Dense Vector Semantics:
More on Low Dimensional Vectors

CSE 597

Outline

- Motivation for dense vector semantics
- Word2Vec (Skip-gram)
- GloVe
- Visualization
- Evaluation

Sparse (High D) versus Dense (Low D) Vectors

- High D (observed weights)
  - Size: $10^3 \leq D \leq 10^4$
  - Constituency: many zero entry cells
- Low D (latent weights)
  - Size: $D = n \times 10^2$, $n \in [1,5]$
  - Constituency: few zero entry cells
- Advantages of a dense representation
  - Fewer weights to learn = reduced training time
  - Better generalization
    - High D: Each vector position represents 1 context
    - Low D: Each vector position represents **generalization** over multiple contexts

Dense Vector Methods

- Shorter, dense vectors perform better than long, sparse vectors
    - More efficient representation for computation, e.g., number of weights
    - Probably stronger generalization, e.g., *car* vs. *automobile*
- Matrix factorization: reduces noise, data sparsity
    - Factor a high-dimensional matrix into a product of matrices
    - Example: SVD
- Neural methods: Directly learn LowD representation
    - Mixed neural log-linear model: Word2Vec and other static methods
    - Many recent advances (e.g., BER and other, contextualized methods)

Language Modeling versus Embeddings

- Neural language model (LM) learns to predict word **sequences**
  - Given *n* words predicts word w_{n+1}
  - Learns distributions over entire vocabulary $|\mathbf{V}|$
  - Learns a hidden layer representation
- Language Modeling
  - Meeting 4: Statistical LM
  - Meeting 9: Feed Forward Neural LM
  - Meeting 13: Recurrent Neural Networks

Idea behind Word2Vec

- Word2Vec learns which context words c show up **near** target t
  - Positive examples: target word and a neighboring context word
  - Negative samples: randomly sample from non-neighboring words
- Each learned embedding is a logistic regression **classifier**
  - Predictors of the target word t are its context words $c_{i:n}$
  - The regression weights on $c_{i:n}$ serve as the embedding for t
- Self Supervision: No need to collect labelled data
  - Supervision signal: the observed words c **near** target t
  - (Language modeling uses the same idea)

Word2Vec: Binary Classification Task

- P(+|t,c) for target words **t** and context words **c** represented as vectors
- Probability P(+|t,c) that **c** is a context word for **t** is derived from similarity of their vector representations: Similarity(t,c) ≈ **t** · **c**
- Use sigmoid: logistic regression

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad P(+|t,c) = \frac{1}{1+e^{-t\cdot c}}$$

- Ensure that P(+|t,c) and P(-|t,c) sum to 1:

$$\begin{aligned} P(-|t,c) &= 1 - P(+|t,c) \\ &= \frac{e^{-t\cdot c}}{1+e^{-t\cdot c}} \end{aligned}$$

Assume Independence of Context Words

- Then: probabilities of the $\mathbf{c_{1:k}}$ context words multiply

$$P(+|t, c_{1:k}) = \prod_{i=1}^{k} \frac{1}{1+e^{-t \cdot c_i}}$$

$$\log P(+|t, c_{1:k}) = \sum_{i=1}^{k} \log \frac{1}{1+e^{-t \cdot c_i}}$$

Positive versus Negative Context Words

- Make ***t*** more like the positive context words c1:c4

a tablespoon of **apricot** preserves or sweet jam
c1 t c2 c3 c4

- Make ***t*** less like the negative context words c5:c8

energy remark deficiency silver **apricot** . . .
c5 c6 c7 c8 t

Negative Sampling

- Create training instances (t, c_{pos})
- For each positive training instance, create k negative samples (t, c_{neg})
  - Each c_{neg} is a random word not similar to (predictive of) t
  - k is usually > 1
- Using all the words in the vocabulary is inefficient
- Negative words are selected by a user-selected weight α on the unigram probability of the word
  - Sample less from words not in the context
  - Gives less weight to distant words

