Distributional Representation of Words

Meaning of a Word

- Definition: **meaning** (Webster dictionary)

1 **a** : the thing one intends to convey especially by language : PURPORT • Do not mistake my *meaning*.

b : the thing that is conveyed especially by language : IMPORT • Many words have more than one *meaning*.

2 : something meant or intended : AIM • a mischievous *meaning* was apparent

3 : significant quality; *especially* : implication of a hidden or special significance • a glance full of *meaning*

4 **a** : the logical connotation of a word or phrase

b : the logical denotation or extension of a word or phrase

From Atomic to Distributional

Atomic representation

similarity between (hotel, motel)

= similarity between (hotel, capacity)

Distributional representation

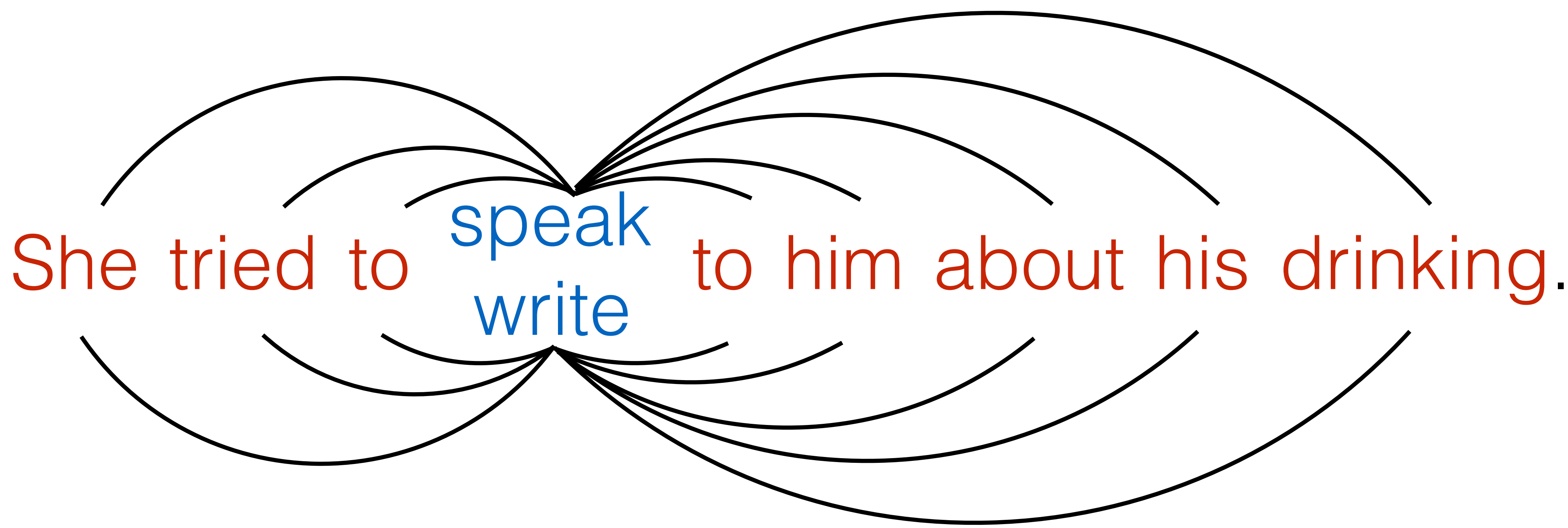
similarity between (hotel, motel)

> similarity between (hotel, capacity)

