Natural Language Processing

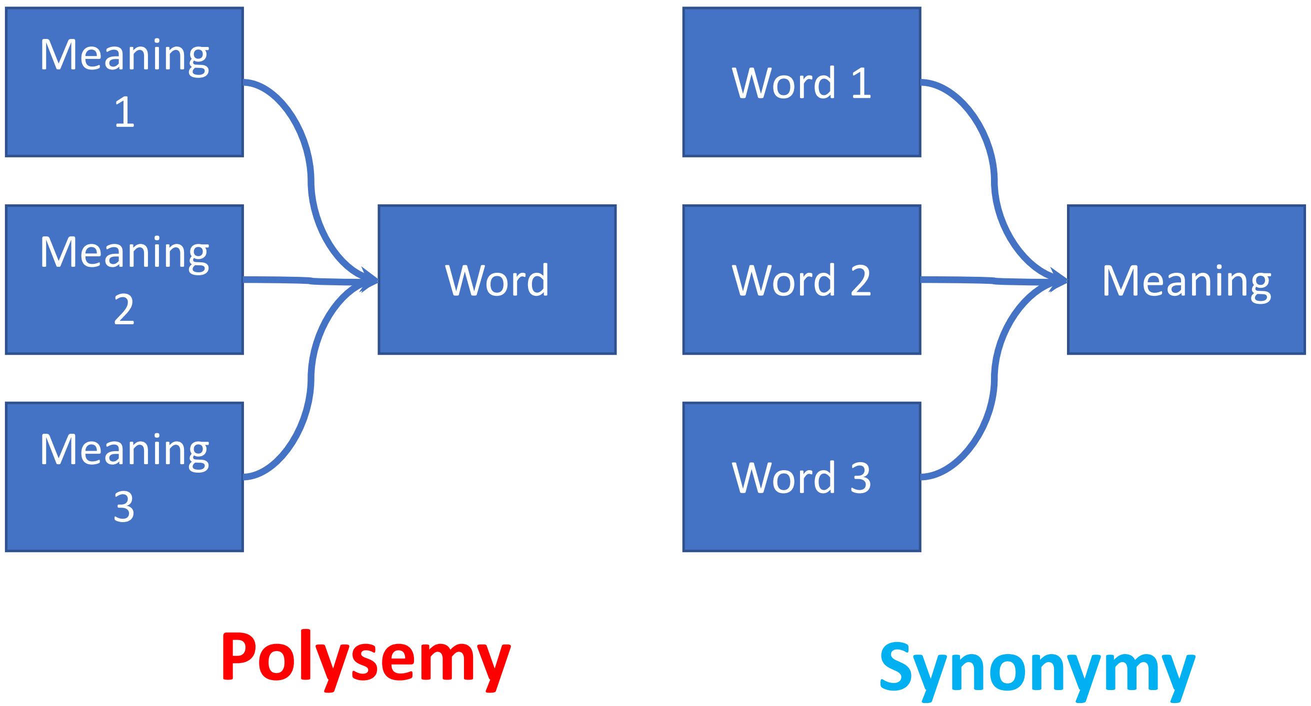
Word Meanings

Dr. Uzair Ahmad

Program

- Word meanings and context
- The distributional hypothesis
- Evaluation of representations

Word meanings: NLP Challenge



The distributional hypothesis

Acquire meaningful representations from unlabeled data

A bottle of _____ is on the table.

Everybody likes _____.

Don't have _____ before you drive.

We make _____ out of corn.

Lin, D. (1998). Automatic retrieval and clustering of similar words. See col (1998), pp.768–774.

The distributional hypothesis

(14.1) A bottle of ____ is on the table.

(14.2) Everybody likes ____.

