Word Senses and Semantics

Slido: https://app.sli.do/event/5e2iag4PJwbU6DB3CXQwKb

Yu Meng
University of Virginia
yumeng5@virginia.edu

Sept 8, 2025

Reminders

- Assignment 1 is due today 11:59pm!
- Assignment 2 is released (due 09/17)

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 & Recurrent Neural Networks (RNNs)
- Week 6: Language Modeling with Transformers
- Week 9: Large Language Models (LLMs) & In-context Learning
- Week 10: Knowledge in LLMs and Retrieval-Augmented Generation (RAG)
- Week 11: LLM Alignment
- Week 12: Reinforcement Learning for LLM Post-Training
- Week 13: LLM Agents + Course Summary
- Week 15 (after Thanksgiving): Project Presentations

(Recap) Language Models = Universal NLP Task Solvers

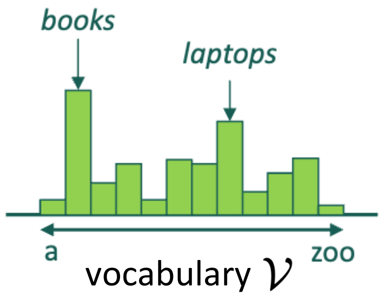
- Every NLP task can be converted into a text-to-text task!
  - Sentiment analysis: The movie’s closing scene is attractive; it was ___ (good)
  - Machine translation: “Hello world” in French is ___ (Bonjour le monde)
  - Question answering: Which city is UVA located in? ___ (Charlottesville)
  - …
- All these tasks can be formulated as a language modeling problem!

(Recap) Language Modeling: Probability Decomposition

- Given a text sequence $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, how can we model $p(\boldsymbol{x})$?
- Autoregressive assumption: the probability of each word only depends on its previous tokens

$$p(\boldsymbol{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)\cdots p(x_n|x_1, \ldots, x_{n-1}) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1})$$

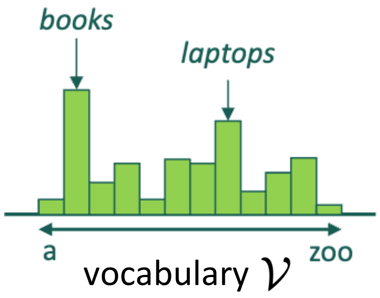
- How to guarantee the probability distributions are valid?
  - Non-negative
    $$p(x_i = w|x_1, \ldots, x_{i-1}) \geq 0, \quad \forall w \in \mathcal{V}$$
  - Summed to 1: $$\sum_{w \in \mathcal{V}} p(x_i = w|x_1, \ldots, x_{i-1}) = 1$$



- The goal of language modeling is to learn the distribution $p(x_i = w|x_1, \ldots, x_{i-1})$!

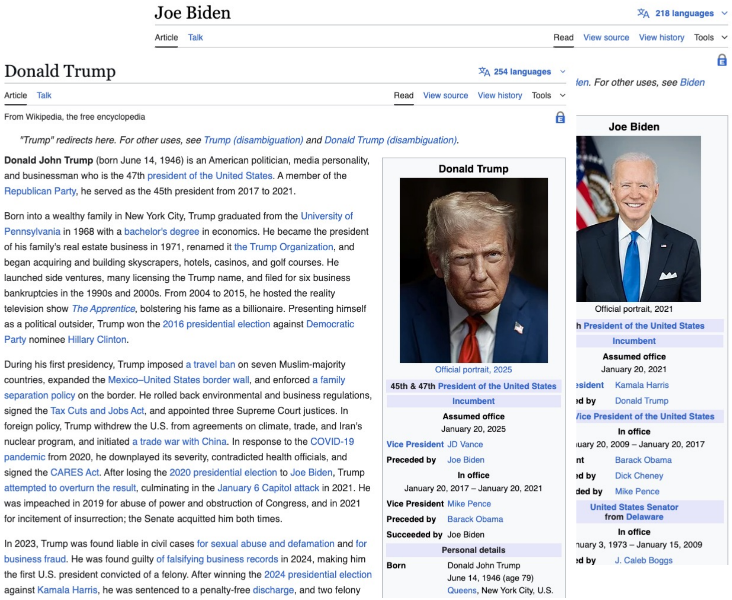
(Recap) Language Models Are Generative Models

- Suppose we have a language model that gives us the estimate of $p(w|x_1, \ldots, x_{i-1})$, we can generate the next tokens one-by-one!
- Sampling: $x_i \sim p(w|x_1, \ldots, x_{i-1})$
- Or greedily: $x_i \leftarrow \arg\max_w p(w|x_1, \ldots, x_{i-1})$
- But how do we know when to stop generation?
- Use a special symbol [EOS] (end-of-sequence) to denote stopping



(Recap) How to Obtain A Language Model?

Learn the probability distribution $p(w|x_1, \ldots, x_{i-1})$ from a training corpus!



Joe Biden

218 languages

Article Talk Read View source View history Tools

Donald Trump

254 languages

Article Talk Read View source View history Tools

From Wikipedia, the free encyclopedia

"Trump" redirects here. For other uses, see Trump (disambiguation) and Donald Trump (disambiguation).

Donald John Trump (born June 14, 1946) is an American politician, media personality, and businessman who is the 47th president of the United States. A member of the Republican Party, he served as the 45th president from 2017 to 2021.

Born into a wealthy family in New York City, Trump graduated from the University of Pennsylvania in 1968 with a bachelor's degree in economics. He became the president of his family's real estate business in 1971, renamed it the Trump Organization, and began acquiring and building skyscrapers, hotels, casinos, and golf courses. He launched side ventures, many licensing the Trump name, and filed for six business bankruptcies in the 1990s and 2000s. From 2004 to 2015, he hosted the reality television show *The Apprentice*, bolstering his fame as a billionaire. Presenting himself as a political outsider, Trump won the 2016 presidential election against Democratic Party nominee Hillary Clinton.

