



Word Embedding (Continued)

Yu Meng

University of Virginia
yumeng5@virginia.edu

Sep 20, 2024

Reminder

- Project proposal is due today 11:59pm!
- Assignment 2 is due next Wed!
- Rivanna instruction:

https://docs.google.com/document/d/1SK_d6ApRPbMcsvyuyGytxBRvVxWzKbwIVRvyOrFu47k/edit?usp=sharing

Join at

slido.com

#2098 474



Overview of Course Contents

Join at

slido.com

#2098 474



- Week 1: Logistics & Overview
- Week 2: N-gram Language Models
- Week 3: Word Senses, Semantics & Classic Word Representations
- **Week 4: Word Embeddings**
- Week 5: Sequence Modeling and Transformers
- Week 6-7: Language Modeling with Transformers (Pretraining + Fine-tuning)
- Week 8: Large Language Models (LLMs) & In-context Learning
- Week 9-10: Knowledge in LLMs and Retrieval-Augmented Generation (RAG)
- Week 11: LLM Alignment
- Week 12: Language Agents
- Week 13: Recap + Future of NLP
- Week 15 (after Thanksgiving): Project Presentations

(Recap) Dense Vectors Pros & Cons

Join at
slido.com
#2098 474



- **(+) Compactness:** Represent a large number of concepts using fewer resources (richer semantic information per dimension); easier to use as features to neural networks
- **(+) Robustness:** Information is spread across many dimensions => more robust to the randomness/noise in individual units
- **(+) Scalability & Generalization:** Efficiently handle large-scale data and generalize to various applications
- **(-) Lack of Interpretability:** (Unlike sparse vectors) difficult to assign a clear meaning to individual dimensions, making model interpretation challenging



(Recap) Distributional Hypothesis

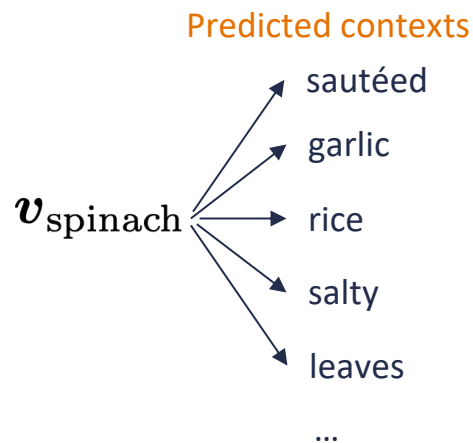
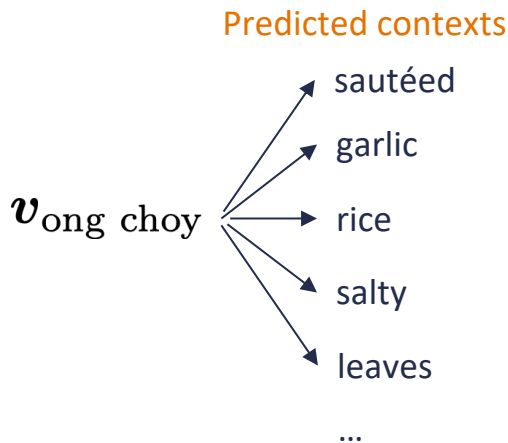
- Words that occur in similar contexts tend to have similar meanings
- A word's meaning is largely defined by the company it keeps (its context)
- Example: suppose we don't know the meaning of "Ong choy" but see the following:
 - Ong choy is delicious **sautéed with garlic**
 - Ong choy is superb **over rice**
 - ... ong choy **leaves** with **salty** sauces
- And we've seen the following contexts:
 - ... spinach **sautéed with garlic over rice**
 - ... chard stems and **leaves** are **delicious**
 - ... collard greens and other **salty** leafy greens
- Ong choy = water spinach!





(Recap) Learning Word Embeddings

- Assume a large text collection (e.g., Wikipedia)
- Hope to learn similar word embeddings for words occurring in similar contexts
- Construct a prediction task: use a center word's embedding to predict its contexts!
- Intuition: If two words have similar embeddings, they will predict similar contexts, thus being semantically similar!





(Recap) Word2Vec Setting

- Input: a corpus D – the larger, the better!
- Training data: word-context pairs (w, c) where w is a center word, and c is a context word
 - Each word in the corpus can act as center word
 - Context words = neighboring words of the center word in a local context window ($\pm l$ words)
- Parameters to learn: $\theta = \{v_w, v_c\}$ – each word has two vectors (center word representation & context word representation)
- The center word representations v_w are usually used as the final word embeddings
- Number of parameters to store: $d \times |V|$
 - d is the embedding dimension; usually 100-300
 - $|V|$ is the vocabulary size; usually $> 10K$
 - Sparse vector representations will have $|V|^2$ parameters!