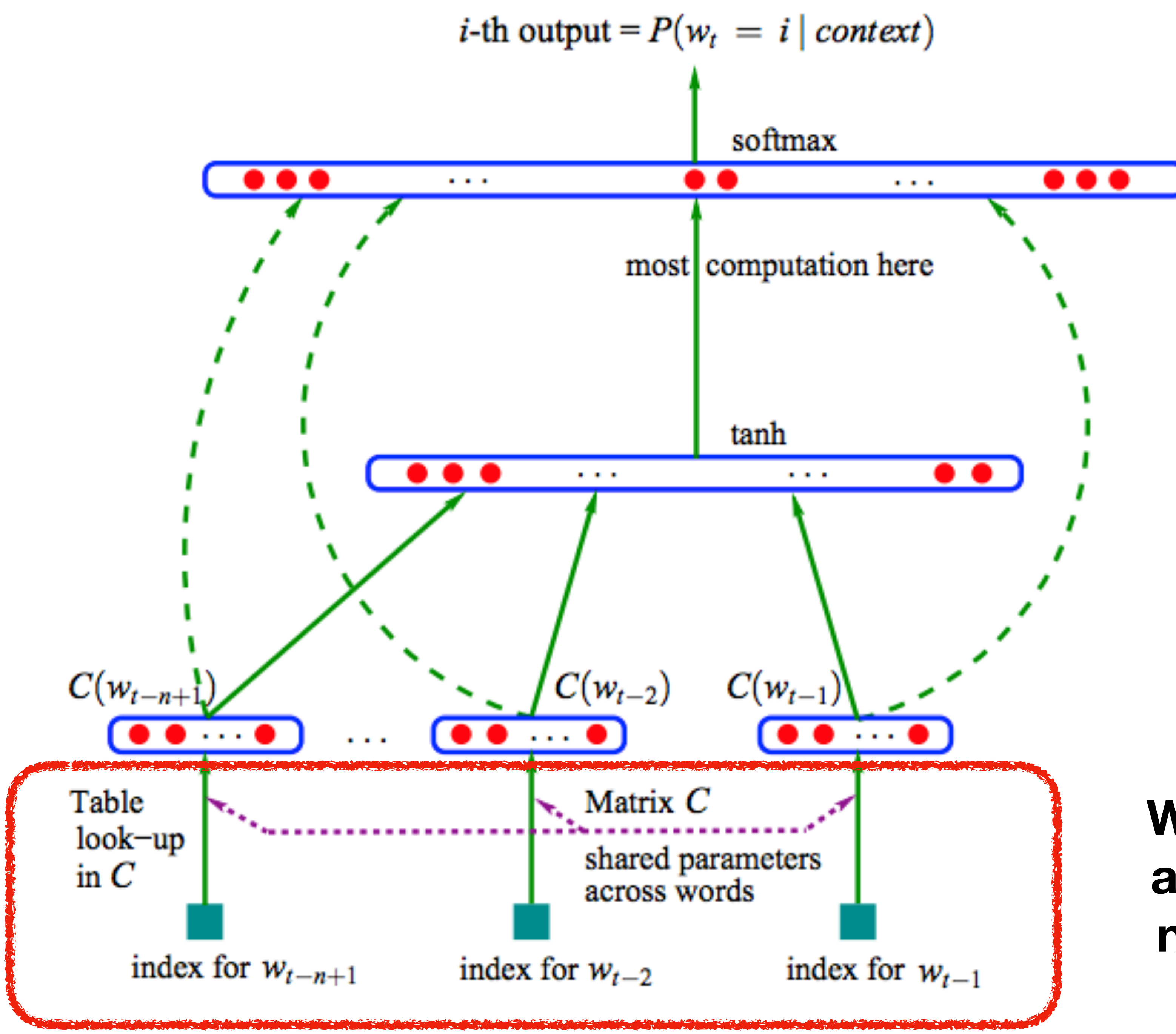
Word Representations by Context

"A word is characterized by the company it keeps."

— Firth 1957



Word Representations via Language Model



Word embedding as parameters in neural networks

Reference:http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf

Simplification via A Log-Bilinear Model

Log-bilinear model

$$P(w|w_1, \ldots, w_{n-1}) \propto \boxed{\exp\left(\sum_{i=1}^{n-1} u_w H_i v_{w_i}\right)}$$

