CSCI 5541: Natural Language Processing

Lecture 5: Distributional Semantics and Word Embeddings

Dongyeop Kang (DK), University of Minnesota

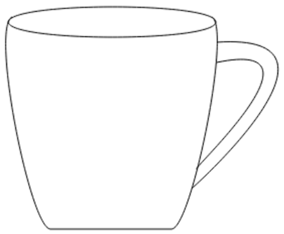
dongyeop@umn.edu | twitter.com/dongyeopkang | dykang.github.io

computer science & engineering

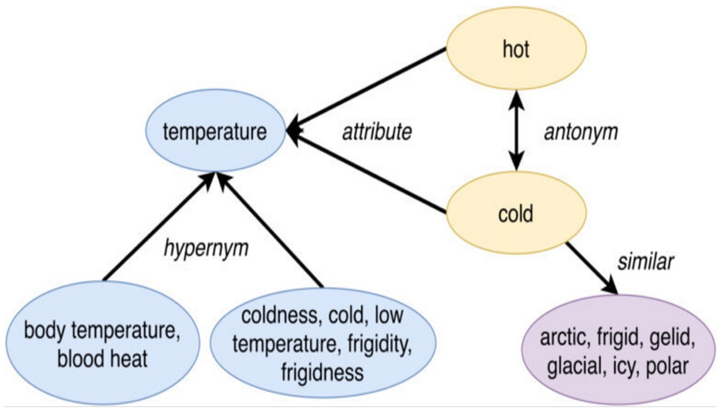
MINNESOTA NLP EST. 2021

University of Minnesota
Driven to Discover®

Decompositional semantics

Shape:  **Color**: green, blue, black, etc

Ontological semantics



Distributional semantics

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

Firth, J. R. 1957:11

```
001.☐ t but of different schools. Cook had discovered a BEEF in his possession a few days earlier and, whe
002.☐ ity to available canned pork products. Tests with BEEF have been largely unsuccessful because of the
003.☐ ermaster Corps program is to find the reasons for BEEF's low palatability and means of overcoming it
004.☐ rads can extend the shelf life (at 35 F) of fresh BEEF from 5 days to 5 or 6 weeks. However, the pro
005.☐ radiation blanching process discolors the treated BEEF and liquid accumulates in prepackaged cuts. C
006.☐ nd liquid accumulates in prepackaged cuts. Cooked BEEF irradiated in the absence of oxygen assumes a
007.☐ the improvised counter of boards laid across two BEEF barrels. There was, of course, no real need t
008.☐ ey of the hut across from him was surmounted by a BEEF barrel with ends knocked out. In this heavy a
009.☐ secret employers their money's worth. A good many BEEF-hungry settlers were accepting the death of W
010.☐ ogrammes and cost-cutting measures are planned to BEEF up performance. Analysts at Paribas are looki
011.☐ rsion into animal feed, produce 600,000 tonnes of BEEF, which alone is worth £1,000 million at whol
012.☐ pay on the French wards. We served them up corned BEEF, cheese, pickles and hot cocoa, and they thre
013.☐ they threw it back at us. "Good gracious, corned BEEF, cheese and bread and butter, they were lucky
014.☐ Greece) Ingredients 1 small packet feta cheese 2 BEEF tomatoes ½ cucumber 1 small jar stoned
015.☐ for the past twenty minutes!" Waiter! Waiter! The BEEF surprise was lovely, but what's the surprise?
016.☐ as hands who rode for a Mr. Wolgast who supplied BEEF to the reservation up at San Carlos. I would
017.☐ "Every week or so I'd see them come in for their BEEF ration. And they're allowed to hunt. They can
018.☐ ll specialize in steaks, chops, chicken and prime BEEF as well as Tom's favorite dish, stuffed shrim
019.☐ close, she said. She had raised a calf, grown it BEEF-fat. She had, with her own work-weary hands,
020.☐ and get ready to bear. She was ready to kill the BEEF, dress it out, and with vegetables from her g
021.☐ 't know what to say. He did say she could get her BEEF and vegetables in cans this summer. He did sa
022.☐ 12. _HAMBURGER PATTIES WITH NUTS_ 1 pound ground BEEF 2 teaspoons grated onion Dash of pepper 1/2 t
023.☐ a highly competitive business more profitable for BEEF, dairy, and sheep men. The target chart quick
024.☐ cle assumes that the rations you are feeding your BEEF, dairy cattle, and sheep are adequately balan
025.☐ -infective properties of this drug. _HOW TO FEED: BEEF CATTLE (FINISHING RATION)_ - To increase rate
026.☐ in the prevention of liver abscesses in feed-lot BEEF cattle. Prevention of bacterial pneumonia, sh
027.☐ founder, and in controlling scours. _HOW TO FEED: BEEF AND DAIRY CALVES_ - 0.2 gram Dynafac per head
028.☐ gain and improves feed efficiency. _HOW TO FEED: BEEF CATTLE_ - 10 milligrams of diethylstilbestrol
029.☐ oves growth rate and feed efficiency of fattening BEEF animals. _HOW TO FEED:_ At the rate of 2-1/2
```

Beef

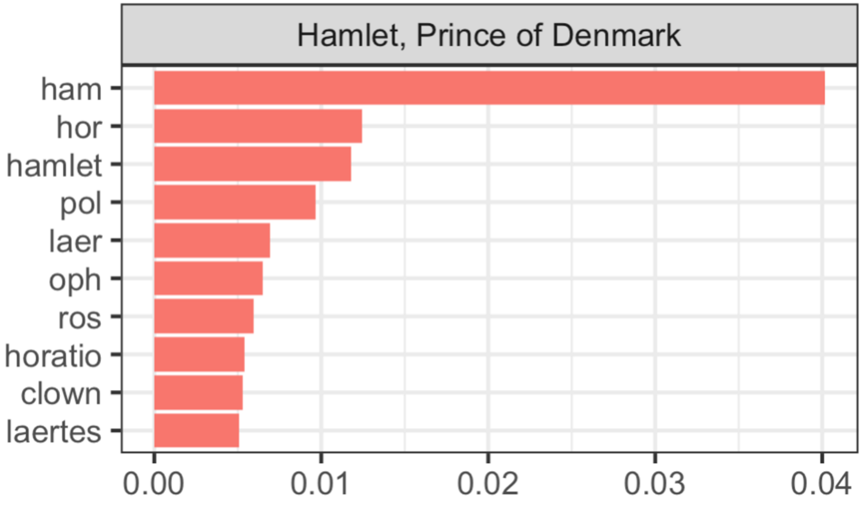
0.7
1.3
-4.5

Recap

	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
…						

$$\cos(x,y) = \frac{\sum_{i=1}^{F} x_i y_i}{\sqrt{\sum_{i=1}^{F} x_i^2}\sqrt{\sum_{i=1}^{F} y_i^2}}$$

$$tfidf(t,d) = tf_{t,d} \times \log\frac{N}{D_t}$$



Distributed prediction-based (type) embeddings

- Count-based method (e.g., Latent Semantic Analysis)
- Prediction-based method (e.g., Skip-gram, CBOW)
- Types of evaluation
- Limitation of word embeddings

Different kinds of encoding "context"

- ~~Count-based~~
  - PMI, TF-IDF
- **Distributed prediction-based (type) embeddings**
  - Word2vec, GloVe, Fasttext
- Distributed contextual (token) embeddings from language models
  - ELMo, BERT, GPT
- Many more variants
  - Multilingual / multi-sense / syntactic embeddings, etc

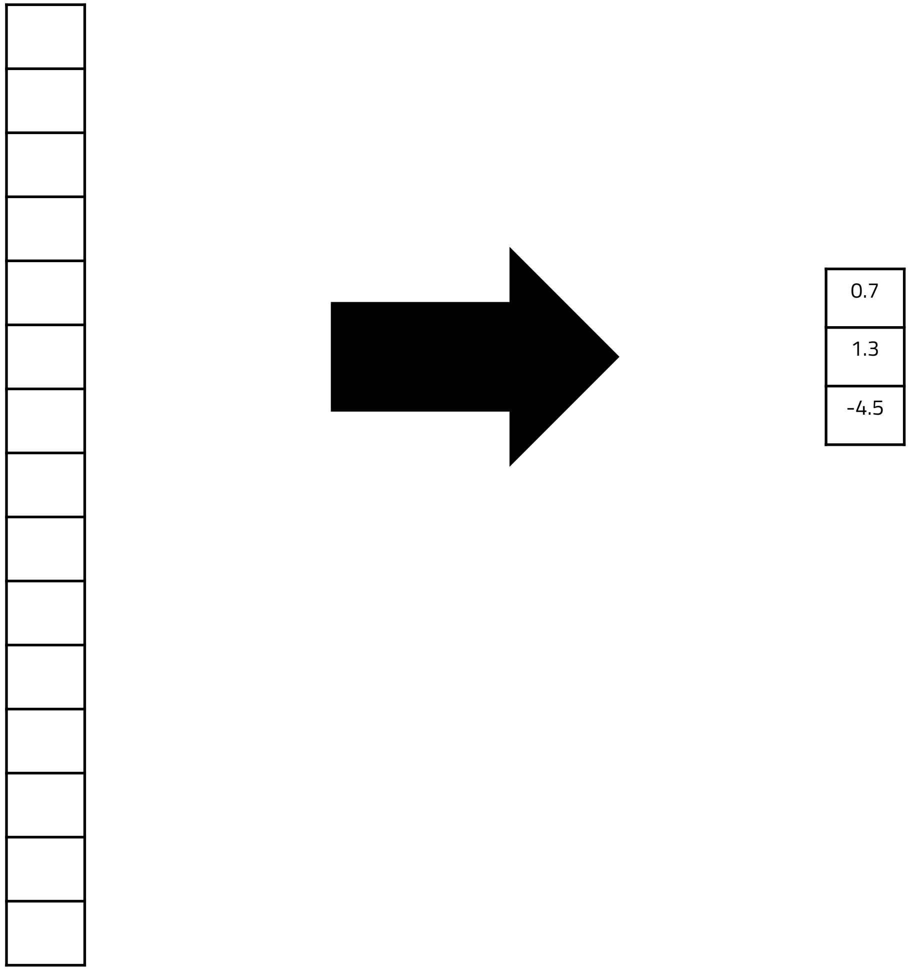
Sparse vectors

“aardvark”

V-dimensional vector, single 1 for the identity of the element

a	0
a	0
aa	0
aal	0
aalii	0
aam	0
Aani	0
aardvark	1
aardwolf	0
...	
zythem	0
Zythia	0
zythum	0
Zyzomys	0
Zyzzogeton	0

Sparse vectors -> Dense vectors



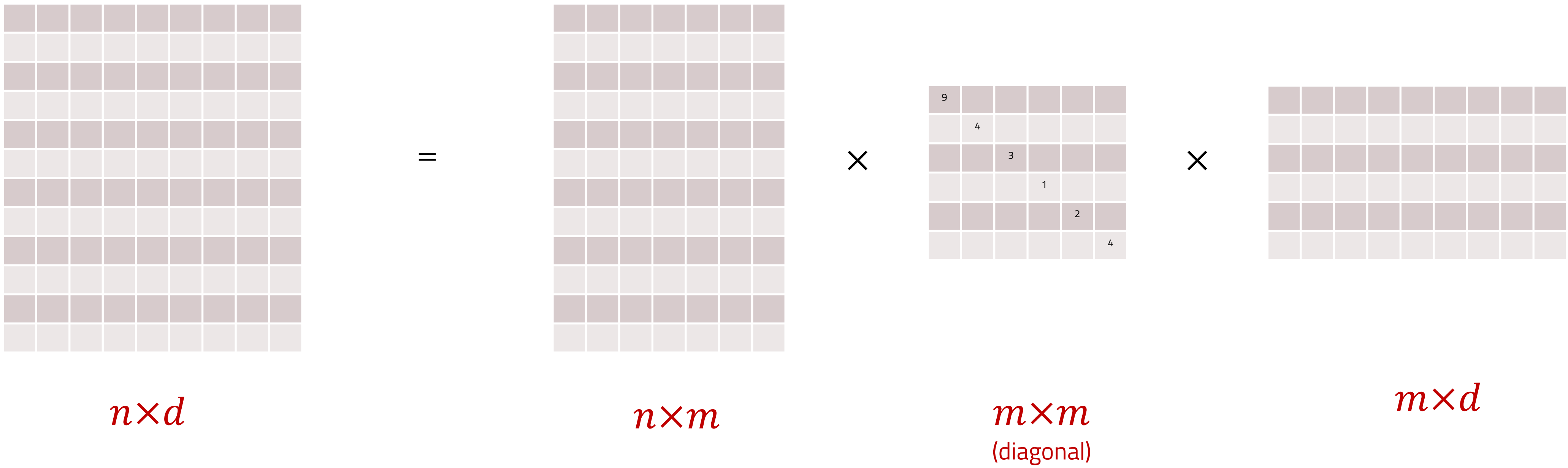
	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41
…						

=

$n \times d$

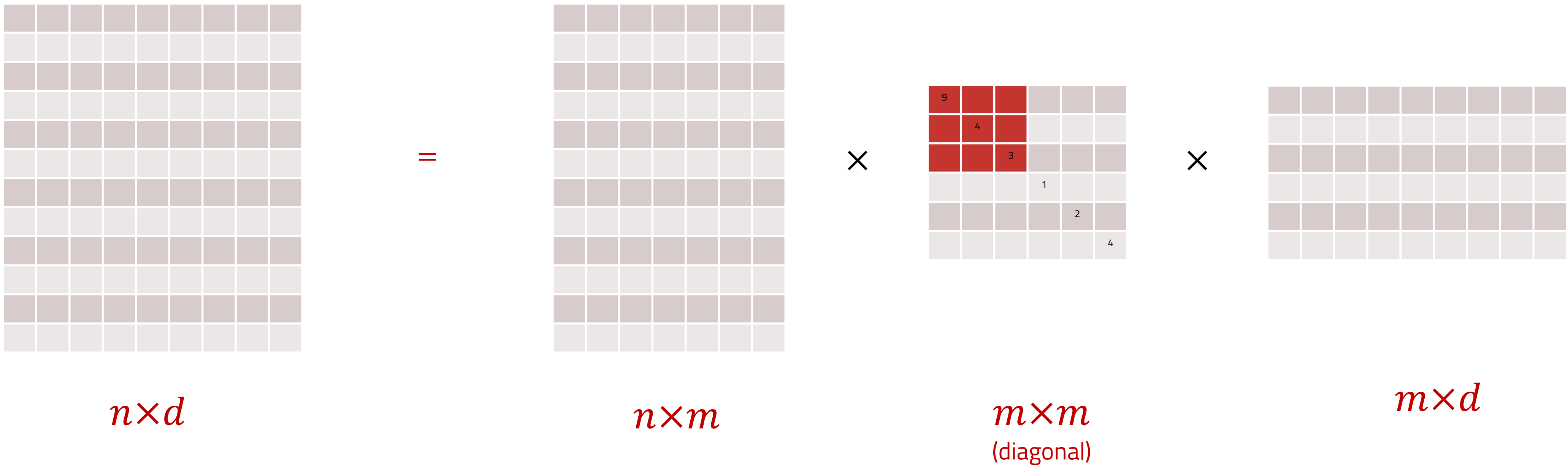
Singular value decomposition (SVD)

- Any $n{\times}d$ matrix X can be decomposed into the product of three matrices
  - where m is the number of linearly independent rows



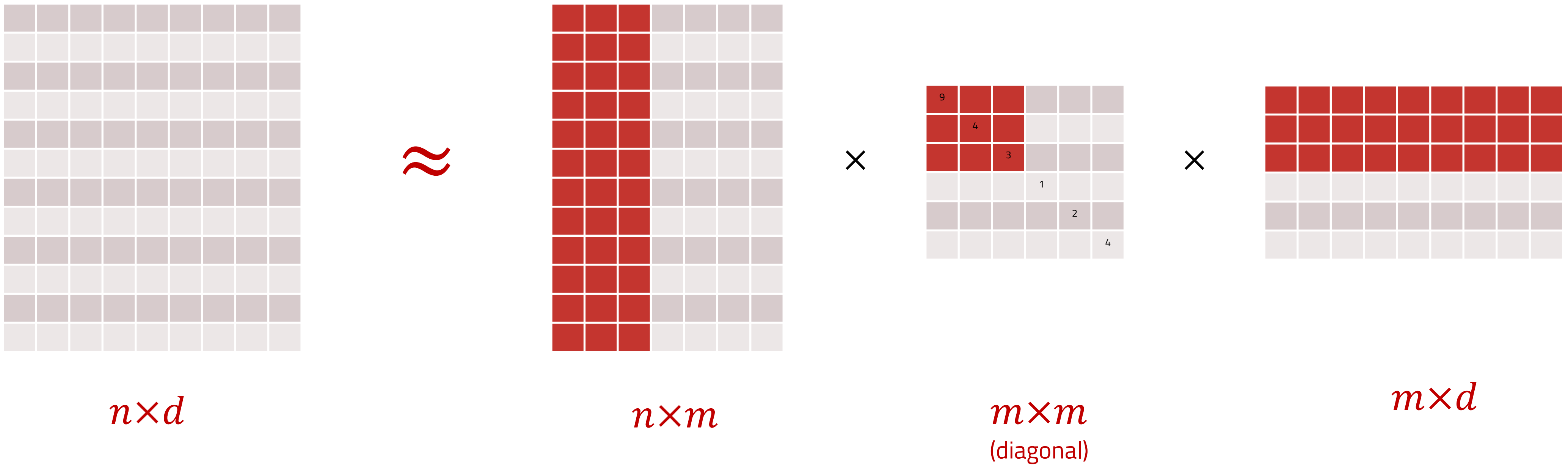
Singular value decomposition (SVD)

- We can approximate the full matrix by only considering the leftmost k terms in the diagonal matrix



Singular value decomposition (SVD)