During his first presidency, Trump imposed a travel ban on seven Muslim-majority countries, expanded the Mexico–United States border wall, and enforced a family separation policy on the border. He rolled back environmental and business regulations, signed the Tax Cuts and Jobs Act, and appointed three Supreme Court justices. In foreign policy, Trump withdrew the U.S. from agreements on climate, trade, and Iran's nuclear program, and initiated a trade war with China. In response to the COVID-19 pandemic from 2020, he downplayed its severity, contradicted health officials, and signed the CARES Act. After losing the 2020 presidential election to Joe Biden, Trump attempted to overturn the result, culminating in the January 6 Capitol attack in 2021. He was impeached in 2019 for abuse of power and obstruction of Congress, and in 2021 for incitement of insurrection; the Senate acquitted him both times.

In 2023, Trump was found liable in civil cases for sexual abuse and defamation and for business fraud. He was found guilty of falsifying business records in 2024, making him the first U.S. president convicted of a felony. After winning the 2024 presidential election against Kamala Harris, he was sentenced to a penalty-free discharge, and two felony

Donald Trump	
Official portrait, 2025	
45th & 47th President of the United States	
Incumbent	
Assumed office January 20, 2025	
Vice President	JD Vance
Preceded by	Joe Biden
In office January 20, 2017 – January 20, 2021	
Vice President	Mike Pence
Preceded by	Barack Obama
Succeeded by	Joe Biden
Personal details	
Born	Donald John Trump June 14, 1946 (age 79) Queens, New York City, U.S.

len. For other uses, see Biden

Joe Biden	
Official portrait, 2021	
h President of the United States	
Incumbent	
Assumed office January 20, 2021	
ısident	Kamala Harris
ıd by	Donald Trump
/ice President of the United States	
In office ıuary 20, 2009 – January 20, 2017	
nt	Barack Obama
ıd by	Dick Cheney
ded by	Mike Pence
United States Senator from Delaware	
In office ıuary 3, 1973 – January 15, 2009	
ıd by	J. Caleb Boggs

Learning target:

$p(w|x_1, \ldots, x_{i-1})$

Text corpora contain rich distributional statistics!

(Recap) N-gram Language Model: Simplified Assumption

- Challenge of language modeling: hard to keep track of all previous tokens!

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | \boxed{x_1, \ldots, x_{i-1}})$$

Long context!
(Can we model long contexts at all?
Yes, but not for now!)

- Instead of keeping track of all previous tokens, assume the probability of a word is only dependent on the previous N−1 words

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1}) \boxed{\approx \prod_{i=1}^{n} p(x_i | x_{i-N+1}, \ldots, x_{i-1})}$$

N-gram assumption

Should N be larger or smaller?

(Recap) How to Learn N-grams?

- Probabilities can be estimated by frequencies (maximum likelihood estimation)!

$$p(x_i|x_{i-N+1},\ldots,x_{i-1}) = \frac{\#(x_{i-N+1},\ldots,x_{i-1},x_i)}{\#(x_{i-N+1},\ldots,x_{i-1})}$$

How many times (counts) the sequences occur in the corpus

- Unigram: $p(x_i) = \dfrac{\#(x_i)}{\#(\text{all word counts in the corpus})}$
- Bigram: $p(x_i|x_{i-1}) = \dfrac{\#(x_{i-1},x_i)}{\#(x_{i-1})}$
- Trigram: $p(x_i|x_{i-2},x_{i-1}) = \dfrac{\#(x_{i-2},x_{i-1},x_i)}{\#(x_{i-2},x_{i-1})}$

(Recap) Unigram Issues: No Word Correlations

- Learned unigram probabilities:

$$p([\text{BOS}]) = \frac{3}{23}, \quad p([\text{EOS}]) = \frac{3}{23}, \quad p(\text{"the"}) = \frac{3}{23}, \quad p(\text{"cat"}) = \frac{3}{23},$$
$$p(\text{"mat"}) = \frac{2}{23}, \quad p(\text{"I"}) = \frac{2}{23}, \quad p(\text{"a"}) = \frac{2}{23}, \quad p(\text{"have"}) = \frac{1}{23},$$
$$p(\text{"like"}) = \frac{1}{23}, \quad p(\text{"is"}) = \frac{1}{23}, \quad p(\text{"on"}) = \frac{1}{23}, \quad p(\text{"and"}) = \frac{1}{23}$$

- Is unigram reliable for estimating the sequence likelihood?

For simplicity, omitting [BOS] & [EOS] in the calculation

$$p(\text{"the the the the"}) = p(\text{"the"}) \times p(\text{"the"}) \times p(\text{"the"}) \times p(\text{"the"}) \approx 0.0003$$
$$p(\text{"I have a cat"}) = p(\text{"I"}) \times p(\text{"have"}) \times p(\text{"a"}) \times p(\text{"cat"}) \approx 0.00004$$

- Why? Unigram ignores the relationships between words!

(Recap) Bigram Issues: Sparsity

- Learned bigram probabilities:

$$p(\text{“I”}|[\text{BOS}]) = \frac{2}{3}, \quad p(\text{“The”}|[\text{BOS}]) = \frac{1}{3}, \quad p([\text{EOS}]|\text{“mat”}) = 1, \quad p([\text{EOS}]|\text{“cat”}) = \frac{1}{3},$$
$$p(\text{“cat”}|\text{“the”}) = \frac{2}{3}, \quad p(\text{“mat”}|\text{“the”}) = \frac{1}{3}, \quad p(\text{“is”}|\text{“cat”}) = \frac{1}{3}, \quad p(\text{“and”}|\text{“cat”}) = \frac{1}{3},$$
$$p(\text{“have”}|\text{“I”}) = \frac{1}{2}, \quad p(\text{“like”}|\text{“I”}) = \frac{1}{2}, \quad p(\text{“a”}|\text{“have”}) = 1, \quad p(\text{“cat”}|\text{“a”}) = \frac{1}{2}$$

- Does bigram address the issue of unigram?

