GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
*Stanford
EMNLP 2014

Presenter: Derrick Blakely

Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe? How does it work?
4. Results
5. Conclusion and Take-Aways

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Word Embedding Algos

1. Matrix factorization methods
    - LSA, HAL, PPMI, HPCA

Word Embedding Algos

1. Matrix factorization methods
    - LSA, HAL, PPMI, HPCA

2. Local context window methods

    - Bengio 2003, C&W 2008/2011, skip-gram & CBOW (aka word2vec)

Matrix Factorization Methods

- Co-occurrence counts ≈ “latent semantics”

Matrix Factorization Methods

- Co-occurrence counts $\approx$ “latent semantics”
- Latent semantic analysis (LSA):
    1. SVD factorization: $C = U\Sigma V^T$
    2. Low-rank approximation: $C_k = U\Sigma_k V^T$

Matrix Factorization Methods

- Co-occurrence counts $\approx$ “latent semantics”
- Latent semantic analysis (LSA):
  1. SVD factorization: $C = U\Sigma V^T$
  2. Low-rank approximation: $C_k = U\Sigma_k V^T$
- Good approximation: the largest k eigenvalues matter a lot more than the smaller ones

Matrix Factorization Methods

- Co-occurrence counts ≈ “latent semantics”
- Latent semantic analysis (LSA):
  1. SVD factorization: $C = U\Sigma V^T$
  2. Low-rank approximation: $C_k = U\Sigma_k V^T$
- Good approximation: the largest k eigenvalues matter a lot more than the smaller ones
- Useful for semantics: C_k models co-occurrence counts

LSA

	doc1	doc2	doc3	doc4	doc5	doc6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

LSA

	doc1	doc2	doc3	doc4	doc5	doc6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

$$\langle doc1, doc2 \rangle = 1*0 + 0*1 + 1*1 + 1*0 + 0*0 = \underline{1}$$

LSA

doc1	doc2	doc3	doc4	doc5	doc6
-1.62	-0.6	-0.44	-0.97	-0.7	-0.26
-0.46	-0.84	-0.30	1	0.35	0.65

LSA

doc1	doc2	doc3	doc4	doc5	doc6
-1.62	-0.6	-0.44	-0.97	-0.7	-0.26
-0.46	-0.84	-0.30	1	0.35	0.65

$\langle doc1, doc2\rangle = (-1.62)(-0.6) + (-0.46)(-0.84) = \underline{1.36}$

Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA

Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA
- Takes advantage of global corpus stats

Matrix Factorization Methods

- Term-term matrix methods: HAL, COALS, PPMI, HPCA
- Takes advantage of global corpus stats
- Not the best approach for word embeddings (but often a reasonable baseline)