Two sets of word vectors

Reference: https://www.cs.toronto.edu/~amnih/papers/threenew.pdf

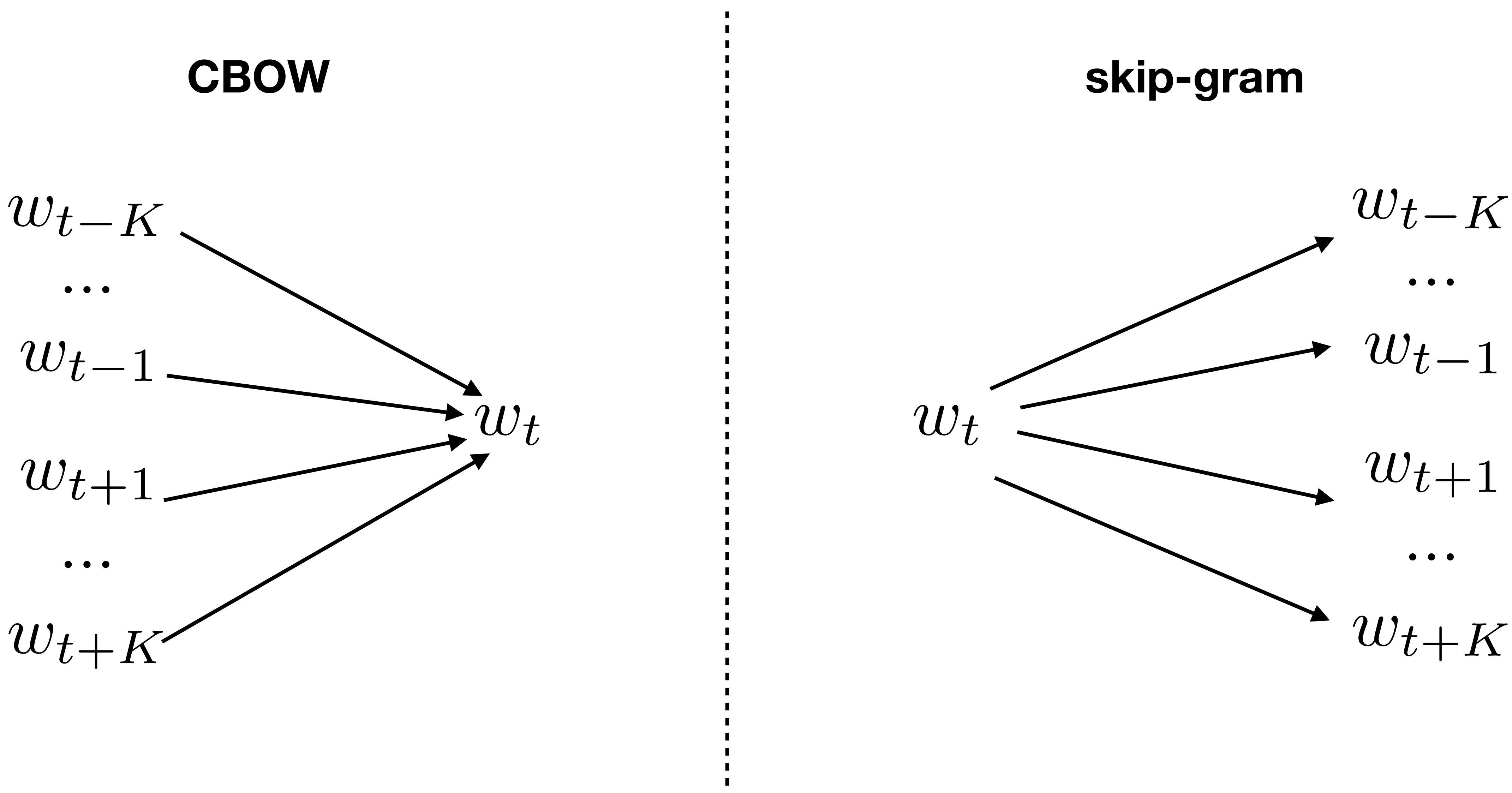
Practical Issues

- Language model only takes past context into consideration
- Future context also matters for word representation
- Cannot scale to large corpora due to normalization.

Our objective is finding good representations of words instead of good language model

Larger Context => Better Representation

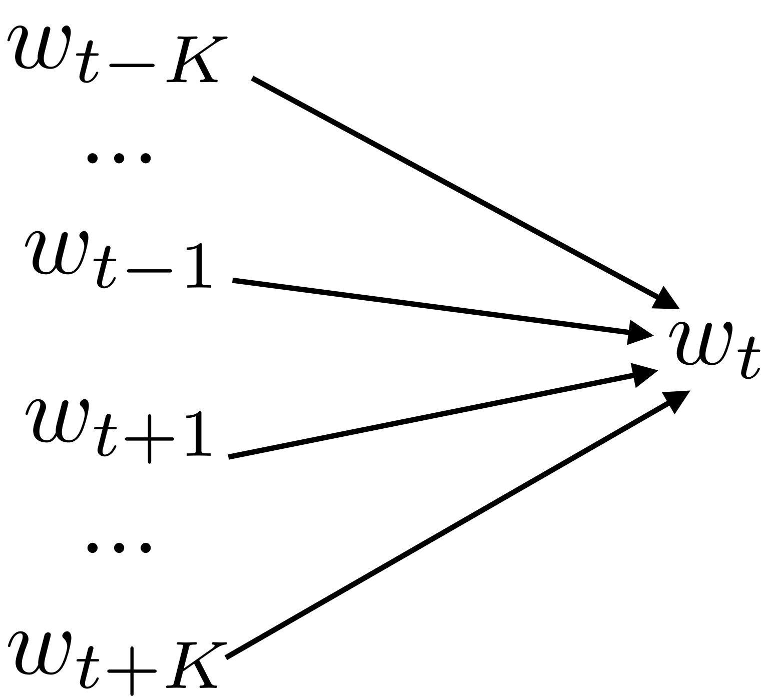
Word2vec: prediction between every word and its context