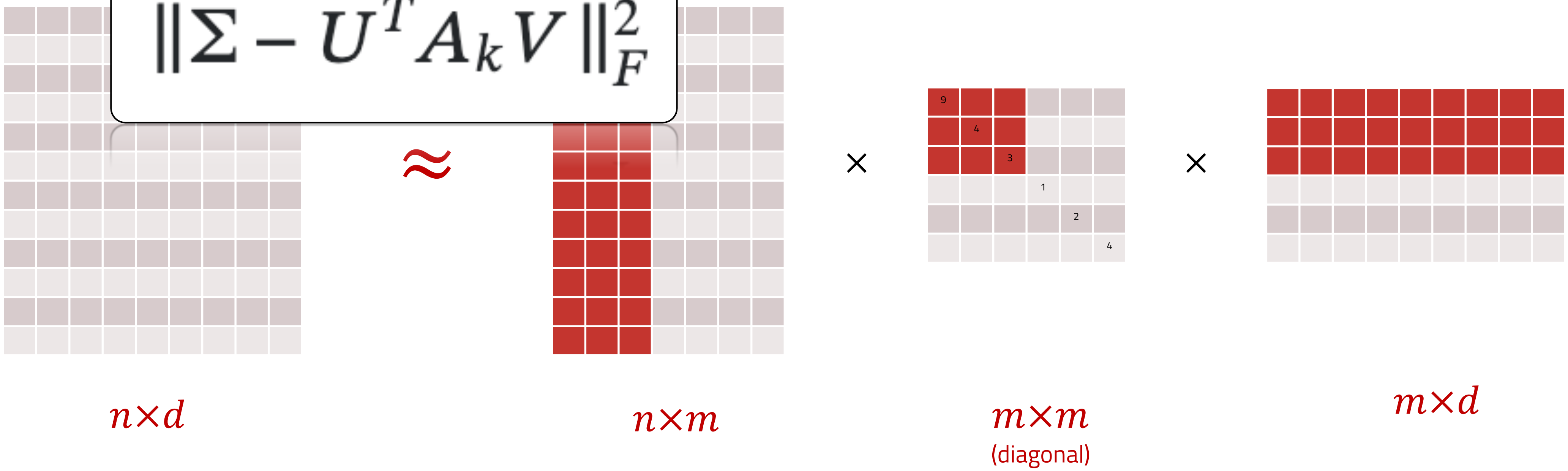
- We can approximate the full matrix by only considering the leftmost k terms in the diagonal matrix



Singular value decomposition (SVD)

- We can approximate the full matrix by only considering the leftmost k terms in the diagonal matrix

$$\|\Sigma - U^T A_k V\|_F^2$$



	Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
knife	1	1	4	2		2
dog				6	12	2
sword	2	2	7	5		5
love	64		135	63		12
like	75	38	34	36	34	41

knife	0.2	0.42	0.22
dog	0.5	1.2	8.6
sword	-0.2	0.7	-2.2
love	9.3	-0.5	0.5
like	0.2	4.3	0.9

$n \times m$

$\times$

0.5		
	0.3	
		2.5

$m \times m$

$\times$

Hamlet	Macbeth	Romeo & Juliet	Richard III	Julius Caesar	Tempest
-0.2	0.7	-2.2	-0.2	0.7	-2.2
-0.2	0.7	-2.2	9.3	-0.5	0.5
9.3	-0.5	0.5	-0.5	0.5	9.3

$m \times d$

Low-dimensional representation
for terms (here 3 dimensions)

Low-dimensional representation
for documents (here 3 dimensions)

knife	0.2	0.42	0.22
dog	0.5	1.2	8.6
sword	-0.2	0.7	-2.2
love	9.3	-0.5	0.5
like	0.2	4.3	0.9

0.5		
	0.3	
		2.5

Hamle t	Macbe th	Romeo & Juliet	Richar d III	Julius Caesar	Tempe st
-0.2	0.7	-2.2	-0.2	0.7	-2.2
-0.2	0.7	-2.2	9.3	-0.5	0.5
9.3	-0.5	0.5	-0.5	0.5	9.3

Latent semantic analysis

- Latent Semantic Analysis/Indexing is this process of applying SVD to the term-document co-occurrence matrix
  - Terms typically weighted by tf-idf
- This is a form of dimensionality reduction
  - for terms, from a D-dimensional sparse vector to a K-dimensional dense one where K << D.
- Similar kinds:
  - Probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999)
  - Nonnegative Matrix Factorization (NMF) (Lee & Seung, 1999)
  - Latent Dirichlet Allocation (LDA) (Blei et al., 2003)

	#1	#2	#3
knife	0.2	0.42	0.22
dog	0.5	1.2	8.6
sword	-0.2	0.7	-2.2
love	9.3	-0.5	0.5
like	0.2	4.3	0.9

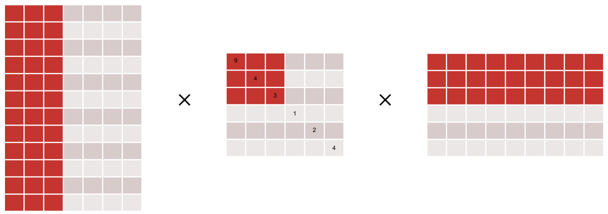
#1	#2	#3	#4
music	how	program	10
film	what	project	30
theater	about	russian	11
mr	their	space	12
this	or	russia	15

(Deerwester et al. 1998)

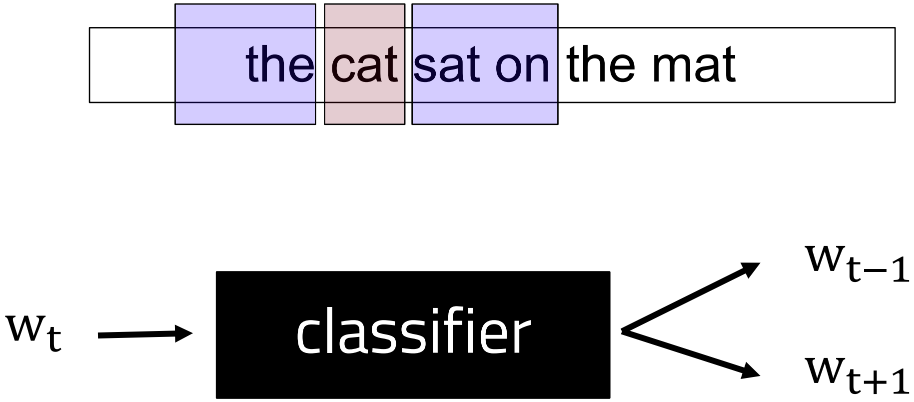
Count-based vs Prediction-based Methods

LSA, **HAL** (Lund & Burgess)
Hellinger-PCA (Rohde et al, Lebret & Collobert)

	Hamlet	Macbeth
knife	1	1
dog		
sword	2	2
love	64	
like	75	38



Skip-gram/CBOW (Mikolovet al)
NLM, HLBL, RNN (Bengioet al; Collobert & Weston; Huang et al; Mnih & Hinton)



Recap: Text Classification

x = “Today’s weather is great”

$$P(y \mid x)$$

y = {positive, negative}

$\hat{y}$ = positive

|Y| = **2**

$x_{<t}$ = “Today’s weather is”

$$P(x_t \mid x_{<t})$$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = great

|X| = **V (vocabulary size)**

$x_{<t}$ = “Today ’s [ ] is great”

$$P(x_t \mid x_{t-2,t-1,\ t+1,\ t+2})$$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = weather

|X| = **V (vocabulary size)**

Recap: Text Classification

x_{t-2} = [ ] .. weather .. ..

x_{t-1} = .. [ ] weather .. ..

$P(x_{t-2} \mid x_t)$

$P(x_{t-1} \mid x_t)$

$x_{<t}$ = "Today 's [ ] is great"

$P(x_t \mid x_{t-2,t-1,\,t+1,\,t+2})$

x_t = {a, aa .. apple .. banana .. great .. good .. zebra ..}

$\hat{x}$ = weather

$P(x_{t+1} \mid x_t)$

$P(x_{t+2} \mid x_t)$

|X| = **V (vocabulary size)**

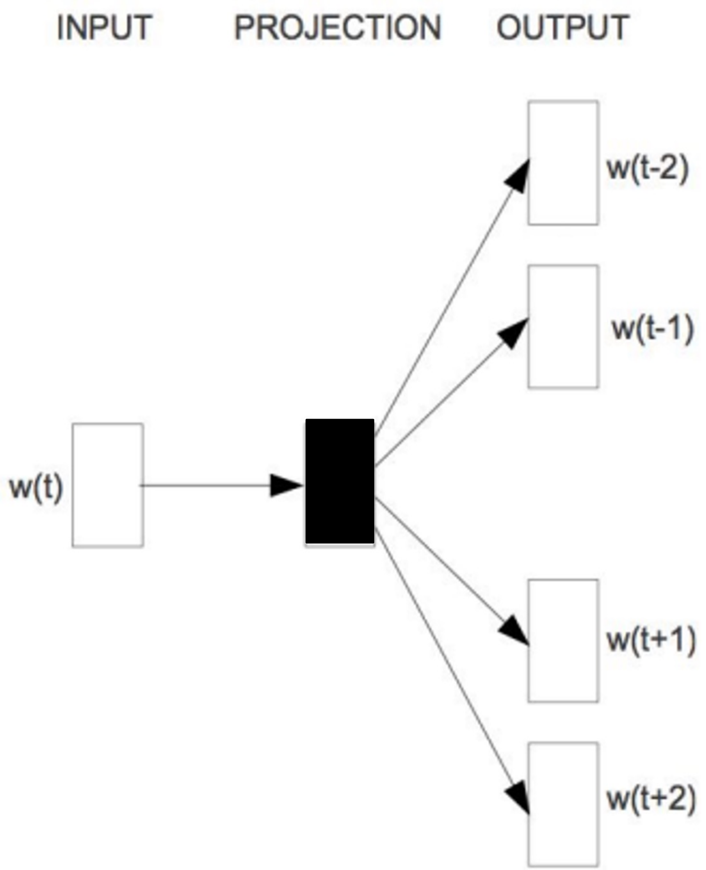
x_{t+1} = ... weather [ ] ..

x_{t+2} = ... weather .. [ ]

Dense vectors from prediction (not count)

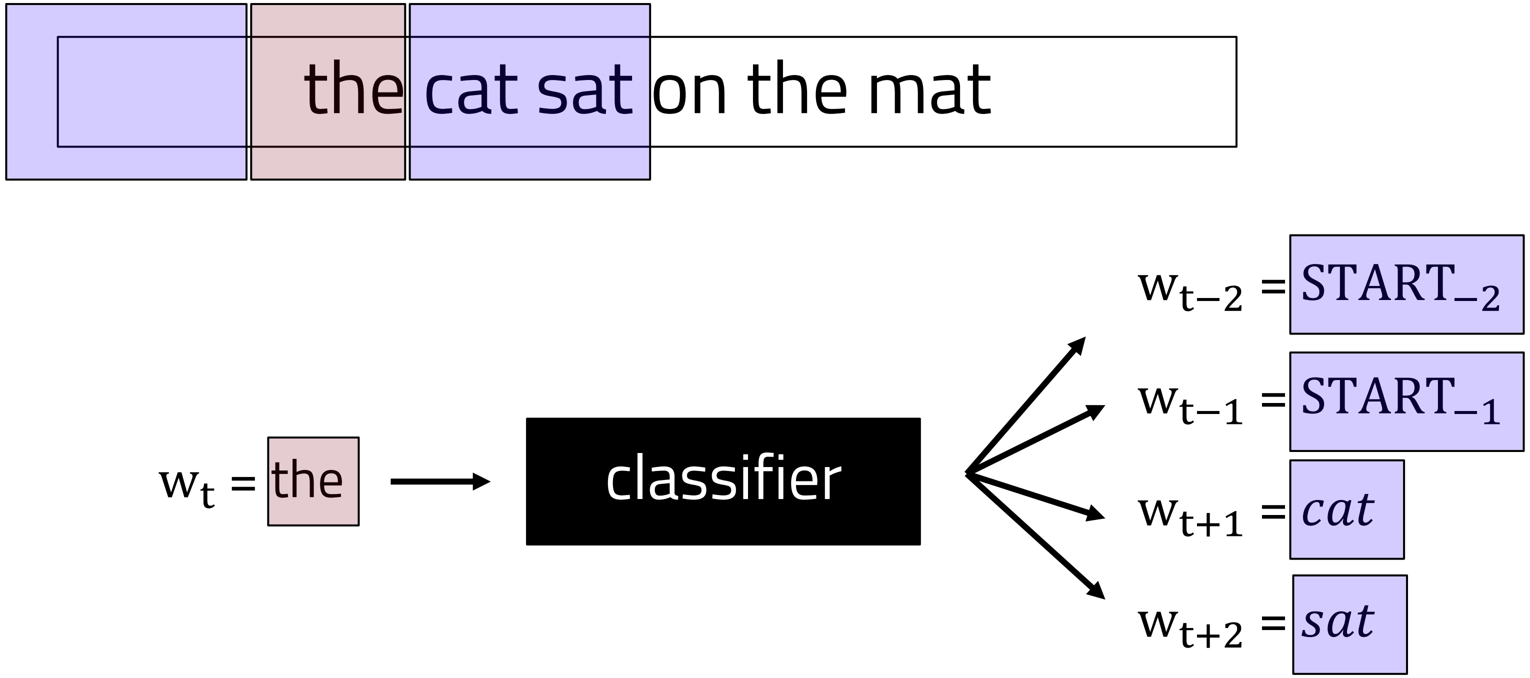
the cat sat on the mat

Skipgram model: given a single word in a sentence, predict the words in a context window around it.



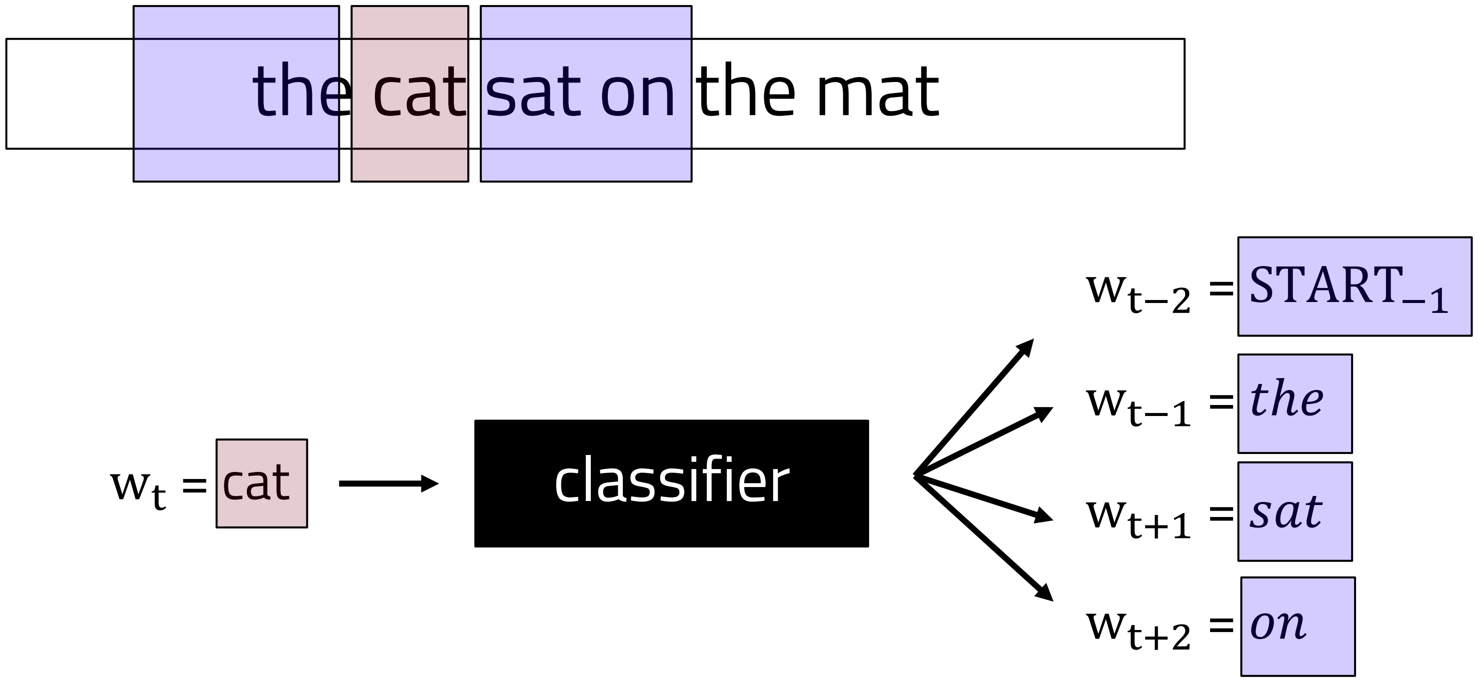
(Mikolove et al., 14)

Dense vectors from prediction (not count)



