CS249: ADVANCED DATA MINING

Text Data: Word Embedding

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

May 10, 2017

Announcements

- Homework 3 due today
  - Due May 10th (11:59pm)

- Midterm Exam
  - In class May 15th
  - Closed-Book Exam, no cell phone
  - Bring a simple electronic calculator
  - You can bring an A4 size reference sheet

Methods Learnt: Last Lecture

	Vector Data	Text Data	Recommender System	Graph & Network
Classification	**Decision Tree**; **Naïve Bayes**; **Logistic Regression** **SVM; NN**			Label Propagation
Clustering	**K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means**	**PLSA; LDA**	Matrix Factorization	SCAN; Spectral Clustering
Prediction	**Linear Regression** **GLM**		Collaborative Filtering	
Ranking				PageRank
Feature Representation		Word embedding		Network embedding

Methods to Learn

	Vector Data	Text Data	Recommender System	Graph & Network
Classification	**Decision Tree**; **Naïve Bayes**; **Logistic Regression** **SVM; NN**			Label Propagation
Clustering	**K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means**	**PLSA; LDA**	Matrix Factorization	SCAN; Spectral Clustering
Prediction	**Linear Regression GLM**		Collaborative Filtering	
Ranking				PageRank
Feature Representation		**Word embedding**		Network embedding

Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary

Why Word Representation?

- Finding Synonyms: words that have the same meaning
  - E.g., movie and film
- Finding polysemy: words with multiple meanings
  - E.g., light
- Document representation
  - E.g., aggregation of all the word representation

How to Represent a Word?

- Challenge
  - Discrete structure
- Simple representation
  - One-hot representation: a vector with one 1 and a lot of zeroes
  - E.g., Motel =