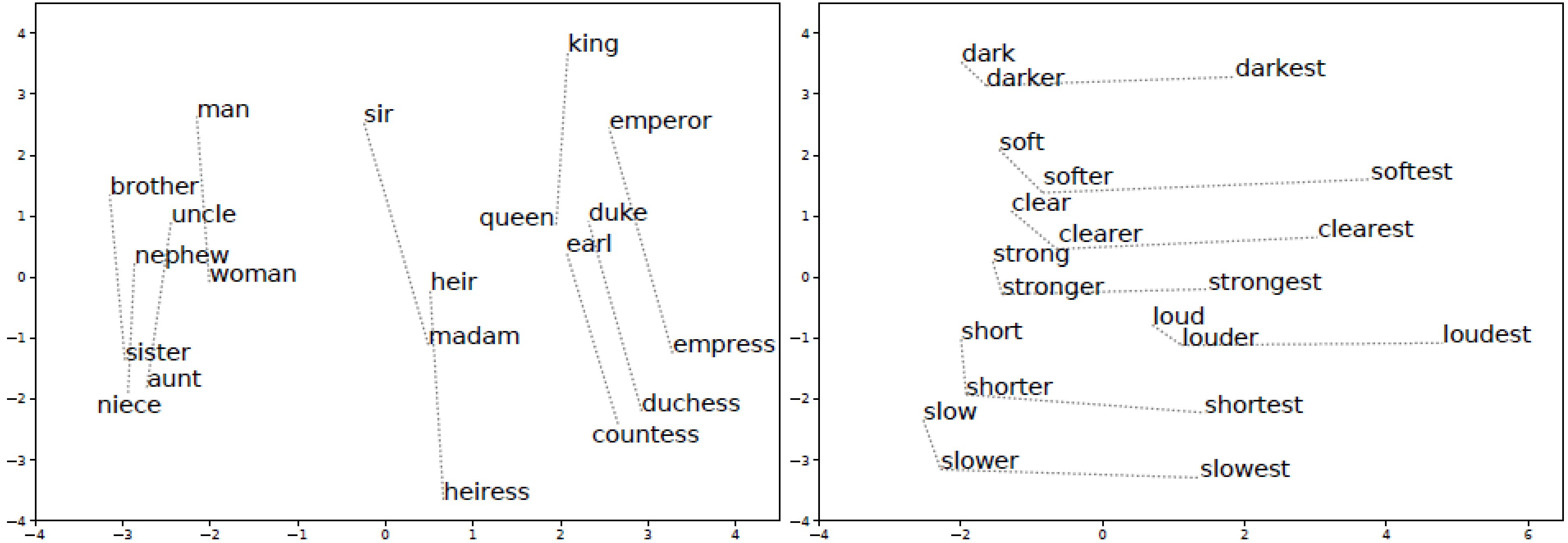
(14.3) Don't have ____ before you drive.

(14.4) We make ____ out of corn.

contextual properties					
	(14.1)	(14.2)	(14.3)	(14.4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
choices	0	1	0	0	
wine	1	1	1	0	

You shall know a word by the company it keeps.
(Firth 1957)