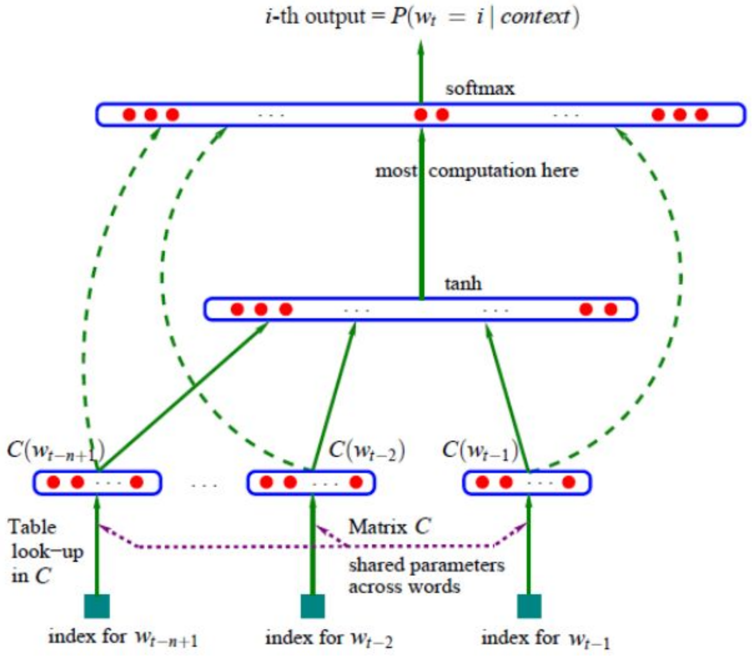
Local Context Window Methods

$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, ..., w_{t-n+1}]$$

Local Context Window Methods

$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}]$$

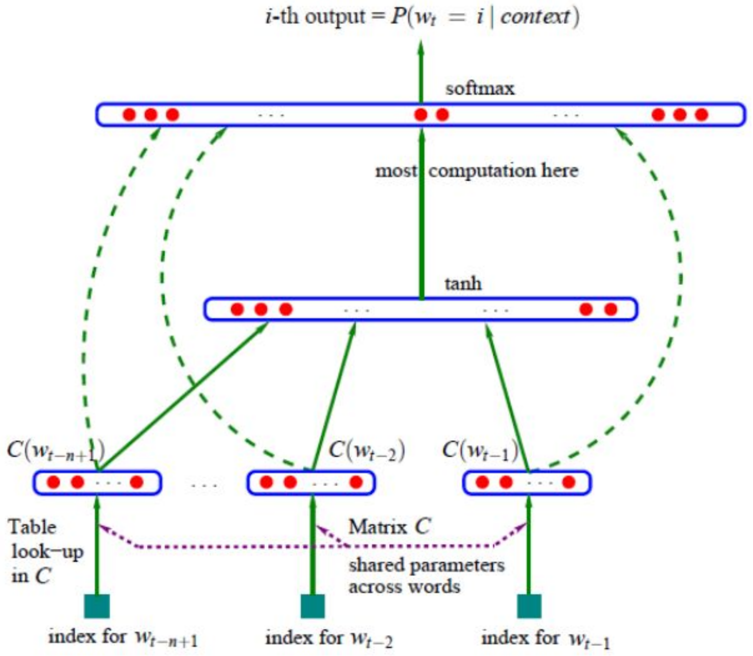
- Bengio, 2003 - neural language model



Local Context Window Methods

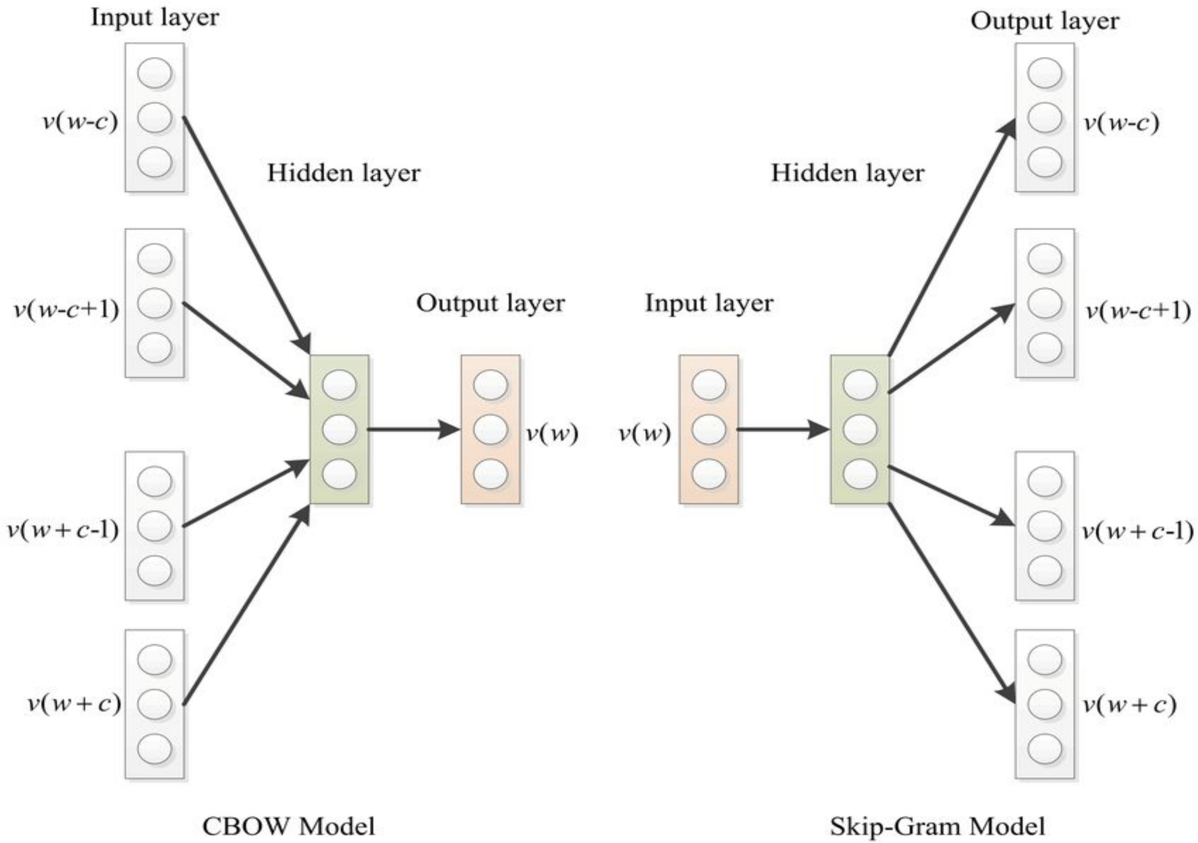
$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}]$$

- Bengio, 2003 - neural language model
- Learning word representations stored lookup table/matrix or network weights



Word2Vec (Mikolov, 2013)

- CBOW
- Skip-gram



Local Context Window Methods

- Tailored to the task of learning useful embeddings

Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)