[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

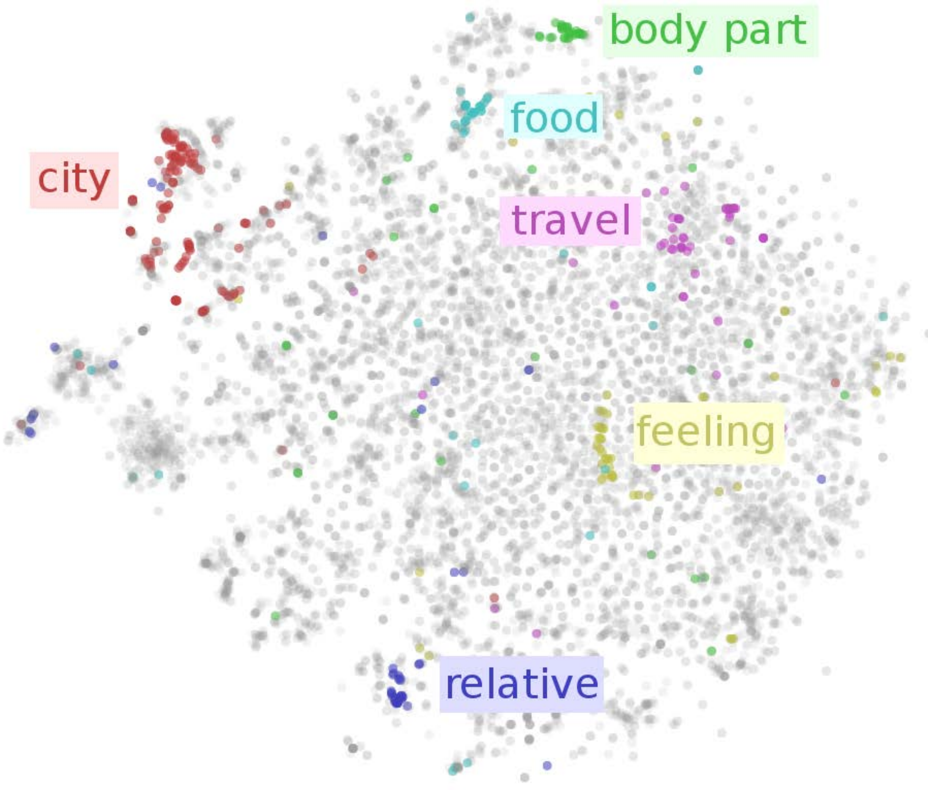
Problem of One-Hot Representation

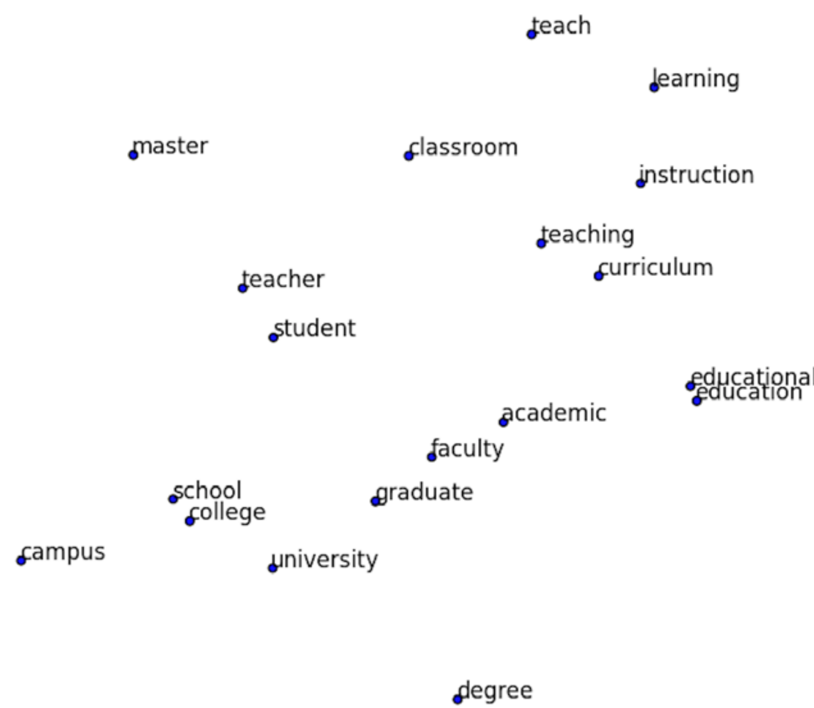
- High dimensionality
  - E.g., for Google news, 13M words
- Sparse
  - Only 1 non-zero value
- Shallow representation
  - E.g.,

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

Word Embedding

- Low dimensional vector representation of every word
  - E.g., motel = [1.3, -1.4] and hotel = [1.2, -1.5]





How to Learn Such Embeddings?

- Using context information!

...he curtains open and the moon shining in on the barely...

...ars and the cold , close moon " . And neither of the w...

...rough the night with the moon shining so brightly , it...

...made in the light of the moon . It all boils down , wr...

...surely under a crescent moon , thrilled by ice-white...

...sun , the seasons of the moon ? Home , alone , Jay pla...

...m is dazzling snow , the moon has risen full and cold...

...un and the temple of the moon , driving out of the hug...

...in the dark and now the moon rises , full and amber a...

...bird on the shape of the moon over the trees in front...

A Naïve Approach

- Build a co-occurrence matrix for words, and apply SVD
  - Example Corpus:
    - I like deep learning.
    - I like NLP.
    - I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

- Issues:
  - Global context
  - SVD is very expensive

Text Data: Word Embedding

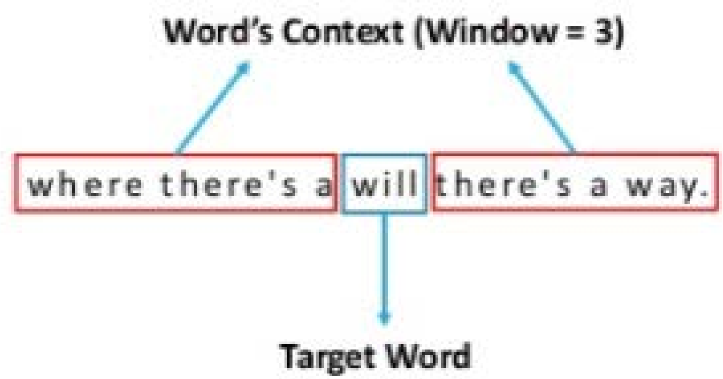
- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary

Word2Vec

- Proposed by Mikolov et al. at Google in 2013
- The most popular word embedding models
- Two architectures are proposed
  - Continuous bag-of-words (CBOW)
  - Skip-gram
- Extremely fast
  - "an optimized single-machine implementation can train on more than 100 billion words in one day"

Main Idea of Word2Vec

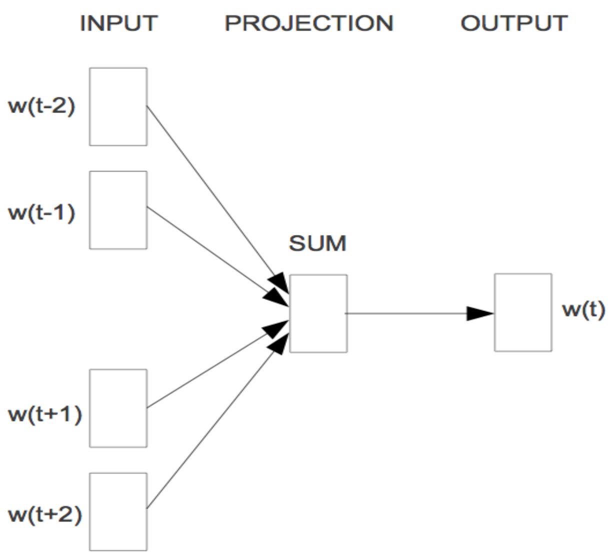
- Consider a local window of a target word



- CBOW: predict the target words given the neighbors
- Skip-gram: predict neighbors given the target words