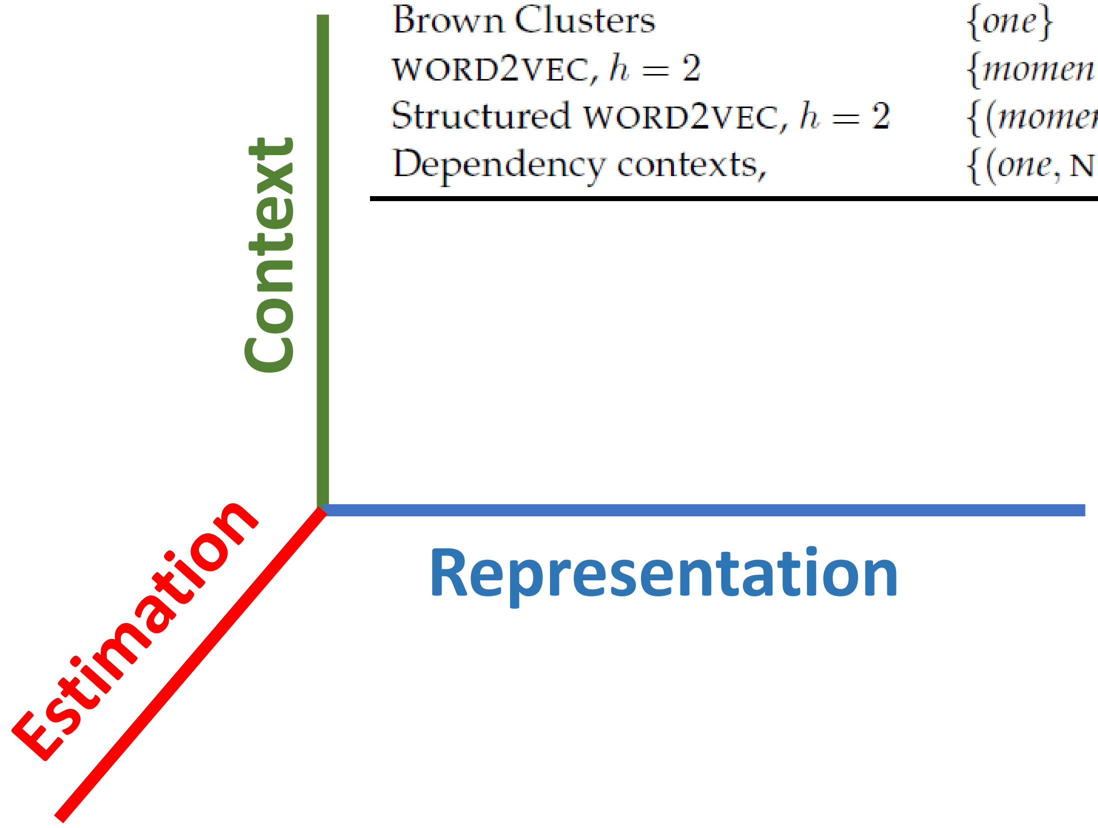
Lexical semantic relationships



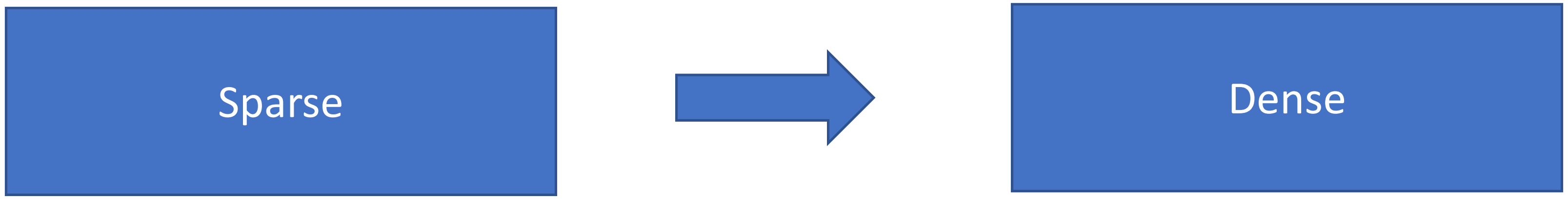
Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. (2014)

