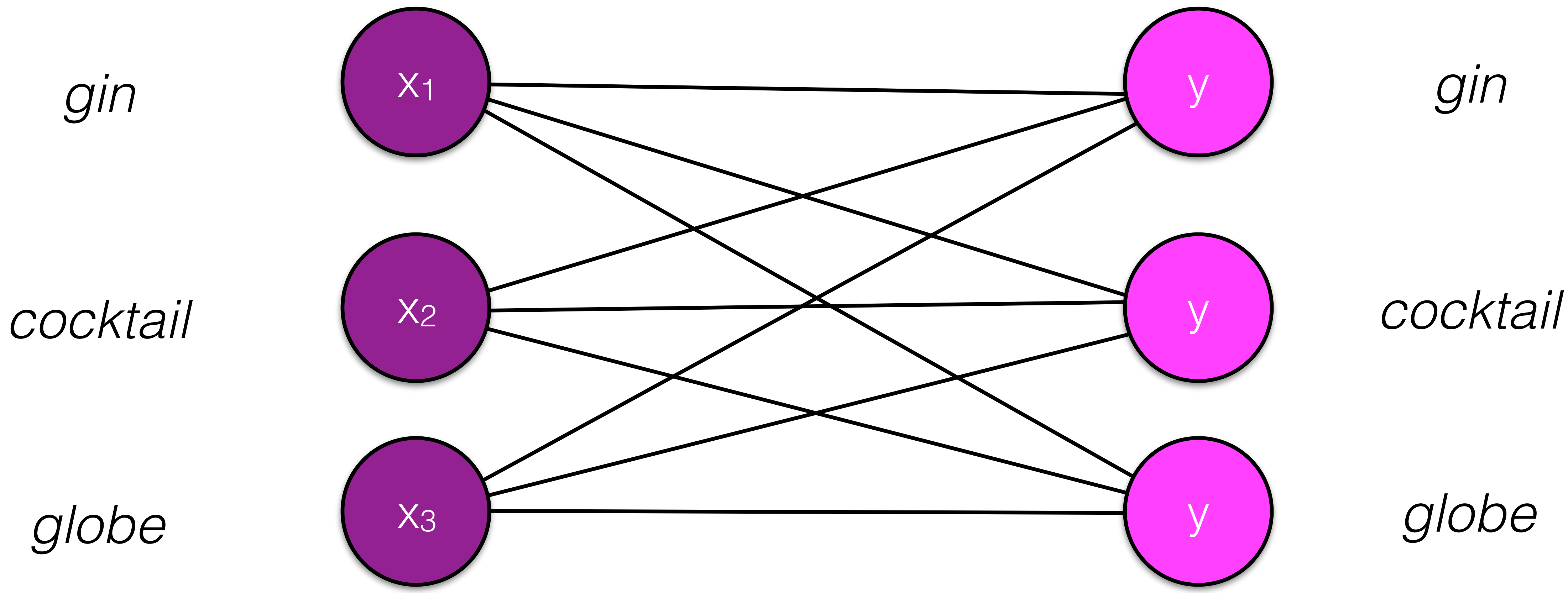
…	…
the	1
a	0
an	0
for	0
in	0
on	0
dog	0
cat	0
…	…