For simplicity, omitting [EOS] in the calculation

$$p(\text{“the the the the”}) = p(\text{“the”}|[\text{BOS}]) \times p(\text{“the”}|\text{“the”}) \times p(\text{“the”}|\text{“the”}) \times p(\text{“the”}|\text{“the”}) = 0$$
$$p(\text{“I have a cat”}) = p(\text{“I”}|[\text{BOS}]) \times p(\text{“have”}|\text{“I”}) \times p(\text{“a”}|\text{“have”}) \times p(\text{“cat”}|\text{“a”}) \approx 0.17$$

- But… $p(\text{“a cat”}) = p(\text{“a”}|[\text{BOS}]) \times p(\text{“cat”}|\text{“a”}) = 0$

Sparsity: Valid bigrams having zero probability due to no occurrence in the training corpus

UNIVERSITY of VIRGINIA

(Recap) Bigram Issues: Sparsity

Bigram counts can be mostly zero even for larger corpora!

Berkeley Restaurant Project Corpus
(>9K sentences)

can you tell me about any good cantonese restaurants close by
tell me about chez panisse
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

Second word

First word

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Lots of zero entries!

Figure source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

(Recap) Learning Trigrams

- Consider the following mini-corpus:

[BOS] The cat is on the mat [EOS]
[BOS] I have a cat and a mat [EOS]
[BOS] I like the cat [EOS]

Treating "The" & "the" as one word

- Trigram estimated from the mini-corpus $p(x_i|x_{i-2}, x_{i-1}) = \frac{\#(x_{i-2}, x_{i-1}, x_i)}{\#(x_{i-2}, x_{i-1})}$

$$p(\text{“like”}|[\text{BOS}], \text{“I”}) = \frac{1}{2}, \quad p(\text{“have”}|[\text{BOS}], \text{“I”}) = \frac{1}{2}, \quad p([\text{EOS}]|\text{“the”}, \text{“mat”}) = 1,$$
$$p(\text{“is”}|\text{“the”}, \text{“cat”}) = \frac{1}{2}, \quad p([\text{EOS}]|\text{“the”}, \text{“cat”}) = \frac{1}{2}, \quad p([\text{EOS}]|\text{“a”}, \text{“mat”}) = 1,$$
$$p(\text{“the”}|\text{“I”}, \text{“like”}) = 1, \quad p(\text{“a”}|\text{“I”}, \text{“have”}) = 1, \quad p(\text{“mat”}|\text{“on”}, \text{“the”}) = 1$$

Sparsity grows compared to bigram! ... there are more trigrams!

(Recap) N-gram Properties

- As N becomes larger
  - Better modeling of word correlations (incorporating more contexts)
  - Sparsity increases
- The number of possible N-grams (parameters) grows exponentially with N!
  - Suppose vocabulary size = 10K words
  - Possible unigrams = 10K
  - Possible bigrams = (10K)^2 = 100M
  - Possible trigrams = (10K)^3 = 1T
  - …

(Recap) N-gram Sparsity

With a larger N, the context becomes more specific, and the chances of encountering any particular N-gram in the training data are lower

```
198015222 the first
194623024 the same
168504105 the following
158562063 the world
…
14112454 the door
-----------------
23135851162 the *
```

Bigram counts

```
197302  close the window
191125  close the door
152500  close the gap
116451  close the thread
87298   close the deal
-----------------
3785230 close the *
```

Trigram counts

```
3380 please close the door
1601 please close the window
1164 please close the new
1159 please close the gate
…
0 please close the first
-----------------
13951 please close the *
```

4-gram counts

Figure source: https://lm-class.org/lectures/05%20-%20language%20models.pdf

(Recap) Overcoming Sparsity in N-gram Language Models

- Unseen N-grams in the training corpus always lead to a zero probability
- The entire sequence will have a zero probability if any of the term is zero!

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_1, \ldots, x_{i-1}) \approx \boxed{\prod_{i=1}^{n} p(x_i|x_{i-N+1}, \ldots, x_{i-1})}$$

All terms must be non-zero

- Can we fix zero-probability N-grams?

(Recap) Add-one Smoothing (Laplace Smoothing)

Add one to all the N-gram counts!

Original counts

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Smoothed counts

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Figure source: https://web.stanford.edu/~jurafsky/slp3/3.pdf

(Recap) Add-*k* Smoothing

- Instead of adding 1 to each count, we add a fractional count *k* (*k* < 1) to all N-grams

Original (no smoothing):

$$p(x_i|x_{i-N+1},\ldots,x_{i-1}) = \frac{\#(x_{i-N+1},\ldots,x_{i-1},x_i)}{\#(x_{i-N+1},\ldots,x_{i-1})}$$

Add-one smoothing:

$$p_{\text{Add-1}}(x_i|x_{i-N+1},\ldots,x_{i-1}) = \frac{\#(x_{i-N+1},\ldots,x_{i-1},x_i)+1}{\#(x_{i-N+1},\ldots,x_{i-1})+|\mathcal{V}|}$$

- Probability of N-grams under add-*k* smoothing

Add-*k* smoothing:

$$p_{\text{Add-}k}(x_i|x_{i-N+1},\ldots,x_{i-1}) = \frac{\#(x_{i-N+1},\ldots,x_{i-1},x_i)+k}{\#(x_{i-N+1},\ldots,x_{i-1})+k|\mathcal{V}|}$$

- How to choose *k*? Use a validation set!