Continuous Bag-of-Words

Predict the target word from bag-of-words context.

$$\max_{u,v} \prod_{t=1}^{T} p\left(w_t \middle| w_{t-K}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+K}\right)$$



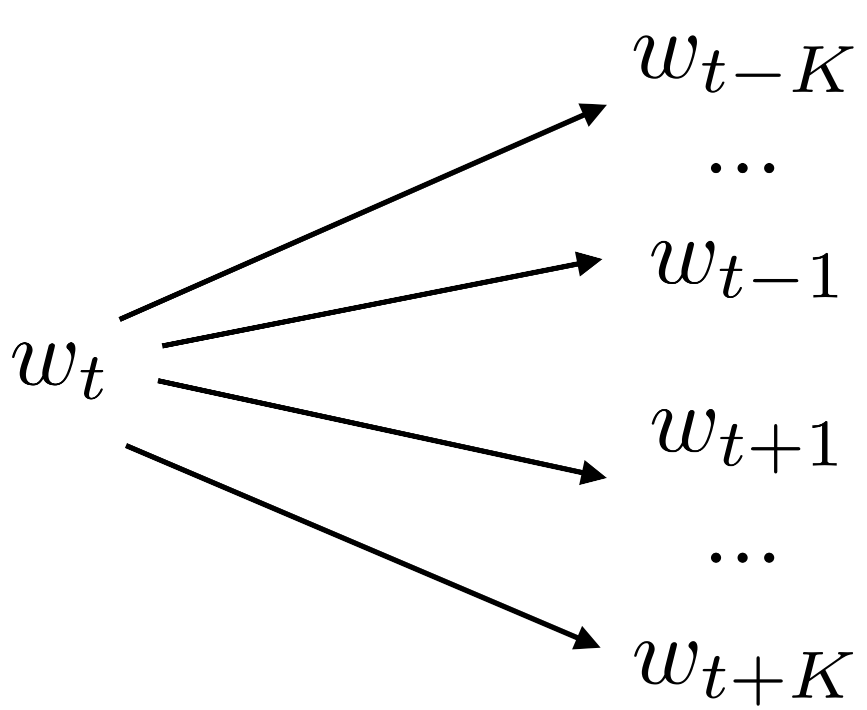
$$p(w_t | w_{t-K}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+K})$$

$$\propto \exp\left[u_{w_t}^{\mathrm{T}} \left(\sum_{k=-K, k\neq 0}^{K} v_{w_{t+k}} \right)\right]$$

Skip-Gram

Predict context words from the a target word.

$$\max_{u,v} \prod_{t=1}^{T} p\left(w_{t-K}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+K} \middle| w_t\right)$$



$$\begin{aligned} & p(w_{t-K}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+K} | w_t) \\ & = \prod_{k=-K, k\neq 0}^{K} p(w_{t-k} | w_t) \\ & \propto \prod_{k=-K, k\neq 0}^{K} \exp\left(u_{w_{t-k}}^{\mathrm{T}} v_{w_t}\right) \end{aligned}$$