the is a point in V-dimensional space

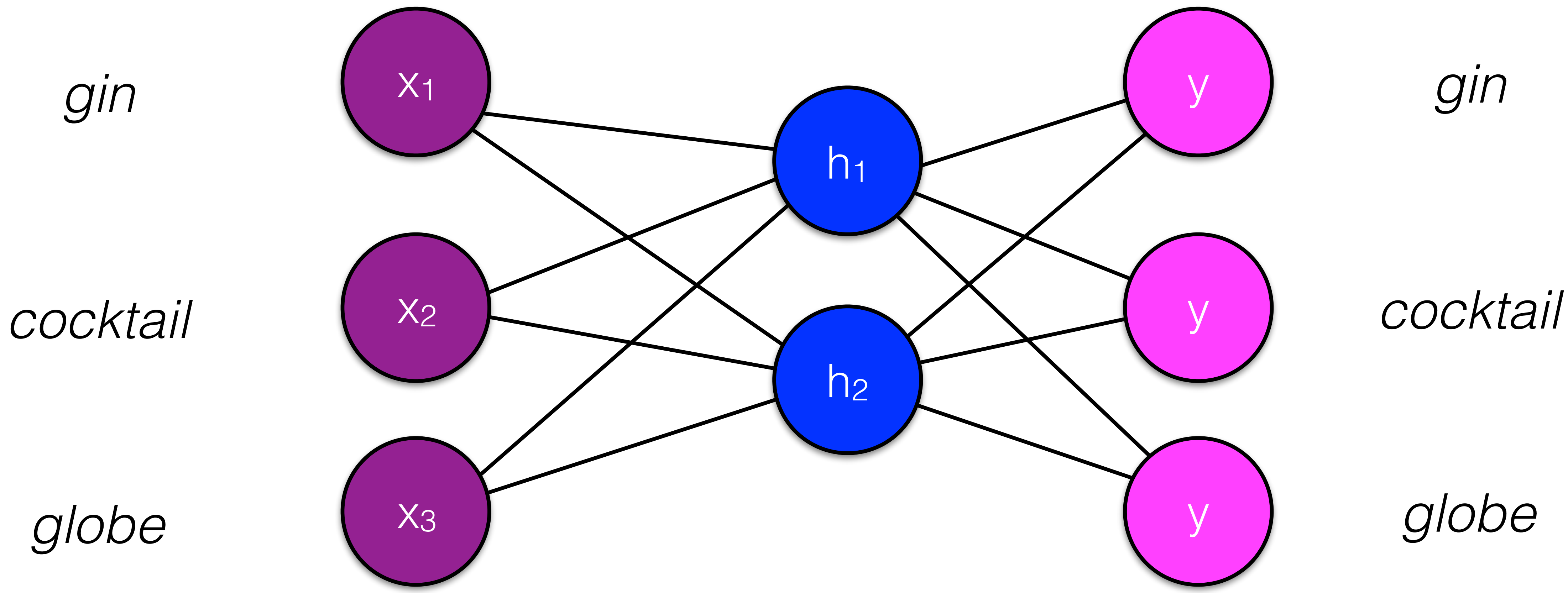
the

4.1
-0.9

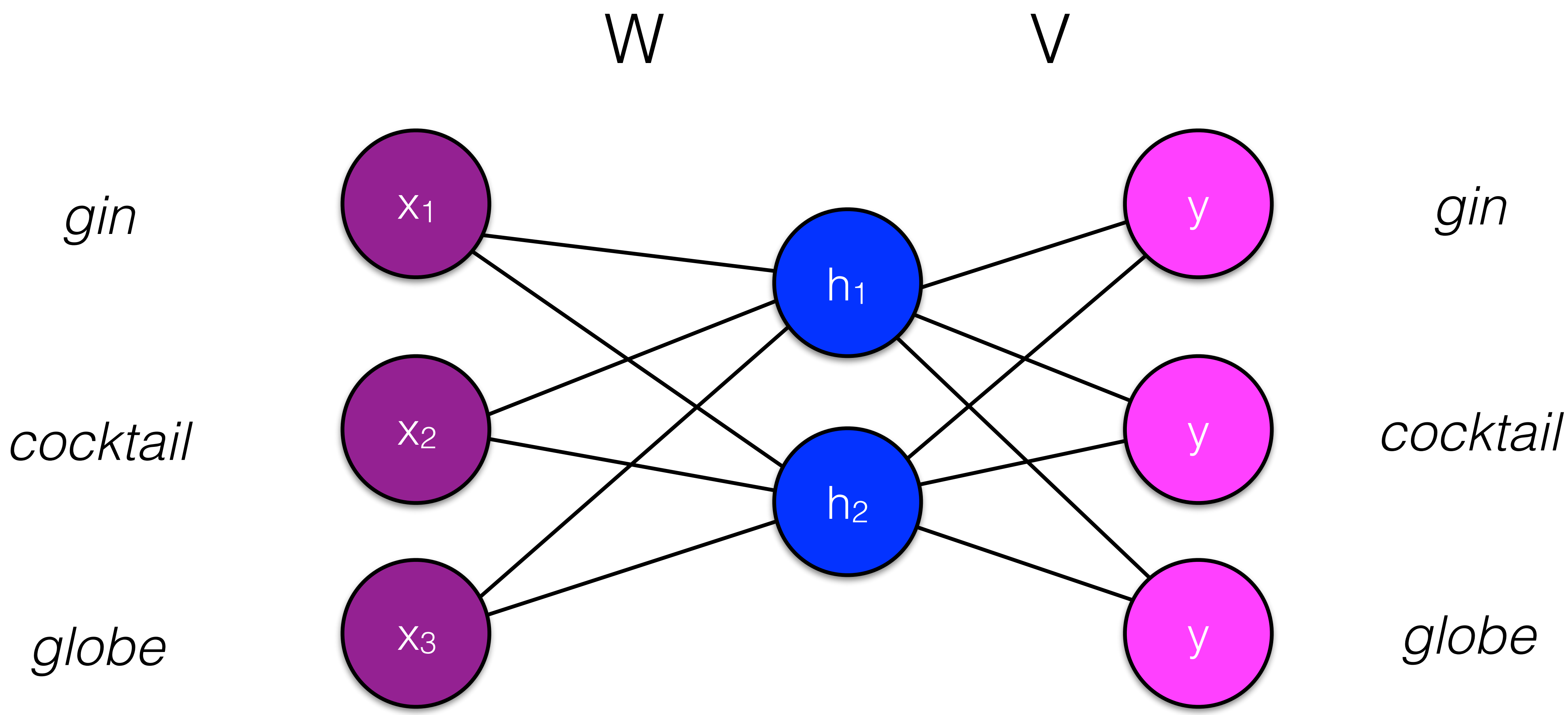
the is a point in 2-dimensional space



Feature	β_1 k="gin"	β_2 k="cocktail"	β_3 k="globe"
gin	1.33	-0.80	-0.54
cocktail	1.21	-1.73	-1.57
globe	0.96	-0.05	0.24