(Recap) Smoothing via Language Model Interpolation

- Intuition: Combine the advantages of different N-grams
  - Lower-order N-grams (e.g., unigrams) capture less context but are also less sparse
  - Higher-order N-grams (e.g., trigrams) capture more context but are also more sparse
- Combine probabilities from multiple N-gram models of different Ns (e.g., unigrams, bigrams, trigrams)

$$p_{\text{Interpolate}}(x_i | x_{i-N+1}, \dots, x_{i-1}) = \underbrace{\lambda_1 p(x_i)}_{\text{Unigram}} + \underbrace{\lambda_2 p(x_i | x_{i-1})}_{\text{Bigram}} + \cdots + \underbrace{\lambda_N p(x_i | x_{i-N+1}, \dots, x_{i-1})}_{\text{N-gram}}$$

$$\sum_{n=1}^{N} \lambda_n = 1$$ Interpolation weights sum to 1

- How to pick λ_n ? Use a validation set!

(Recap) Smoothing via Backoff

- Start with the highest-order N-gram available
- If that N-gram is not available (has a zero count), use the lower-order (N-1)-gram
- Continue backing off to lower-order N-grams until we reach a non-zero N-gram

$$p_{\text{Backoff}}(x_i|x_{i-N+1},\ldots,x_{i-1}) = \begin{cases} p_{\text{Backoff}}(x_i|x_{i-N+1},\ldots,x_{i-1}) & \text{If } \#(x_{i-N+1},\ldots,x_{i-1},x_i) > 0 \\ \alpha \cdot p_{\text{Backoff}}(x_i|x_{i-N+2},\ldots,x_{i-1}) & \text{Otherwise} \end{cases}$$

α (<1): discount factor that adjusts the lower-order probability

(N-1)-gram probability

- Is it possible that even after backing off to unigram, the probability is still zero?

(Recap) Out-of-vocabulary Words

- Unigrams will have a zero probability for words not occurring in the training data!
- Simple remedy: reserve a special token [UNK] for unknown/unseen words
- During testing, convert unknown words to [UNK] -> use [UNK]'s probability
- How to estimate the probability of [UNK]?
- During training, replace all rare words with [UNK], and estimate its probability as if it is a normal word
- How to determine rare words? Threshold based on counts in the training corpus
- Example: set a fixed vocabulary size of 10K, and words outside the most frequent 10K will be converted to [UNK] in training

(Recap) How to Evaluate Language Models?

- What language models should be considered “good”?
  - A perfect language model should be able to correctly predict every word in a corpus
  - We hope the language model can assign a high probability to the next word
  - Better language model = “less surprised” by the next word
- Just use the next word probability assigned by a language model as the metric!
- Does the choice of the evaluation corpus matter?

(Recap) Perplexity

- Perplexity (abbreviation: PPL) is an **intrinsic** evaluation metric for language models
- PPL = the per-word inverse probability on a test sequence $\boldsymbol{x}_{\text{test}} = [x_1, x_2, \dots, x_n]$

$$\text{PPL}(\boldsymbol{x}_{\text{test}}) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i | x_{i-N+1}, \dots, x_{i-1})}}$$

- A lower PPL = a better language model (less surprised/confused by the next word)

$$\text{PPL}(\boldsymbol{x}_{\text{test}}) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i)}}$$

Unigram

$$\text{PPL}(\boldsymbol{x}_{\text{test}}) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i | x_{i-1})}}$$

Bigram

$$\text{PPL}(\boldsymbol{x}_{\text{test}}) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i | x_{i-2}, x_{i-1})}}$$

Trigram

Perplexity can be used to evaluate general language models (e.g., large language models) too

Perplexity: Log-Scale Computation

- Computation of PPL in the raw probability scale can cause numerical instability

$$\mathrm{PPL}(\boldsymbol{x}_{\text{test}}) = \sqrt[n]{\boxed{\prod_{i=1}^{n} \frac{1}{p(x_i|x_{i-N+1}, \ldots, x_{i-1})}}}$$

Multiplication of many small probability values!

Example: (1/10) ^ 100 = 10^-100 -> risks of underflow (round to 0)

- PPL is usually computed in the log-scale in practice

$$\mathrm{PPL}(\boldsymbol{x}_{\text{test}}) = \exp\left(\log\left(\sqrt[n]{\prod_{i=1}^{n} \frac{1}{p(x_i|x_{i-N+1}, \ldots, x_{i-1})}}\right)\right) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \boxed{\log p(x_i|x_{i-N+1}, \ldots, x_{i-1})}\right)$$