(Recap) Word2Vec Training Data Example

- Input sentence: “there is a cat on the mat”
- Suppose context window size = 2
- Word-context pairs as training data:
 - (there, is), (there, a)
 - (is, there), (is, a), (is, cat)
 - (a, there), (a, is), (a, cat), (a, on)
 - (cat, is), (cat, a), (cat, on), (cat, the)
 - (on, a), (on, cat), (on, the), (on, mat)
 - (the, cat), (the, on), (the, mat)
 - (mat, on), (mat, the)
- “Skip-gram”: skipping over some context words to predict the others!
- Training data completely derived from the raw corpus (no human labels!)

there is a cat on the mat
there is a cat on the mat
there is a cat on the mat
there is a cat on the mat
there is a cat on the mat
there is a cat on the mat
there is a cat on the mat



(Recap) Word2Vec Objective (Skip-gram)

- Intuition: predict the contexts words using the center word (semantically similar center words will predict similar contexts words)
- Objective: using the parameters $\theta = \{v_w, v_c\}$ to maximize the probability of predicting the context word c using the center word w

$$\max_{\theta} \prod_{(w,c) \in \mathcal{D}} p_{\theta}(c|w)$$

Probability expressed as a function of the model parameters

- How to parametrize the probability?



(Recap) Word2Vec Parameterized Objective

- Word2Vec objective: $\max_{\theta} \prod_{(w,c) \in \mathcal{D}} p_{\theta}(c|w)$

- Assume the log probability (i.e., logit) is proportional to vector dot product

$$\log p_{\theta}(c|w) \propto \mathbf{v}_c \cdot \mathbf{v}_w$$

- The final probability distribution is given by the softmax function:

$$p_{\theta}(c|w) = \frac{\exp(\mathbf{v}_c \cdot \mathbf{v}_w)}{\sum_{c' \in |\mathcal{V}|} \exp(\mathbf{v}_{c'} \cdot \mathbf{v}_w)} \quad \rightarrow \quad \sum_{c' \in |\mathcal{V}|} p_{\theta}(c'|w) = 1$$

- Word2Vec objective (log-scale):

$$\max_{\theta} \sum_{(w,c) \in \mathcal{D}} \log p_{\theta}(c|w) = \sum_{(w,c) \in \mathcal{D}} \left(\mathbf{v}_c \cdot \mathbf{v}_w - \log \sum_{c' \in |\mathcal{V}|} \exp(\mathbf{v}_{c'} \cdot \mathbf{v}_w) \right)$$



Word2Vec Negative Sampling

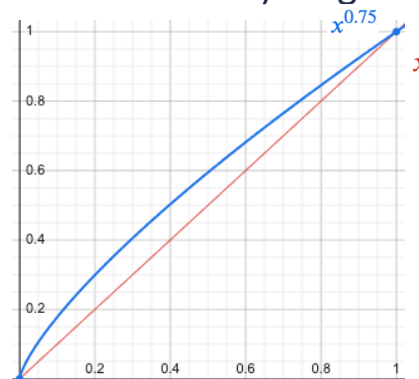
- Challenges with the original objective: **Sum over the entire vocabulary – expensive!**

$$\max_{\theta} \sum_{(w,c) \in \mathcal{D}} \log p_{\theta}(c|w) = \sum_{(w,c) \in \mathcal{D}} \left(\mathbf{v}_c \cdot \mathbf{v}_w - \log \sum_{c' \in |\mathcal{V}|} \exp(\mathbf{v}_{c'} \cdot \mathbf{v}_w) \right)$$

- Randomly sample a few negative terms from the vocabulary to form a negative set N
- How to sample negatives? Based on the (power-smoothed) unigram distribution

$$p_{\text{neg}}(w) \propto \left(\frac{\#(w)}{\sum_{w' \in \mathcal{V}} \#(w')} \right)^{0.75}$$