Positive and Negative Sampling

- Aim for high weights to predict the positive class:

$$log\,\sigma(c1 \cdot t) + log\,\sigma(c2 \cdot t) + log\,\sigma(c3 \cdot t) + log\,\sigma(c4 \cdot t)$$

- Aim for low weights to predict the negative class:

$$log\,\sigma(c5 \cdot t) + log\,\sigma(c6 \cdot t) + log\,\sigma(c7 \cdot t) + log\,\sigma(c8 \cdot t)$$

Training Objective: Skipgram with Negative Sampling

- Training objective: maximize the following

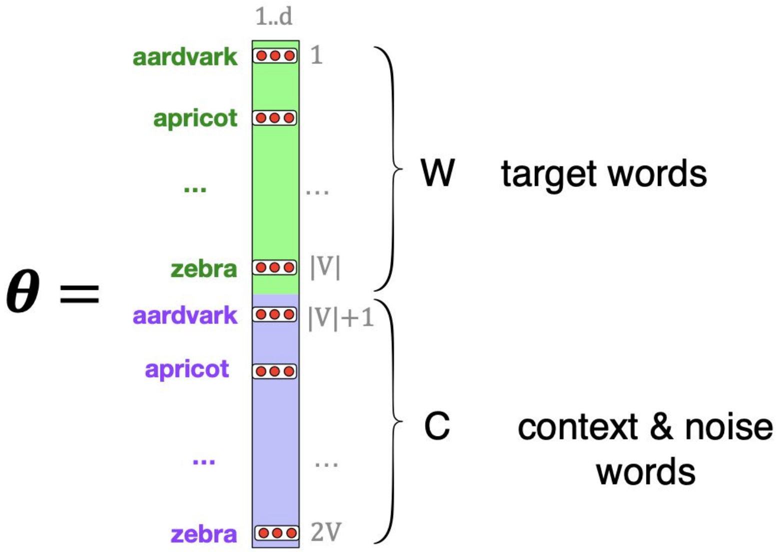
$$L_{CE} = -\left[\log \sigma(c_{pos} \cdot w) + \sum_{i=1}^{k} \log \sigma(-c_{neg_i} \cdot w)\right]$$

- Maximize the dot product of the word (***w***) with the context words (***c***)
- Minimize the dot products of the word with the ***k*** negative sampled non-neighbor words
- Sample w_i from V by $p(w)$

Skipgram Learns Two Embeddings

- θ is a matrix of vectors length 2|V| for the embeddings W and C
- Final embedding for a word $\boldsymbol{i}$ is sum of vectors $\boldsymbol{w}_{\boldsymbol{i}} + \boldsymbol{c}_{\boldsymbol{i}}$, or just $\boldsymbol{w}_{\boldsymbol{i}}$

