Introduction to Word Embedding

Yu Meng
University of Virginia
yumeng5@virginia.edu

Sep 16, 2024

UNIVERSITY of VIRGINIA

Join at
slido.com
#2998 538

Announcement

- Assignment 1 grades posted; reference answer released
- Contact Wenqian (pvc7hs@virginia.edu) if you have questions about your grade

Join at
slido.com
#2998 538

Overview of Course Contents

- 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) Word Semantics & Senses

- Understanding word semantics & senses help us build better language models!
- Word semantics is complex
  - Polysemy: a single word having multiple meanings
  - Multi-faceted: word meanings entail various aspects (e.g., valence, arousal, dominance)
- Many types of word relations: synonyms, antonyms, hyponyms & hypernyms…
- Word relations are usually not binarized (e.g., perfect synonyms are rare); word similarity is usually a more flexible measure

(Recap) Classic Word Representations

- Large-scale lexical databases (WordNet) were constructed in early NLP developments
- WordNet consists of manually curated synsets linked by relation edges
- WordNet can be used as a database for word sense disambiguation
- WordNet has significant limitations:
  - Require significant efforts to construct and maintain/update
  - Limited coverage of domain-specific terms & low-resource language
  - Only support individual words and their meanings

(Recap) Document Similarity

- Document vector representation with word frequencies:

$$\boldsymbol{v}_{d_1} = [1, 114, 36, 20] \quad \boldsymbol{v}_{d_2} = [0, 80, 58, 15] \quad \boldsymbol{v}_{d_3} = [7, 62, 1, 2] \quad \boldsymbol{v}_{d_4} = [13, 89, 4, 3]$$

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- “fool” and “wit” occur much more frequently in d_1 and d_2 than d_3 and d_4
- d_1 and d_2 are comedies $\cos(\boldsymbol{v}_{d_1}, \boldsymbol{v}_{d_2}) = 0.95$ $\quad \cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.81$
- Word frequencies in documents do reflect the semantic similarity between documents!

Figure source: https://web.stanford.edu/~jurafsky/slp3/6.pdf

(Recap) Words Represented with Documents

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

$$\boldsymbol{v}_{\text{battle}} = [1, 0, 7, 13]$$
$$\boldsymbol{v}_{\text{good}} = [114, 80, 62, 89]$$
$$\boldsymbol{v}_{\text{fool}} = [36, 58, 1, 4]$$
$$\boldsymbol{v}_{\text{wit}} = [20, 15, 2, 3]$$

$$\cos(\boldsymbol{v}_{\text{fool}}, \boldsymbol{v}_{\text{wit}}) = 0.93$$
$$\cos(\boldsymbol{v}_{\text{fool}}, \boldsymbol{v}_{\text{battle}}) = 0.09$$

Previously:

$$\boldsymbol{v}_{\text{battle}} = [1, 0, 0, 0]$$
$$\boldsymbol{v}_{\text{good}} = [0, 1, 0, 0]$$
$$\boldsymbol{v}_{\text{fool}} = [0, 0, 1, 0]$$
$$\boldsymbol{v}_{\text{wit}} = [0, 0, 0, 1]$$

$$\cos(\boldsymbol{v}_{\text{fool}}, \boldsymbol{v}_{\text{wit}}) = 0$$
$$\cos(\boldsymbol{v}_{\text{fool}}, \boldsymbol{v}_{\text{battle}}) = 0$$

Document co-occurrence statistics provide coarse-grained contexts

(Recap) Word Co-occurrence

- Word-word matrix with ±4 word window

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

- “digital” and “information” both co-occur with “computer” and “data” frequently
- “cherry” and “strawberry” both co-occur with “pie” and “sugar” frequently
- Word co-occurrence statistics reflect word semantic similarity!
- Issues? Sparsity!

(Recap) Raw Frequency Is Biased

- On the one hand, high frequency can imply semantic similarity
- On the other hand, there are words with universally high frequencies

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- Can we reweight the raw frequencies so that distinctively high frequency terms are highlighted?