Rare words get a bit boost in sampling probability

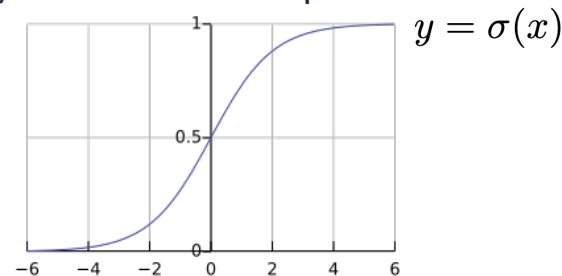




Word2Vec Negative Sampling

- Formulate a binary classification task; predict whether (w, c) is a real context pair:

$$p_{\theta}(\text{True}|c, w) = \sigma(\mathbf{v}_c \cdot \mathbf{v}_w) = \frac{1}{1 + \exp(-\mathbf{v}_c \cdot \mathbf{v}_w)}$$



- Maximize the binary classification probability for real context pairs, and minimize for negative (random) pairs

$$\max_{\theta} \sum_{(w,c) \in \mathcal{D}} \left(\log \sigma(\mathbf{v}_c \cdot \mathbf{v}_w) - \sum_{c' \in \mathcal{N}} \log \sigma(\mathbf{v}_{c'} \cdot \mathbf{v}_w) \right)$$

↓
↓

Real context pair
Negative context pair



Word2Vec Optimization

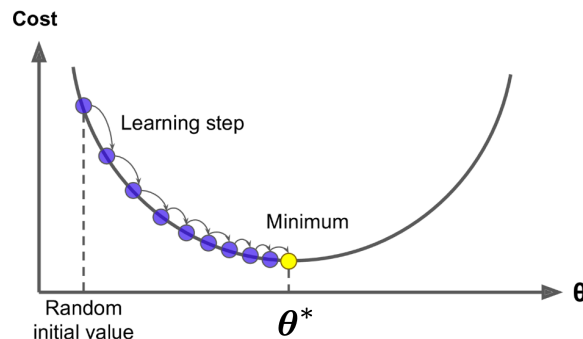
- How to optimize the following objective?

$$\max_{\theta} \sum_{(w,c) \in \mathcal{D}} \left(\log \sigma(\mathbf{v}_c \cdot \mathbf{v}_w) - \sum_{c' \in \mathcal{N}} \log \sigma(\mathbf{v}_{c'} \cdot \mathbf{v}_w) \right)$$

- Stochastic gradient descent (SGD)!
- First, initialize parameters $\theta = \{\mathbf{v}_w, \mathbf{v}_c\}$ with random d -dimensional vectors
- In each step: update parameters in the direction of the gradient of the objective (weighted by the learning rate)

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \nabla_{\theta} \mathcal{L} \Big|_{\theta = \theta^{(t)}}$$

← Learning rate Loss function →



Word2Vec Hyperparameters

Join at

slido.com

#2098 474



- Word embedding dimension d (usually 100-300)
 - Larger d provides richer vector semantics
 - Extremely large d suffers from inefficiency and curse of dimensionality
- Local context window size l (usually 5-10)
 - Smaller l learns from immediately nearby words – more syntactic information
 - Bigger l learns from longer-ranged contexts – more semantic/topical information
- Number of negative samples k (usually 5-10)
 - Larger k usually makes training more stable but also more costly
- Learning rate η (usually 0.02-0.05)

Agenda

- Sparse vs. Dense Vectors
- Word Embeddings: Overview
- Word2Vec Training
- **Word Embedding Properties & Evaluation**
- Other Word Embedding Methods
- Word Embedding Limitations & Summary