Jurafsky &
Martin, Fig 6.13

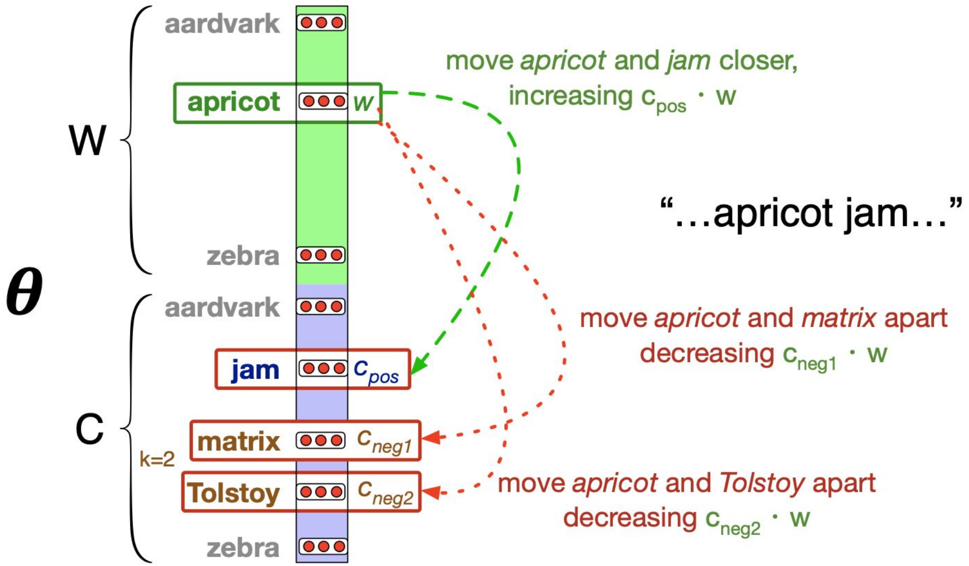


Word2Vec

Training Skipgram: Stochastic Gradient Descent

- Initialize weights in matrices W and C, then apply SGD

Jurafsky & Martin, Fig 6.14



Stochastic Gradient Descent for Skipgram

- Derivative of the loss function

$$\frac{\partial L_{CE}}{\partial w} = [\sigma(c_{pos} \cdot w) - 1]c_{pos} + \sum_{i=1}^{k}[\sigma(c_{\text{neg}_i} \cdot w)]c_{neg_i}$$

- Weight update rule

$$w^{t+1} = w^t - \eta[\sigma(c_{pos} \cdot w^t) - 1]c_{pos} + \sum_{i=1}^{k}[\sigma(c_{\text{neg}_i} \cdot w^t)]c_{neg_i}$$

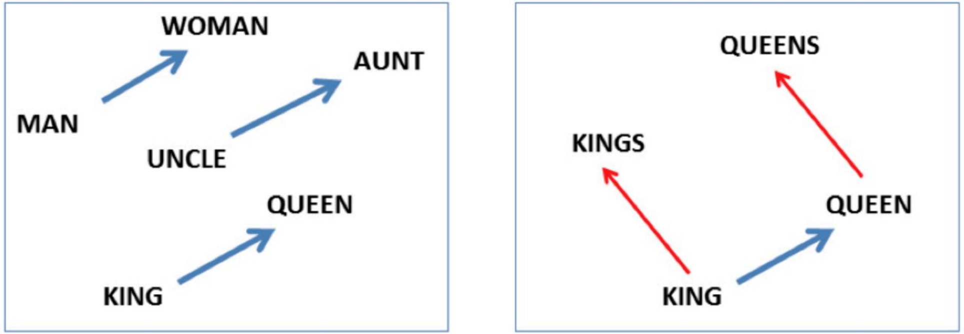
Example Word Embedding Similarities

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	graffiti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

Vector Math and Word Embeddings

Considerations

- How well do 2D pictures represent ND Vectors (100 < N < 1000)?
- What % of V can be structured into analogical relations?
- What if vector methods differ? (Word2Vec = local; GloVE = global)



$$\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$$

GloVe

GloVe: Global Vectors for Word Representation

- Based on addressing limitations of SVD and Word2Vec
- SVD: global context matrix factorization
  - Global co-occurrence statistics distinguish large-scale differences, e.g., stop words
  - Performs poorly on word analogy tasks
- Word2Vec: word classification by local contexts
  - Captures analogical meaning well
  - Ignores global statistics

GloVe Intuition

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Given a pair of words *i, j* and any other word *k*

- (1) > 1 if k is more likely given i than given j
- (1) < 1 if k is more likely given j than given i
- (1) ≈ 1 if k is equally likely in either context