Pain of Normalization

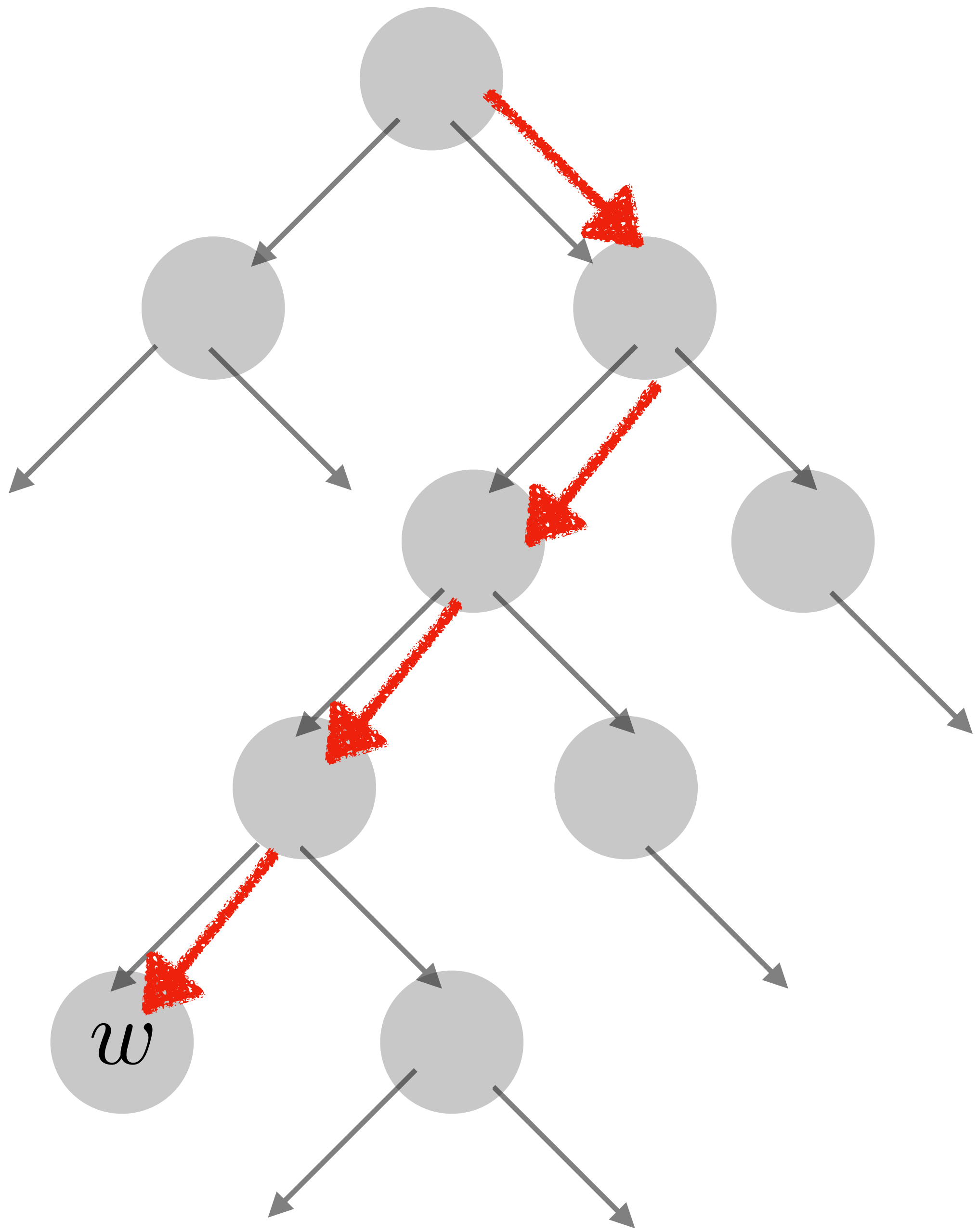
Prediction as V-class classification given a hidden variable θ

$$
\begin{aligned}
p(w|\theta) &\propto \exp\left(u_w^{\mathrm{T}}\theta\right) \\
&= \frac{\exp\left(u_w^{\mathrm{T}}\theta\right)}{\sum_{w \in \text{vocabulary}} \exp\left(u_w^{\mathrm{T}}\theta\right)}
\end{aligned}
$$

curse of dimensionality O(V)

Solved via hierarchical softmax and negative sampling

Hierarchical Softmax



- Constructed a Huffman tree for words
- Each node is associated with a vector
- The probability of going left/right given θ

$$p(\text{left}|\theta, \text{node}) = \sigma(\theta^{\mathrm{T}} u_{\text{node}})$$

$$p(\text{right}|\theta, \text{node}) = \sigma(-\theta^{\mathrm{T}} u_{\text{node}})$$

$\sigma(\cdot)$ is the sigmoid function

$$p(w|\theta) = \prod_{l \in \text{path}} p(l|\theta, \text{parent}(l))$$

Computational complexity O(log V)

Reference: https://www.cs.toronto.edu/~amnih/papers/threenew.pdf

Negative Sampling

V-ary classification -> Binary Classification

label Y	observed samples	model probability
positive	real (w, θ) from data	$p(+\vert(w,\theta)) = \sigma(u_w^{\mathrm{T}}\theta)$
negative	randomly generated w from uniform distribution	$p(-\vert(w,\theta)) = \sigma(-u_w^{\mathrm{T}}\theta)$

$$\max_{u,v} \sum_{w,\theta} \left(\log p(+|(w,\theta)) + k\mathbb{E}_{w'\sim\mathrm{unigram}} \log p(-|(w',\theta))\right)$$