Log probabilities are numerically stable

Example: log(1/10) = -2.3

Perplexity: Important Intrinsic Metric

PPL is an important metric to benchmark the development of language models

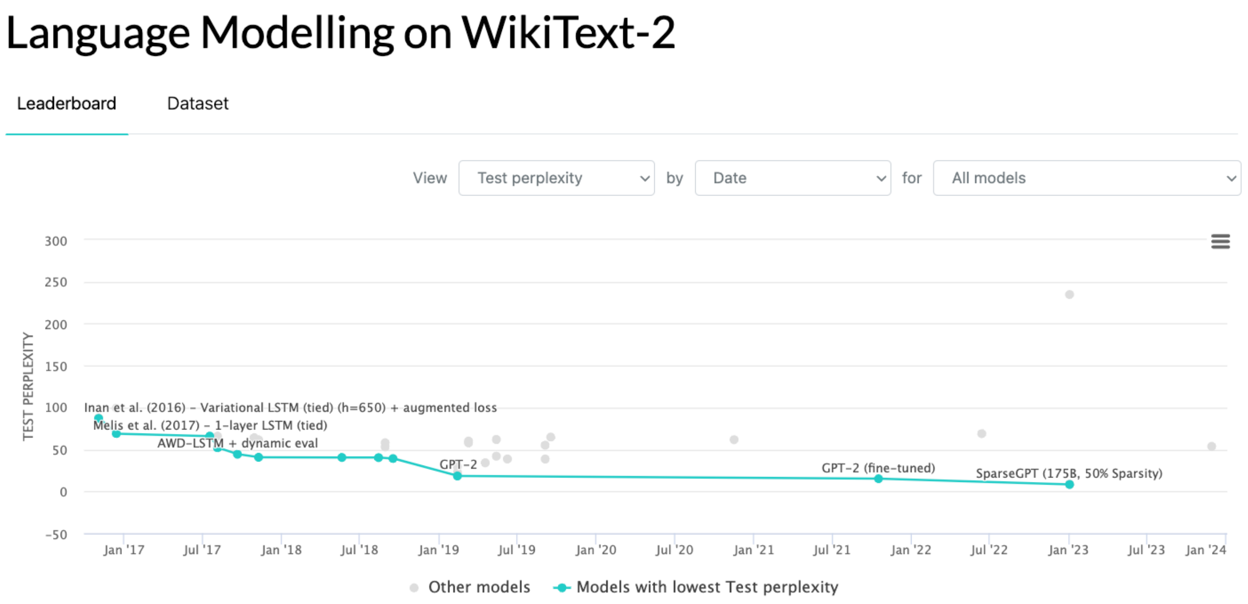


Figure source: https://paperswithcode.com/sota/language-modelling-on-wikitext-2

Intrinsic vs. Extrinsic Evaluation

- **Intrinsic metrics** (e.g., perplexity) directly measure the quality of language modeling per se, independent of any application
- **Extrinsic metrics** (e.g., accuracy) measure the language model's performance for specific tasks/applications (e.g., classification, translation)
- Intrinsic evaluations are good during the development to iterate quickly and understand specific properties of the model
- Extrinsic evaluations are essential to validate that the model improves the performance of an application in a real-world scenario
- Both intrinsic and extrinsic evaluations are commonly used to evaluation language models (they may not be always positively correlated!)

Extrinsic Evaluations for SOTA Language Models

Math reasoning, question answering, general knowledge understanding…

🤗 Open LLM Leaderboard

Model	BBH	MATH Lvl 5	GPQA	MUSR	MMLU-PRO
MaziyarPanahi/calme-2.1-rys-78b	59.47	36.4	19.24	19	49.38
MaziyarPanahi/calme-2.2-rys-78b	59.27	37.92	20.92	16.83	48.73
MaziyarPanahi/calme-2.1-qwen2-72b	57.33	36.03	17.45	20.15	49.05
MaziyarPanahi/calme-2.2-qwen2-72b	56.8	41.16	16.55	16.52	49.27
Qwen/Qwen2-72B-Instruct	57.48	35.12	16.33	17.17	48.92
alpindale/magnum-72b-v1	57.65	35.27	18.79	15.62	49.64
meta-llama/Meta-Llama-3.1-70B-Instruct	55.93	28.02	14.21	17.69	47.88
abacusai/Smaug-Qwen2-72B-Instruct	56.27	35.35	14.88	15.18	46.56
MaziyarPanahi/calme-2.2-llama3-70b	48.57	22.96	12.19	15.3	46.74
NousResearch/Hermes-3-Llama-3.1-70B	53.77	13.75	14.88	23.43	41.41
tenyx/Llama3-TenyxChat-70B	49.62	22.66	6.82	12.52	46.78

Figure source: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

Summary: Language Modeling

- Language modeling is the core problem in NLP
- Every NLP task can be formulated as language modeling
- (Autoregressive) language models can be used to generate texts
- Language model distributions are estimated (trained) on a training corpus

Summary: N-gram Language Models

- N-gram language models simplifies the (general) language modeling assumption: the probability of a word is only dependent on the previous N−1 words
- Lower-order N-grams (small N) capture less context information/word correlations
- Higher-order N-grams (bigger N) suffer from more sparsity and huge parameter space
- Smoothing techniques can be used to address sparsity in N-gram language models
  - Add-one smoothing
  - Add-*k* smoothing
  - Language model interpolation
  - Backoff

Summary: Language Model Evaluation

- Training/validation/test split required before training & evaluating language models
- Perplexity measures how "confused" the language model is about the next word
- Lower perplexity on the test set = better language model
- Perplexity is the commonly used intrinsic evaluation metric for language modeling
- Perplexity is practically computed in the log scale
- Both intrinsic and extrinsic evaluations are important

Agenda

- Introduction to Word Senses & Semantics
- Classic Word Representations
- Vector Space Model Basics

UNIVERSITY of VIRGINIA

Why Care About Word Semantics?

- Understanding word meanings helps us build better language models!
- Recall the example from N-gram lectures:

$$\begin{aligned}&[\text{BOS}]\ \text{The cat is on the mat}\ [\text{EOS}]\\&[\text{BOS}]\ \text{I have a cat and a mat}\ [\text{EOS}]\\&[\text{BOS}]\ \text{I like the cat}\ [\text{EOS}]\end{aligned}$$

$$p(\text{“cat”} \mid \text{“the”}) = \frac{2}{3}, \quad p(\text{“mat”} \mid \text{“the”}) = \frac{1}{3},$$

- Sparsity: many valid bigram counts are zero – count-based measures do not account for word semantics!
- If we know “cat” is semantically similar to “dog”, then $p(\text{“dog”} \mid \text{“the”}) \approx p(\text{“cat”} \mid \text{“the”})$

What Types of Word Semantics Exist in NLP?