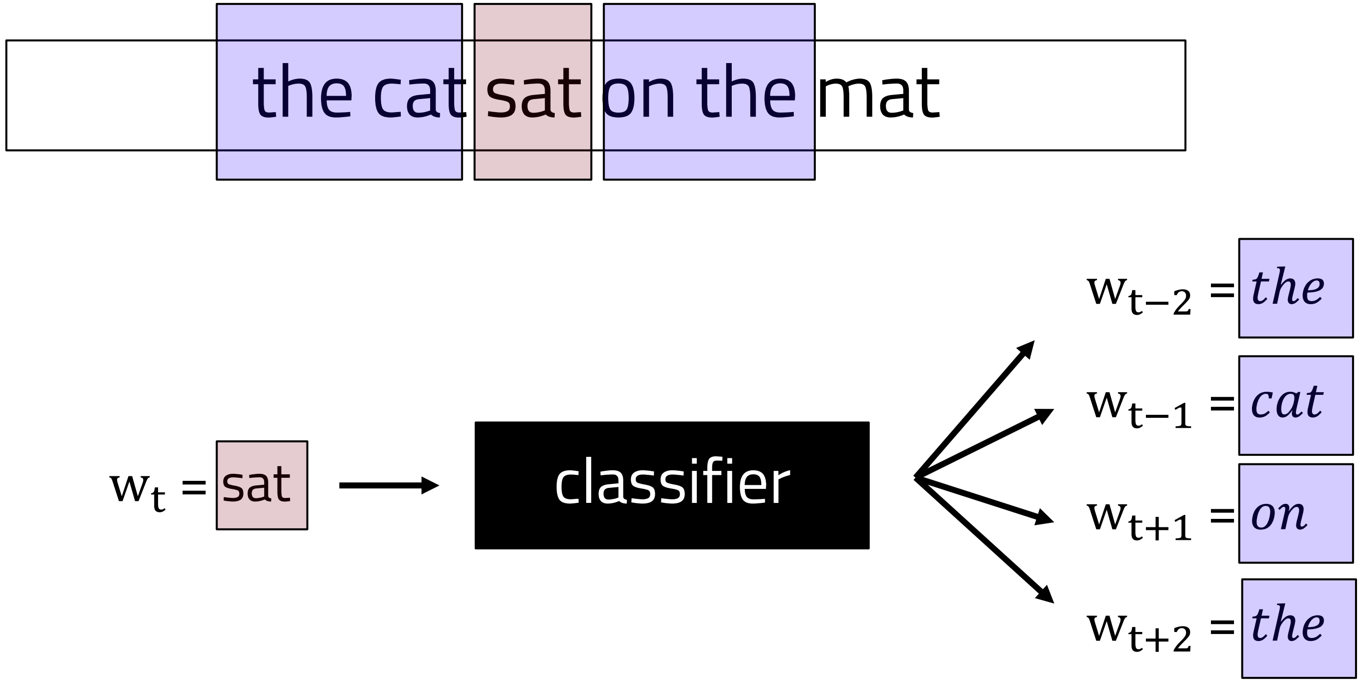
Context window size = 2

Dense vectors from prediction (not count)



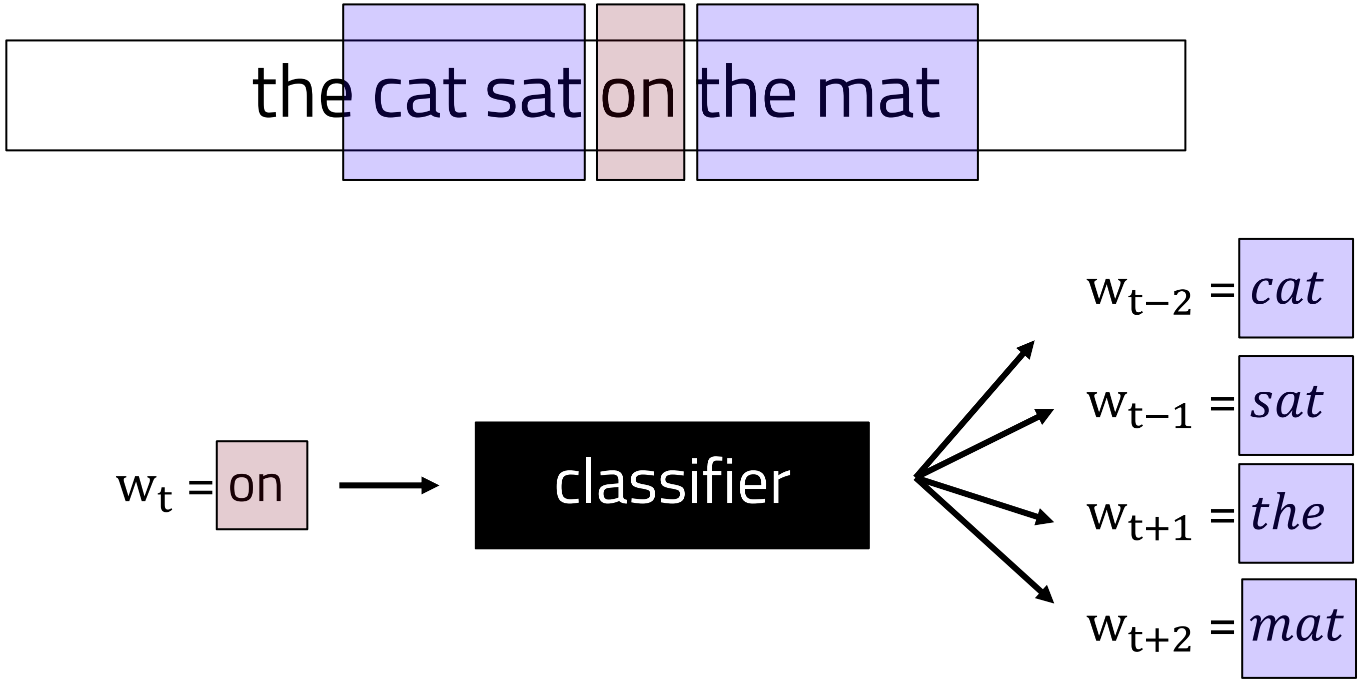
Context window size = 2

Dense vectors from prediction (not count)



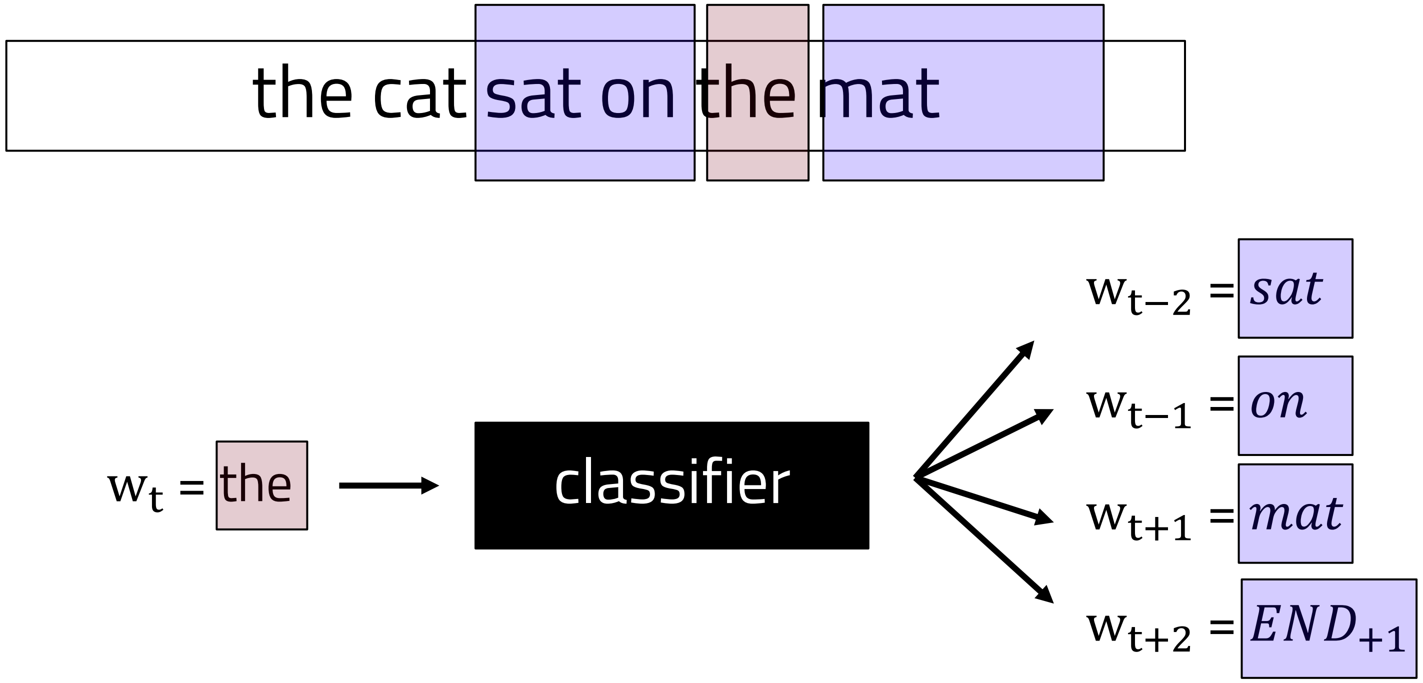
Context window size = 2

Dense vectors from prediction (not count)



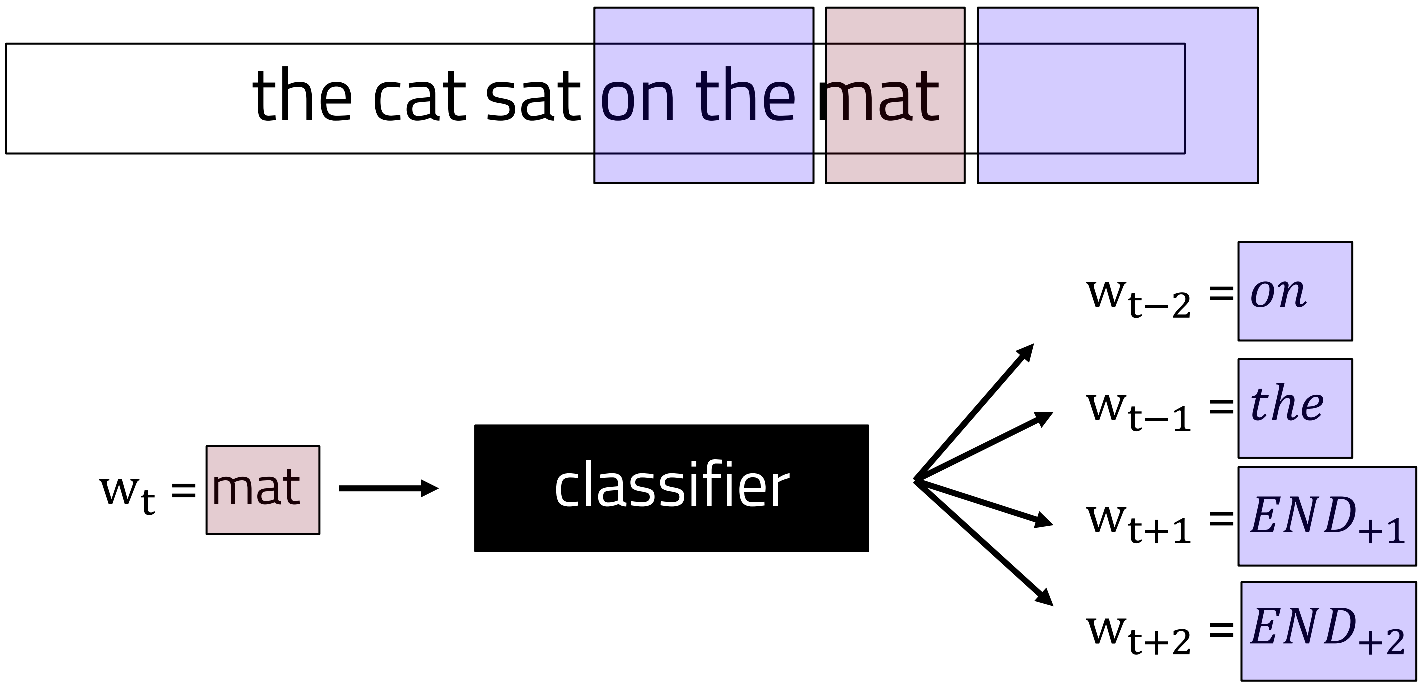
Context window size = 2

Dense vectors from prediction (not count)



Context window size = 2

Dense vectors from prediction (not count)



Context window size = 2

Dense vectors from prediction (not count)

the cat sat on the mat

the cat sat on the mat

Dense vectors from prediction (not count)