CBOW

- Predicting target using neighbors

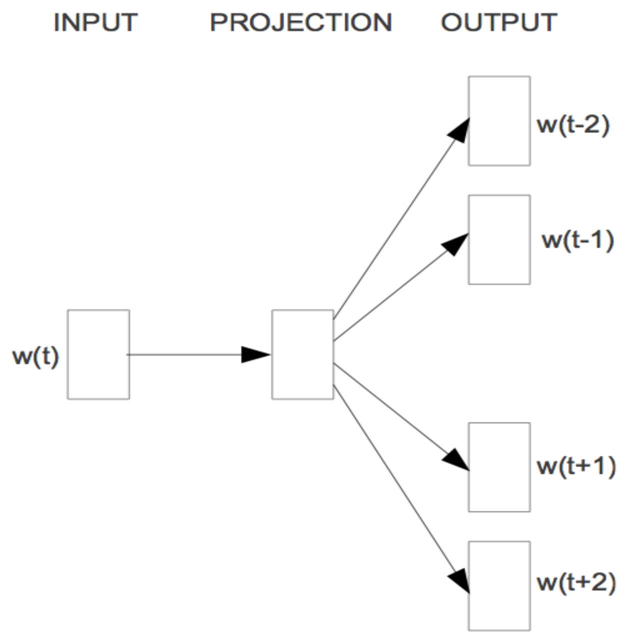


$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} logp(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n})$$

More details can be found in: http://www.1-4-5.net/~dmm/ml/how_does_word2vec_work.pdf

Skip-Gram

- Predicting neighbors using target



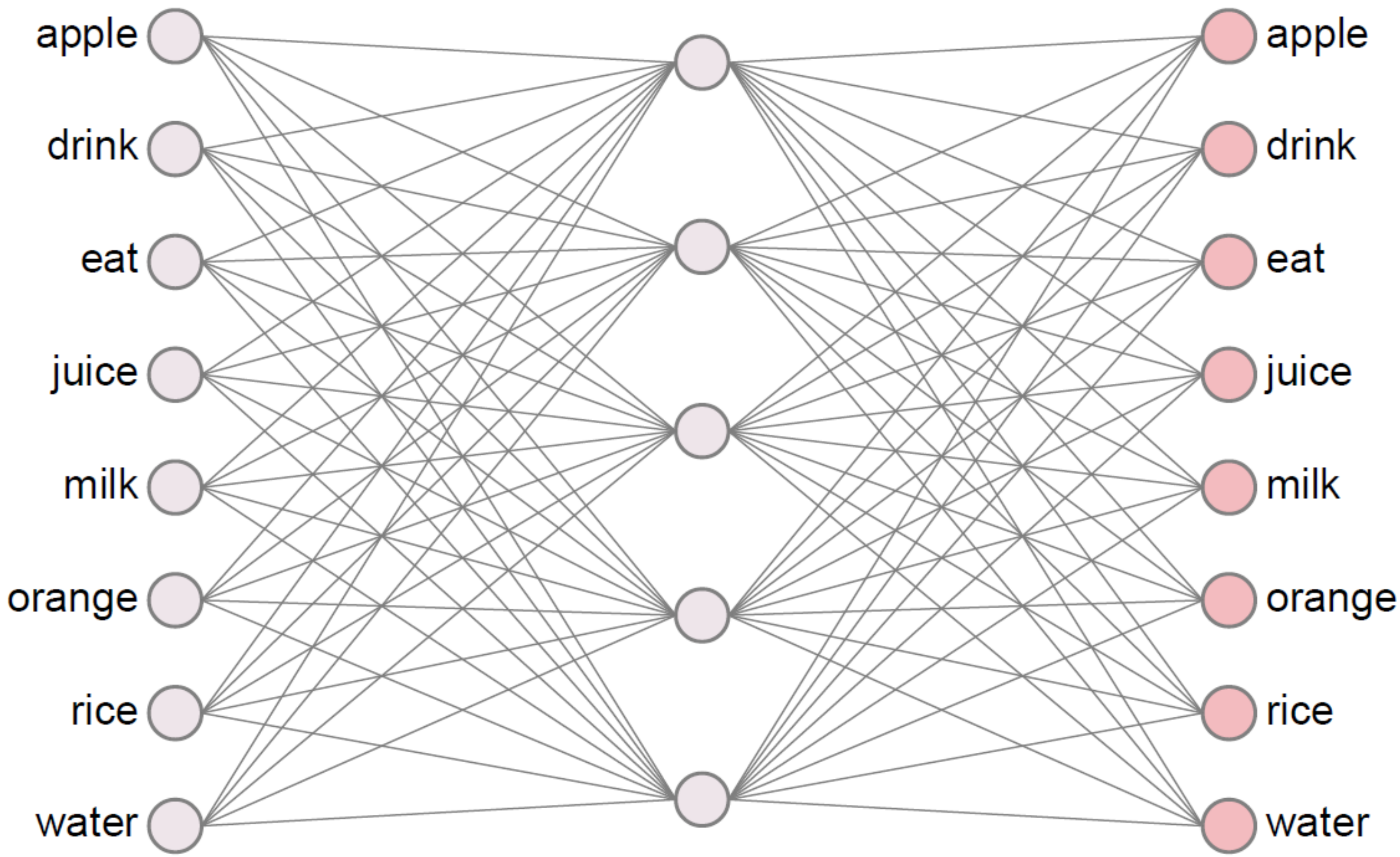
$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\sum_{-n\le j\le n, j\ne 0} logp(w_{t+j}|w_t)$$

The Conditional Probability

- $p(w_{t+j}|w_t)$: the probability to see w_{t+j} in target word w_t's neighborhood
  - Intuition: w_t's embedding should be closer to w_{t+j}'s embedding
  - Every word has two copies of embedding
    - One serves as the role of target (***v***), and the other serves as the role of context (***u***)

$$p(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w=1}^{W} \exp\left(u_w^T v_c\right)}$$