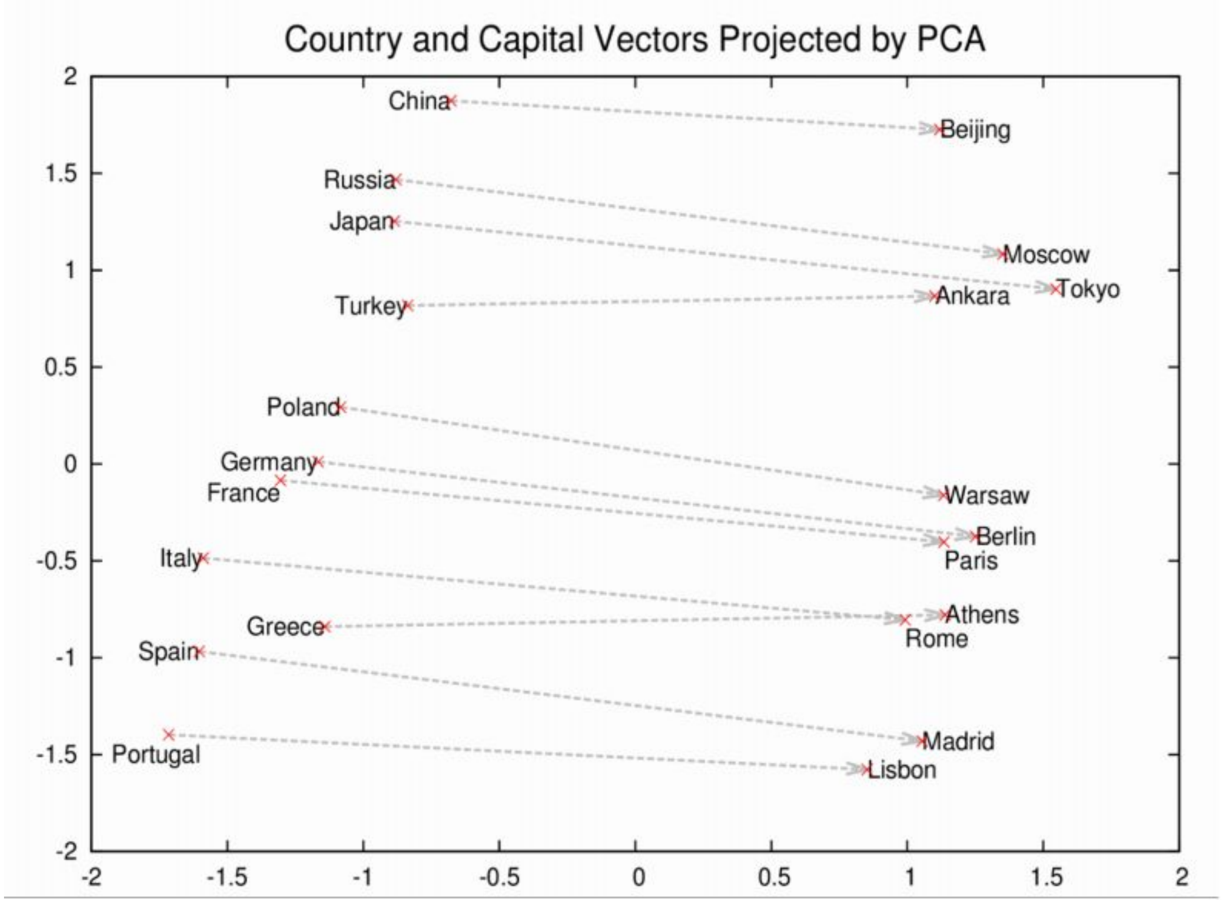
Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)
- Don’t utilize global corpus statistics

Local Context Window Methods

- Tailored to the task of learning useful embeddings
- Explicitly penalize models that poorly predict contexts given words (or words given contexts)
- Don’t utilize global corpus statistics
- Intuitively, a more globally-aware model should be able to do better

Good Embedding Spaces have Linear Substructure



Linear Substructure

- Want to “capture the meaningful linear substructures” prevalent in the embedding space

Linear Substructure

- Want to "capture the meaningful linear substructures" prevalent in the embedding space
- Analogy tasks reflect linear relationships between words in the embedding space.

Linear Substructure

- Want to “capture the meaningful linear substructures” prevalent in the embedding space
- Analogy tasks reflect linear relationships between words in the embedding space.

Paris - France + Germany = Berlin

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec

Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
- Take advantage of global stats

Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
- Take advantage of global stats
- Analogies need linear substructure

Motivation

- Learn embeddings useful for downstream tasks and outperform word2vec
- Take advantage of global stats
- Analogies need linear substructure
- Embedding algos should exploit this substructure

Observation 1: Linear Substructure

- Analogy property is linear

Observation 1: Linear Substructure

- Analogy property is linear
- Vector differences seem to encode concepts

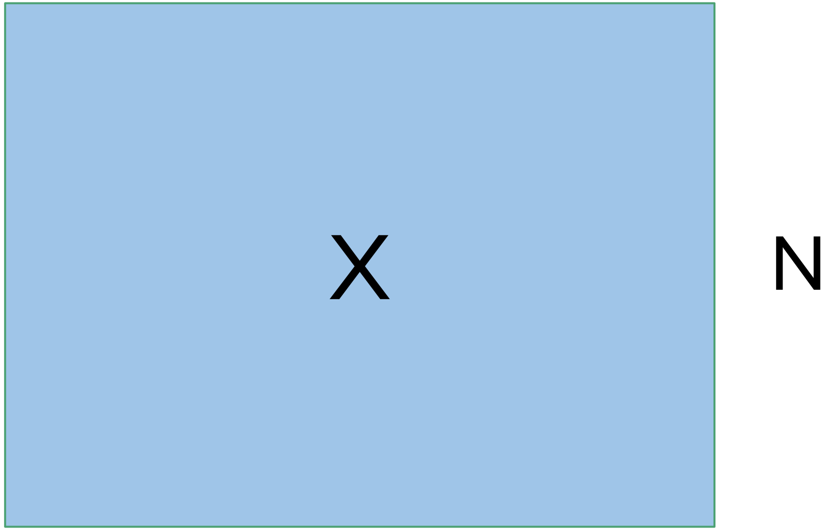
Observation 1: Linear Substructure

- Analogy property is linear
- Vector differences seem to encode concepts
- Man – woman should encode concept of gender