Word representations

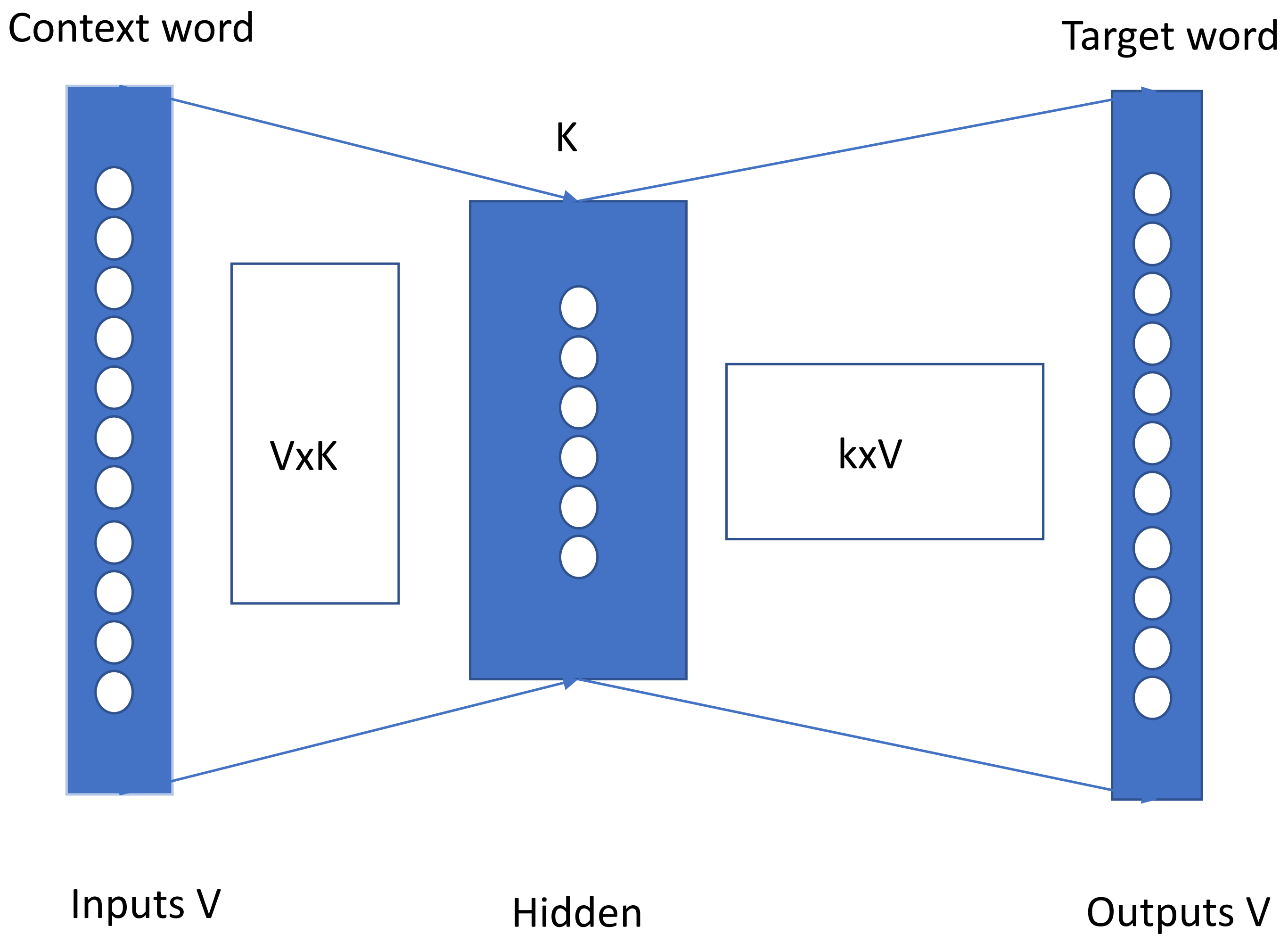
Brown Clusters	$\{one\}$
WORD2VEC, $h = 2$	$\{moment, one, English, complications\}$
Structured WORD2VEC, $h = 2$	$\{(moment, -2), (one, -1), (English, +1), (complications, +2)\}$
Dependency contexts,	$\{(one, \text{NSUBJ}), (English, \text{DOBJ}), (moment, \text{ACL}^{-1})\}$