A Neural Network Point of View



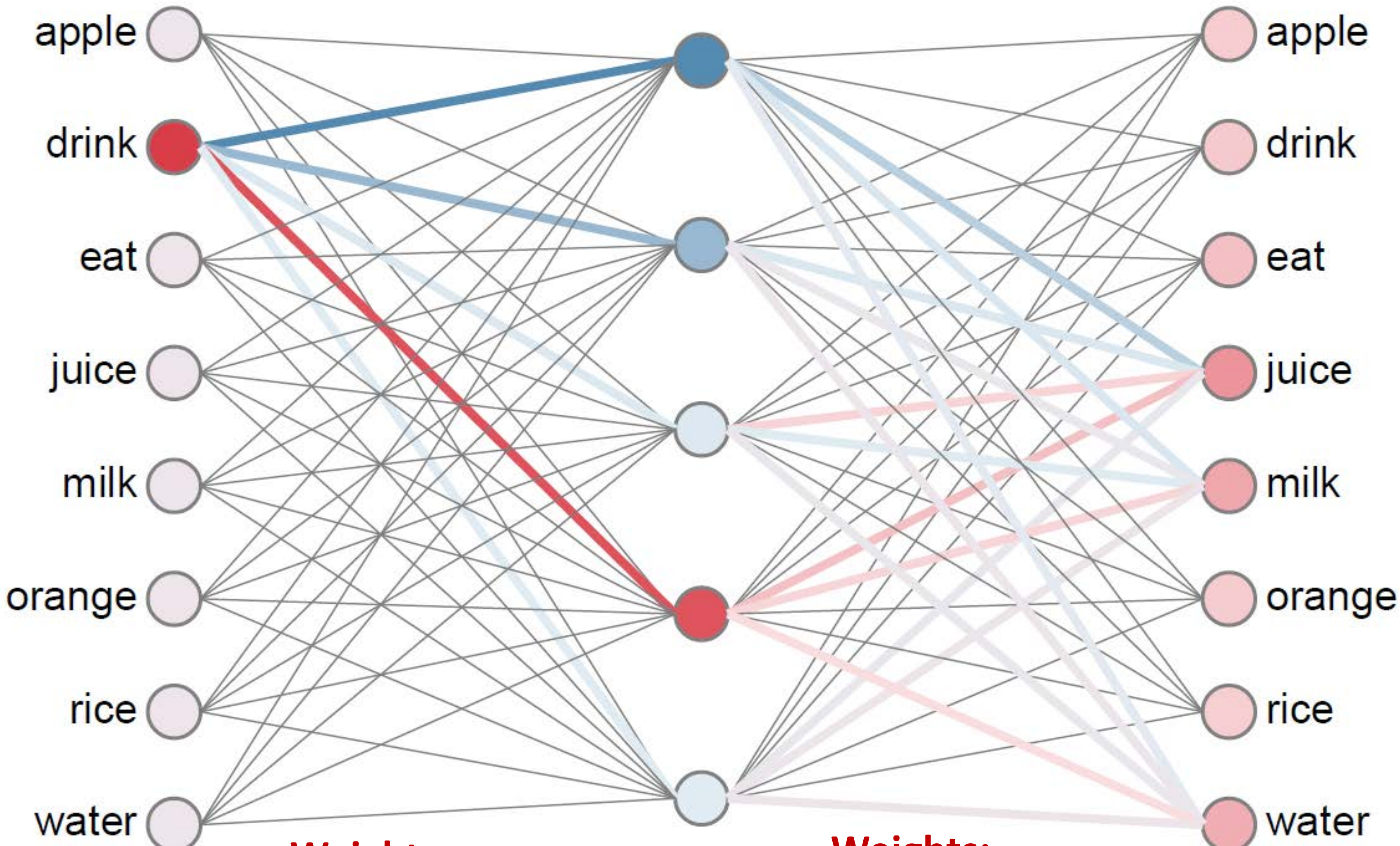
Input Layer:
one-hot vector

Hidden Layer:
Linear (Identity)