gin
cocktail
globe
x1
x2
x3
h1
h2
y
y
y
gin
cocktail
globe

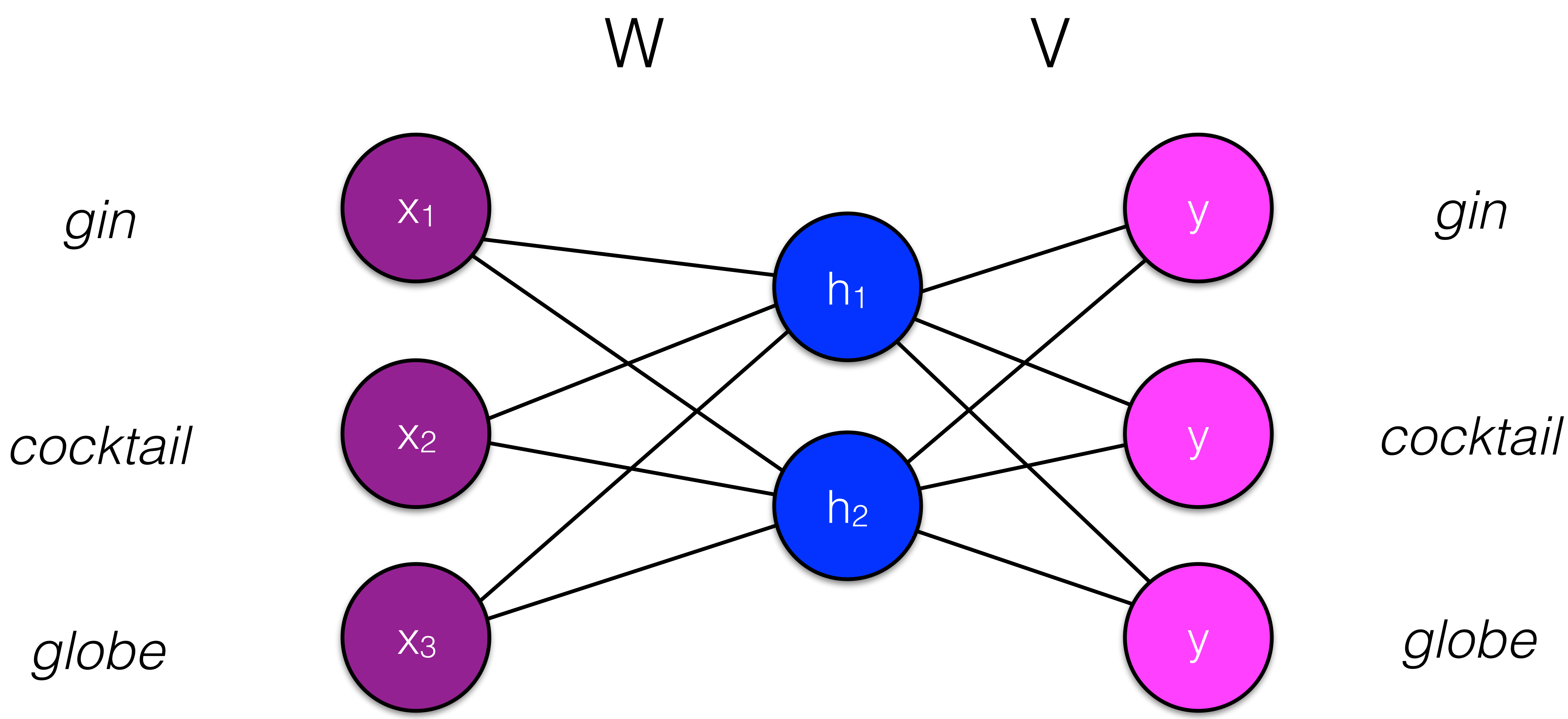


	x
gin	0
cocktail	1
globe	0

W	
-0.5	1.3
0.4	0.08
1.7	3.1

V		
4.1	0.7	0.1
-0.9	1.3	0.3

y
1
0
0



Only one of the inputs is nonzero.

= the inputs are really $W_{cocktail}$

W	
-0.5	1.3
0.4	0.08
1.7	3.1

V		
4.1	0.7	0.1
-0.9	1.3	0.3

x	W	
	0.13	0.56
	-1.75	0.07
	0.80	1.19
	-0.11	1.38
	-0.62	-1.46
	-1.16	-1.24
	0.99	-0.26
	-1.46	-0.85
	0.79	0.47
	0.06	-1.21
	-0.31	0.00
1	-1.01	-2.52
	-1.50	-0.14