Transition



Neural word embeddings

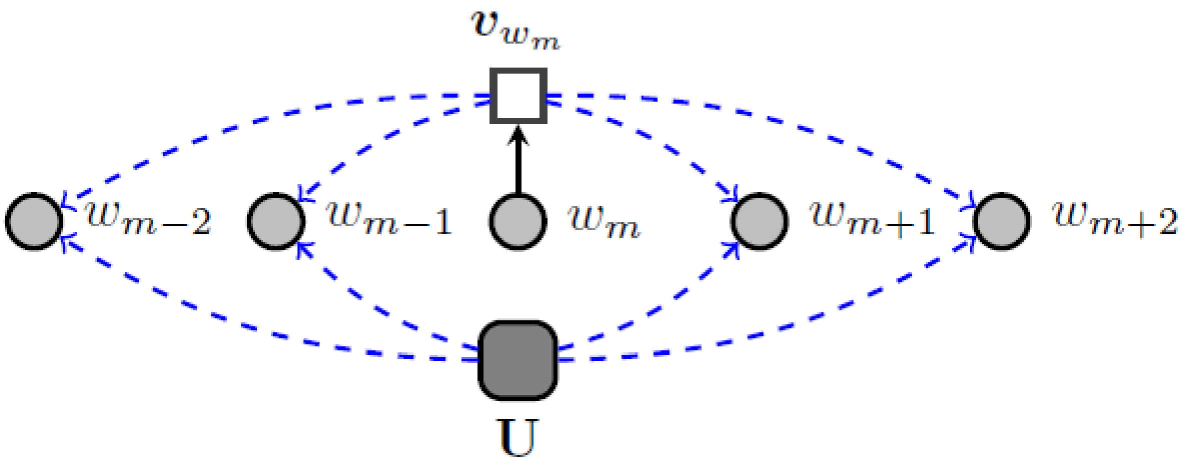


$$y_i = P(w_i \mid w_{context})$$

$$y_i = \frac{exp^{o_i}}{\sum_{n=1}^{V} exp^{o_n}}$$

Neural word embeddings

- Skipgram Model
  - The context is predicted from the word $P(w_c \mid w_m)$



Skip-gram word2Vec training data

```
... lemon,   a [tablespoon of apricot jam,       a] pinch ...
                c1          c2     w     c3      c4
```

positive examples +

w	c_{pos}
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

w	c_{neg}	w	c_{neg}
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

Skip-gram word2vec: Intuition

```
... lemon,   a [tablespoon of apricot  jam,        a] pinch ...
                c1            c2     w       c3          c4
```

P(+ |apricot, jam) > P(- |apricot, regression)

P(+ |w, c) > P(- |w, ~c)