w_t = the → classifier

$w_{t-2} = START_{-2}$

$w_{t-1} = START_{-1}$

$w_{t+1} = cat$

$w_{t+2} = sat$

w_t = the → classifier

$w_{t-2} = sat$

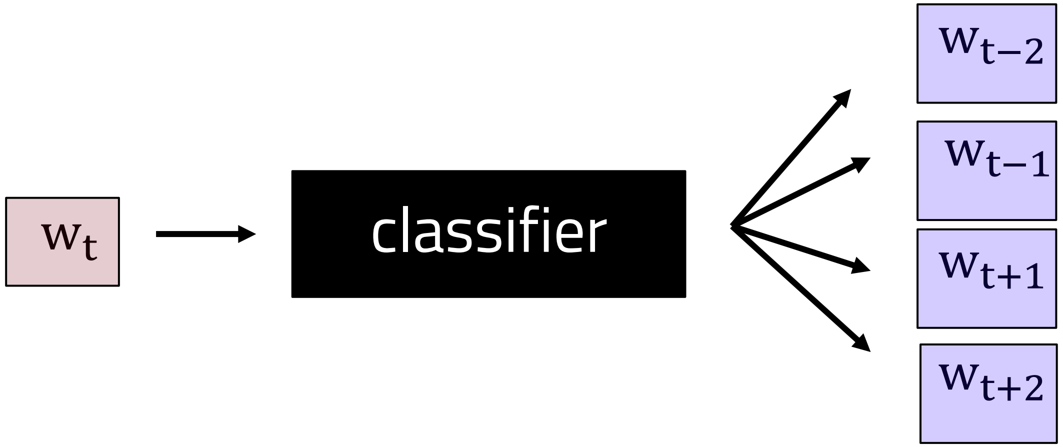
$w_{t-1} = on$

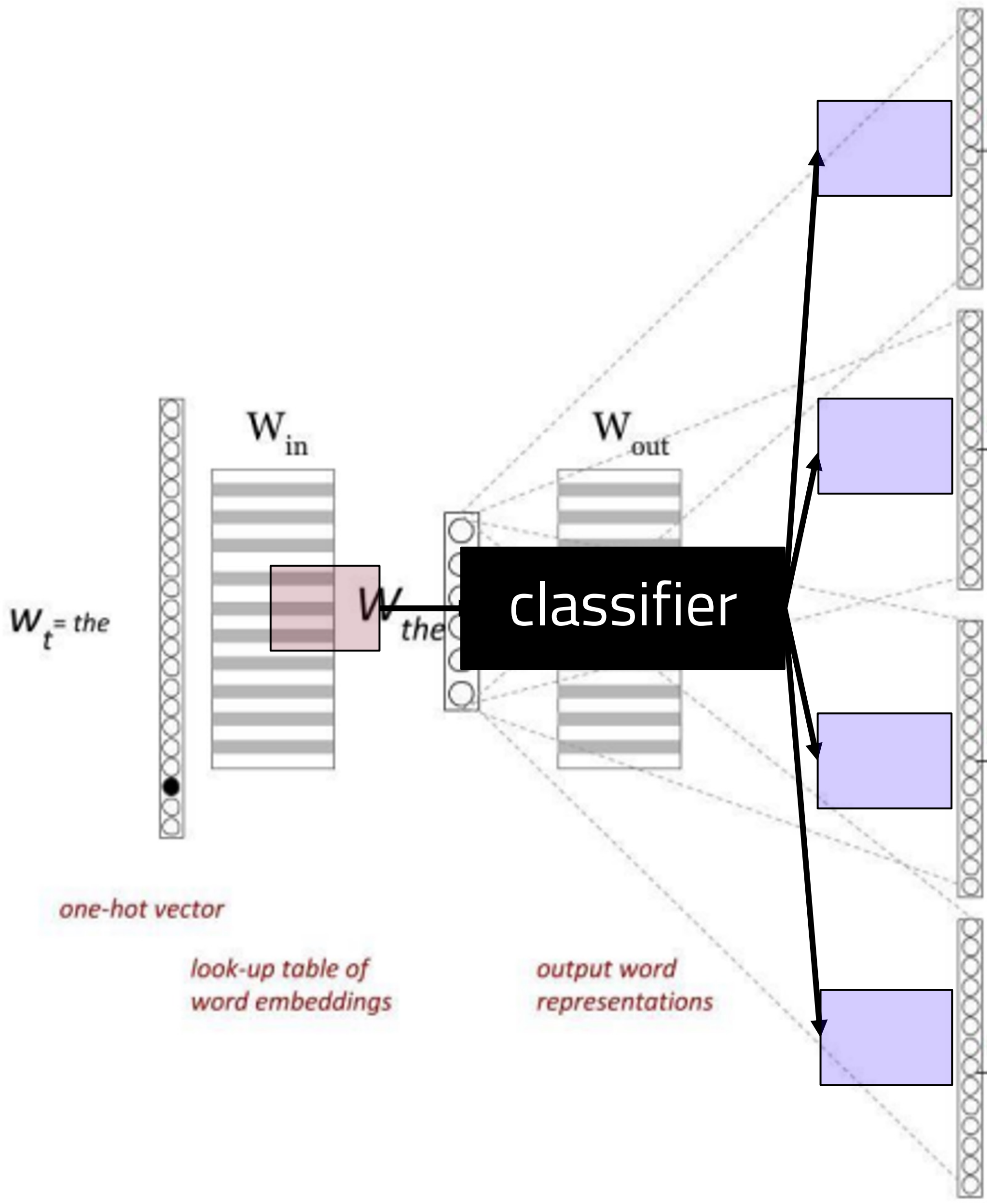
$w_{t+1} = mat$

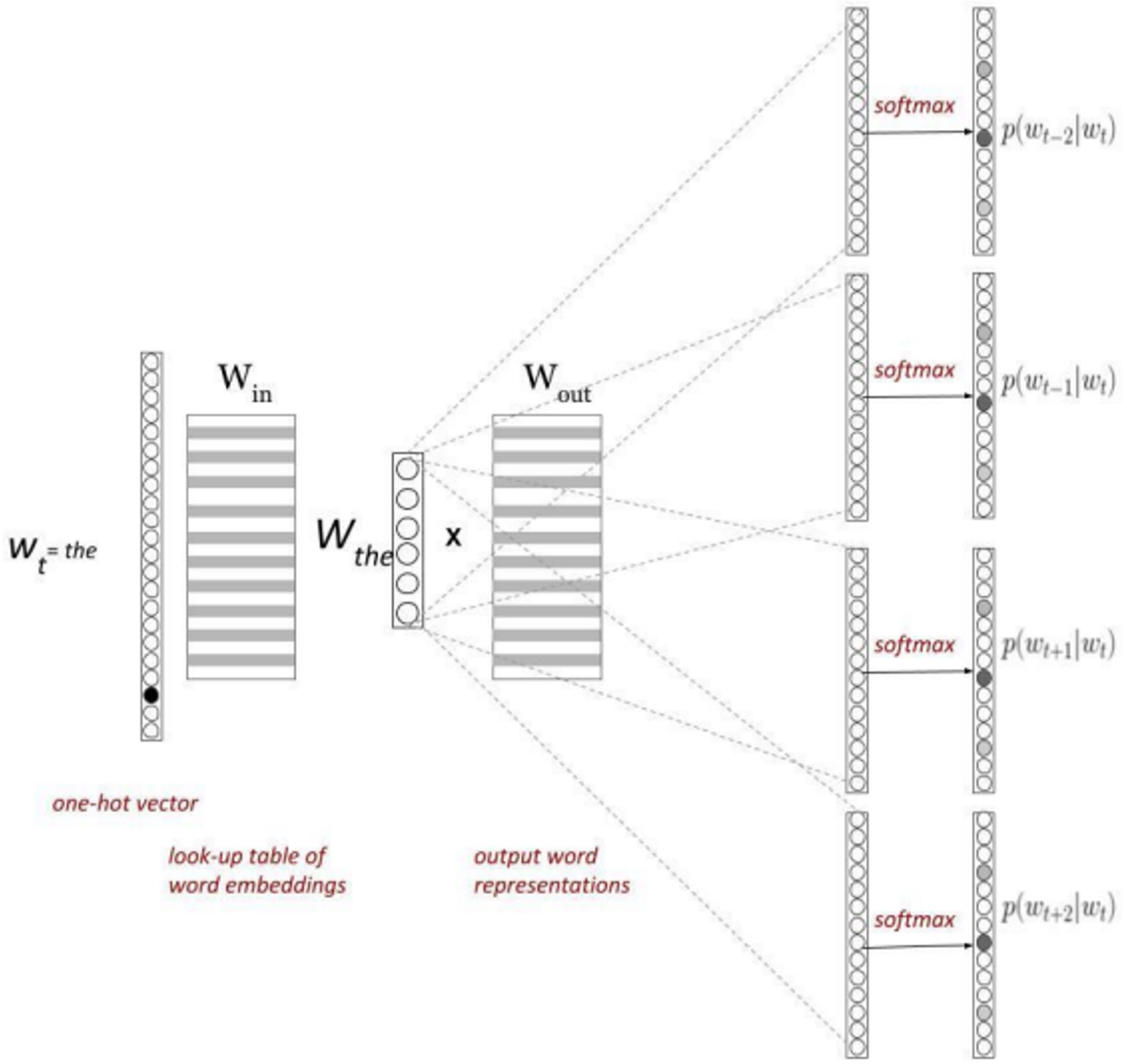
$w_{t+2} = END_{+1}$

Context window size = 2

Dense vectors from prediction (not count)




W_{in}
W_{out}
w_t = the
w_{the}
classifier
one-hot vector
look-up table of
word embeddings
output word
representations



W_t = the
W_{in}
W_{the}
x
W_{out}
softmax
$p(w_{t-2}|w_t)$
softmax
$p(w_{t-1}|w_t)$
softmax
$p(w_{t+1}|w_t)$
softmax
$p(w_{t+2}|w_t)$
one-hot vector
look-up table of
word embeddings
output word
representations

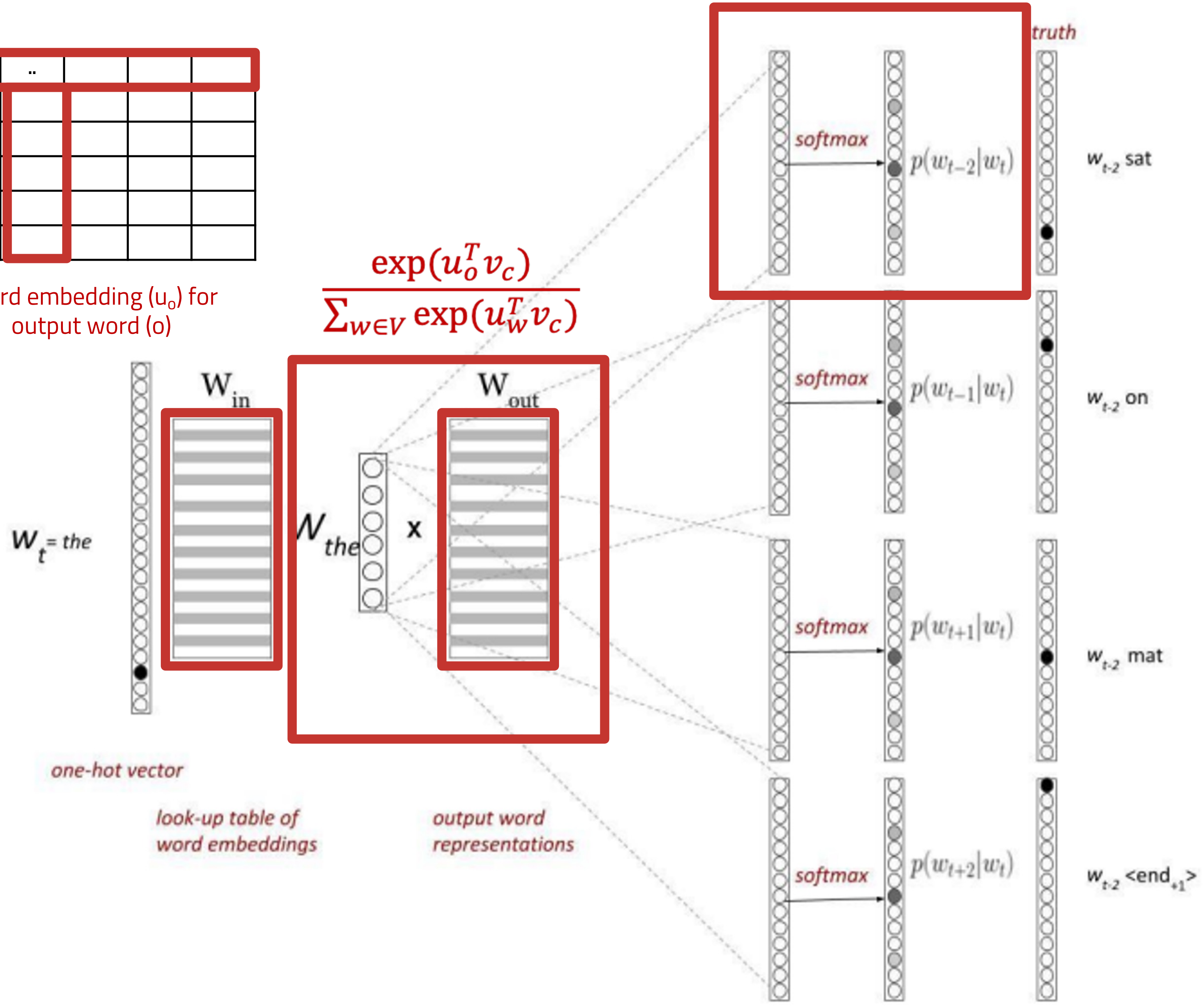
V

the	cat	mat	on	sat	..			
5.2	1.5	...						
0.5	0.4	...						
-6.2	0.6	..						
0.5	-3.4	..						
...								

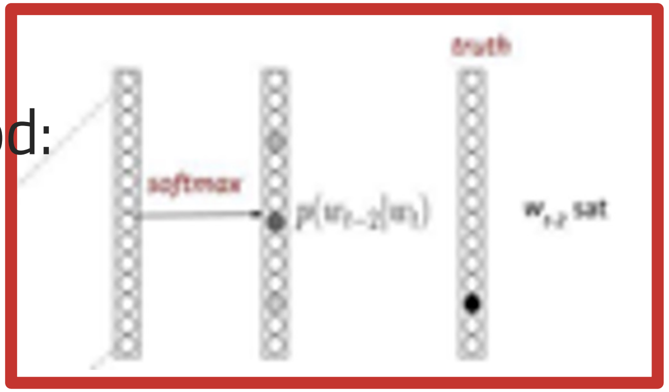
Word embedding (v_c) for center word (c) “the”

Word embedding (u_o) for output word (o)

$$\frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$



The objective function $J(\theta)$ is the average negative log likelihood:



$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\le j\le m, j\ne 0} \boxed{\log P(w_{t+j} \mid w_t; \theta)}$$

All word vectors

For a center word c and a context word o :

$$x_i = \quad P(\, o \mid c\,) = \frac{\exp\boxed{(u_o^T v_c)}}{\boxed{\sum_{w\in V}\exp(u_w^T v_c)}}$$

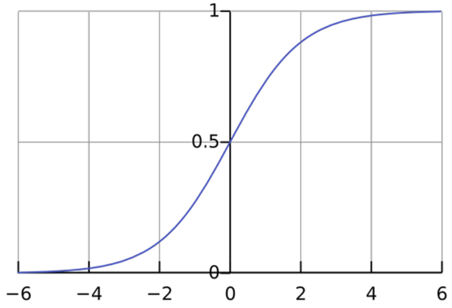
Dor product compares similarity of o and c . $u^T v = u \cdot v = \sum_{i=1}^{n} u_i v_i$

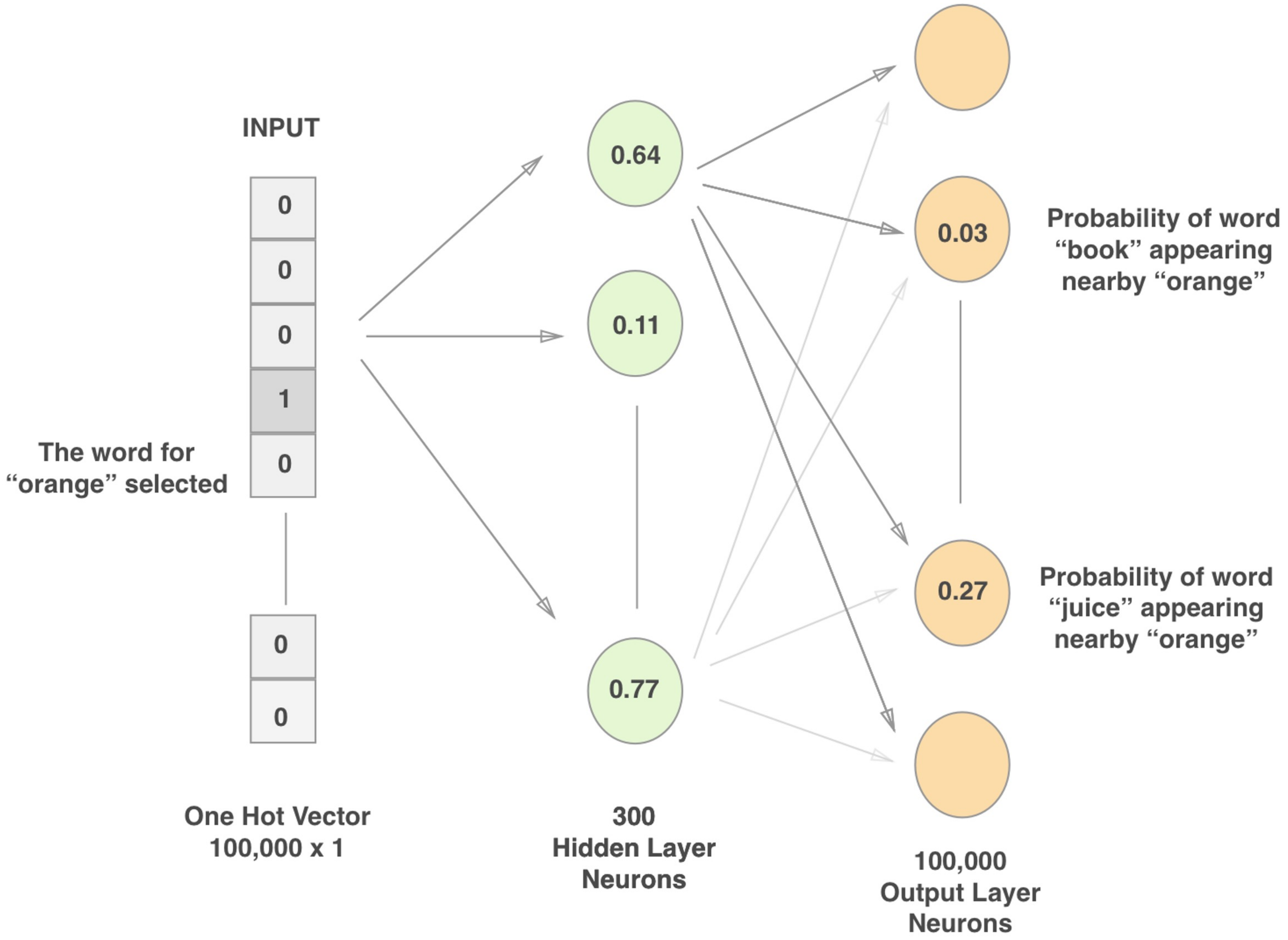
Normalize over entire vocabulary to give probability distribution

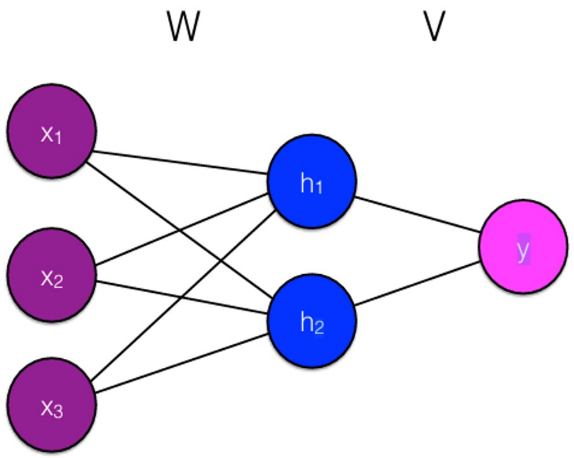
“soft” because still assigns some probability to smaller x_i

$$\boxed{soft}\boxed{max}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}\exp(x_j)} = p_i$$

“max” because amplifies probability of largest x_i

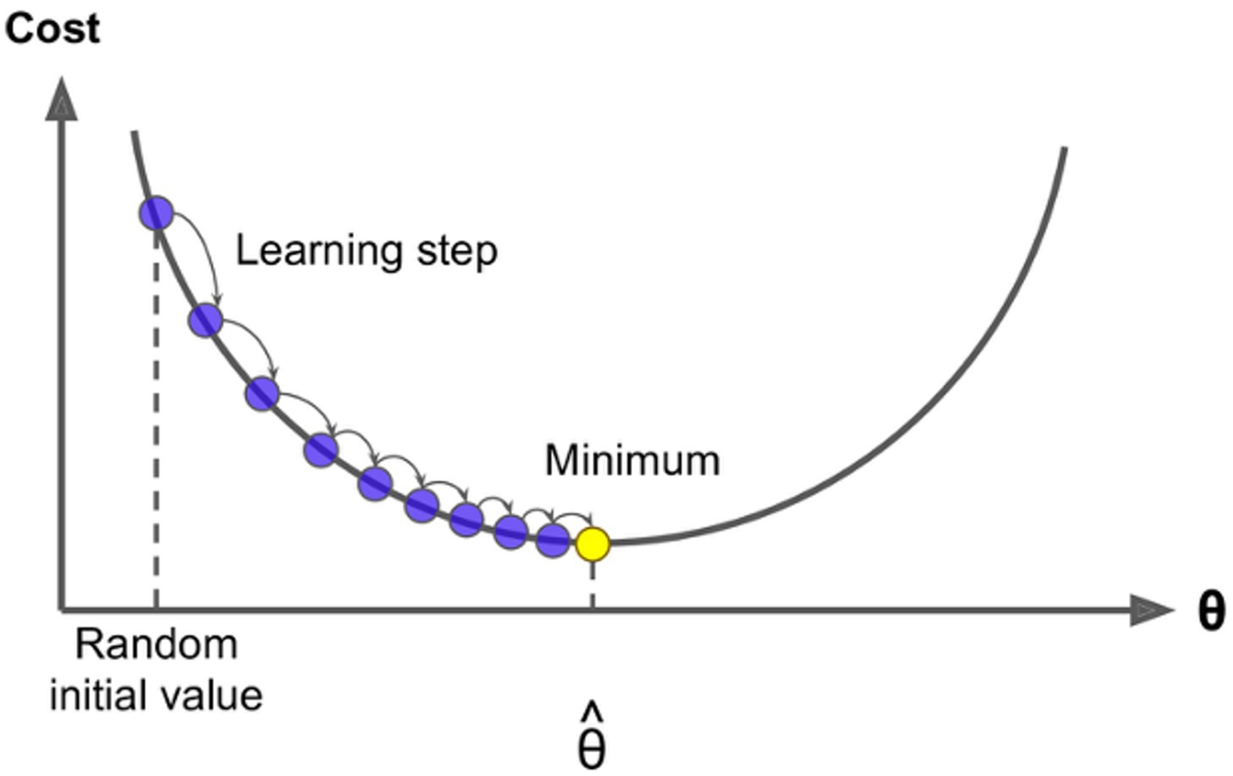



INPUT
0
0
0
1
0
0
0
The word for
"orange" selected
One Hot Vector
100,000 x 1
0.64
0.11
0.77
300
Hidden Layer
Neurons
0.03
0.27
100,000
Output Layer
Neurons
Probability of word
"book" appearing
nearby "orange"
Probability of word
"juice" appearing
nearby "orange"