Output Layer:
softmax

Demo

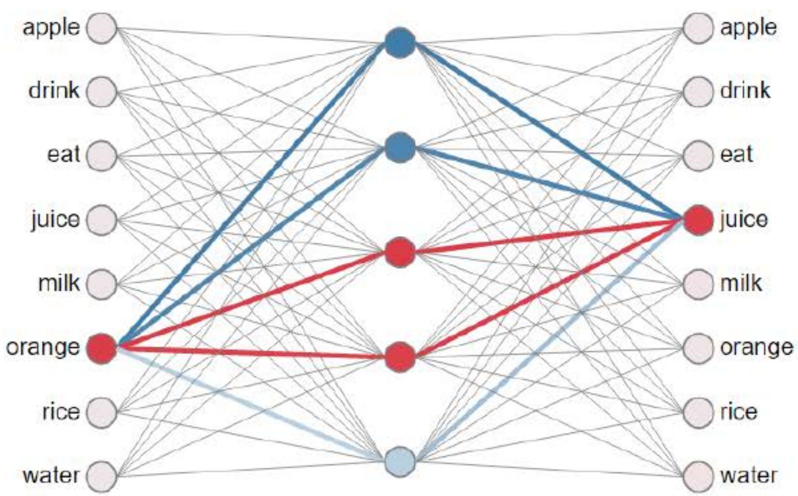
- https://ronxin.github.io/wevi/

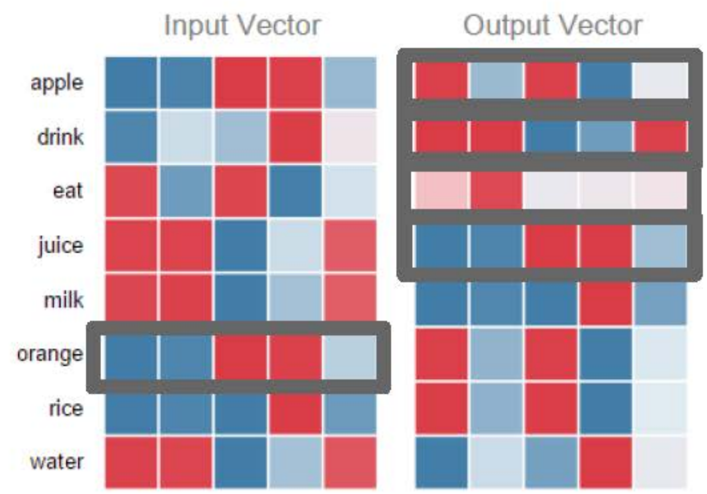


Weights:
Target Embedding

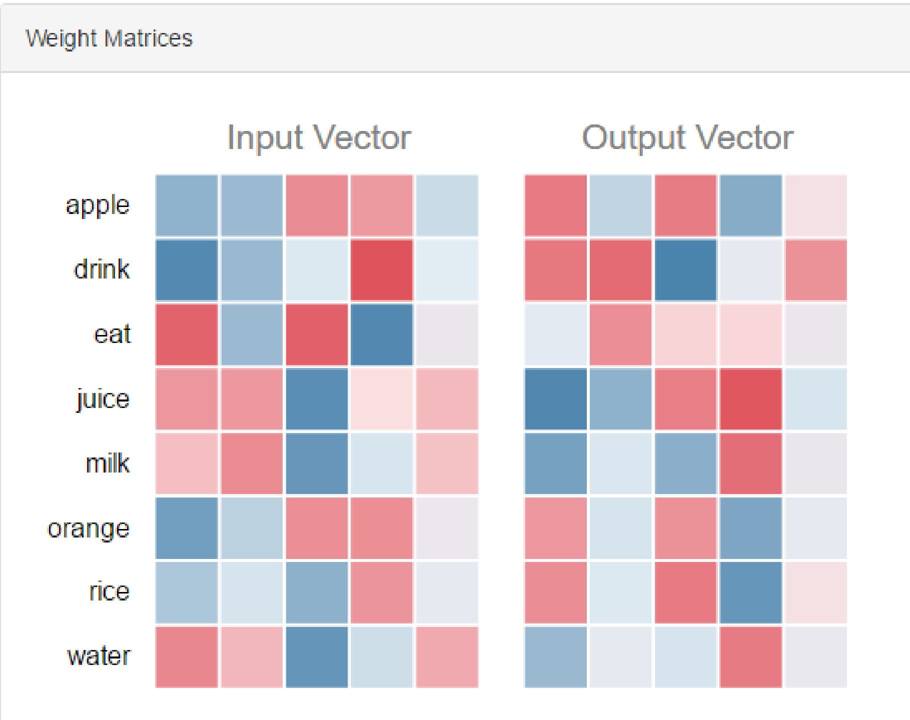
Weights:
Context Embedding

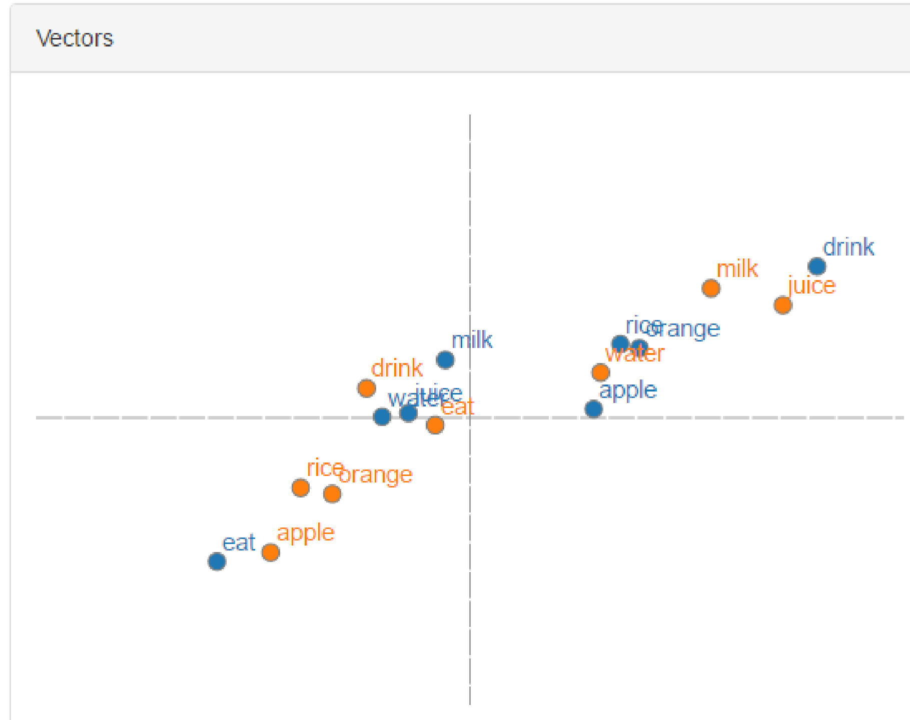
Embedding vs. NN Weights





Embedding Visualization





Negative Sampling for Skip-Gram

- The original objective is not scalable for large size vocabulary!