$$P(+|w,c) = \frac{1}{1+e^{-w.c}}$$

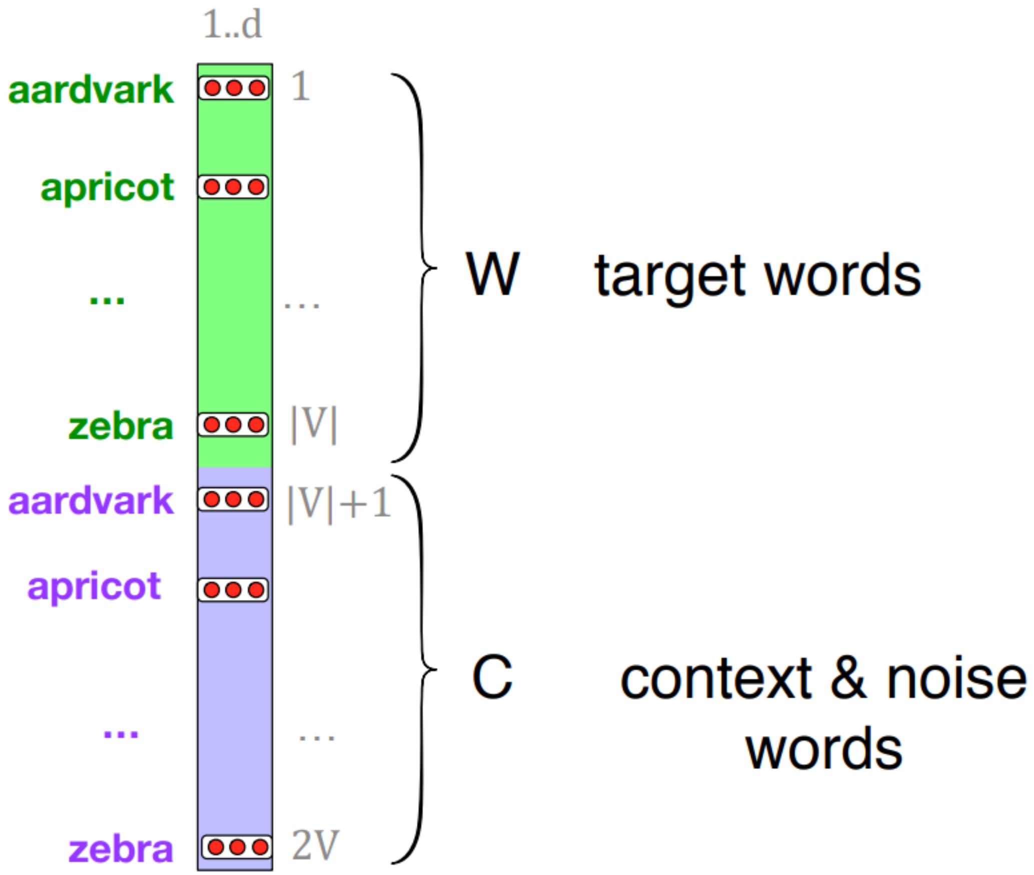
$$P(-|w,c) = \frac{1}{1+e^{w.c}}$$

Skip-gram word2vec and Logistic Regression

$$P(+|w, c_{1:L}) = \prod_{i=1}^{L} \frac{1}{1+e^{w.c_i}}$$

$$P(-|w, c_{1:L}) = \sum_{i=1}^{L} log \frac{1}{1+e^{-w.c_i}}$$

$\boldsymbol{\theta} =$



SLP: C6

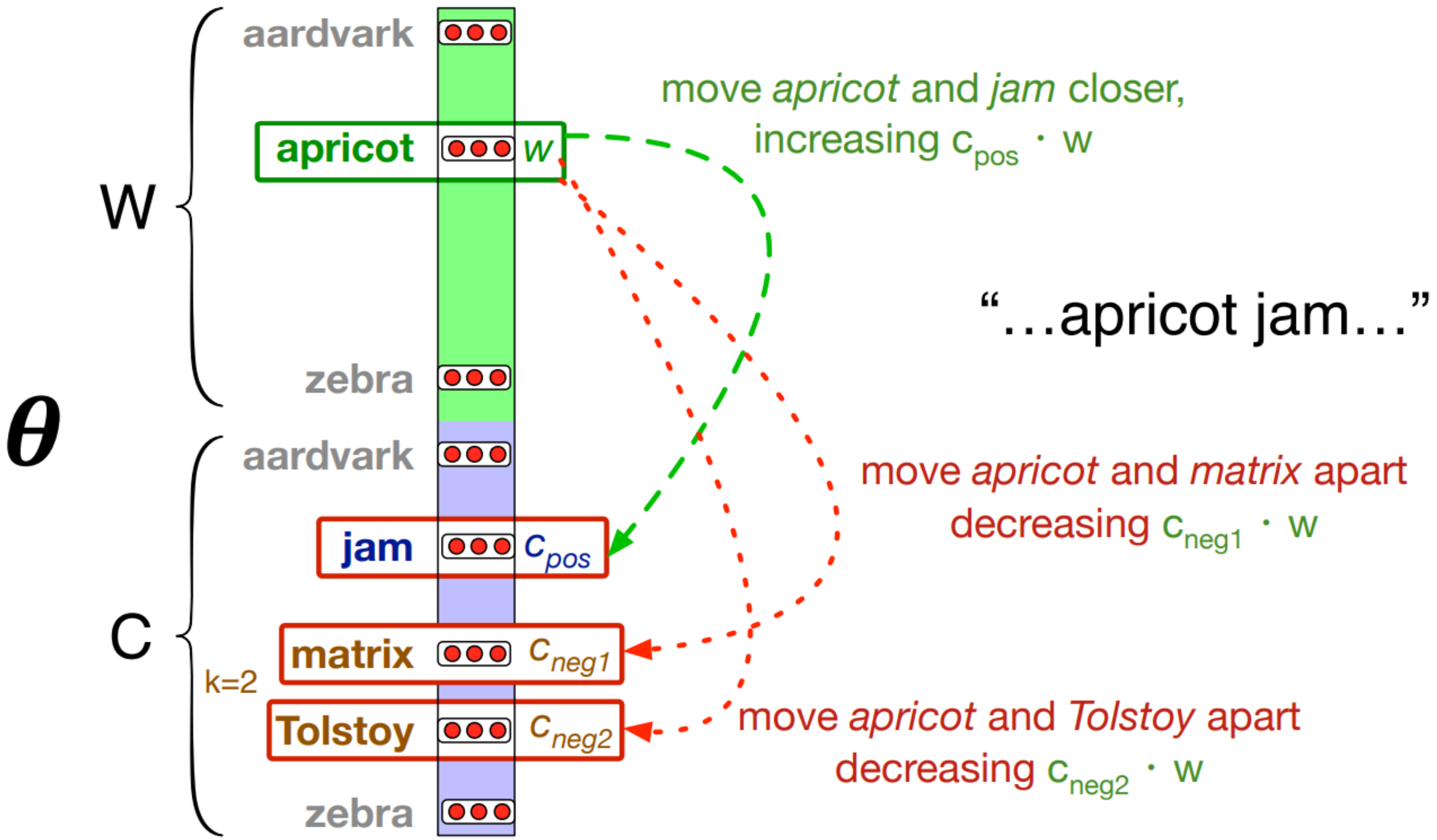
Skip-gram Loss Function

- The goal of the learning algorithm is to adjust those embeddings to
  - **Maximize the similarity of the target word, context word pairs (w, cpos)**
  - **Minimize the similarity of the (w, cneg)**

$$
\begin{aligned}
L_{CE} &= -\log\left[P(+|w,c_{pos})\prod_{i=1}^{k}P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log\left(1-P(+|w,c_{neg_i})\right)\right] \\