W
V
x1
x2
x3
h1
h2
y

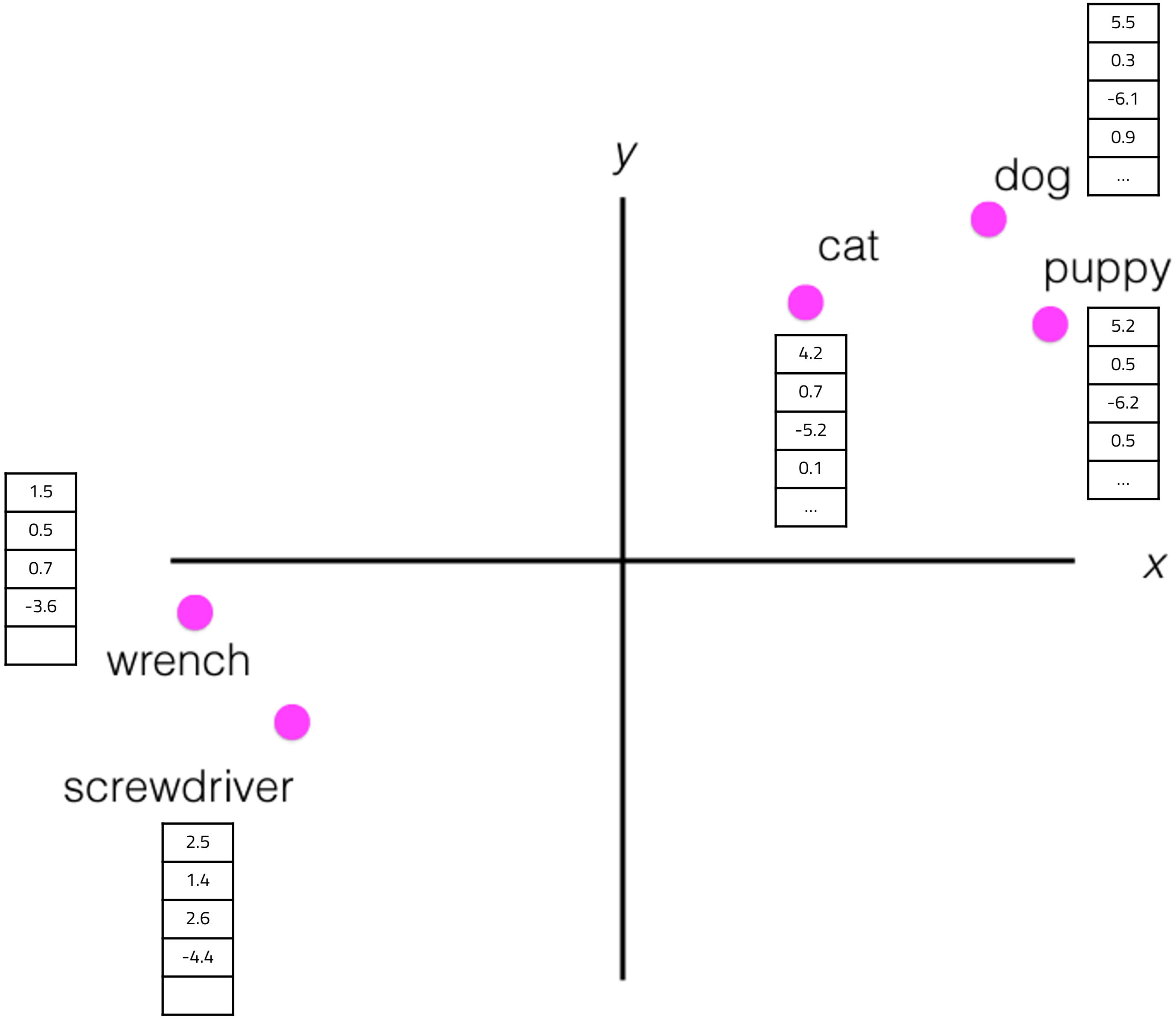
Minimize the objective function $J(\theta)$ using *gradient descent*

Idea: for current value of θ , calculate gradient of $J(\theta)$ then take small step in direction of negative gradient. Repeat this until convergence



Two kinds of training data

- The labeled data for specific tasks
    - Labeled sentiment for movie reviews (~2K labels/reviews, ~1.5 words)
    - Used for supervised models
- Unlabeled text for representation learning
    - Trillions of words (Wikipedia, web text, books, etc)
    - Used for word distributed representations



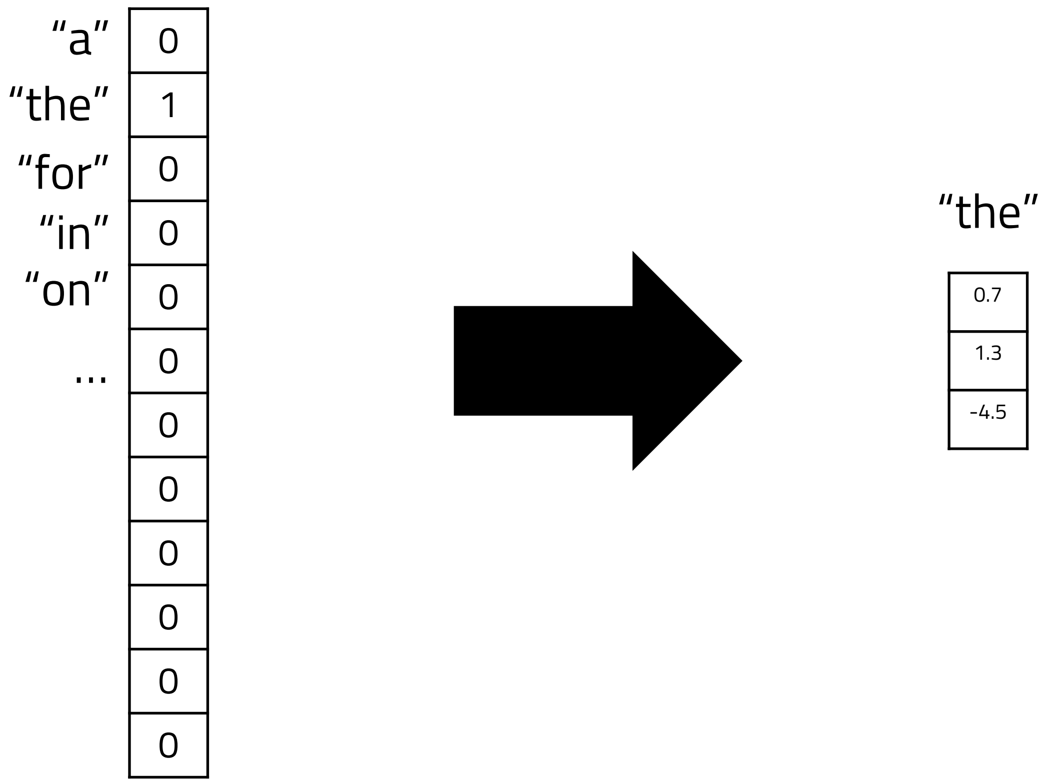
y
x
dog
5.5
0.3
-6.1
0.9
...
cat
4.2
0.7
-5.2
0.1
...
puppy
5.2
0.5
-6.2
0.5
...
1.5
0.5
0.7
-3.6
wrench
screwdriver
2.5
1.4
2.6
-4.4

Why *dog* and *cat* are in similar positions

the	black	**dog**	jumped	on	the	table
the	black	**cat**	jumped	on	the	table
the	black	**puppy**	jumped	on	the	table

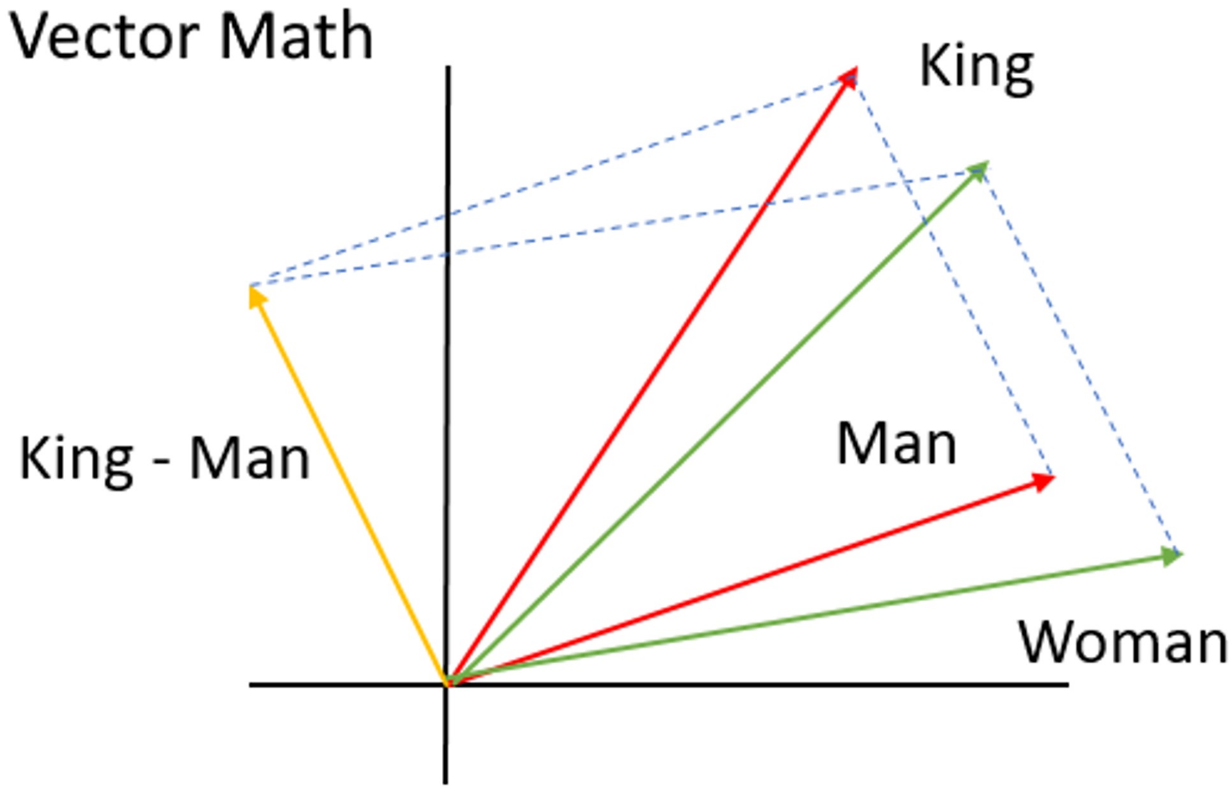
the	black	**wrench**	jumped	on	the	table
the	black	**shoe**	jumped	on	the	table

Dimensionality reduction



V-dimensional space (1-hot)
Representations for all words are completely independent

3-dimensional space
Representations are not structured



v(“King”) – v(“Man”) - v(“Woman”) =

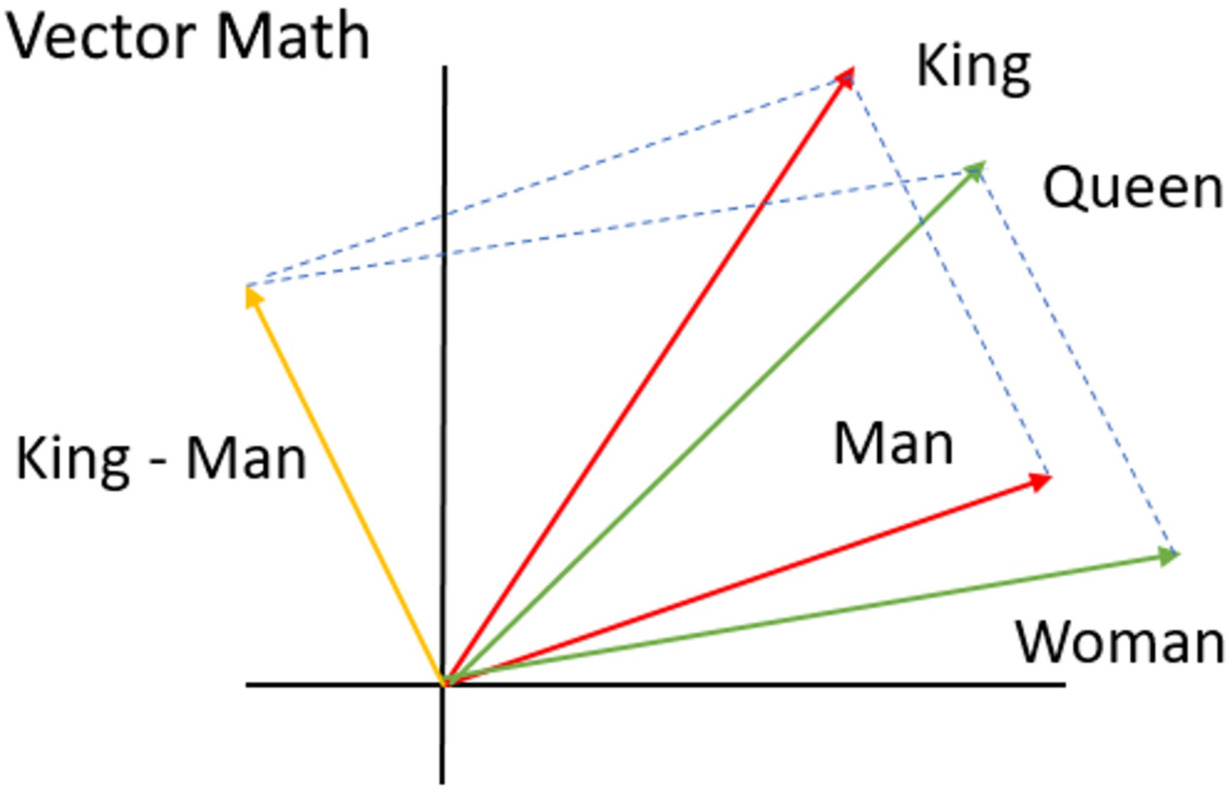
$$\begin{bmatrix} 0.7 \\ 1.3 \\ -4.5 \\ \dots \end{bmatrix} - \begin{bmatrix} 5.2 \\ 0.5 \\ -6.2 \\ 0.5 \\ \dots \end{bmatrix} + \begin{bmatrix} 4.2 \\ 0.7 \\ -5.2 \\ 0.1 \\ \dots \end{bmatrix} = \begin{bmatrix} 5.2 \\ 0.5 \\ -6.2 \\ 0.5 \\ \dots \end{bmatrix}$$

Closest vector

the	king	man	on	sat	..			
5.2	1.5	...						
0.5	0.4	...						
-6.2	0.6	..						
0.5	-3.4	..						
...								

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

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



v("King") – v("Man") - v("Woman") =

$$\begin{bmatrix} 0.7 \\ 1.3 \\ -4.5 \\ \dots \end{bmatrix} - \begin{bmatrix} 5.2 \\ 0.5 \\ -6.2 \\ 0.5 \\ \dots \end{bmatrix} + \begin{bmatrix} 4.2 \\ 0.7 \\ -5.2 \\ 0.1 \\ \dots \end{bmatrix} = \begin{bmatrix} 5.2 \\ 0.5 \\ -6.2 \\ 0.5 \\ \dots \end{bmatrix}$$

Closest vector

the	king	man	on	sat	..	queen		
5.2	1.5	…						
0.5	0.4	…						
-6.2	0.6	..						
0.5	-3.4	..						
…								

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

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

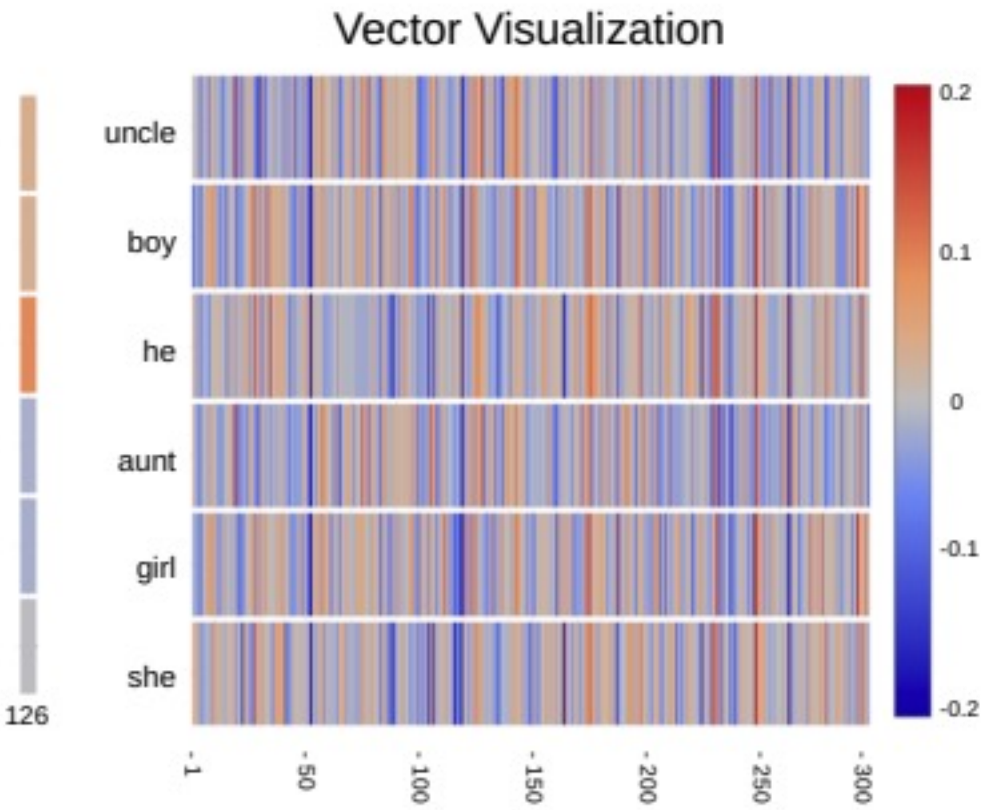


Figure 2: Embedding vectors for three male words ("uncle", "boy", "he") and three female words ("aunt", "girl", "she"). Component 126, shown magnified at left, is positive for the male words and negative for the female words.