	-0.14	0.01
	-0.13	-1.76
	-1.08	-0.56
	-0.17	-0.74
	0.31	1.03
	-0.24	-0.84
	-0.79	-0.18

$$x^\top W =$$

-1.01	-2.52

This is the embedding of the context

Word embeddings

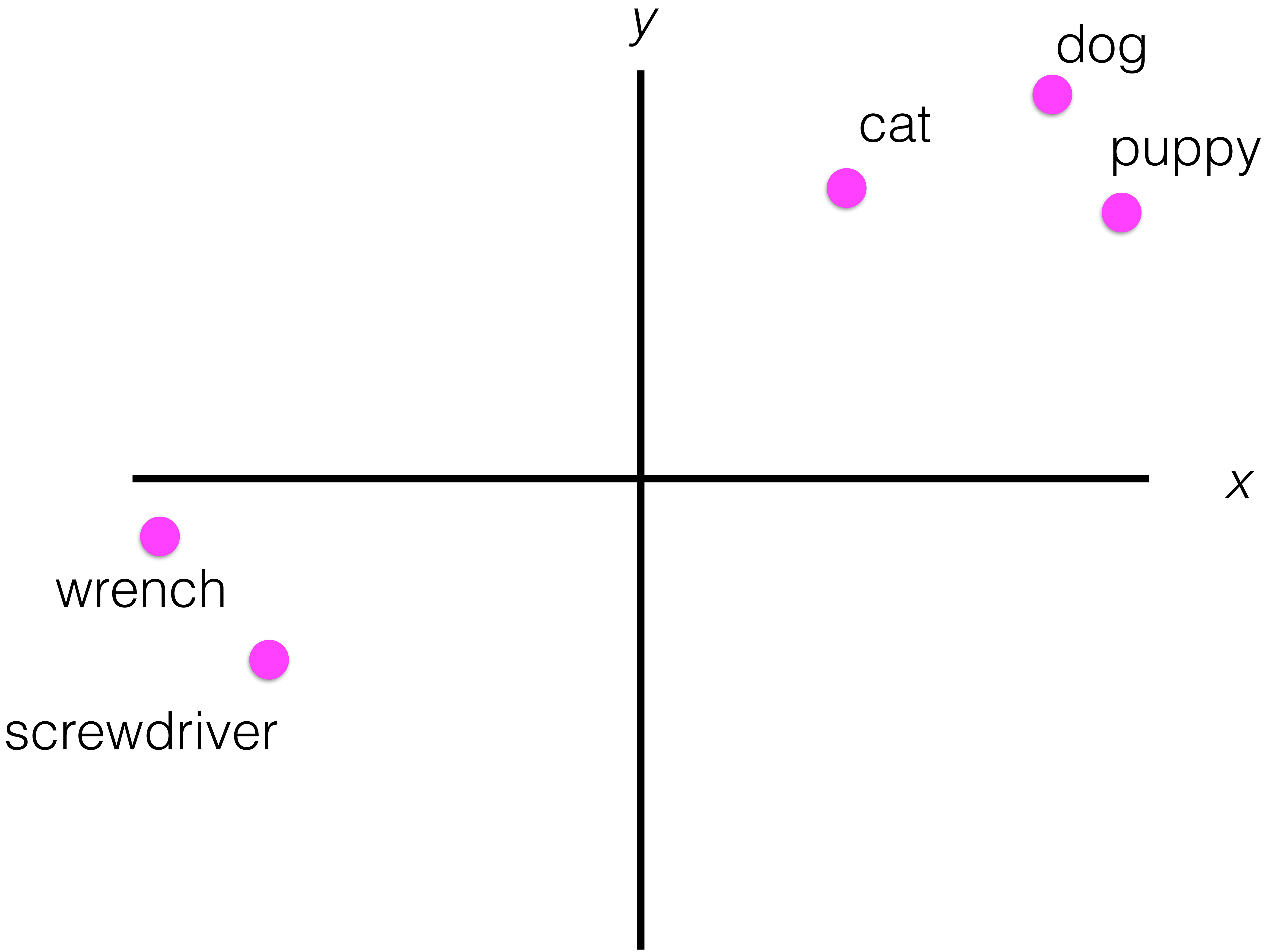
- Can you predict the output word from a vector representation of the input word?
- Rather than seeing the input as a one-hot encoded vector specifying the word in the vocabulary we're conditioning on, we can see it as indexing into the appropriate row in the weight matrix W