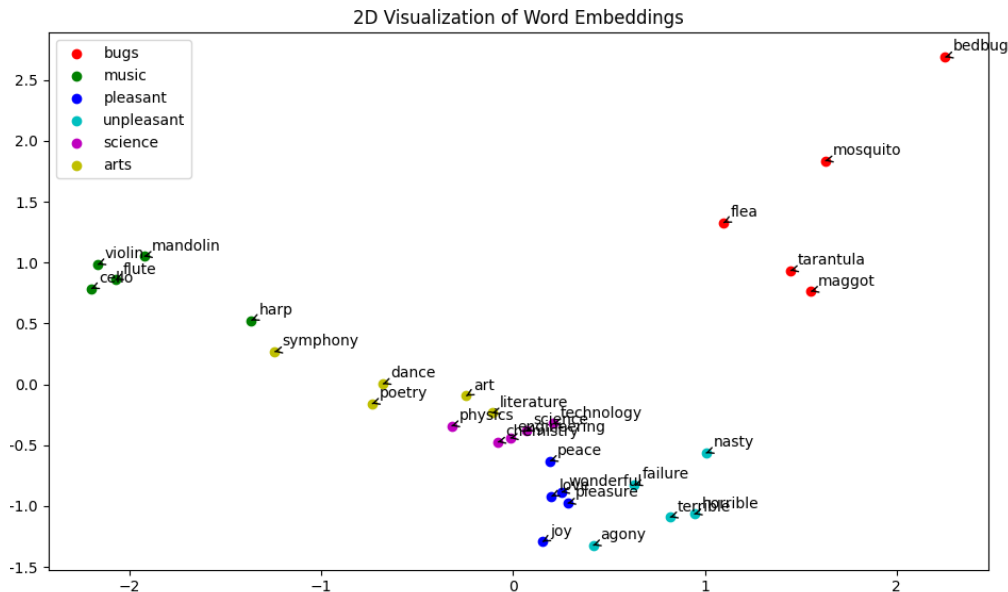
Join at
slido.com
#2098 474



Word Similarity



- Measure word similarity with cosine similarity between embeddings $\cos(\mathbf{v}_{w_1}, \mathbf{v}_{w_2})$
- Higher cosine similarity = more semantically close





Word Similarity Evaluation

- An **intrinsic** word embedding evaluation
- Measure how well word vector similarity correlates with human judgments
- Example dataset: WordSim353 (353 word pairs with their similarity scores assessed by humans)

Word 1	Word 2	Human (mean)
tiger	cat	7.35
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

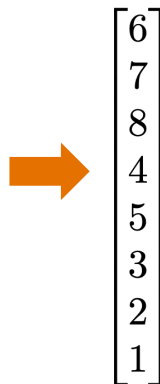


Correlation Metric

Spearman rank correlation: measure the correlation between two rank variables

Word 1	Word 2	Human (mean)
tiger	cat	7.35
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

Rank by human



$$r = \frac{\text{Cov}[R[X], R[Y]]}{\sigma_{R[X]} \sigma_{R[Y]}}$$

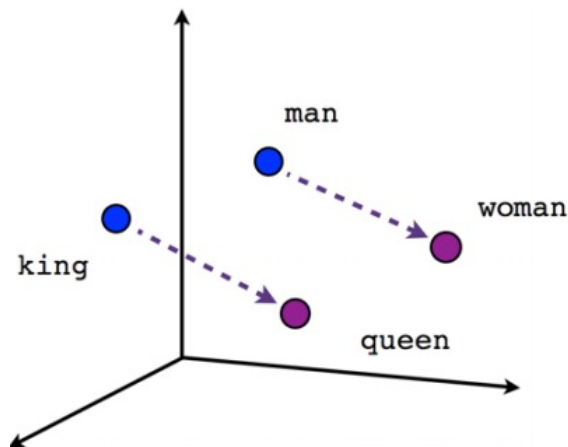
Covariance
Standard deviations



Word Analogy

- Word embeddings reflect intuitive semantic and syntactic analogy
- Example: man : woman :: king : ? $\mathbf{v}_{\text{queen}} \approx \mathbf{v}_{\text{woman}} - \mathbf{v}_{\text{man}} + \mathbf{v}_{\text{king}}$
- General case: find the word such that $a : b :: c : ?$
- Find the word that maximizes the cosine similarity

$$\begin{aligned}
 w &= \arg \max_{w' \in \mathcal{V}} \cos(\mathbf{v}_b - \mathbf{v}_a + \mathbf{v}_c, \mathbf{v}_{w'}) \\
 &= \arg \max_{w' \in \mathcal{V}} \frac{(\mathbf{v}_b - \mathbf{v}_a + \mathbf{v}_c) \cdot \mathbf{v}_{w'}}{|\mathbf{v}_b - \mathbf{v}_a + \mathbf{v}_c| |\mathbf{v}_{w'}|}
 \end{aligned}$$



Word Analogy Evaluation

 Join at
slido.com
#2098 474


- Word analogy is another **intrinsic** word embedding evaluation
- Encompass various types of word relationships
- Usually use accuracy as the metric

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

 Figure source: <https://arxiv.org/pdf/1301.3781>



Extrinsic Evaluation of Word Embeddings

- Word embeddings can be used as input features to task-specific NLP models
- Example 1: Text classification (topic/sentiment classification)
 - Sentence/document embeddings are obtained by applying sequence modeling architectures on top of word embeddings
 - Classification accuracy is used as the extrinsic metric
- Example 2: Named entity recognition (NER)
 - Find and classify entity names (e.g., person, organization, location) in text
 - Concatenated word embeddings can be used to represent spans of words (entities)
 - Precision/recall/F1 are used as the extrinsic metrics
- Word embedding demo