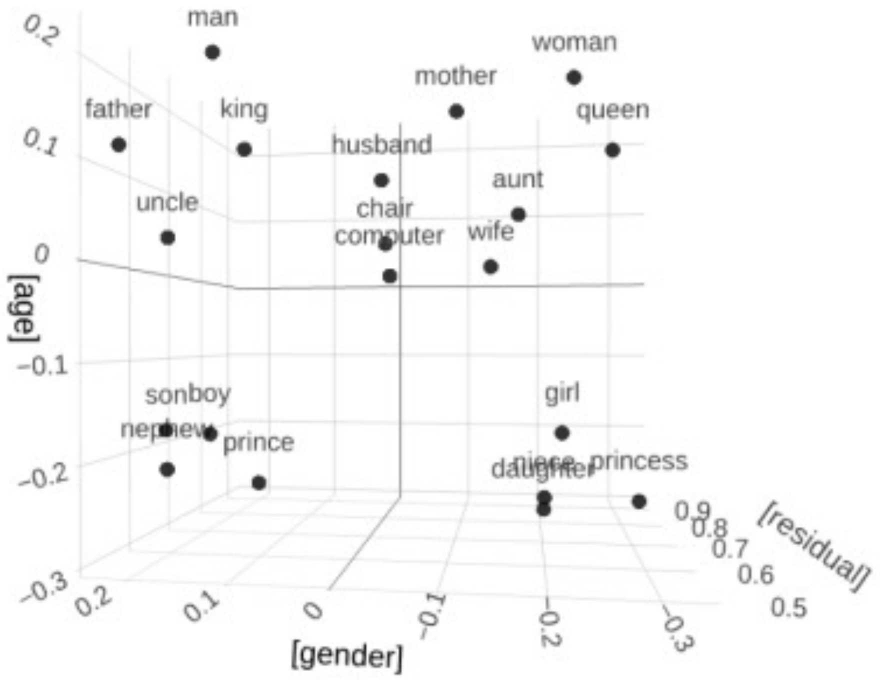


Figure 3: Words plotted in our 3D semantic space. Male words appear in the positive (left) half of the x-axis; female words in the negative (right) half. Adult words are in the positive (top) half of the y-axis; youth words in the negative (bottom) half. The third dimension is the "semantic residual", explained in the main text.

Interactive Visualizations of Word Embeddings for K-12 Students. AAAI-22

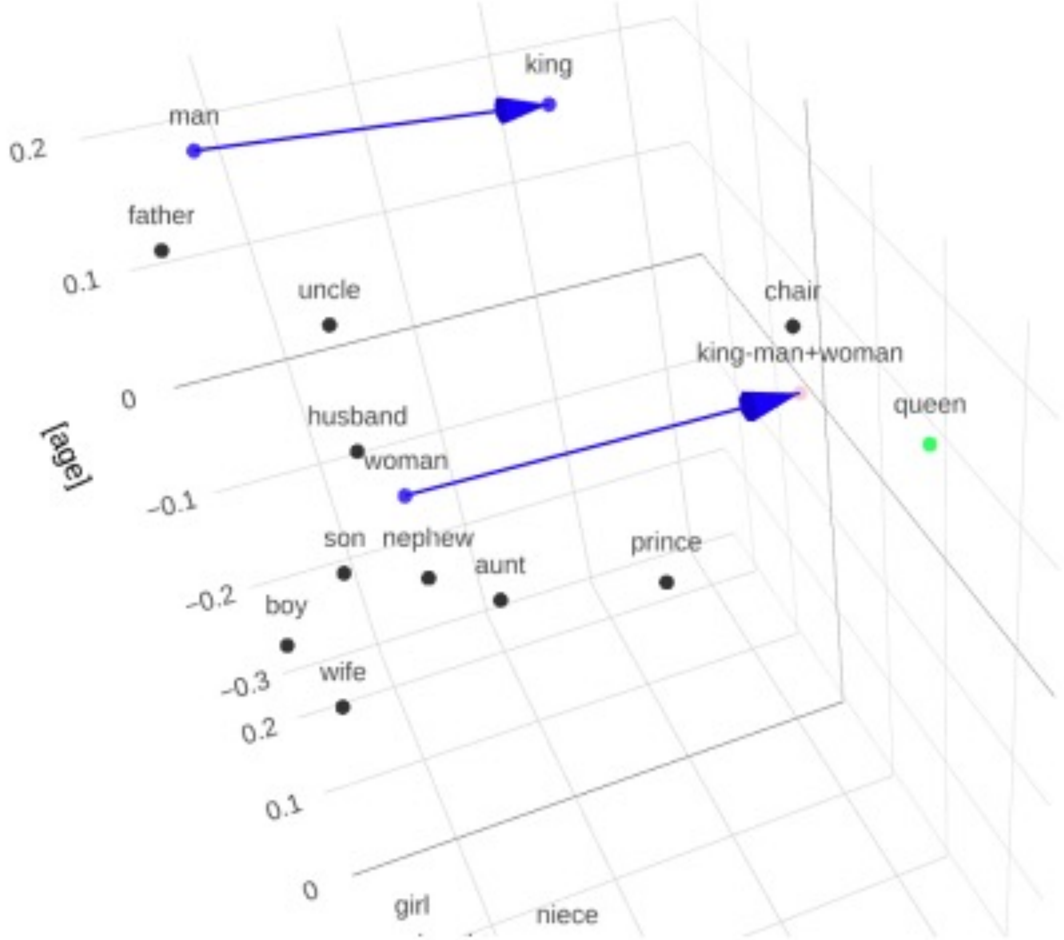


Figure 5: Analogy by vector arithmetic: “man” is to “king” as “woman” is to “king − man + woman” = “queen”.

Low-dimensional, distributed representations

- Two similar words (e.g., synonyms or words under the same class) have similar distributional properties

- In neural models, replace the initial V-dimensional sparse vector with much smaller k-dimensional dense vectors

- Low-dimensional, dense word representations are extraordinarily powerful and are a large part of why neural network models have been so successful for NLP

Count-based vs Prediction-based Methods

LSA, HAL (Lund & Burgess)

Hellinger-PCA (Rohde et al, Lebret & Collobert)

- ❑ Fast training
- ❑ Efficient usage of statistics
- ❑ Primarily used to capture word similarity
- ❑ Disproportionate importance given to large counts

Skip-gram/b (Mikolovet al)

NLM, **HLBL**, **RNN** (Bengioet al; Collobert & Weston; Huang et al; Mnih & Hinton)

- ❑ Scales with corpus size
- ❑ Inefficient usage of statistics
- ❑ Generated improved performance on other tasks
- ❑ Can capture complex patterns beyond word similarity

Count-based and Prediction-based Methods

- Strong connection between count-based methods and prediction-based methods (Levy and Goldberg 2014)

- Skip-gram objective is equivalent to matrix factorization with PMI and discount for number of samples k

$$M_{w,c} = \mathrm{PMI}(w,c) - \log(k)$$

Neural Word Embedding as Implicit Matrix Factorization, (Levy & Goldberg, 2014)

Other techniques and embeddings not covered

- Contrastive learning with negative samples
- Other variants
  - ~~Word2Vec (Mikolove et al., 14)~~
    https://code.google.com/archive/p/word2vec/
  - GloVe (Pennington et al., 14)
    http://nlp.stanford.edu/projects/glove/
  - FastText (Bojanowski et al.' 17)
    http://www.fasttext.cc/

Word2Vec Demo

- Pre-trained word2vec models:
  - https://code.google.com/archive/p/word2vec/
- Gensim:
  - https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html
- Online demos:
  - http://nlp.polytechnique.fr/word2vec
  - http://vectors.nlpl.eu/explore/embeddings/en/
  - https://remykarem.github.io/word2vec-demo/

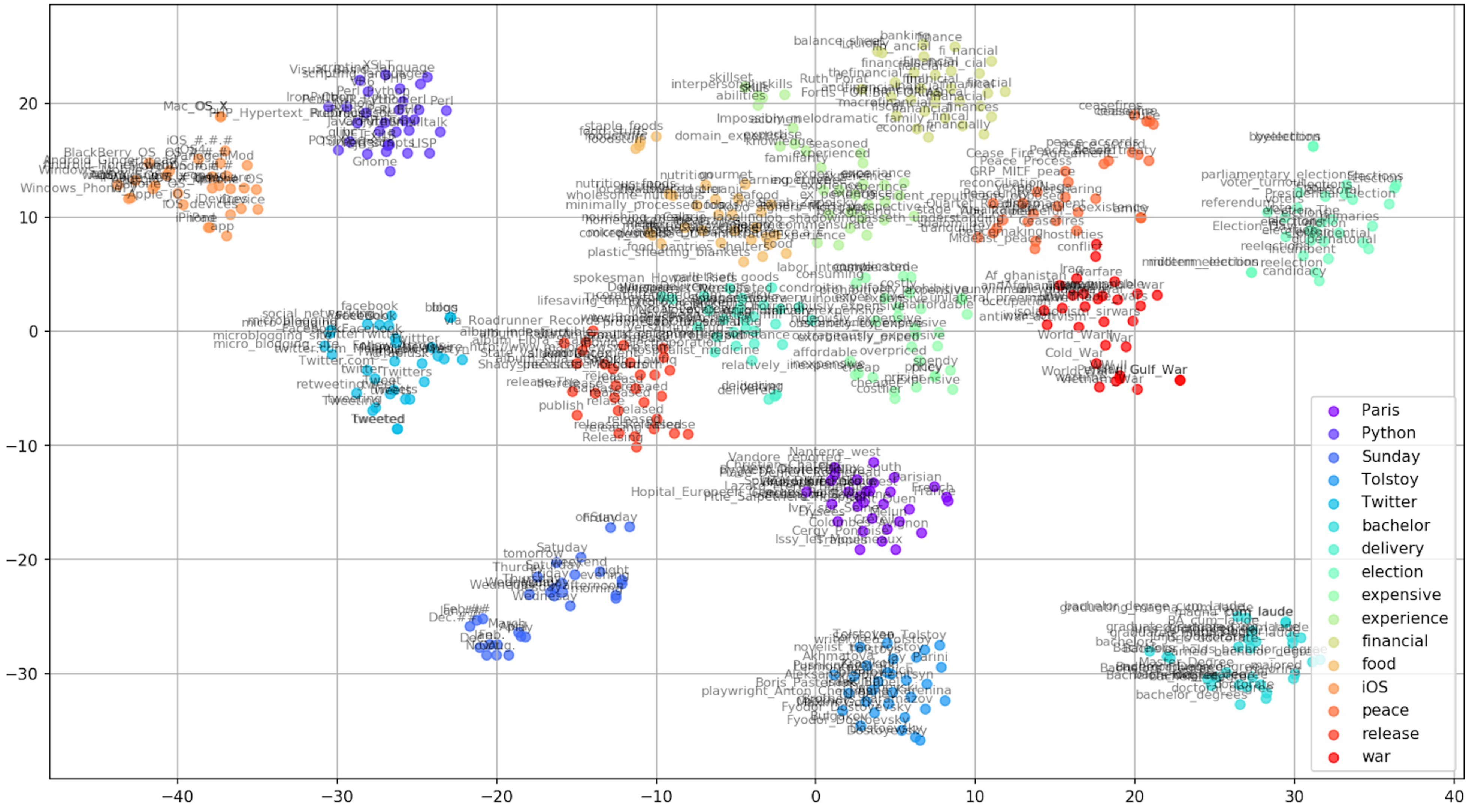
Types of Evaluation

Types of Evaluation

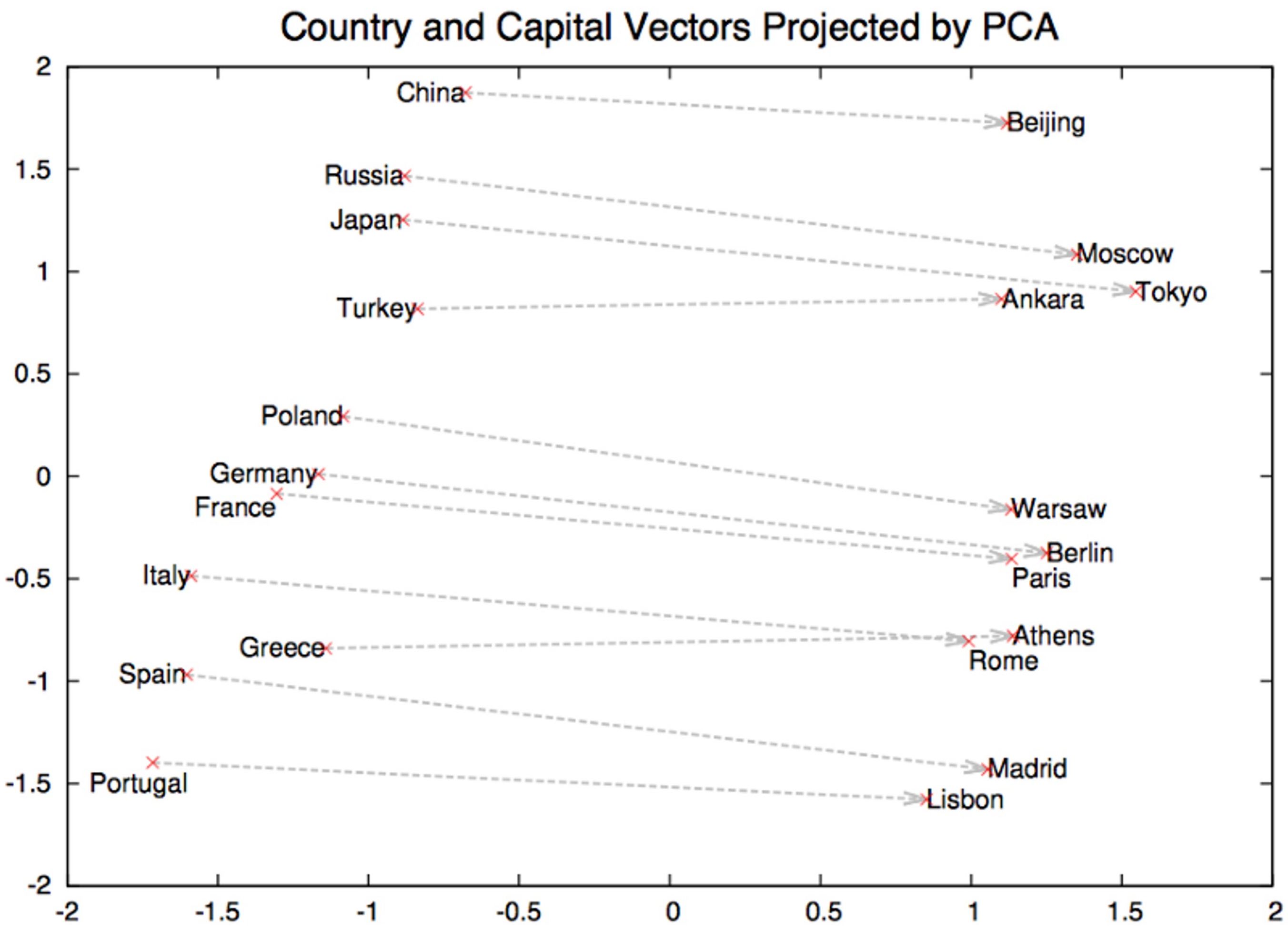
- Intrinsic vs Extrinsic
  - Intrinsic: How good is it based on its features?
  - Extrinsic: How useful is it downstream?

- Qualitative vs. Quantitative
  - Qualitative: Examine the characteristics of examples.
  - Quantitative: Calculate statistics

Visualization of Embeddings



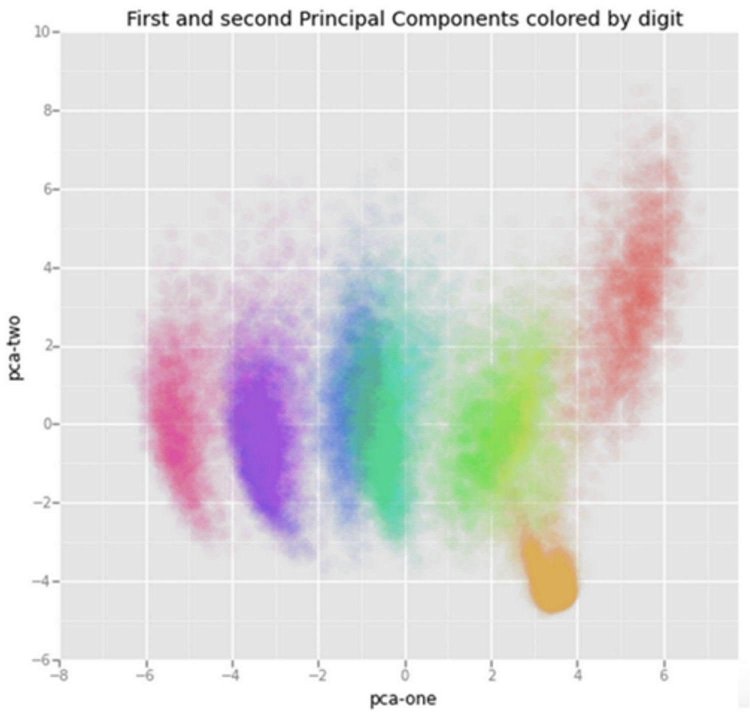
Visualization of Embeddings

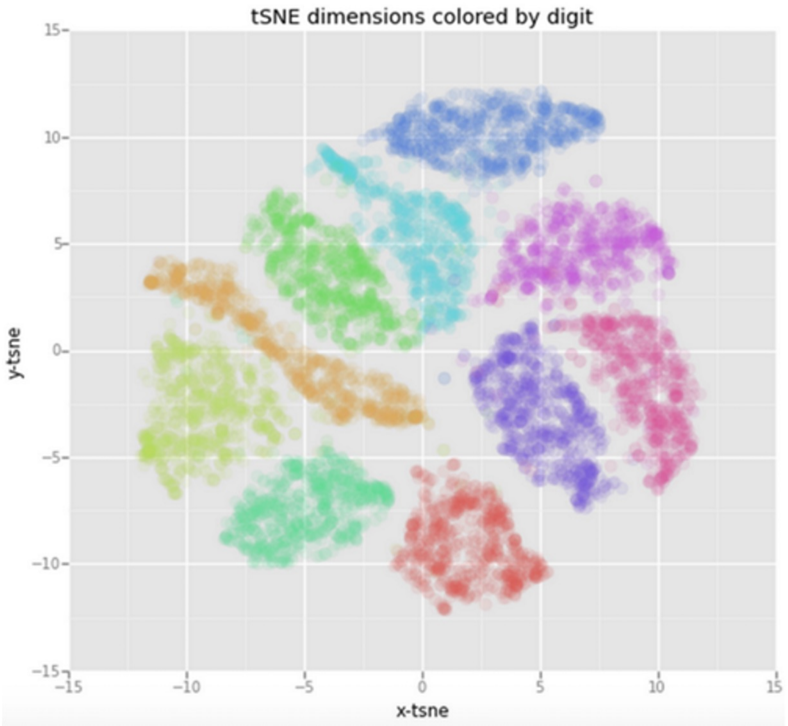


Linear and Non-linear Projection

- Non-linear projections group things that are close in high-dimensional space
  - e.g. SNE/t-SNE (van der Maaten and Hinton 2008) group things that give each other a high probability according to a Gaussian