Word embeddings

- Similarly, V has one H-dimensional vector for each element in the vocabulary (for the words that are being predicted)

V			
gin	cocktail	cat	globe
4.1	0.7	0.1	1.3
-0.9	1.3	0.3	-3.4

This is the embedding of the word


y
dog
cat
puppy
x
wrench
screwdriver

- Why this behavior? *dog*, *cat* show up in similar positions

the	black	cat	jumped	on	the	table
the	black	dog	jumped	on	the	table
the	black	puppy	jumped	on	the	table
the	black	skunk	jumped	on	the	table
the	black	shoe	jumped	on	the	table

- Why this behavior? *dog*, *cat* show up in similar positions

the	black	[0.4, 0.08]	jumped	on	the	table
the	black	[0.4, 0.07]	jumped	on	the	table
the	black	puppy	jumped	on	the	table
the	black	skunk	jumped	on	the	table
the	black	shoe	jumped	on	the	table

To make the same predictions, these numbers need to be close to each other.

Dimensionality reduction

…	…
the	1
a	0
an	0
for	0
in	0
on	0
dog	0
cat	0
…	…

the is a point in V-dimensional space;
representations for all words are completely independent

the

4.1
-0.9

the is a point in 2-dimensional space
representations are now structured

Analogical inference

- Mikolov et al. 2013 show that vector representations have some potential for analogical reasoning through vector arithmetic.

$$\text{apple} - \text{apples} \approx \text{car} - \text{cars}$$

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

Mikolov et al., (2013), “Linguistic Regularities in Continuous Space Word Representations” (NAACL)

Science AAAS

REPORT

Semantics derived automatically from language corpora contain human-like biases

Aylin Caliskan[1,*], Joanna J. Bryson[1,2,*], Arvind Narayanan[1,*]

+ See all authors and affiliations

Science 14 Apr 2017:
Vol. 356, Issue 6334, pp. 183-186
DOI: 10.1126/science.aal4230

Low-dimensional distributed representations

- Low-dimensional, dense word representations are extraordinarily powerful (and are arguably responsible for much of gains that neural network models have in NLP).

- Lets your representation of the input share statistical strength with words that behave similarly in terms of their distributional properties (often synonyms or words that belong to the same class).