Agenda

- Sparse vs. Dense Vectors
- Word Embeddings: Overview
- Word2Vec Training
- Word Embedding Properties & Evaluation
- **Other Word Embedding Methods**
- Word Embedding Limitations & Summary

Join at

slido.com

#2098 474



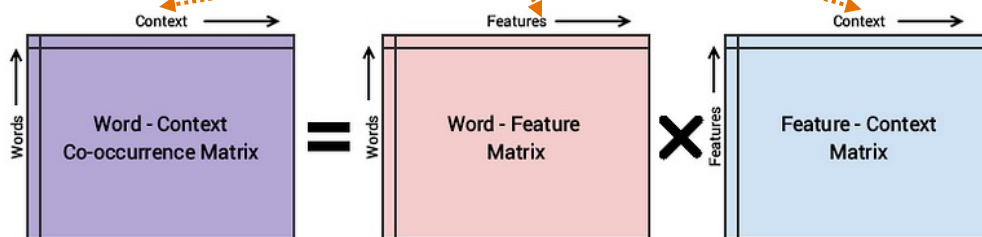


GloVe: Global Vectors for Word Representation

- Core insight: ratios of co-occurrence probabilities can encode meaning components

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

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



A (reweighted) matrix factorization problem



FastText: Incorporating Subword Information

- Motivation: treating each word as a whole ignores the internal structure of words
- Solution: representing words with character N-grams
- Example (assume character trigram):
 - the word “where” will be decomposed into: <wh, whe, her, ere, re>
 - The word “her” will be represented as <her>
- Each word is represented by the sum of the vectors of its character N-grams
- Use the same training objective as Word2Vec
- Benefit: more robust representations for rare words

Word Embedding: Further Reading

Join at
slido.com
#2098 474



- [Neural Word Embedding as Implicit Matrix Factorization](#) [Levy & Goldberg, 2014]
- [Distributed Representations of Sentences and Documents](#) [Le & Mikolov, 2014]
- [Poincaré Embeddings for Learning Hierarchical Representations](#) [Nickel & Kiela, 2017]
- [Word Translation without Parallel Data](#) [Conneau et al., 2018]

Agenda

- Sparse vs. Dense Vectors
- Word Embeddings: Overview
- Word2Vec Training
- Word Embedding Properties & Evaluation
- Other Word Embedding Methods
- Word Embedding Limitations & Summary

Join at

slido.com

#2098 474





Word Embedding Limitations

- **Static representations (context independence):** A word is always assigned a single vector representation regardless of its context
 - Words can have multiple meanings (polysemy)
 - Example: “bank” can mean a financial institution or the side of a river
- **Shallow representations:** Word embedding learning only focus on local context (a fixed window size of nearby words)
 - Cannot capture complex syntactic or long-range dependencies
 - Example: “The book that the president, who everyone admires, recommended is fascinating.”
– distant subject (“book”) and adjective (“fascinating”)
- **Single-word representations:** Can only represent single words rather than larger linguistic units (phrases, sentences, paragraphs)
 - Many tasks require modeling relationships & compositionality between larger text chunks
 - Example: “They sell delicious hot dogs.” – “hot dogs” should be understood as an entire unit

Summary: Sparse vs. Dense Vectors

Join at

slido.com

#2098 474



- Sparse vectors are derived based on frequencies/counts
 - High-dimensional – inefficiency in training & storage
 - Lots of zero dimensions – do not reflect semantics
- Dense vectors distribute information across multiple/all dimensions
 - Fewer dimensions; most dimensions are non-zero
 - More compact, robust, scalable, and efficient
 - Less interpretable

Summary: Word Embedding Learning

Join at
slido.com
#2098 474



- Distributional hypothesis
 - Words that occur in similar contexts tend to have similar meanings
 - Infer semantic similarity based on context similarity
- Word embeddings
 - Construct a prediction task: use a center word's embedding to predict its contexts
 - Two words with similar embeddings will predict similar contexts => semantically similar
 - Word embedding is a form of self-supervised learning

Summary: Word2Vec

Join at

slido.com

#2098 474



- Two variants: Skip-gram and CBOW
- Skip-gram: predict the words in a local context window surrounding the center word
- Employ negative sampling to improve training efficiency
- Use SGD to optimize vector representations
- Word embedding applications & evaluations
 - Word similarity
 - Word analogy
 - Use as input features to downstream tasks (e.g., text classification; NER)



Thank You!

Yu Meng

University of Virginia

yumeng5@virginia.edu