Observation 1: Linear Substructure

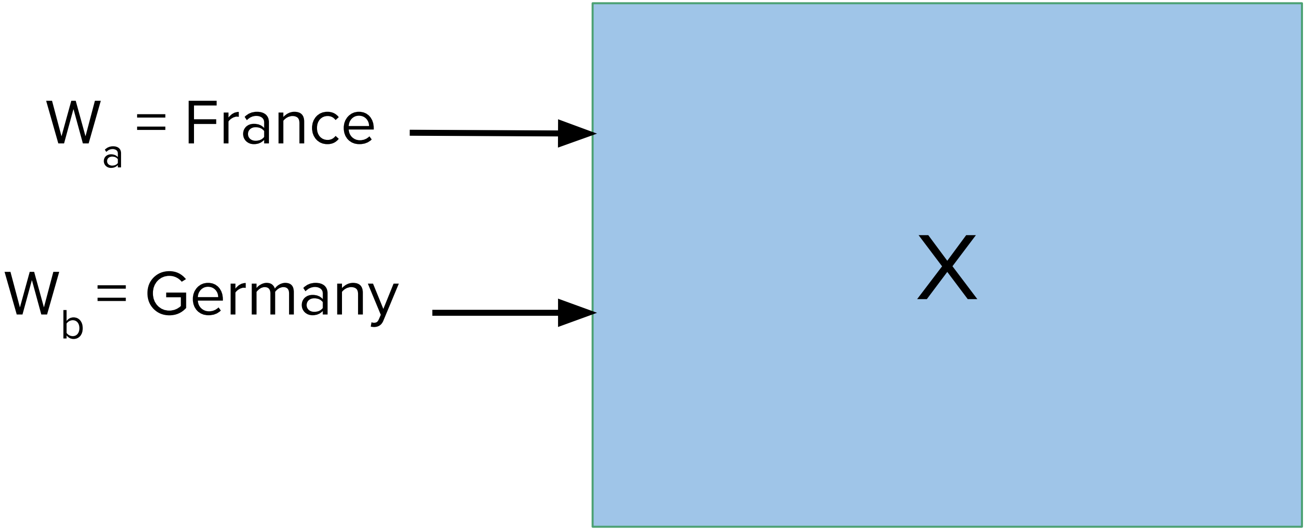
- Analogy property is linear
- Vector differences seem to encode concepts
- Man – woman should encode concept of gender
- France – Germany should encode them being different countries