PCA



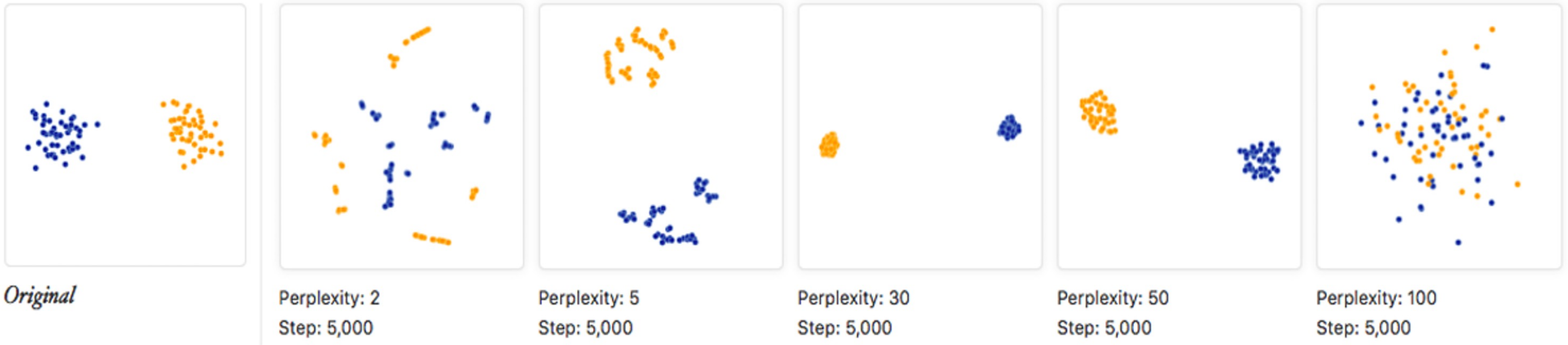


T-SNE

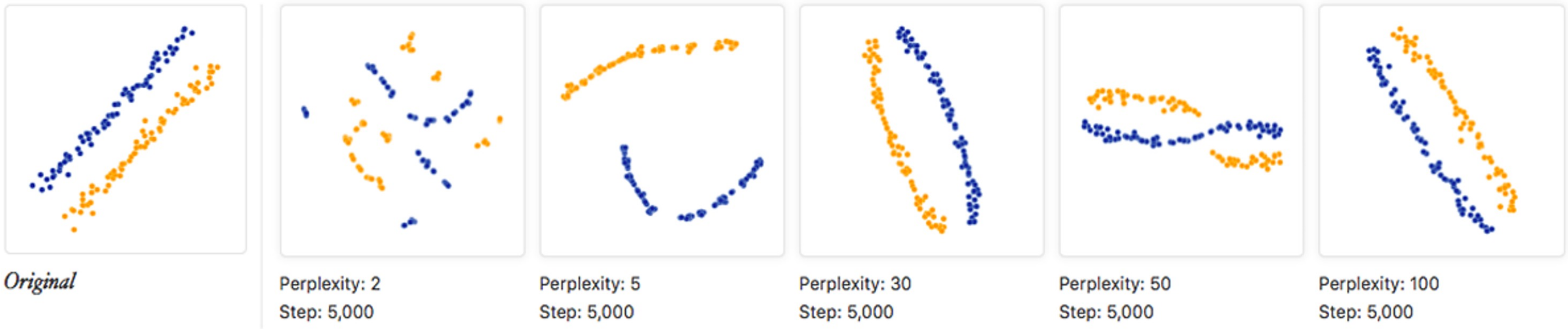
Image from Derksen (2016)

t-SNE Visualization can be Misleading! (Wattenberg et al. 2016)

Settings matter



Linear correlations cannot be interpreted



Intrinsic Evaluation of Embeddings

- **Relatedness**: The correlation between embedding cosine similarity and human eval of similarity?
- **Analogy**: Find x for "a is to b, as x is to y".

- **Categorization**: Create clusters based on the embeddings, and measure purity of clusters.
- **Selectional Preference**: Determine whether a noun is a typical argument of a verb.

(categorization from Schnabel et al 2015)

Intrinsic evaluation:

Ask humans how similar two words are

Relatedness:
correlation (Spearman/Pearson) between vector similarity of pair of words and human judgments

Word 1	Word 2	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)

WordSim-353 dataset (Finkelstein et al., 2002)

Intrinsic evaluation:

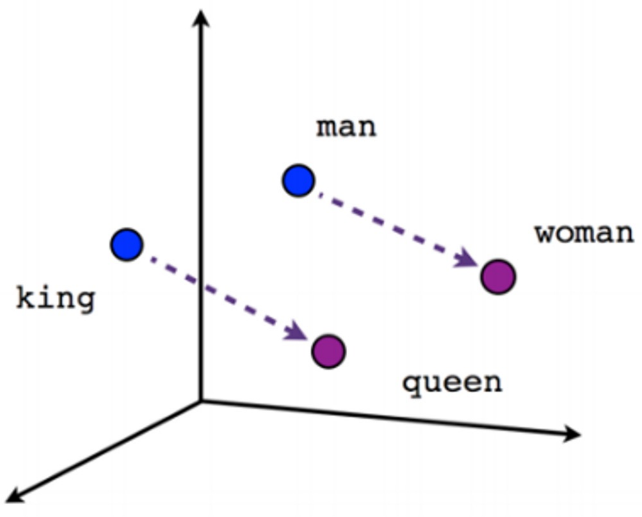
Analogical reasoning (Mikolov et al., 2013).

For analogy Germany : Berlin :: France : ?,
find closest vector to v("Berlin") – v("Germany")+v("France")

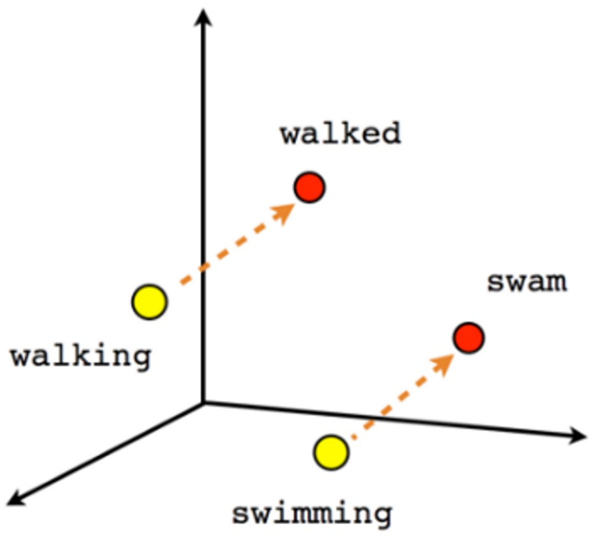
possibly	impossibly	Certain	Uncertain
generating	generated	Shrinking	Shrank
think	thinking	Look	Looking
Baltimore	Maryland	Minneapolis	Minnesota
shrinking	shrank	Slowing	Slowed
Rabat	Morocco	Astana	Kazakhstan

Intrinsic evaluation:

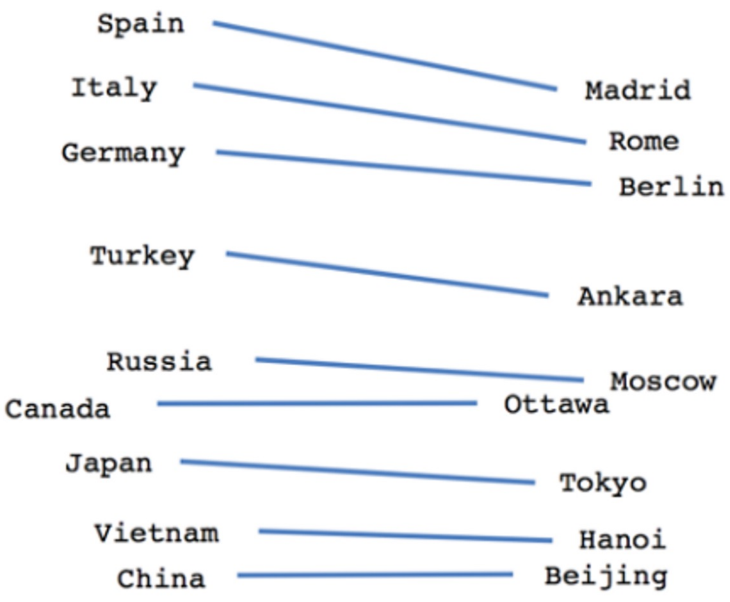
Analogical reasoning (Mikolov et al., 2013).



Male-Female



Verb tense



Country-Capital

Analogical reasoning test

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

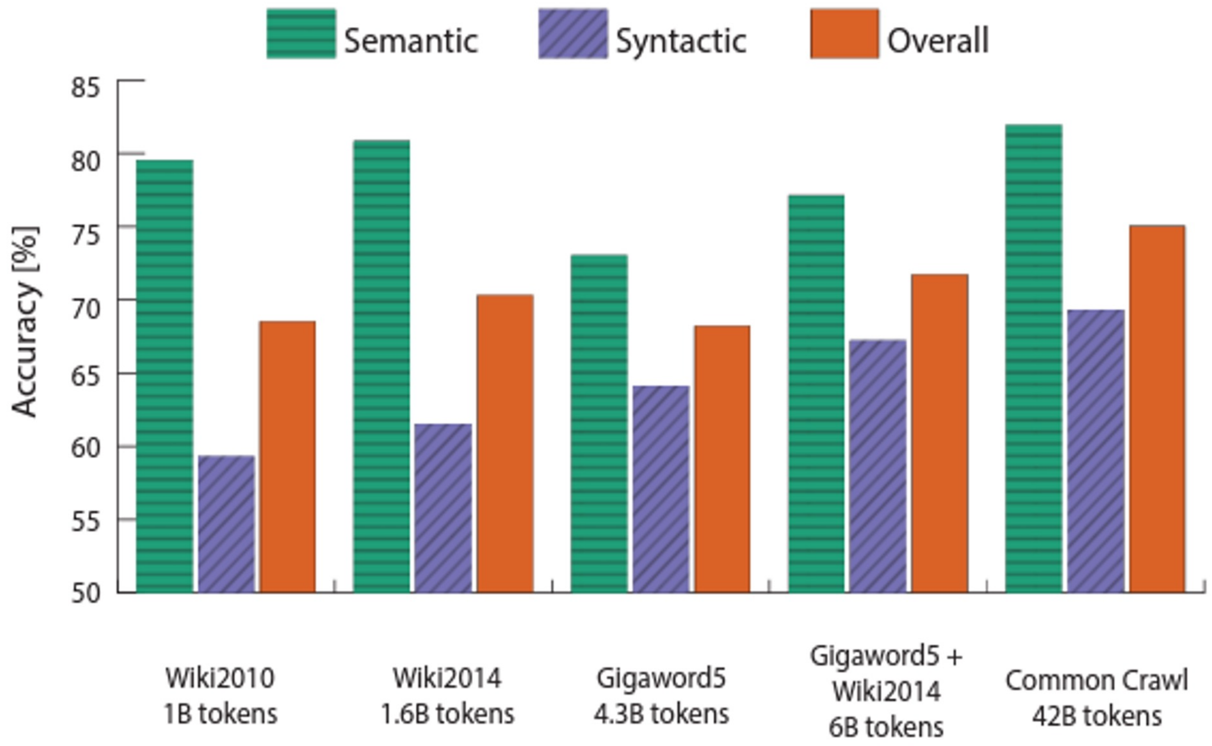
Mikolov et al. 2013

Analogical reasoning test

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

Mikolov et al. 2013

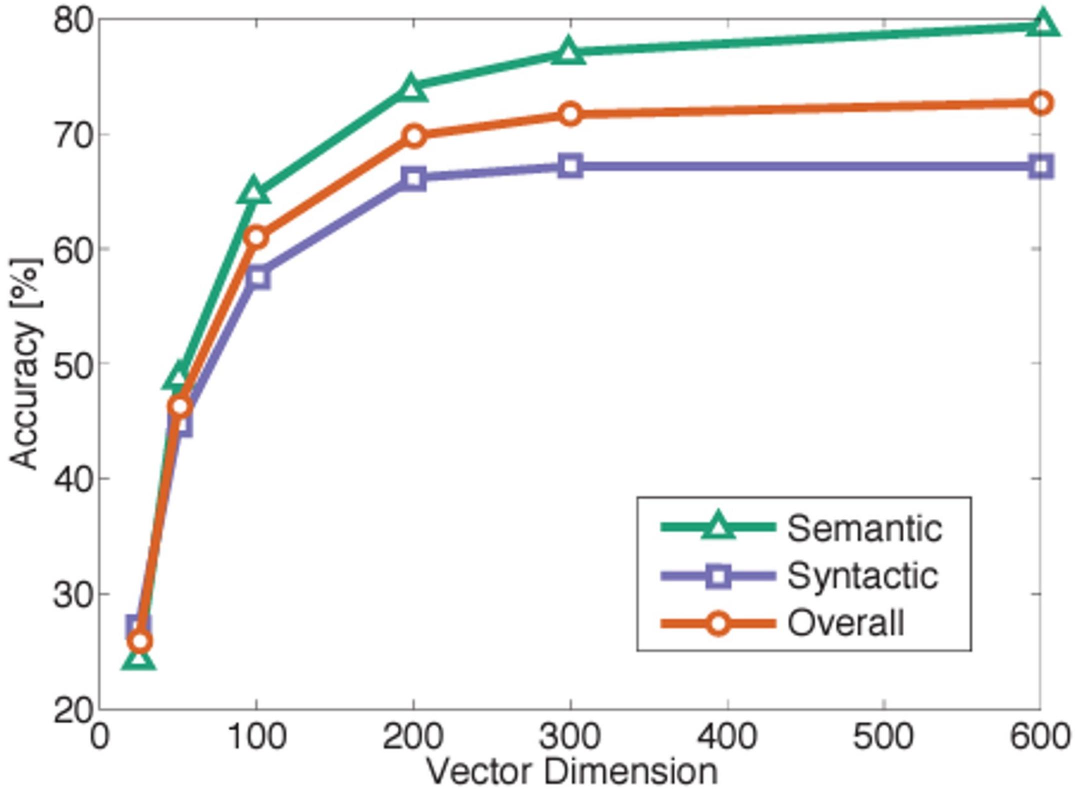
Analogy evaluation and hyper-parameters



- More data helps
- Wikipedia is better than news text

Mikolov et al. 2013

Analogy evaluation and hyper-parameters

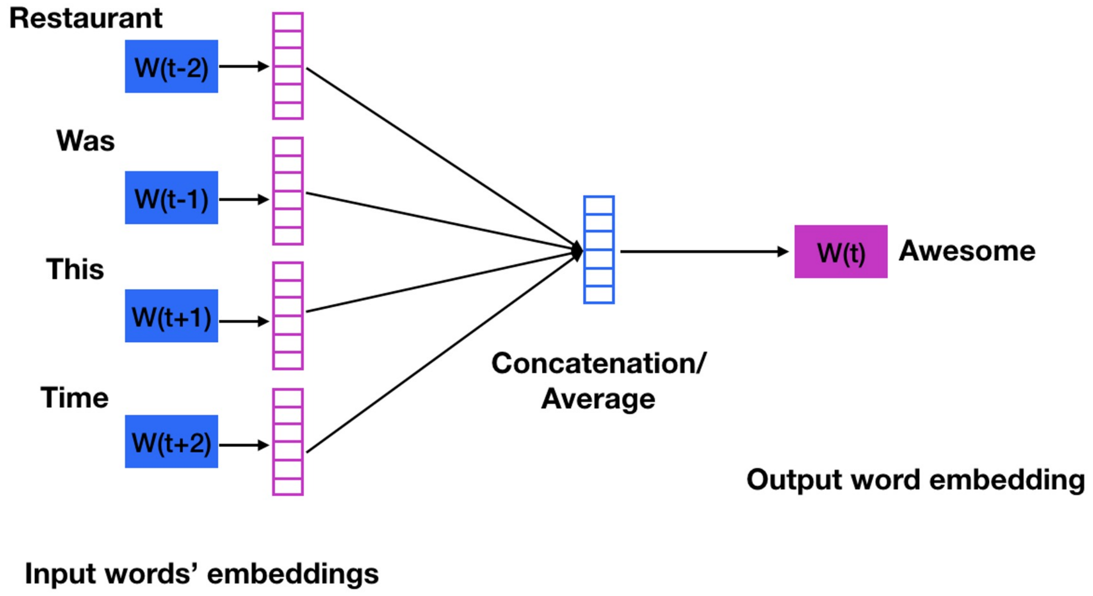


- Dimensionality
- Good dimension is ~300

Mikolov et al. 2013

Extrinsic Evaluation

Be aware and use the best one for the task



Method	Fine-grained	Binary
DAN		
- Word2vec	46.2	84.5
- GloVe	46.9	85.7

Sentiment classification

Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
$\boxed{\text{SVD}}$	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	**88.7**	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
$\boxed{\text{CBOW}}$	93.1	88.2	82.2	81.1
GloVe	**93.2**	88.3	**82.9**	**82.2**

Named Entity Recognition: identifying references to a person, organization or location:

When are Pre-trained Embeddings Useful?