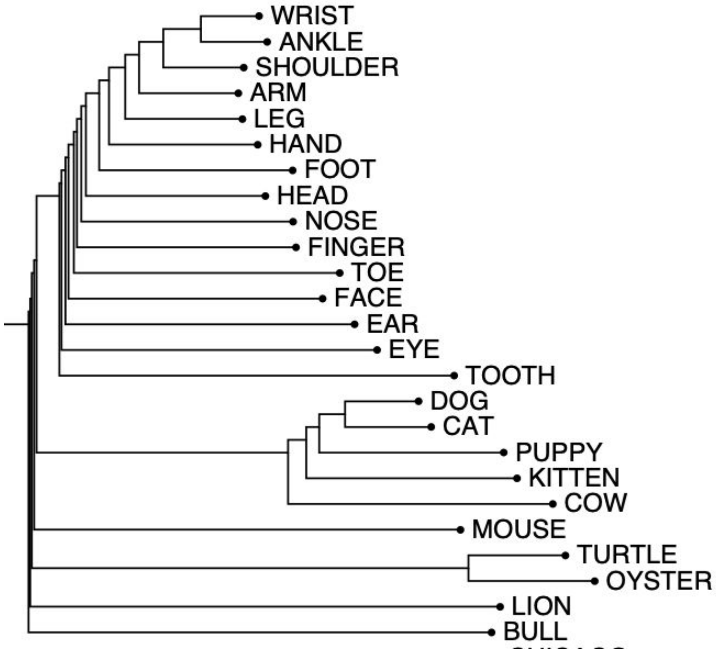
$$\frac{P(k|i)}{P(k|j)} \quad (1)$$

GloVe

- Global log bi-linear regression model
- Source code and trained vectors are available
- Introduces a weighted least squares training objective for the word-word co-occurrence matrix $\boldsymbol{X}$

Visualizing N-dimensional Embeddings

- Look at words close to a target word t
  - Sort by cosine similarity to t
  - *frogs, toad, litoria, leptodactylidae, lizard*
- Hierarchical clustering

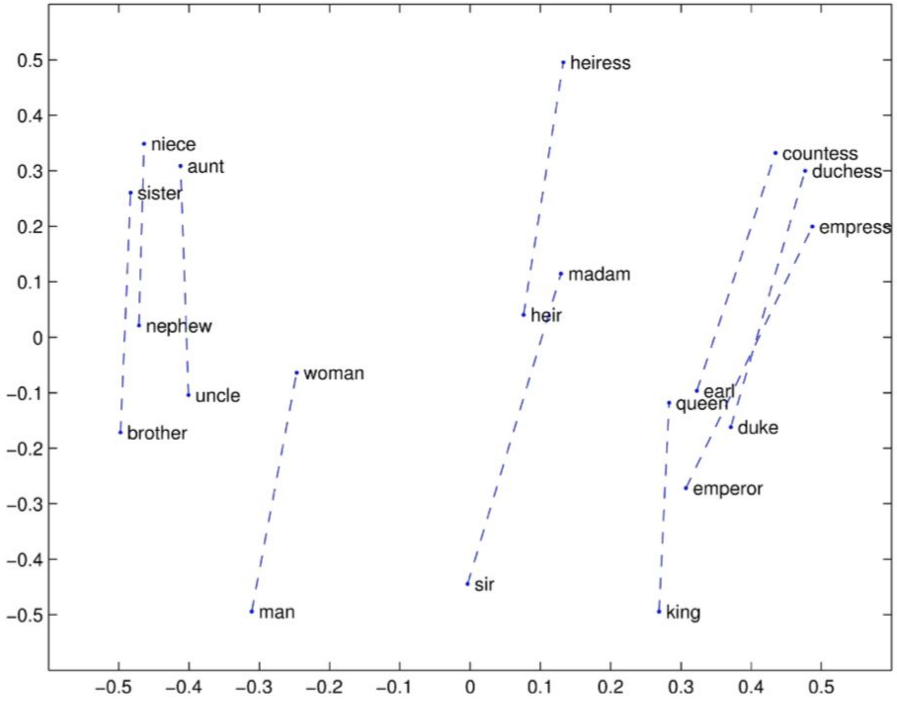


Collapsing to 2D Plots

- Principal Component Analysis (PCA)
- t-SNE (van der Maaten):
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
  - A dimensionality reduction technique for visualization of high-dimensional datasets
  - Can handle non-linear relationships among data clusters

t-SNE Visualization of GloVe Vectors

- Similar to early "parallelogram" proposal for analogical reasoning (Rummelhart & Abramson, 1973)
- Many limitations to casting word meaning in terms of analogical cases
  - Does not work well for infrequent words, more complex relations