Co-occurrence Matrix

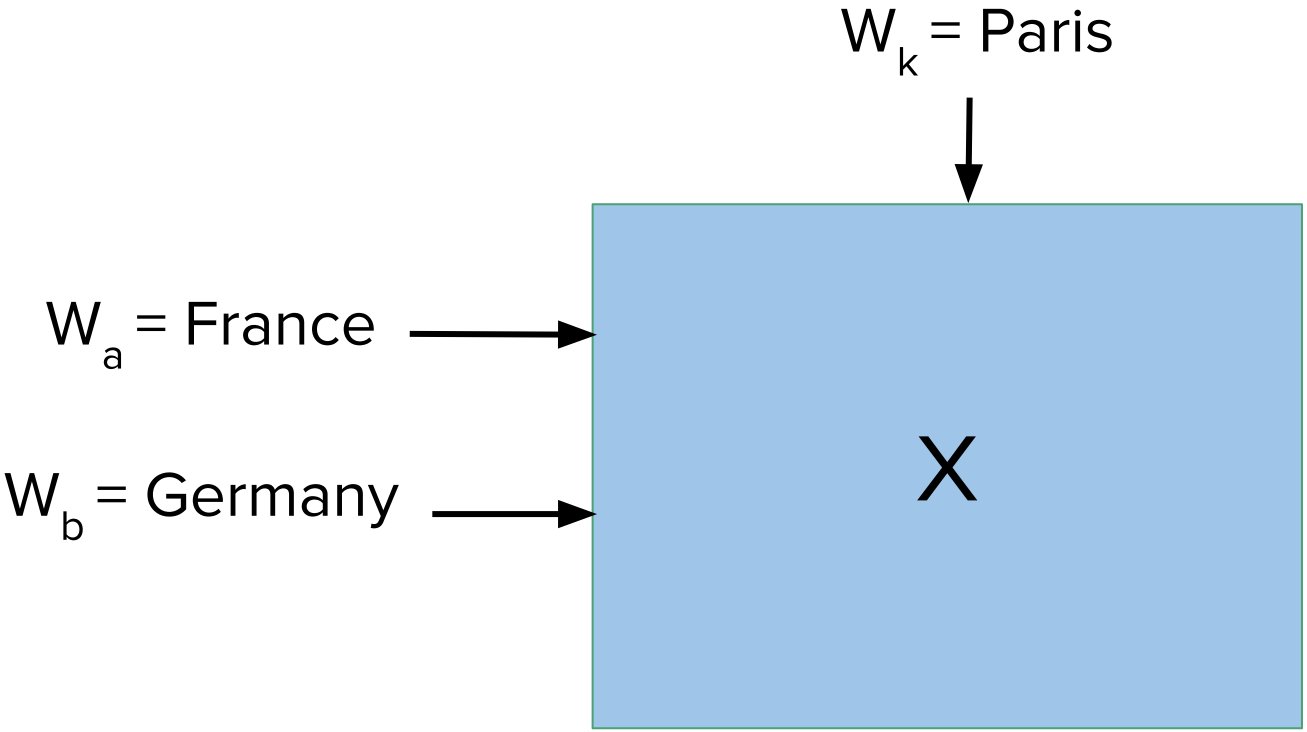


$$N = |V|$$

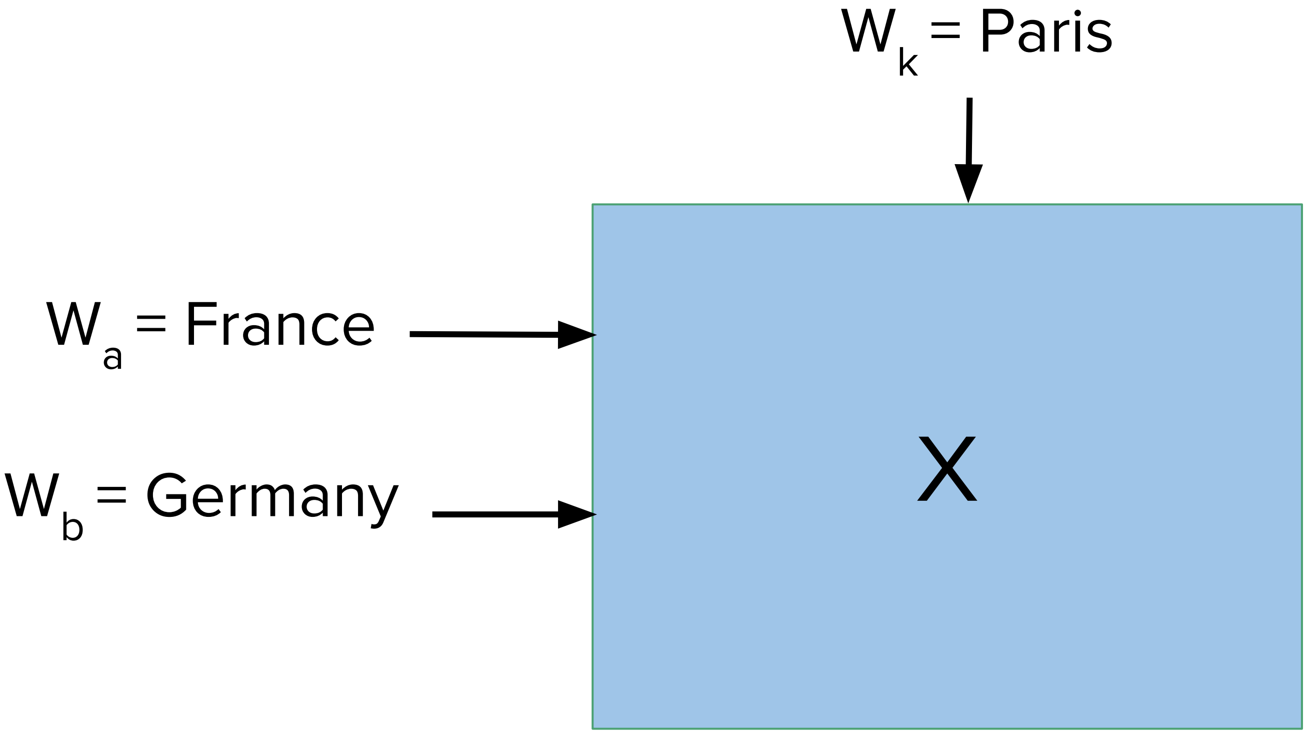
Co-occurrence Matrix



Co-occurrence Matrix



Co-occurrence Matrix



X_{ak} = Num times *k* appear in contexts with *a*

X_a = Num contexts with *a*

Observation 2: Co-Occurrence Ratios Matter

Observation 2: Co-Occurrence Ratios Matter

	k = paris
Pr[k | france]	large
Pr[k | germany]	small
Pr[k | france]/ Pr[k | germany]	large

Observation 2: Co-Occurrence Ratios Matter

	k = paris	**k = berlin**
Pr[k | france]	large	small
Pr[k | germany]	small	large
Pr[k | france]/ Pr[k | germany]	large	small

Observation 2: Co-Occurrence Ratios Matter

	k = paris	k = berlin	k = europe
Pr[k | france]	large	small	large
Pr[k | germany]	small	large	large
Pr[k | france]/ Pr[k | germany]	large	small	≈ 1

Observation 2: Co-Occurrence Ratios Matter

	k = paris	k = berlin	k = europe	k = ostrich
Pr[k ǀ france]	large	small	large	small
Pr[k ǀ germany]	small	large	large	small
Pr[k ǀ france]/ Pr[k ǀ germany]	large	small	≈ 1	≈ 1