In practice, replaced by random samples in SGD

Only Cooccurrence Matters

Predicting surrounding words of each word

=> cooccurrence directly

A series of many genres, including fantasy, drama, coming of age,…

(series, genres)
(of, genres)
(many, genres)
(including, genres)
(fantasy, genres)
(drama, genres)

target words

context words

	…	genres	…
…	…	…	…
series	…	+1	…
of	…	+1	…
many	…	+1	…
including	…	+1	…
fantasy	…	+1	…
drama	…	+1	…
…	…	…	…

Low-rank Representation

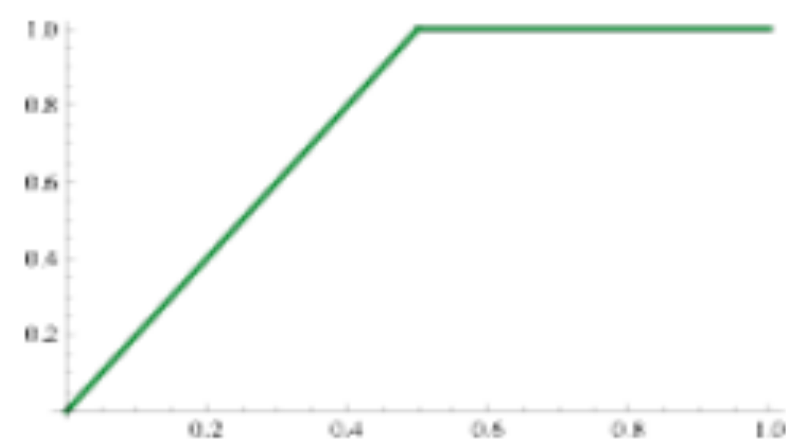
Sparsity => low rank for robustness

$$\min_{u,v} \frac{1}{2} \sum_{w_1,w_2} f(\hat{p}(w_1, w_2)) \left(u_{w_1}^{\mathrm{T}} v_{w_2} - \log \hat{p}(w_1, w_2)\right)$$

$\hat{p}(w_1, w_2)$ is the empirical probability of cooccurrence

More frequent samples are more robust

$f \sim$



Word Similarity

Nearest neighbors for "**frog**"