- **Synonyms**: words with similar meanings
  - "happy" & "joyful"
- **Antonyms**: words with opposite meanings
  - "hot" & "cold"
- **Hyponyms** & **hypernyms**: one word is a more specific instance of another
  - "rose" is a hyponym of "flower"
  - "flower" is a hypernym of "rose"
- **Polysemy**: A single word having multiple related meanings
  - "mouse" can mean small rodents or the device that controls a cursor
- The study of these aspects of word meanings is called **lexical semantics** in linguistics

Lemmas

- **Lemma**: the base or canonical form of a word, from which other forms can be derived
  - "run" "runs" "ran" and "running" all share the lemma "run"
  - "better" and "best" share the lemma "good"
- **Lemmatization**: reducing words to their lemma
  - Allows models to recognize that different forms of a word carry the same meaning
  - An important pre-processing step in early NLP models
  - Contemporary LLMs (sort of) perform lemmatization through tokenization (later lectures!)

Synonyms

- Word that have the same meaning in some or all contexts
- Two words are synonyms if they can be substituted for each other
- Perfect synonym is very rare!
  - Typically, words are slightly different in notions of politeness, connotation, genre/style…
  - “Child” vs. “kid”: “child” is often more formal/neutral; “kid” is more informal/casual
  - “Slim” vs. “skinny”: “slim” is often more positive in connotation than “skinny”
  - “Big” vs. “Large”: “big sister” is a common phrase but “large sister” is not

Antonyms

- Words that have opposite meanings
- Gradable antonyms: exist on the ends of a spectrum or scale
  - “Hot” vs. “cold”
  - “Tall” vs. “short”
- Complementary antonyms: the presence of one directly excludes the other
  - “Alive” vs. “dead”
  - “True” vs. “false”
- Relational antonyms: express a relationship between two dependent entities
  - “Teacher” vs. “student”
  - “Buyer” vs. “seller”

Hyponyms & Hypernyms

- Describe hierarchical relationships between words based on specificity and generality
- **Hypernym** is a word that is more general/broader in meaning and can encompass a variety of more specific words
- **Hyponym** is a word that is more specific in meaning and falls under a broader category
- “Vehicle” is a hypernym for “car” “bicycle” “airplane” “boat” etc.
- “Car” “bicycle” “airplane” “boat” are hyponyms of “vehicle”
- **Hypernym/hyponym** relationship is usually transitive
  - A is a hypernym of B; B is a hypernym of C => A is a hypernym of C

Polysemy & Senses

- **Polysemy**: a single word has multiple related meanings
  - "**Light**": "This bag is **light**" / "Turn on the **light**" / "She made a **light** comment"
- **Sense**: a particular meaning or interpretation of a word in a given context
- Word relations (e.g., synonyms, antonyms, hypernyms/hyponyms) are defined between word senses!
- **Word sense disambiguation (WSD):** determine which sense of a word is being used in a specific context
  - She went to the **bank** to deposit money
  - She lives by the river **bank**
- WSD can be challenging especially when the context is short/insufficient
  - Is the query "mouse info" looking for a pet or a tool?

Word Sense Disambiguation

WSD can be an interesting/challenging test case even for the strong (multimodal) LLMs

Image generated by Nano Banana under the user prompt: *“generate an image of a baseball player caring for his bat in the cave where he lives with all the other bats”*

Example source: https://lm-class.org/lectures/04%20-%20word%20embeddings.pdf

Word Similarity

- Most words may not have many perfect synonyms, but usually have lots of similar words
  - “cat” is not a synonym of “dog”, but they are similar in meaning

vanish	disappear	9.8
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

Word similarity (on a scale from 0 to 10) manually annotated by humans

- We’ll introduce word embeddings to automatically learn word similarity next week!

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

Word Relatedness & Semantic Field

- **Word relatedness**: the meaning of words can be related in ways other than similarity
  - Functional relationship: “doctor” and “hospital” – doctors work in hospitals
  - Thematic relationship: “bread” and “butter” – often used together in the context of food
  - Conceptual relationship: “teacher” and “chalkboard” – both part of the educational context
- **Semantic field**: a set of words which cover a particular semantic domain and bear structured relations with each other
  - Semantic field of “houses”: door, roof, kitchen, family, bed…
  - Semantic field of “restaurants”: waiter, menu, plate, food, chef…
  - Semantic field of “hospitals”: surgeon, nurse, anesthetic, scalpel…

UNIVERSITY of VIRGINIA

Connotation

- Subjective/cultural/emotional associations that words carry beyond their literal meanings
  - Youthful (positive) vs. childish (negative)
  - Confident (positive) vs. arrogant (negative)
  - Economical (positive) vs. cheap (negative)
- Connotation can be described via three dimensions:
  - Valence: the pleasantness of the stimulus
  - Arousal: the intensity of emotion provoked by the stimulus
  - Dominance: the degree of control exerted by the stimulus

Connotation

- Valence: the pleasantness of the stimulus
  - High: “happy” / “satisfied”; low: “unhappy” / “annoyed”
- Arousal: the intensity of emotion provoked by the stimulus
  - High: “excited”; low: “calm”
- Dominance: the degree of control exerted by the stimulus
  - High: “controlling”; low: “influenced”

	Valence	Arousal	Dominance
courageous	8.05	5.5	7.38
music	7.67	5.57	6.5
heartbreak	2.45	5.65	3.58
cub	6.71	3.95	4.24

Earliest work on representing words with multi-dimensional vectors!