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

GloVe

- Model using a loss function that leverages:
    1. Global co-occurrence counts and their ratios
    2. Linear substructure for analogies

GloVe

- Model using a loss function that leverages:
  1. Global co-occurrence counts and their ratios
  2. Linear substructure for analogies
- Software package to build embedding models

GloVe

- Model using a loss function that leverages:
  1. Global co-occurrence counts and their ratios
  2. Linear substructure for analogies
- Software package to build embedding models
- Downloadable pre-trained word vectors created using a massive corpus

Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

$$F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}/X_a}{X_{bk}/X_b}$$

Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

$$F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}/X_a}{X_{bk}/X_b}$$

- Suppose F(france, germany, paris) = *small*

Deriving the GloVe model

- Co-occurrence values in matrix X should be the starting point

$$F(w_a, w_b, w_k) = \frac{P[k|a]}{P[k|b]} = \frac{X_{ak}/X_a}{X_{bk}/X_b}$$

- Suppose F(france, germany, paris) = *small*
- ➜ update the vectors

Factoring in Vector Differences

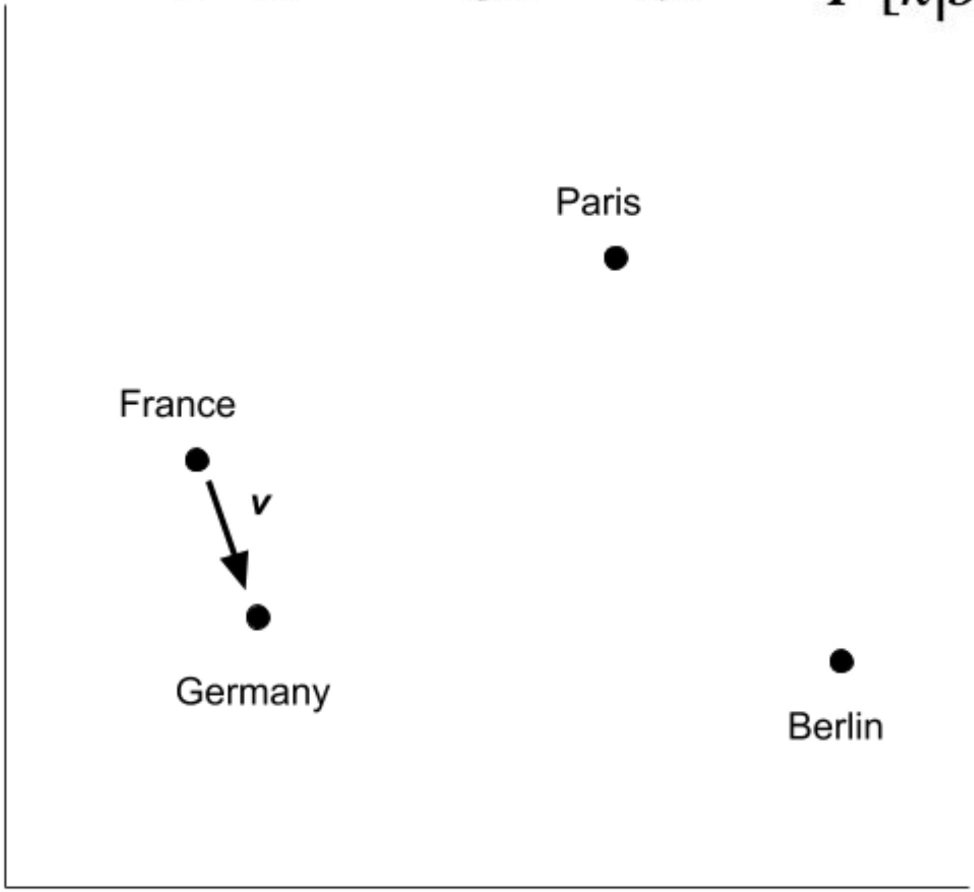
- The difference between the France and Germany vectors is what matters with w.r.t analogies

Factoring in Vector Differences

- The difference between the France and Germany vectors is what matters with w.r.t analogies