(Recap) TF-IDF Weighting

The TF-IDF weighted value characterizes the "salience" of a term in a document

$$\text{TF-IDF}(w, d) = \text{TF}(w, d) \times \text{IDF}(w)$$

TF-IDF weighted

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.246	0	0.454	0.520
good	0	0	0	0
fool	0.030	0.033	0.0012	0.0019
wit	0.085	0.081	0.048	0.054

$$\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.10 \quad \cos(\boldsymbol{v}_{d_3}, \boldsymbol{v}_{d_4}) = 0.99$$

Raw counts

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

$$\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.81 \quad \cos(\boldsymbol{v}_{d_3}, \boldsymbol{v}_{d_4}) = 0.99$$

Join at
slido.com
#2998 538

(Recap) How to Define Documents?

- The concrete definition of documents is usually open to different design choices
  - Wikipedia article/page
  - Shakespeare play
  - Book chapter/section
  - Paragraph/sentence
  - …
- Larger documents provide broader context; smaller ones provide focused insights
- Depends on the analysis need: interested in global trends across documents (e.g., news articles) vs. more local patterns (e.g., specific sections of a legal document)?

Probability-Based Weighting

- TF-IDF weighting scheme is based on heuristics
- Can we weigh the raw counts with probabilistic approaches?
- Intuition: the association between two words can be reflected by **how much they co-occur more than by chance**

context word — summed counts

center word — summed counts

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

Figure source: https://web.stanford.edu/~jurafsky/slp3/6.pdf

Word Association Based on Probability

- In probability theory, when two random variables A & B are independent, we have

Joint probability $p(A, B) = p(A)p(B)$

- When two words co-occur by chance, we expect their probabilities to satisfy the independence assumption: $p(w_1, w_2) = p(w_1)p(w_2)$
- When $p(w_1, w_2) > p(w_1)p(w_2)$, two words co-occur more often than would be expected by chance
- How to develop a probabilistic metric to characterize this association?

Pointwise Mutual Information (PMI)

- PMI compares the probability of two words co-occurring with the probabilities of the words occurring independently

$$\text{PMI} = \log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)} = \log_2 \frac{\#(w_1, w_2)}{\#(w_1)\#(w_2)}$$

- PMI = 0: Two words co-occur as expected by chance => no particular association
- PMI > 0: Two words co-occur more often than by chance => the higher the PMI, the stronger the association between the words
- PMI < 0: Two words co-occur less often than expected by chance => negative associations; not much actionable insight
- Positive PMI (PPMI): replaces all negative PMI values with zero

$$\text{PPMI} = \max\left(\log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)}, 0\right)$$

PPMI Example

Raw counts

	computer	data	result	pie	sugar
cherry	2	8	9	442	25
strawberry	0	0	1	60	19
digital	1670	1683	85	5	4
information	3325	3982	378	5	13

PPMI-weighted matrix

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Issue: biased toward infrequent events (rare words tend to have very high PMI values)

PPMI with Power Smoothing

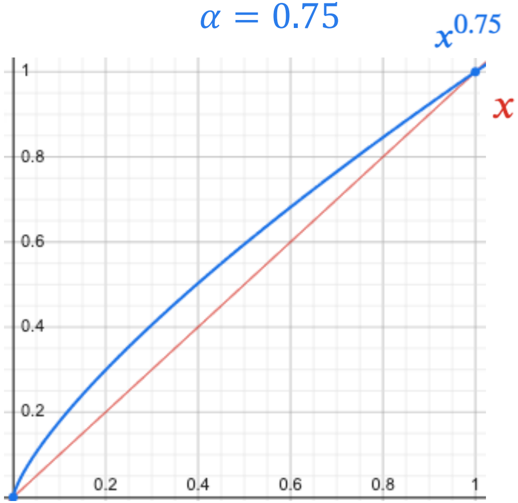
Power smoothing: Manually boost low probabilities by raising to a power α

$$\text{PPMI} = \max\left(\log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)}, 0\right)$$

Original: $$p(w) = \frac{\#(w)}{\sum_{w' \in \mathcal{V}} \#(w')}$$

Power smoothed: ($\alpha < 1$) $$p_\alpha(w) = \frac{\#(w)^\alpha}{\sum_{w' \in \mathcal{V}} \#(w')^\alpha}$$

$\alpha = 0.75$



PPMI with Add-*k* Smoothing

- Another way of increasing the counts of rare occurrences is to apply add-*k* smoothing

	computer	data	result	pie	sugar
cherry	2	8	9	442	25
strawberry	0	0	1	60	19
digital	1670	1683	85	5	4
information	3325	3982	378	5	13

Add a constant *k* to all counts

- The larger the *k* (*k* can be larger than 1), the more we boost the probability of rare occurrences