$$p(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w=1}^{W} \exp\left(u_w^T v_c\right)}$$

- For each target, for every positive word, sample k negative words

$$log\sigma(u_{w_o}^{T} v_{w_c}) + \sum_{i=1}^{k} E_{w_i \sim \underline{P_n(w)}}[log\sigma(-u_{w_i}^{T} v_{w_c})]$$

$P_n(w)$: **"Negative" Distribution**

More on Negative Samples

Positive | **Negative, e.g., k=3**

Source Text

Training Samples

The quick brown fox jumps over the lazy dog. ⟹ (the, quick) (the, brown)

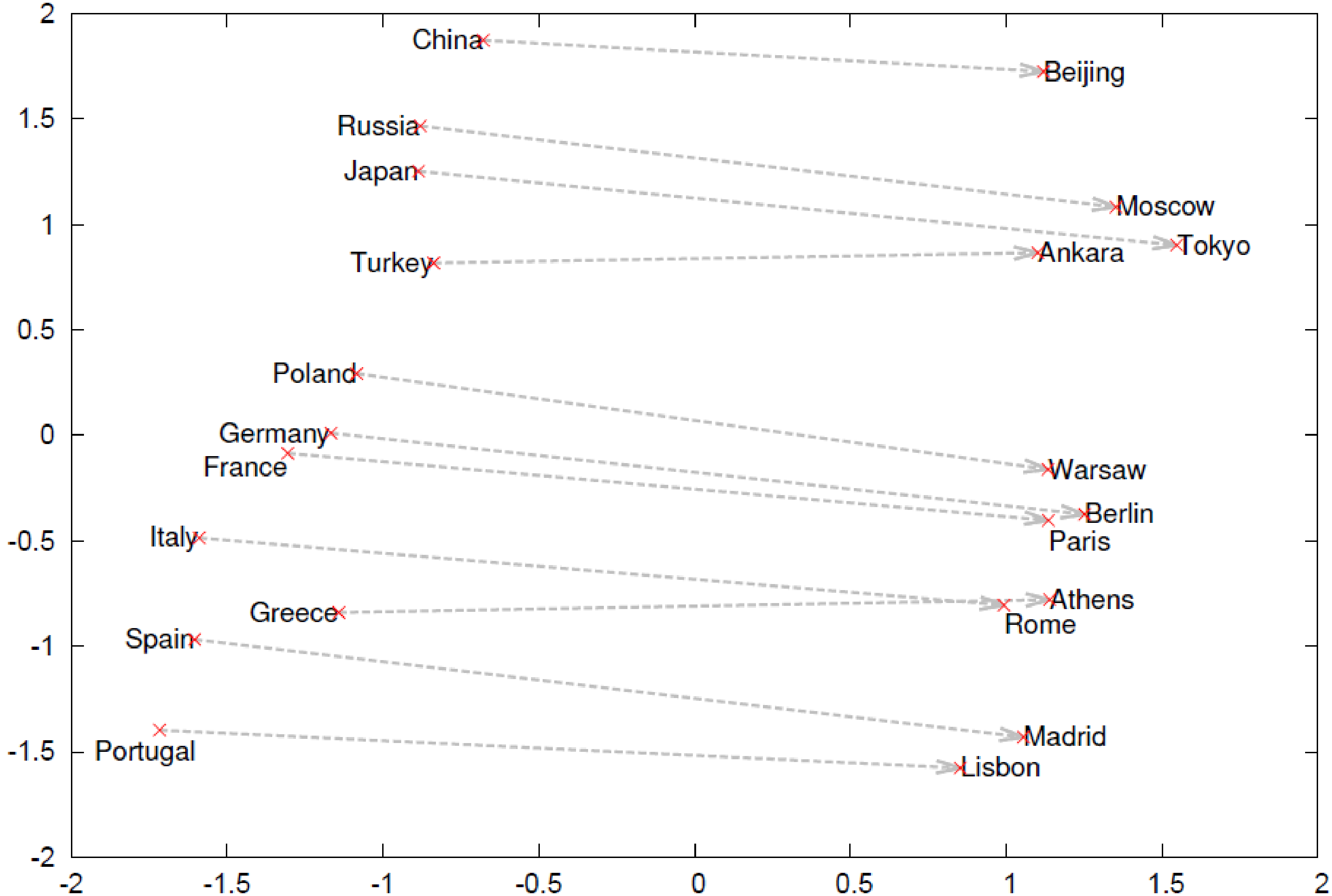
The quick brown fox jumps over the lazy dog. ⟹ (quick, the) (quick, brown) (quick, fox) → (quick, dog) (quick, sky) (quick, flower)