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

Agenda

- Introduction to Word Senses & Semantics
- Classic Word Representations
- Vector Space Model Basics

WordNet

- Word semantics is complex (multiple senses, various relations)!
- How did people represent word senses and relations in early NLP developments?
- WordNet: A manually curated large lexical database
- Three separate databases: one each for nouns, verbs and adjectives/adverbs
- Each database contains a set of lemmas, each one annotated with a set of senses
- Synset (synonym set): The set of near-synonyms for a sense
- Word relations (hypernym, hyponym, antonym) defined between synsets

WordNet: https://wordnet.princeton.edu/

WordNet Relations

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	$breakfast^1 \rightarrow meal^1$
Hyponym	Subordinate	From concepts to subtypes	$meal^1 \rightarrow lunch^1$
Instance Hypernym	Instance	From instances to their concepts	$Austen^1 \rightarrow author^1$
Instance Hyponym	Has-Instance	From concepts to their instances	$composer^1 \rightarrow Bach^1$
Part Meronym	Has-Part	From wholes to parts	$table^2 \rightarrow leg^3$
Part Holonym	Part-Of	From parts to wholes	$course^7 \rightarrow meal^1$
Antonym		Semantic opposition between lemmas	$leader^1 \iff follower^1$
Derivation		Lemmas w/same morphological root	$destruction^1 \iff destroy^1$

Noun relations

Relation	Definition	Example
Hypernym	From events to superordinate events	$fly^9 \rightarrow travel^5$
Troponym	From events to subordinate event	$walk^1 \rightarrow stroll^1$
Entails	From verbs (events) to the verbs (events) they entail	$snore^1 \rightarrow sleep^1$
Antonym	Semantic opposition between lemmas	$increase^1 \iff decrease^1$

Verb relations

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

UNIVERSITY of VIRGINIA

WordNet as a Graph

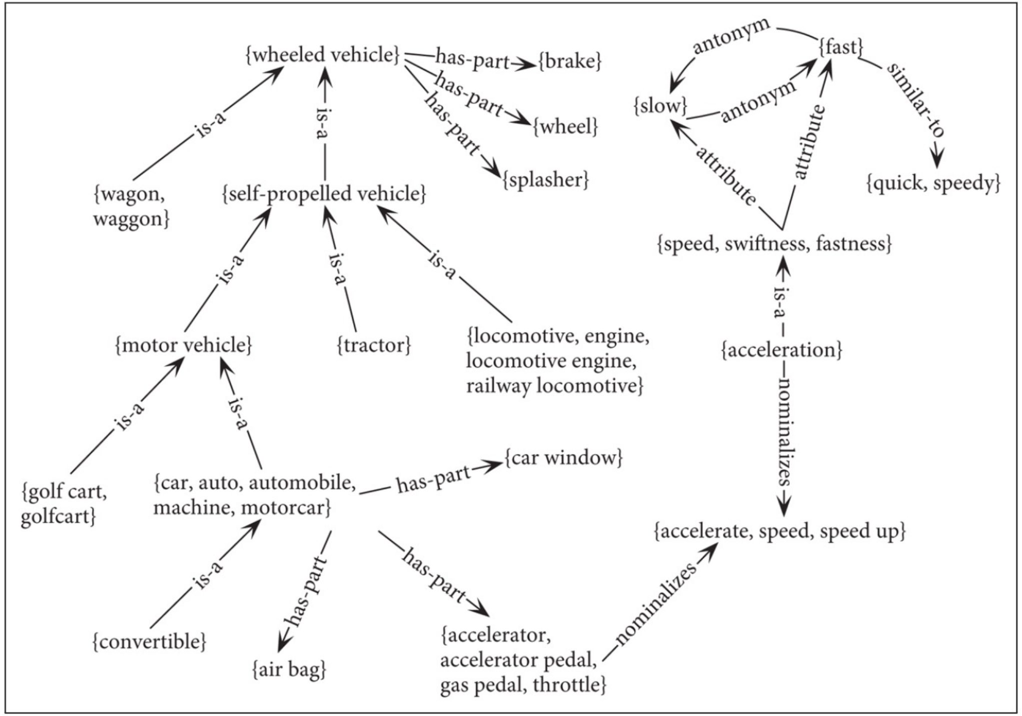


Figure source: https://academic.oup.com/edited-volume/42643/chapter/358151233

WordNet Demo

Category	Unique Strings
Noun	117798
Verb	11529
Adjective	22479
Adverb	4481

Figure source: https://lm-class.org/lectures/04%20-%20word%20embeddings.pdf

Word to search for: light [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **light**, visible light, visible radiation ((physics) electromagnetic radiation that can produce a visual sensation) *"the light was filtered through a soft glass window"*
  - *direct hyponym* | *full hyponym*
  - *domain category*
  - *direct hypernym* | *inherited hypernym* | *sister term*
  - *part holonym*
  - *derivationally related form*
- S: (n) **light**, light source (any device serving as a source of illumination) *"he stopped the car and turned off the lights"*
- S: (n) **light** (a particular perspective or aspect of a situation) *"although he saw it in a different light, he still did not understand"*
- S: (n) luminosity, brightness, brightness level, luminance, luminousness, **light** (the quality of being luminous; emitting or reflecting light) *"its luminosity is measured relative to that of our sun"*
- S: (n) **light** (an illuminated area) *"he stepped into the light"*
  - *direct hypernym* | *inherited hypernym* | *sister term*
  - *derivationally related form*
- S: (n) **light**, illumination (a condition of spiritual awareness; divine illumination) *"follow God's light"*
- S: (n) **light**, lightness (the visual effect of illumination on objects or scenes as created in pictures) *"he could paint the lightest light and the darkest dark"*
- S: (n) **light** (a person regarded very fondly) *"the light of my life"*
- S: (n) **light**, lighting (having abundant light or illumination) *"they played as long as it was light"; "as long as the lighting was good"*
- S: (n) **light** (mental understanding as an enlightening experience) *"he finally saw the light"; "can you shed light on this problem?"*
- S: (n) sparkle, twinkle, spark, **light** (merriment expressed by a brightness or gleam or animation of countenance) *"he had a sparkle in his eye"; "there's a perpetual twinkle in his eyes"*
- S: (n) **light** (public awareness) *"it brought the scandal to light"*
- S: (n) Inner Light, **Light**, Light Within, Christ Within (a divine presence

WordNet web browser: http://wordnetweb.princeton.edu/perl/webwn

WordNet for Word Sense Disambiguation

- All words WSD task: map all input words (nouns/verbs/adjectives/adverbs) to WordNet senses
- Strong baseline: map to the first sense in WordNet (most frequent)
- Modern approaches: sequence modeling architectures (later lectures!)