frogs

toad

litoria

leptodactylidae

rana

lizard

eleutherodactylus

litoria

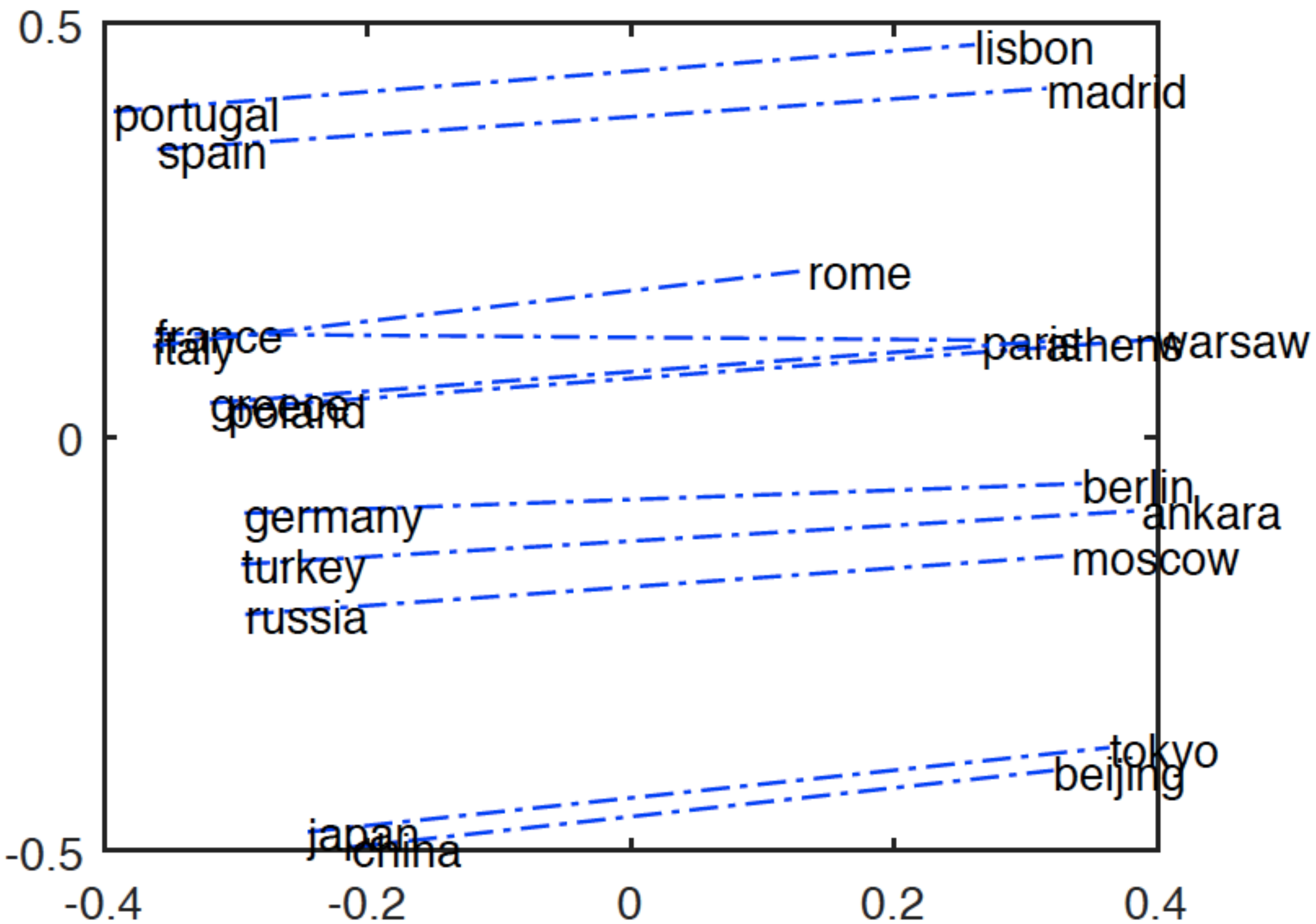
leptodactylidae

rana

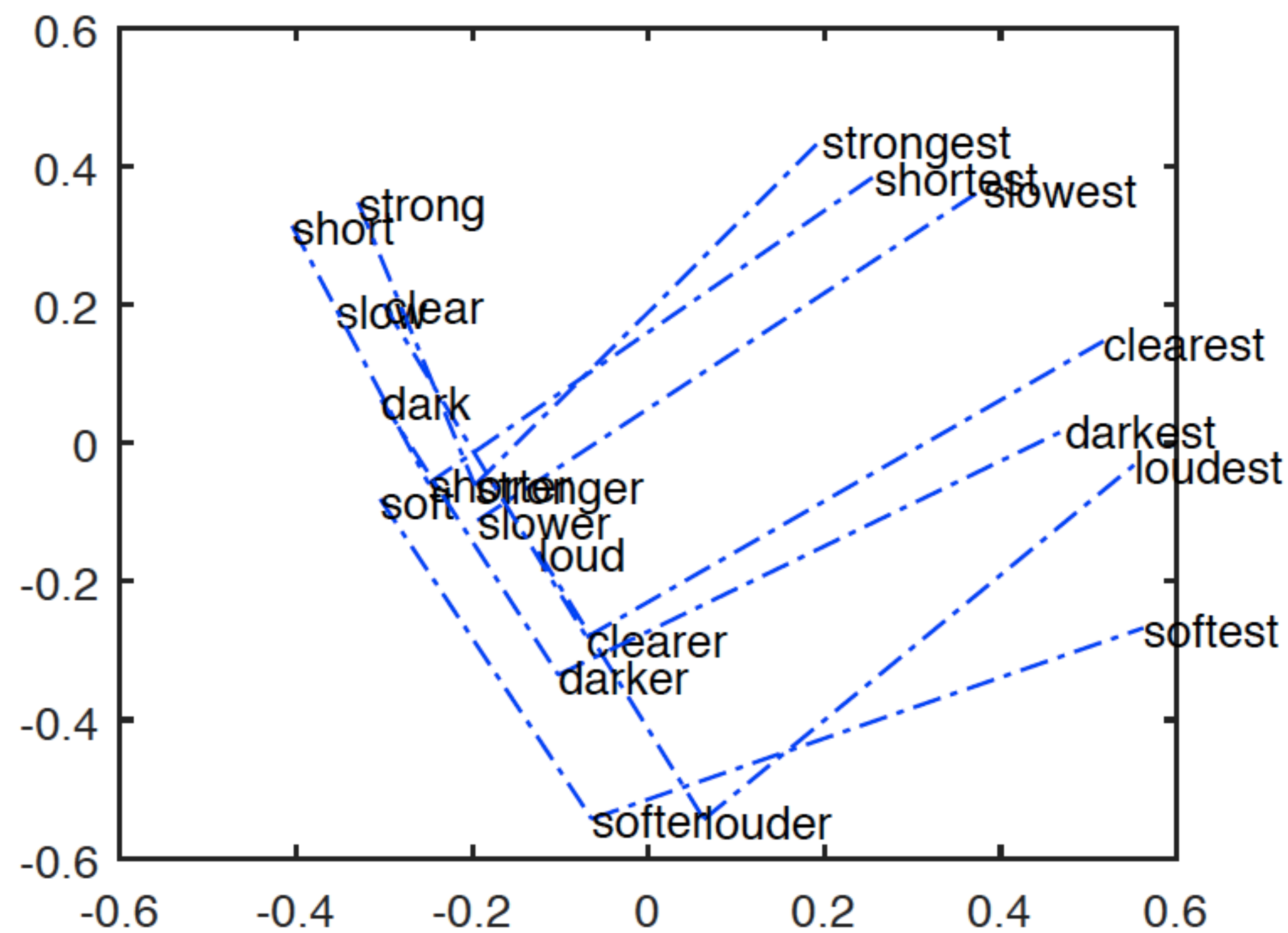
eleutherodactylus

Word Analogy

countries-capticals



comparative-superlative



Demystify

- Skip-gram negative sampling as matrix factorization

Reference: https://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization

- Information-theoretic explanation of SGNS

Reference: http://www.eng.biu.ac.il/goldbej/files/2012/05/ACL_2017.pdf

- Estimating word vectors as a latent variable in a generative LM.

Reference: https://arxiv.org/abs/1502.03520

Only cooccurrence matters!

Recall Skip-Gram Negative Sampling

Let w be the current word, let c be one of its context

Skip-gram tries to predict c by w, denote $M_{w,c} = u_w^{\mathrm{T}} v_c$

$$\begin{aligned}
L_{\mathrm{SG}}^{\mathrm{NS}}(M) &= \sum_{w,c} \#(w,c)(\log\sigma(M_{wc}) + k\mathbf{E}\log\sigma(-M_{w'c})) \\
&= \sum_{w,c} \#(w,c)\log\sigma(M_{wc}) \\
&\quad + \sum_{c} \#(c)k \sum_{w'} \frac{\#(w')}{|D|} \log\sigma(-M_{w'c}) \\
&= \sum_{w,c} \#(w,c)\log\sigma(M_{wc}) \\
&\quad + \sum_{w,c} \#(w,c)k \frac{\#(c)\#(w)}{|D|\#(w,c)} \log\sigma(-M_{wc}) = \sum_{w,c} \#(w,c) L_{wc}(M_{wc}),
\end{aligned}$$

decomposed into independent elements, barring the low-rank constraint

Implicit Matrix Factorization

For each objective,

$$L_{wc}(m) = \left(\log \sigma(m) + k \frac{\#(c)\#(w)}{|D|\#(w,c)} \log \sigma(-m) \right).$$