Two kinds of “training” data

- The labeled data for a specific task (e.g., labeled sentiment for movie reviews): ~ 2K labels/reviews, ~1.5M words → used to train a supervised model

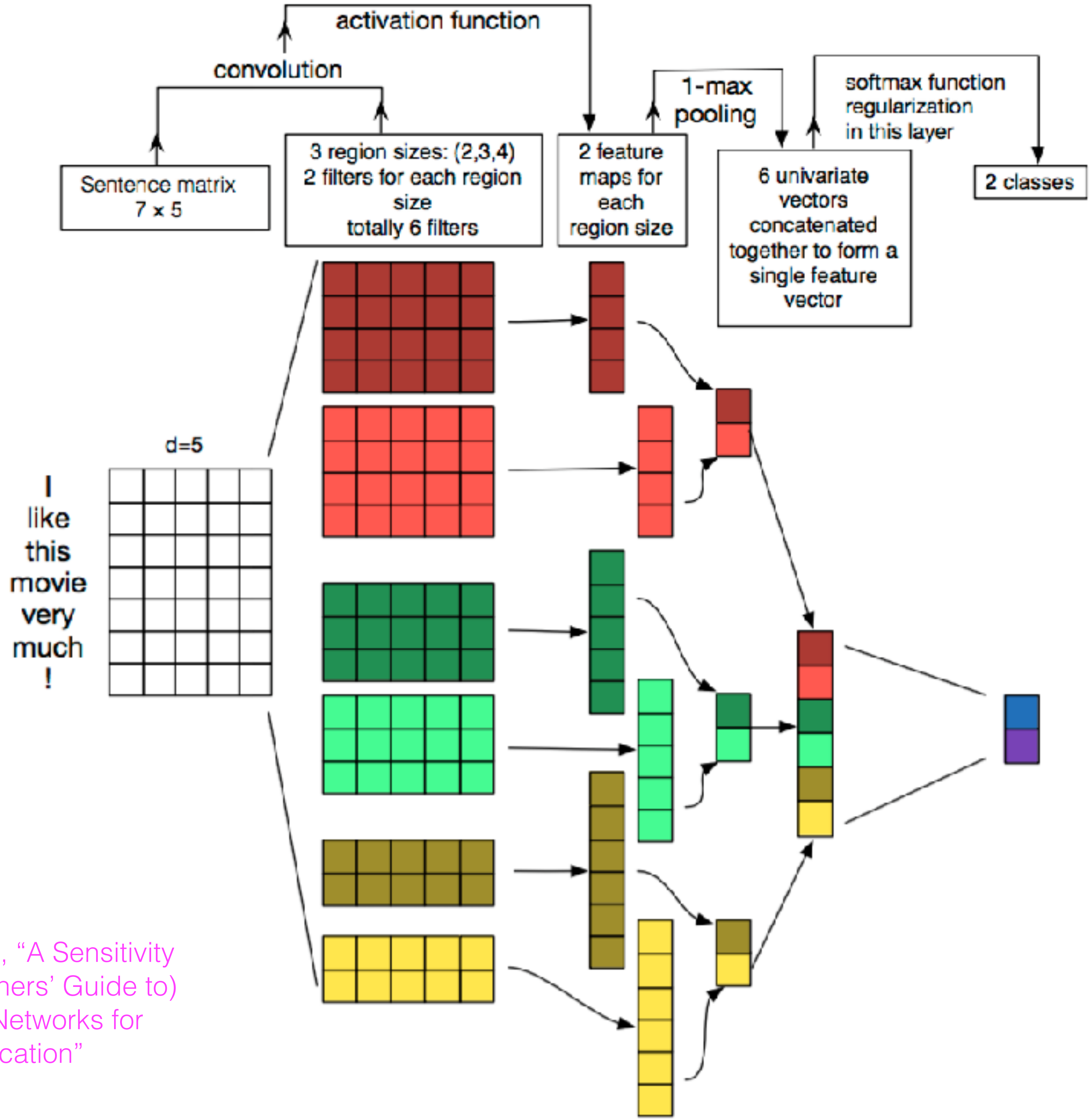
- General text (Wikipedia, the web, books, etc.), ~ trillions of words → used to train word distributed representations