&= -\left[\log\sigma(c_{pos}\cdot w)+\sum_{i=1}^{k}\log\sigma(-c_{neg_i}\cdot w)\right]
\end{aligned}
$$

Skip-gram Learning the weights

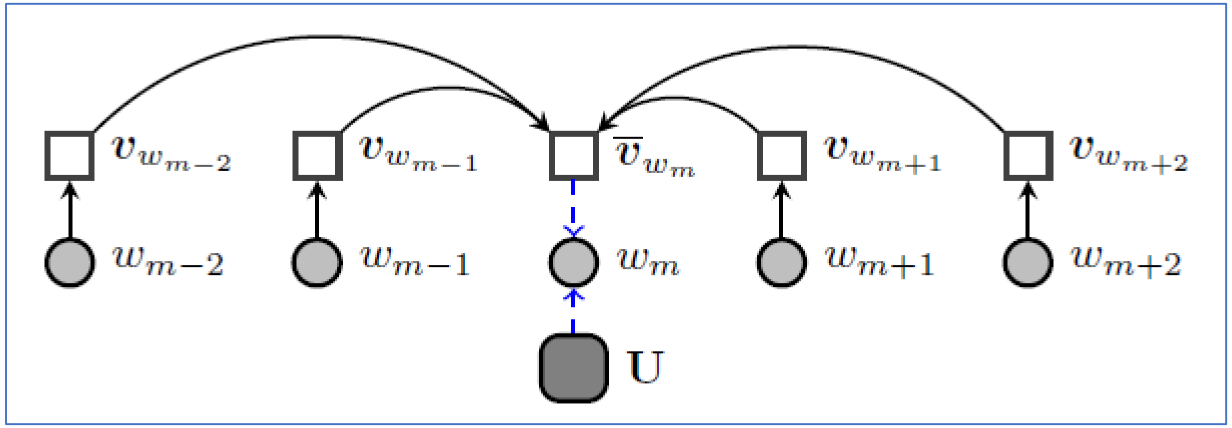
- One step of GD



Neural word embeddings

- Continuous bag-of-words (CBOW)
  - Simplified context
    - Immediate neighborhood of size h

$$\overline{v}_m = \frac{1}{2h} \sum_{n=1}^{h} v_{w_{m+n}} + v_{w_{m-n}}$$



  - $P(w_i \mid w_{context})$

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. In Proceedings of International Conference on Learning Representations.

Neural word embeddings

- Continuous bag-of-words (CBOW)

the corpus likelihood

$$\log \mathrm{p}(\boldsymbol{w}) \approx \sum_{m=1}^{M} \log \mathrm{p}(w_m \mid w_{m-h}, w_{m-h+1}, \ldots, w_{m+h-1}, w_{m+h})$$