$$F(w_a - w_b,\ w_k) = \frac{P[k|a]}{P[k|b]}$$

$$F(w_a - w_b, w_k) = \frac{P[k|a]}{P[k|b]}$$

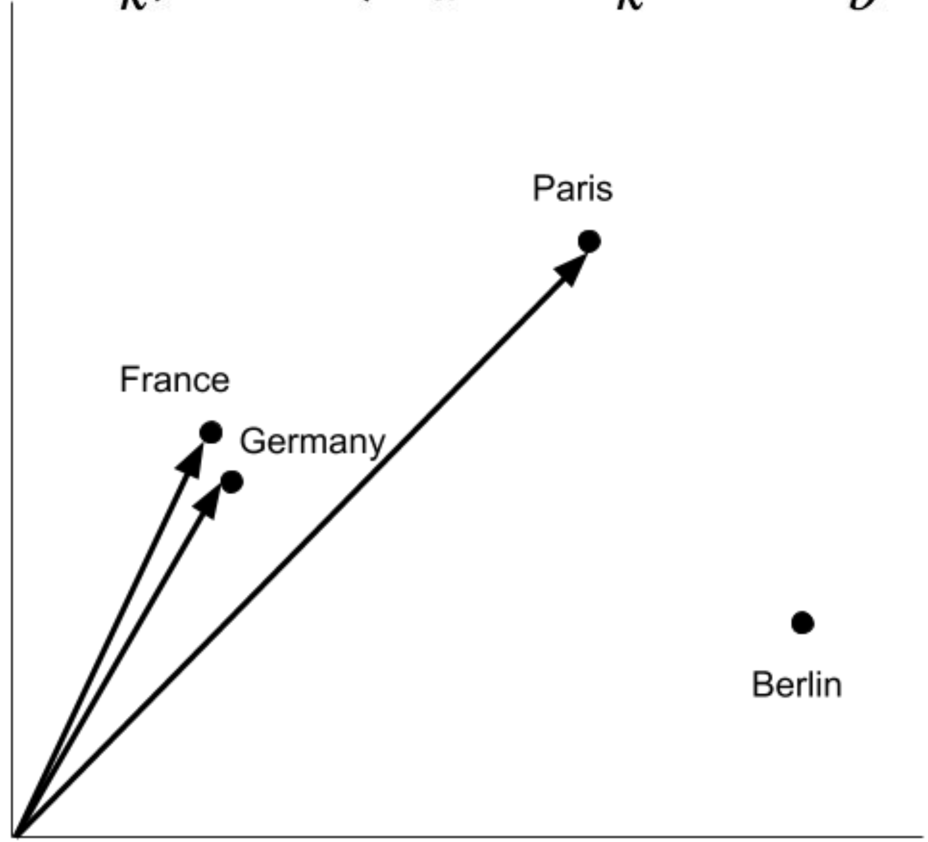


Real-Valued Input is Less Complicated

Real-Valued Input is Less Complicated

$$F((w_a - w_b)^T w_k) = F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]}$$

$$F((w_a - w_b)^T w_k) = F(w_a \cdot w_k - w_b \cdot w_k) = \frac{P[k|a]}{P[k|b]}$$



Reframing with Softmax

$$F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]} = \frac{exp(w_a \bullet w_k)}{exp(w_b \bullet w_k)}$$

Reframing with Softmax

$$F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]} = \frac{exp(w_a \bullet w_k)}{exp(w_b \bullet w_k)}$$

$$\Rightarrow$$

$$w_a \bullet w_k = log(P[k|a])$$

Reframing with Softmax

$$F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]} = \frac{exp(w_a \bullet w_k)}{exp(w_b \bullet w_k)}$$

$$\Rightarrow$$

$$w_a \bullet w_k = log(P[k|a])$$

$$= log(\frac{X_{ak}}{X_a})$$

Reframing with Softmax

$$F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]} = \frac{exp(w_a \bullet w_k)}{exp(w_b \bullet w_k)}$$

$$\Longrightarrow$$

$$\begin{aligned} w_a \bullet w_k &= log(P[k|a]) \\ &= log(\frac{X_{ak}}{X_a}) \\ &= log(X_{ak}) - log(X_a) \end{aligned}$$

Reframing with Softmax

$$F(w_a \bullet w_k - w_b \bullet w_k) = \frac{P[k|a]}{P[k|b]} = \frac{exp(w_a \bullet w_k)}{exp(w_b \bullet w_k)}$$

$$\Longrightarrow$$

$$w_a \bullet w_k = log(P[k|a])$$

$$= log(\frac{X_{ak}}{X_a})$$

$$= log(X_{ak}) - log(X_a)$$

$$\Longrightarrow \quad w_a \bullet w_k + bias - log(X_{ak}) = 0$$

Least Squares Problem

Least Squares Problem

$$w_a \bullet w_k + bias - log(X_{ak}) = 0$$

Least Squares Problem

$$w_a \bullet w_k + bias \ - log(X_{ak}) = 0$$

$$\Rightarrow$$

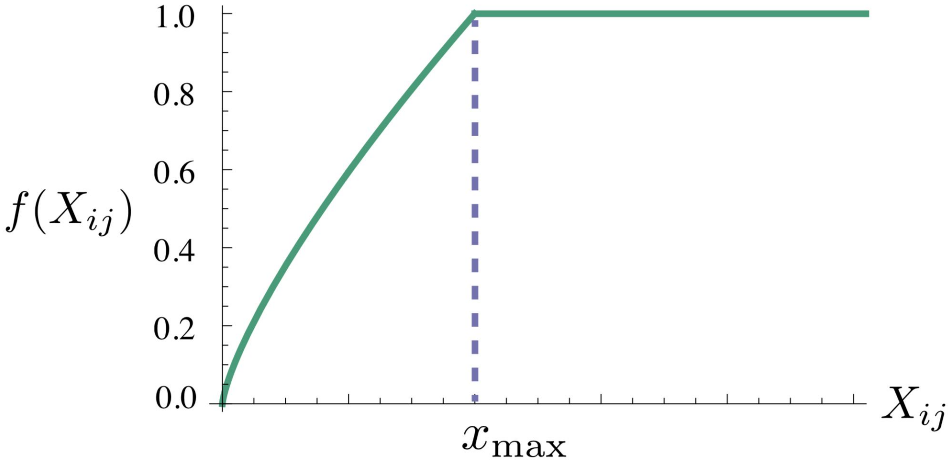
$$(w_i \bullet w_j + biases \ - log(X_{ij}))^2$$