The quick brown fox jumps over the lazy dog. ⟹ (brown, the) (brown, quick) (brown, fox) (brown, jumps)

The quick brown fox jumps over the lazy dog. ⟹ (fox, quick) (fox, brown) (fox, jumps) (fox, over)

A Potential Application

- Relation detection



Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary

Combining Two Worlds

- Matrix factorization for global word-word co-occurrence matrix
  - E.g., SVD
  - Global matrix factorization
- Make predictions within local context windows
  - E.g., word2vec
  - Local context window

Objective Function

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

X_{ij}: *number of times word j appears in the contex of word i*

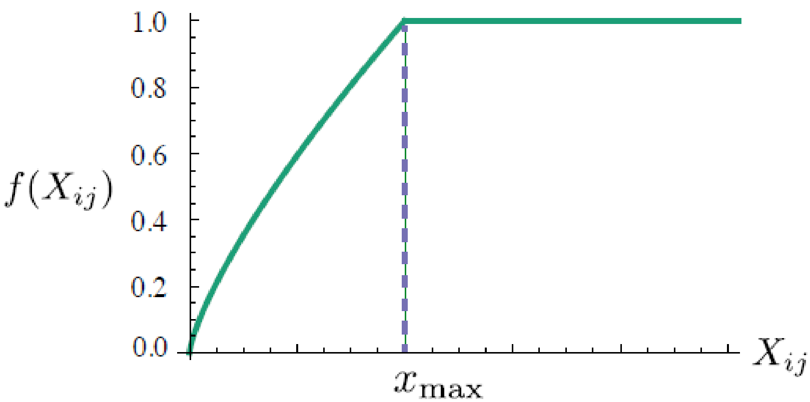
w_i: *word vector for word i*

$\tilde{w}_j$: *context word vector for word j*

b_i: *bias term for word i*

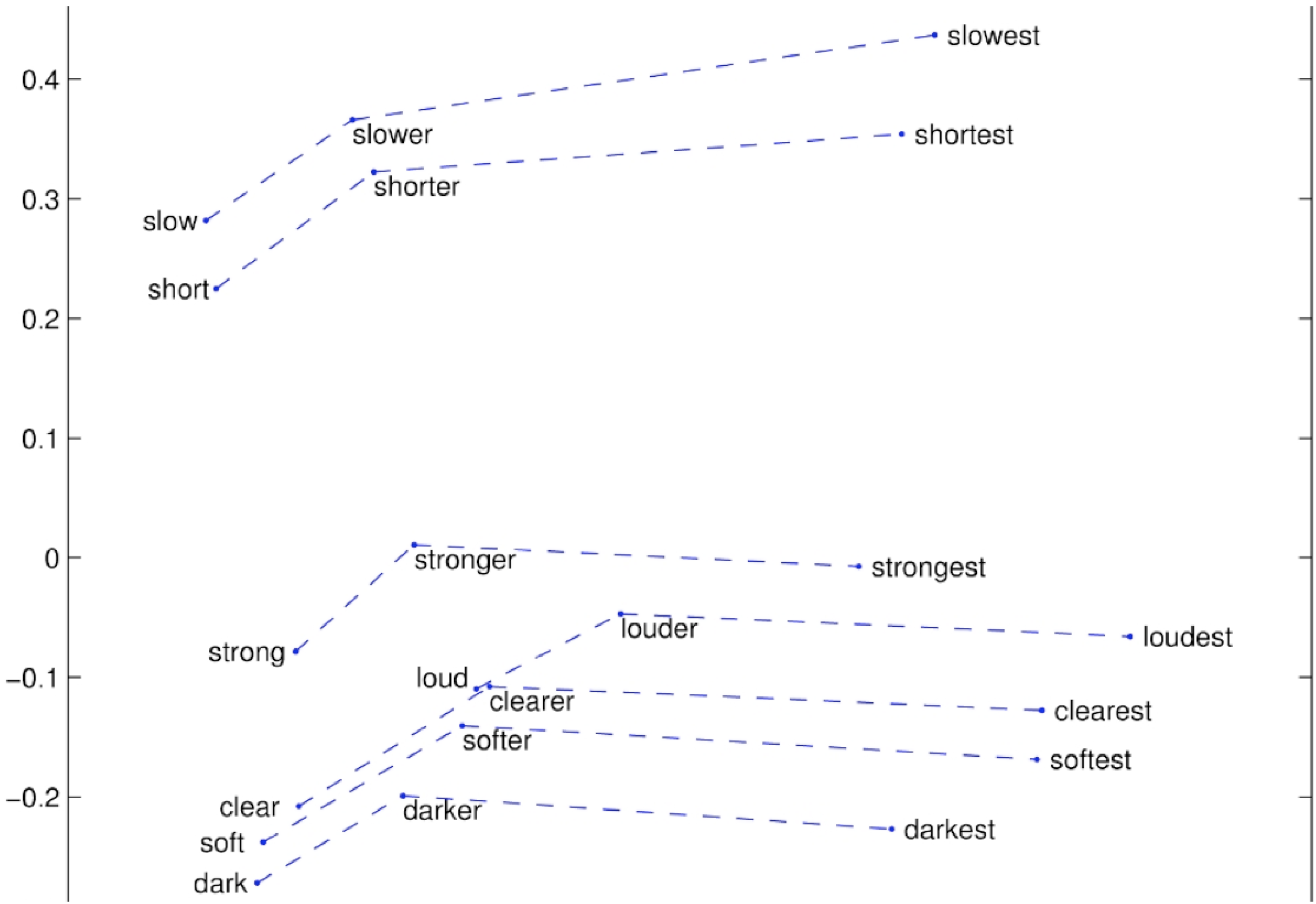
$\tilde{b}_j$: *bias term for context word j*