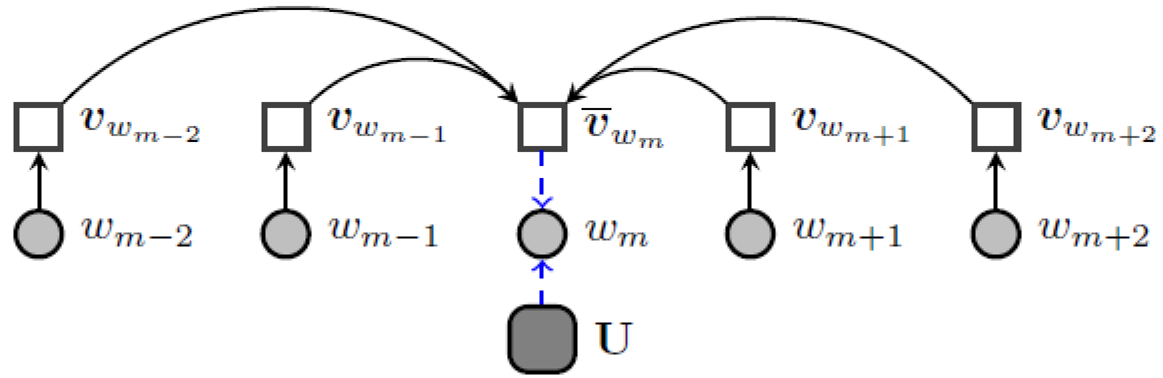
$$= \sum_{m=1}^{M} \log \frac{\exp\left(\boldsymbol{u}_{w_m} \cdot \overline{\boldsymbol{v}}_m\right)}{\sum_{j=1}^{V} \exp\left(\boldsymbol{u}_j \cdot \overline{\boldsymbol{v}}_m\right)}$$

$$= \sum_{m=1}^{M} \boldsymbol{u}_{w_m} \cdot \overline{\boldsymbol{v}}_m - \log \sum_{j=1}^{V} \exp\left(\boldsymbol{u}_j \cdot \overline{\boldsymbol{v}}_m\right).$$

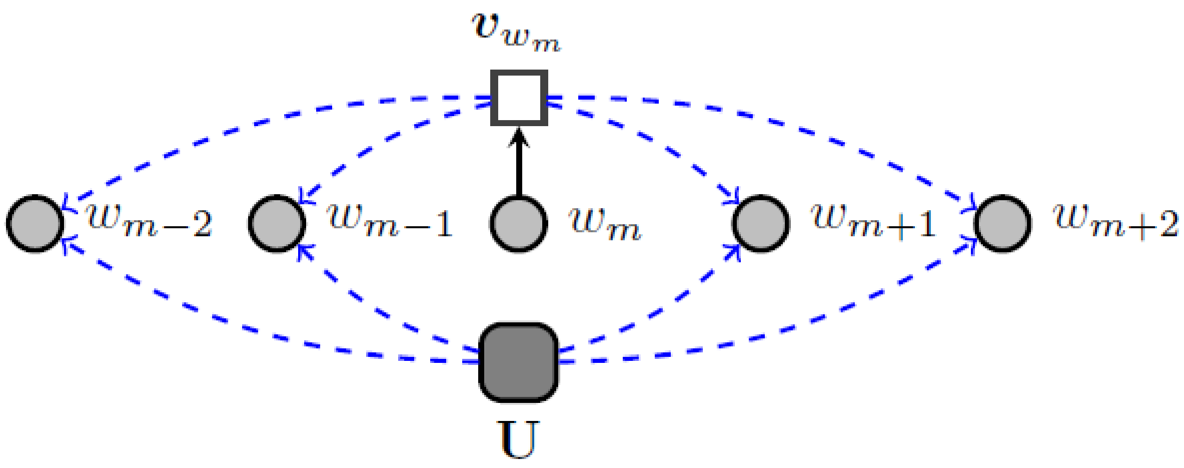
Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. In Proceedings of International Conference on Learning Representations.

Neural word embeddings

CBOW



Skipgram



Evaluating word embeddings

- Intrinsic (intuition based)
  - Word similarity
    - WordSim353 dataset
  - Word analogies
    - King : queen :: man : **?**
- Extrinsic (Empirical evidence)
  - downstream tasks
    - Sequence labeling
    - Document classification

Levy, O., Y. Goldberg, and I. Dagan (2015). Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3, 211–225

Summary

- Distributed representations
  - Latent Semantic Analysis
  - Brown clusters
  - Neural word embeddings
- Evaluation methods
  - Intrinsic
  - Extrinsic