Without low rank constraint, the optimal is given by,

$$\begin{aligned} \hat{M}_{wc} &= \arg\max_{m \in R} \left(\log \sigma(m) + k \frac{\#(c)\#(w)}{|D|\#(w,c)} \log \sigma(-m) \right) \\ &= \log \left(\frac{|D|\#(w,c)}{\#(c)\#(w)} \right) - \log k. = \boxed{\log \left(\frac{p_{W,C}(w,c)}{p_W(w)p_C(c)} \right)} - \log k. \end{aligned}$$

point-wise mutual information

With low rank constraint, **weighted SVD of PMI matrix**.

Information-Theoretic Explanation

For each word/context pair (W, C) and the label $Y \in \{+, -\}$, the probability is paramterized by the matrix $M = (M_{w,c})$

$$p(Y = 1 | W = w, C = c; M) = \sigma(M_{w,c})$$

Theorem 1: The value of the SGNS objective with k negative samples at the PMI matrix satisfies

Jenson Shannon Mutual Information

$$L_{\mathrm{SG}}^{\mathrm{NS}}(\mathrm{PMI}) = \mathrm{JSMI}_{\frac{1}{k+1}}(W, C)$$

Theorem 2: The difference between the SGNS objective at the PMI matrix and the SGNS objective at a given matrix M can be written as

$$L_{\mathrm{SG}}^{\mathrm{NS}}(\mathrm{PMI}) - L_{\mathrm{SG}}^{\mathrm{NS}}(M) = KL\left(p_{PMI}(Y|W, C) \| P_M(Y|W, C)\right)$$

A Generative Model

Each word is parametrized by a vector v_w

Each sentence is generated via the following process

$$P(w \text{ emitted at time } t | c_t) \propto \exp(v_w^{\mathrm{T}} c_t)$$

c_t is a slowly-moving random walk on a unit sphere.

Word vectors are parameters from a generative model, PMI SVD is the inference procedure from real data.