Bias in Input Data Preserved in ML Models

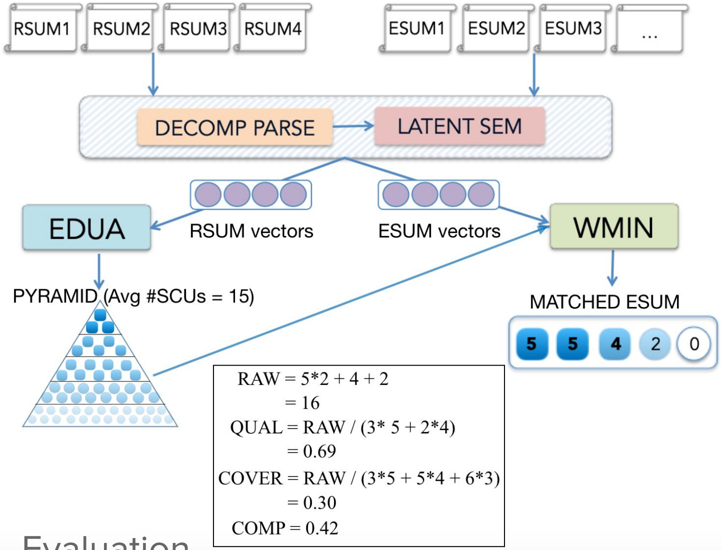
- Bias in Word2Vec embeddings (Bolukbasi et al., 2016)
  - man/computer, woman/X; X predicted to be homemaker
  - father/doctor, mother/Y: Y predicted to be nurse
- Amplification of bias: gendered terms become even more gendered in embedding space
- Association bias:
  - Embeddings for conventionally European-American person names have higher cosine similarities to positive terms
  - Embeddings for African-American person names have higher cosine similarities to negative terms

Intrinsic Evaluation of Word Embeddings

- Word pairs, out of context:
  - WordSim-353 (Finkelstein et al., 2002): ratings from 0 to 10 for 353 noun pairs
  - SimLex-999 (Hill et al., 2015): ratings include verbs and adjectives
- Word pairs, with sentence contexts:
  - Stanford Contextual Word Similarity (SCWS) (Huang et al., 2012)
  - Word-in-Context (WiC) (Pilehvar and Camacho-Collados, 2019)
- Semantic Textual Similarity (STS) datasets (2012-2014)
  - Sentence level similarity
  - Human ratings on 6 pt scale

Extrinsic Evaluation of Word Embeddings

- Performance in the context of an NLP application
  - Rarely done: Most NLP methods initialize with word embeddings, then learn application specific updates
- PSU NLP Lab: PyrEval



Evaluation

PSU Lab’s PyrEval: Intrinsic Evaluation of Emeddings

WTMF Corpus				
	WTMF		GloVe+SIF	
Test Data	Sent	Win	Sent	Win
STS12	**0.7258**	0.6851	0.6859	0.6812
STS13	**0.7405**	0.6901	0.6426	0.6311
STS14	**0.7187**	0.7012	0.6299	0.6149
Gigaword Subset				
	WTMF		GloVe+SIF	
Test Data	Sent	Win	Sent	Win
STS12	0.6400	**0.6482**	0.6256	0.6256
STS13	0.5909	**0.6224**	**0.6214**	**0.6214**
STS14	**0.6835**	**0.6835**	0.6223	0.6223

PSU Lab’s PyrEval: Extrinsic Evaluation of Embeddings

Data	Manual		Rubric	
	WTMF	GloVe	WTMF	GloVe
Orig	0.57	0.59	0.53	0.48
No Outliers	0.62	0.63	0.70	0.52
Rescored	0.65	0.65	0.70	0.52

Summary

- Low dimension word vectors can be efficiently trained (e.g., Word2Vec, GloVe)
- Quality of resulting embeddings depends on how context is handled
  - Local
  - Global
- Applications include
  - Word similarity
  - Lexical relations