	1	2	3	4	…	50
the	0.418	0.24968	-0.41242	0.1217	…	-0.17862
,	0.013441	0.23682	-0.16899	0.40951	…	-0.55641
.	0.15164	0.30177	-0.16763	0.17684	…	-0.31086
of	0.70853	0.57088	-0.4716	0.18048	…	-0.52393
to	0.68047	-0.039263	0.30186	-0.17792	…	0.13228
…	…	…	…	…	…	…
chanty	0.23204	0.025672	-0.70699	-0.04547	…	0.34108
kronik	-0.60921	-0.67218	0.23521	-0.11195	…	0.85632
rolonda	-0.51181	0.058706	1.0913	-0.55163	…	0.079711
zsombor	-0.75898	-0.47426	0.4737	0.7725	…	0.84014
sandberger	0.072617	-0.51393	0.4728	-0.52202	…	0.23096

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

Using dense vectors

- In neural models (CNNs, RNNs, LM), replace the V-dimensional sparse vector with the much smaller K-dimensional dense one.
- Can also take the derivative of the loss function with respect to those representations to optimize for a particular task.

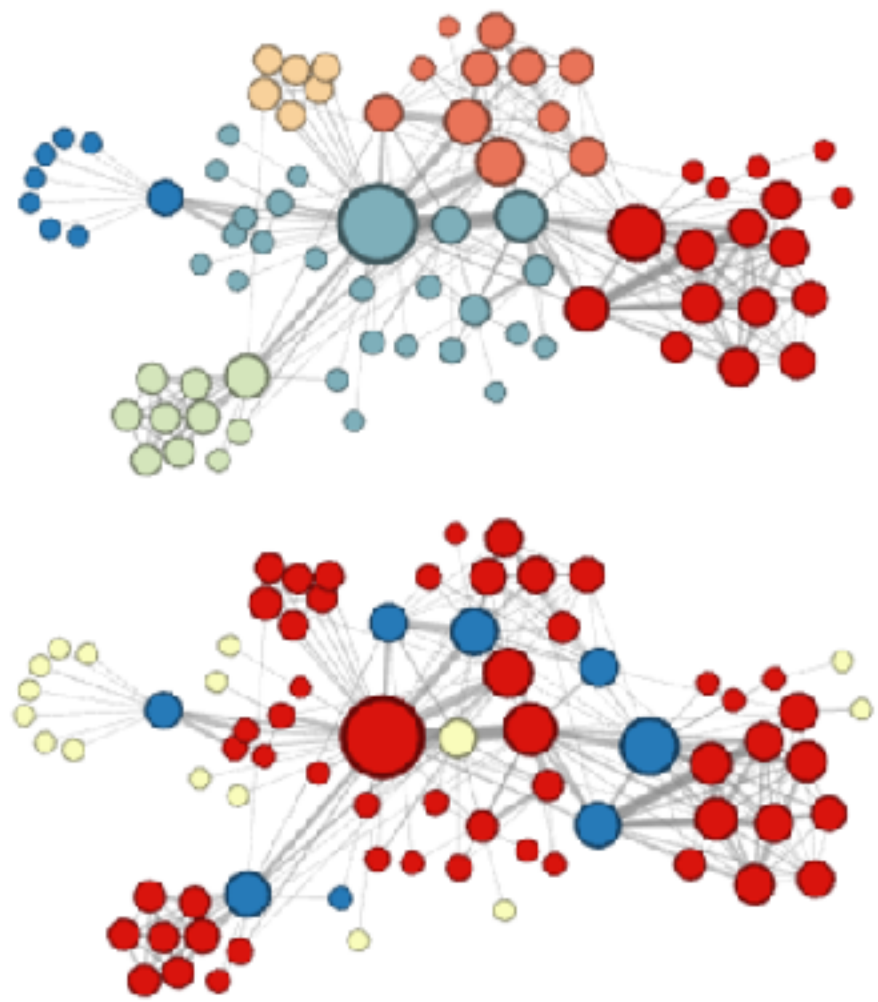


Zhang and Wallace 2016, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification”

emoji2vec

Eisner et al. (2016), “emoji2vec: Learning Emoji Representations from their Description”

node2vec



Grover and Leskovec (2016), “node2vec: Scalable Feature Learning for Networks”

Trained embeddings

- Word2vec
  https://code.google.com/archive/p/word2vec/

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

- Levy/Goldberg dependency embeddings
  https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/

7.embeddings/ WordEmbeddings.ipynb

- Training your own word embeddings using Gensim
- Explore Glove embeddings for finding nearest neighbors and analogies
- Which analogies work and which ones fail? Report one of each at end of class.