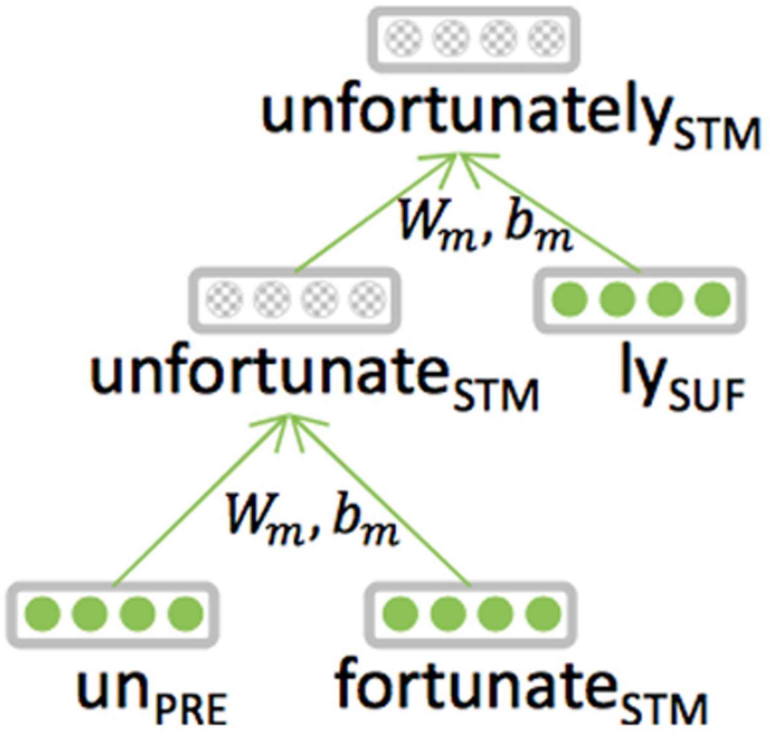
- Basically, when training data is insufficient
  - E.g. Low-resource languages
- **Very useful**: tagging, parsing, text classification
- **Less useful:** machine translation
- **Basically not useful:** language modeling

Limitations of Word Embeddings

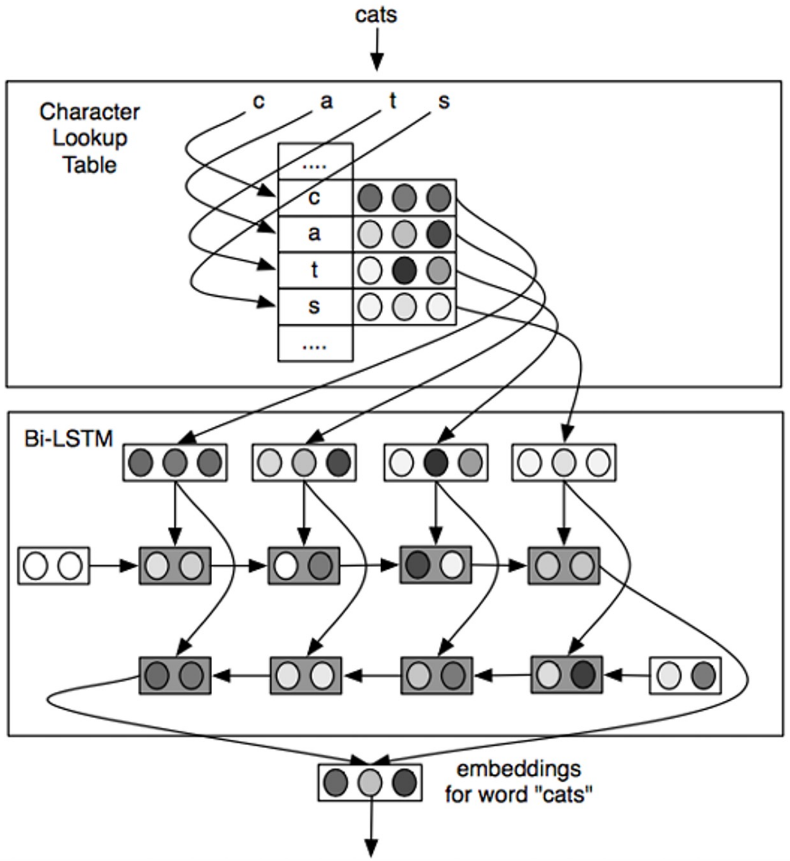
Limitations of Embeddings

- Sensitive to **superficial differences** (dog / dogs)
  - E.g. misspellings: “minuscule” → “miniscule”
  - E.g. compounded/prefixed/suffixed words split into “wrong” subwords “descheduled” ⇒ [“des”, “##ched”, “##uled”]
- **Not necessarily coordinated** with knowledge or across languages
- Can encode **bias** (encode stereotypical gender roles, racial biases)

Sub-word Embeddings

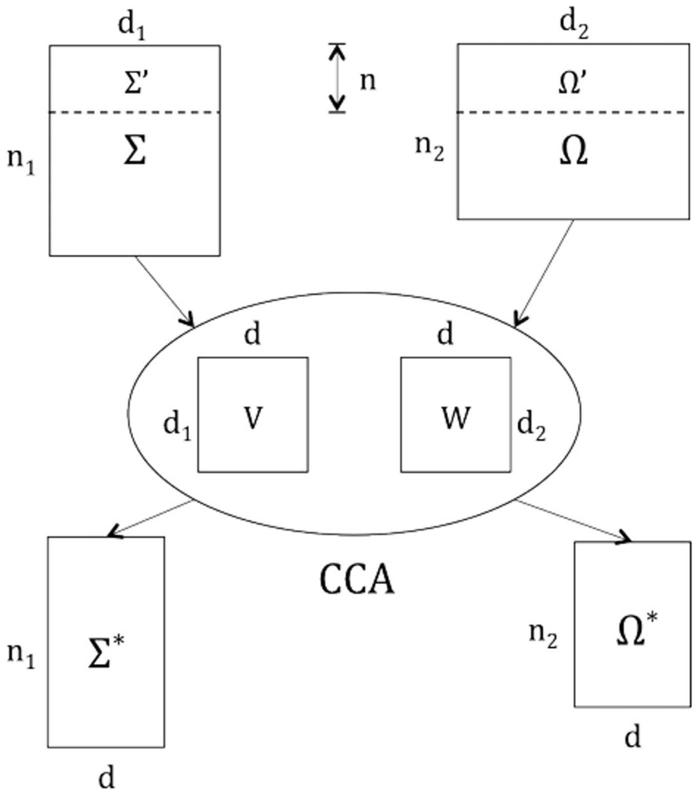


Morpheme-based (Luong et al. 2013)

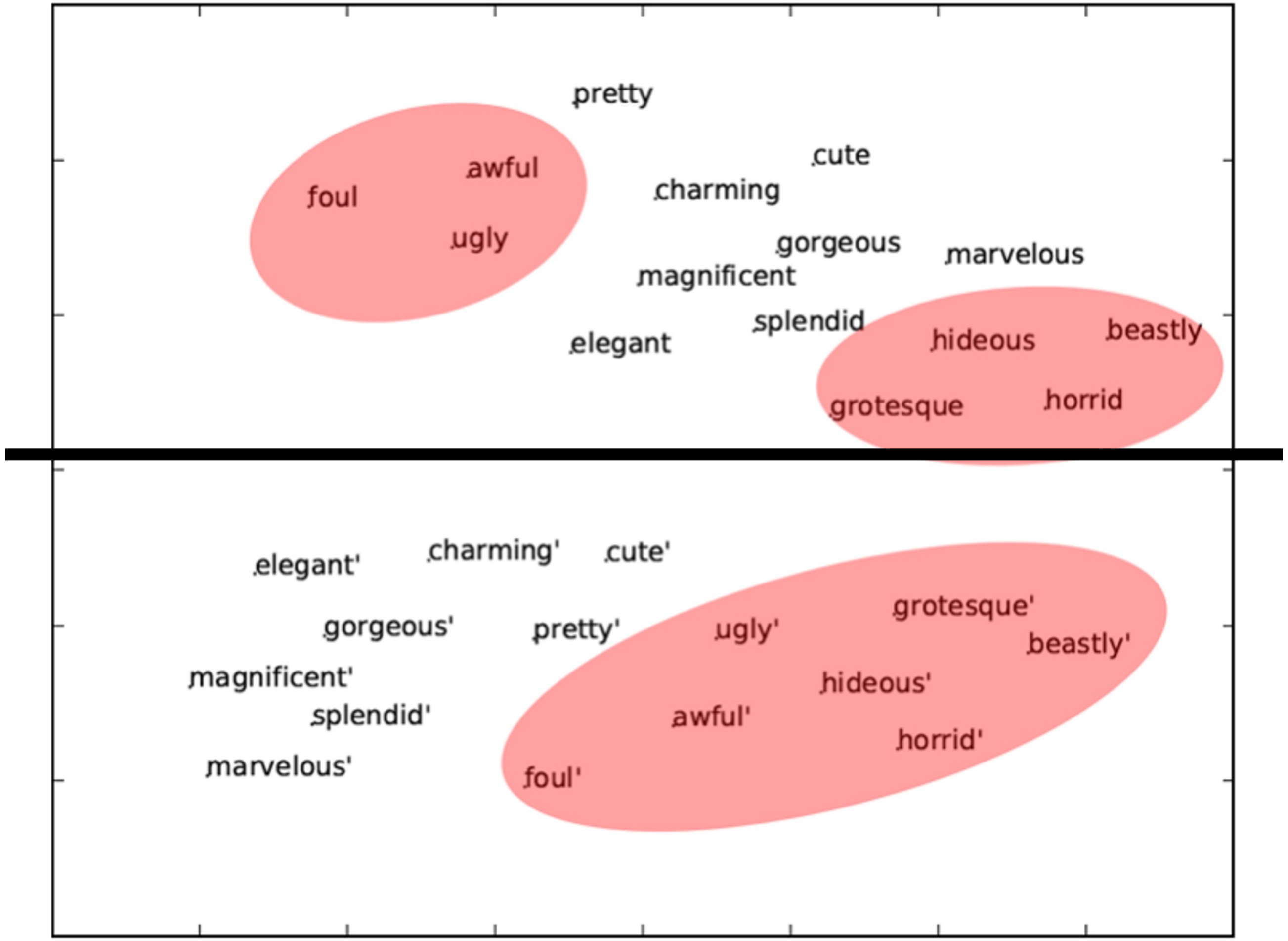


Character-based (Ling et al. 2015)

Multilingual Coordination of Embeddings using dictionaries



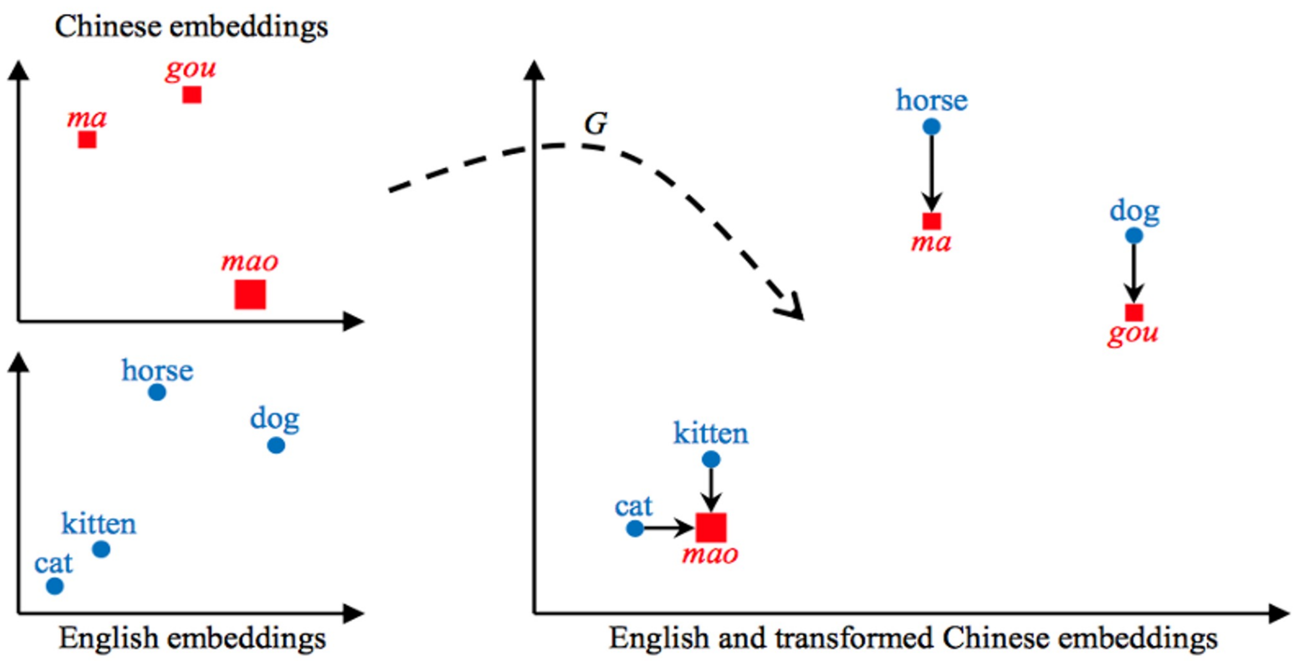
Improving Vector Space Word Representations Using Multilingual Correlation (Faruqui & Dyer, 2014)



Monolingual (top) and multilingual (bottom) word projections of the antonyms (shown in red) and synonyms of “beautiful”

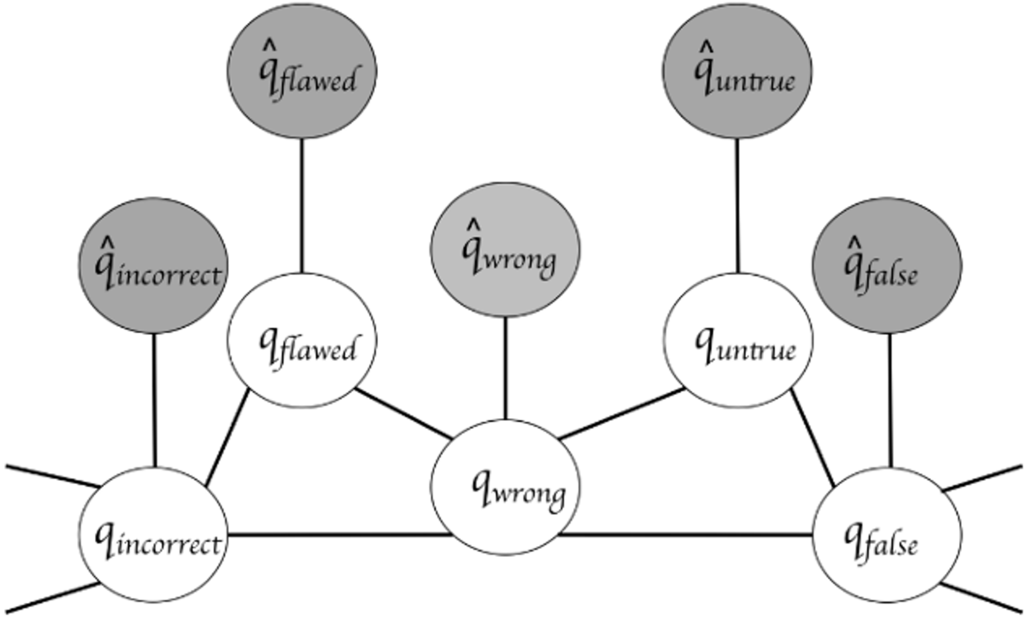
Unsupervised Coordination of Embeddings

- In some cases, we can do it with no dictionary at all!
  - Just use identical words, e.g. the digits (Artexte et al. 2017)
  - Or, just match distributions (Zhang et al. 2017)



Retrofitting of Embeddings to Existing Lexicons

- Make word vectors to match with existing lexicon like WordNet (Faruqui et al. 2015)



$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

De-biasing Word Embeddings

Word embeddings reflect bias in statistics

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

(Bolukbasi et al. 2016)

De-biasing Word Embeddings

Gender stereotype *she-he* analogies.

sewing-carpentry	register-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	hairdresser-barber

Gender appropriate *she-he* analogies.

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

Identify pairs to “neutralize”, find the direction of the trait to neutralize, and ensure that they are neutral in that direction

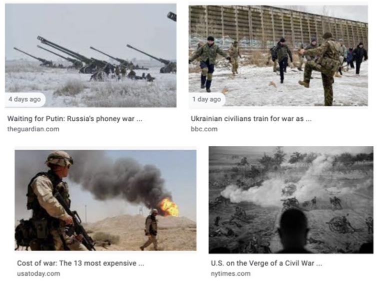
(Bolukbasi et al. 2016)

semantic to model temporal word analogy or relatedness (Szymanski, 2017; Rosin et al., 2017) or to capture the dynamics of semantic relations (Kutuzov et al., 2017)

Different kinds of encoding "context"

- ~~Count-based~~
  - PMI, TF-IDF
- ~~Distributed prediction-based (type) embeddings~~
  - Word2vec, GloVe, Fasttext
- **Distributed contextual (token) embeddings from language models**
  - ELMo, BERT, GPT
- ~~Many more variants~~
  - Multilingual / multi-sense / syntactic embeddings, etc

Questions



- We've just learned how to learn the meaning of "bank" from data as a dense vector. What if meaning of "bank" can be different by context? Can we learn the vectors dynamically adaptable by context?
- How do you interpret the vector? You only know the "relationship" between words but not meaning of word itself. Does each dimension of the vector in distributional semantics correspond to "component" in the decompositional semantics?
- Some words like "war" include various information. Can we quantify the abstract nature of words in distributed representations?