TF-IDF vs. PMI Weighting

- TF-IDF
  - Measures the importance of a word in a document relative to other documents (corpus)
  - Context granularity: document level
  - Based on heuristics
  - High TF-IDF = frequent in a document but infrequent across the corpus
- PMI:
  - Measures the strength of association between two words
  - Context granularity: word pair level (usually based on local context windows)
  - Based on probability assumptions
  - High PMI = words co-occur more often than expected by chance, a strong association

Join at
slido.com
#2998 538

Agenda

- Sparse vs. Dense Vectors
- Word Embeddings: Overview
- Word2Vec

Count-based Vector Limitations

- Count-based vectors are **sparse** (lots of zeros)
  - Zero values in the vectors do not carry any semantics
- Count-based vectors are **long** (many dimensions)
  - Vector dimension = vocabulary size (usually > 10K)
  - "Curse of dimensionality": metrics (e.g. cosine) become less meaningful in high dimensions

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Many more words!

Dense Vectors

- More efficient & effective vector representations?
- **Dense** vectors!
  - Most/all dimensions in the vectors are non-zero
  - Usually floating-point numbers; each dimension could be either positive or negative
  - Dimension much smaller than sparse vectors (i.e., << 10K)
- Also called "**distributed** representations"
  - The information is **distributed** across multiple units/dimensions
  - Each unit/dimension participates in representing multiple pieces of information
  - Analogous to human brains: the brain stores and processes information in a distributed manner: instead of having a single neuron/region represent a concept, information is represented across a network of neurons

UNIVERSITY of VIRGINIA

Join at
slido.com
#2998 538

Dense Vector Example

- One dimension might (partly) contribute to distinguishing animals ("cat" "dog") from vehicles ("car" "truck")
- One dimension might (partly) capture some aspect of size
- Another might (partly) represent formality or emotional tone
- …
- Each of these dimensions is not exclusively responsible for any single concept, but together, they combine to form a rich and nuanced representation of words!

$$\boldsymbol{v}_{\text{good}} = [-1.34, 2.58, 0.37, 4.32, -3.21, \ldots]$$
$$\boldsymbol{v}_{\text{nice}} = [-0.58, 1.97, 0.20, 3.13, -2.58, \ldots]$$

Only showing two decimal places
(typically they are floating point numbers!)

Dense Vectors Pros & Cons

- **(+) 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

Agenda

- Sparse vs. Dense Vectors
- Word Embeddings: Overview
- Word2Vec

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!

Example source: https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf

UNIVERSITY of VIRGINIA

Join at
slido.com
#2998 538

Word Embeddings: General Idea

- Learn dense vector representations of words based on distributional hypothesis
- Semantically similar words (based on context similarity) will have similar vector representations
- **Embedding**: a mapping that takes elements from one space and represents them in a different space

$$\boldsymbol{v}_{\text{to}} = [1,0,0,0,0,0,\dots]$$
$$\boldsymbol{v}_{\text{by}} = [0,1,0,0,0,0,\dots]$$
$$\boldsymbol{v}_{\text{that}} = [0,0,1,0,0,0,\dots]$$
$$\boldsymbol{v}_{\text{good}} = [0,0,0,1,0,0,\dots]$$
$$\boldsymbol{v}_{\text{nice}} = [0,0,0,0,1,0,\dots]$$
$$\boldsymbol{v}_{\text{bad}} = [0,0,0,0,0,1,\dots]$$

2D visualization of a word embedding space

Figure source: https://web.stanford.edu/~jurafsky/slp3/6.pdf

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!

Predicted contexts

$\boldsymbol{v}_{\text{ong choy}} \rightarrow$ sautéed, garlic, rice, salty, leaves, …

Predicted contexts

$\boldsymbol{v}_{\text{spinach}} \rightarrow$ sautéed, garlic, rice, salty, leaves, …

Word Embedding Is Self-Supervised Learning

- **Self-supervised learning**: a model learns to predict parts of its input from other parts of the same input

Input: *Ong choy is superb over rice*

Prediction task: Ong choy → is / superb / over / rice

- Self-supervised learning vs. supervised learning:
  - Self-supervised learning: **no human-labeled data** – the model learns from unlabeled data by generating supervision through the structure of the data itself
  - Supervised learning: **use human-labeled data** – the model learns from human annotated input-label pairs

Thank You!

Yu Meng
University of Virginia
yumeng5@virginia.edu