Final GloVe Model

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

Final GloVe Model

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



Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

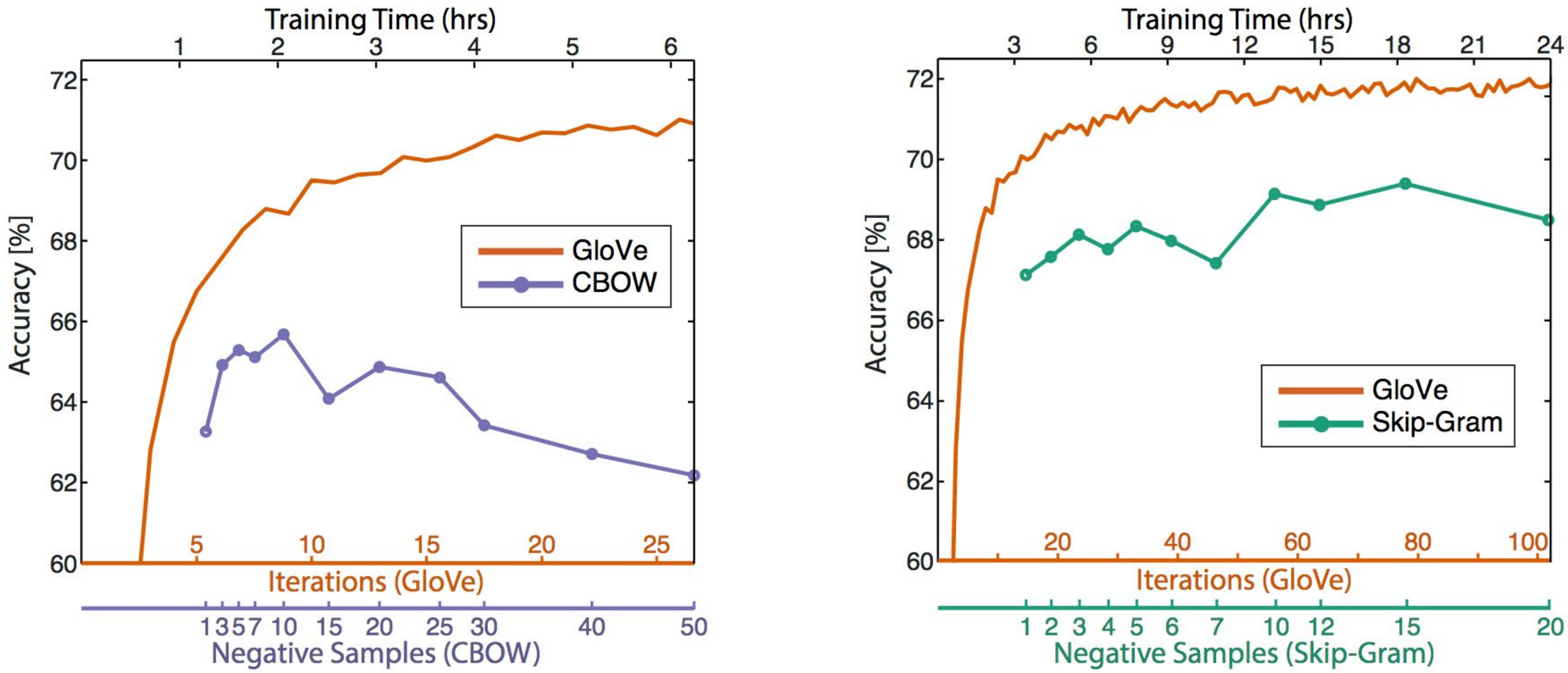
Performance on Analogy Tasks

- Semantic: “Paris is France as Berlin is to _____”
- Syntactic: “Fly is to flying as dance is to _____”

Performance on Analogy Tasks

Model	Dimensions	Corpus Size	Semantic	Syntactic	Total
CBOW	1000	6B	57.3	68.9	63.7
Skip-Gram	1000	GB	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**

Speed


Training Time (hrs)
Accuracy [%]
GloVe
CBOW
Iterations (GloVe)
Negative Samples (CBOW)
Training Time (hrs)
Accuracy [%]
GloVe
Skip-Gram
Iterations (GloVe)
Negative Samples (Skip-Gram)

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online
- Doesn’t allow different entities to be embedded (a la StarSpace)

Limitations

- Superior for analogy tasks; almost the same performance as word2vec and fastText for retrieval tasks
- Co-occurrence construction takes 1 hr +
- Each training iterations takes 10 minutes +
- Not online
- Doesn’t allow different entities to be embedded (a la StarSpace)

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Roadmap

1. Background
2. Motivation of GloVe
3. What is GloVe?
4. Results
5. Conclusion and Take-Aways

Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats

Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost

Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms

Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms
- Simpler models can work well (SVD-L performed very well)

Conclusion and Take-Aways

- Embeddings algos all take advantage of co-occurrence stats
- Leveraging global stats can provide performance boost
- Keep the linear substructure in mind when designing embedding algorithms
- Simpler models can work well (SVD-L performed very well)
- More iterations seems to be most important for embedding models:
- faster iterations ➔ train on a larger corpus ➔ create better embeddings
- Demonstrated by word2vec, GloVe, and SVD-L

Questions?