$f(X_{ij})$: *a weighting function to punish high frequencies*



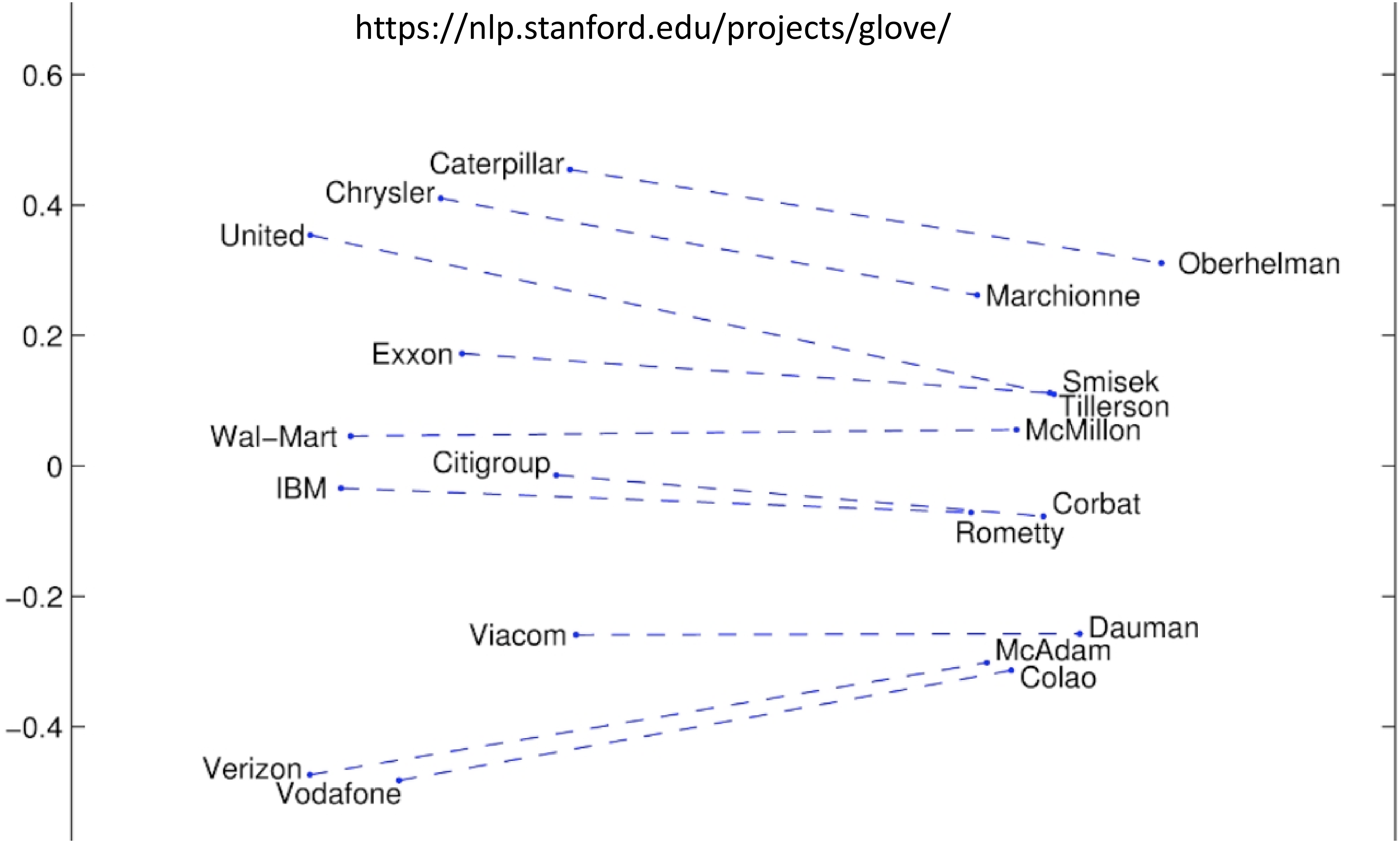
$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}.$$

Some Interesting Results: Superlatives



Some Interesting Results: Company-CEO

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



Text Data: Word Embedding

- Introduction to Word Representation
- Word2vec: CBOW and Skip-Gram
- GloVe: Global Vectors for Word Representation
- Summary

Summary

- Word embedding
  - A low-dimensional vector representation for words
- Word2vec
  - Local context-based prediction: CBOW and Skip-Gram
- Glove
  - Matrix decomposition on local context co-occurrence matrix

References

- Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 1–12.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS, 1–9.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532–1543.