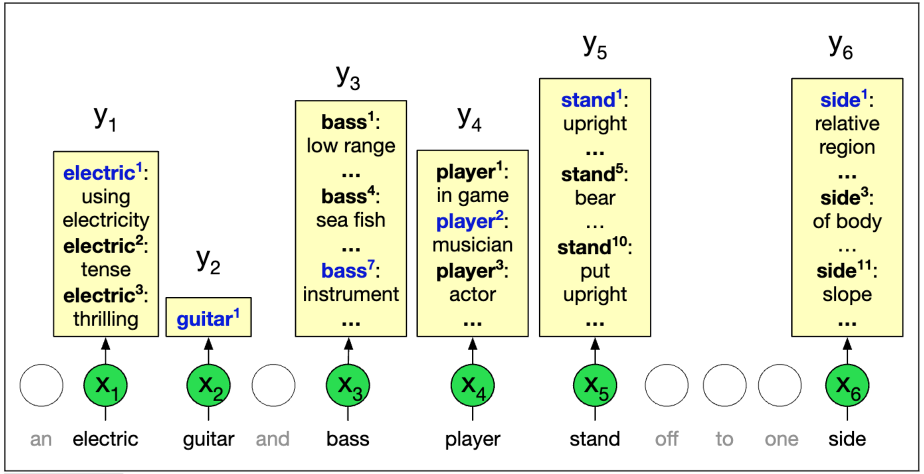


Figure source: https://web.stanford.edu/~jurafsky/slp3/G.pdf

WordNet Limitations

- Require significant efforts to construct and maintain/update
  - Hard to keep up with rapidly evolving language usage
- Limited coverage of domain-specific terms & low-resource language
  - No coverage of specialized, domain-specific terms (e.g., medical, legal, or technical)
- Only support individual words and their meanings
  - Do not account for idiomatic expressions, phrasal verbs, or collocations

A more automatic, scalable, and contextualized word semantic learning approach is needed!

Agenda

- Introduction to Word Senses & Semantics
- Classic Word Representations
- Vector Space Model Basics

Motivation: Representing Texts with Vectors

- Word similarity computation is important for understanding semantics

Word similarity (on a scale from 0 to 10) manually annotated by humans

vanish	disappear	9.8
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

Word semantics can be multi-faceted

	Valence	Arousal	Dominance
courageous	8.05	5.5	7.38
music	7.67	5.57	6.5
heartbreak	2.45	5.65	3.58
cub	6.71	3.95	4.24

- How to represent words numerically? Using multi-dimensional vectors!

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

Vector Semantics

- Represent a word as a point in a multi-dimensional semantic space
- A desirable vector semantic space: words with similar meanings are nearby in space

2D visualization of a desirable high-dimensional vector semantic space

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

Vector Space Basics

- Vector notation: an N-dimensional vector $\boldsymbol{v} = [v_1, v_2, \ldots, v_N] \in \mathbb{R}^N$
- Vector dot product/inner product:

$$\text{dot product}(\boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{v} \cdot \boldsymbol{w} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n = \sum_{i=1}^{N} v_i w_i$$

- Vector length/norm:

$$|\boldsymbol{v}| = \sqrt{\boldsymbol{v} \cdot \boldsymbol{v}} = \sqrt{\sum_{i=1}^{N} v_i^2}$$

Other (less commonly-used) vector norms:
Manhattan norm, *p*-norm, infinity norm…

- Cosine similarity between vectors:

$$\cos(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{|\boldsymbol{v}||\boldsymbol{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

Vector Space Basics: Example

- Consider two 4-dimensional vectors $\boldsymbol{v} = [1, 0, 1, 0] \in \mathbb{R}^4 \quad \boldsymbol{w} = [0, 1, 1, 0] \in \mathbb{R}^4$
- Vector dot product/inner product:

$$\boldsymbol{v} \cdot \boldsymbol{w} = \sum_{i=1}^{N} v_i w_i = 1$$

- Vector length/norm:

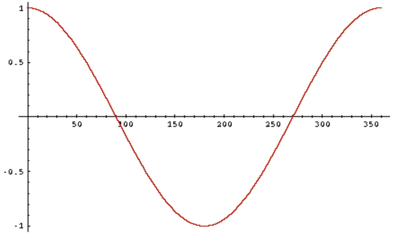
$$|\boldsymbol{v}| = \sqrt{\sum_{i=1}^{N} v_i^2} = \sqrt{2} \quad |\boldsymbol{w}| = \sqrt{\sum_{i=1}^{N} w_i^2} = \sqrt{2}$$

- Cosine similarity between vectors:

$$\cos(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{|\boldsymbol{v}||\boldsymbol{w}|} = \frac{1}{2}$$

Vector Similarity

- Cosine similarity is the most commonly used metric for similarity measurement
  - Symmetric: $\cos(\boldsymbol{v}, \boldsymbol{w}) = \cos(\boldsymbol{w}, \boldsymbol{v})$
  - Not influenced by vector length
  - Has a normalized range: [-1, 1]
  - Intuitive geometric interpretation



Cosine function values under different angles

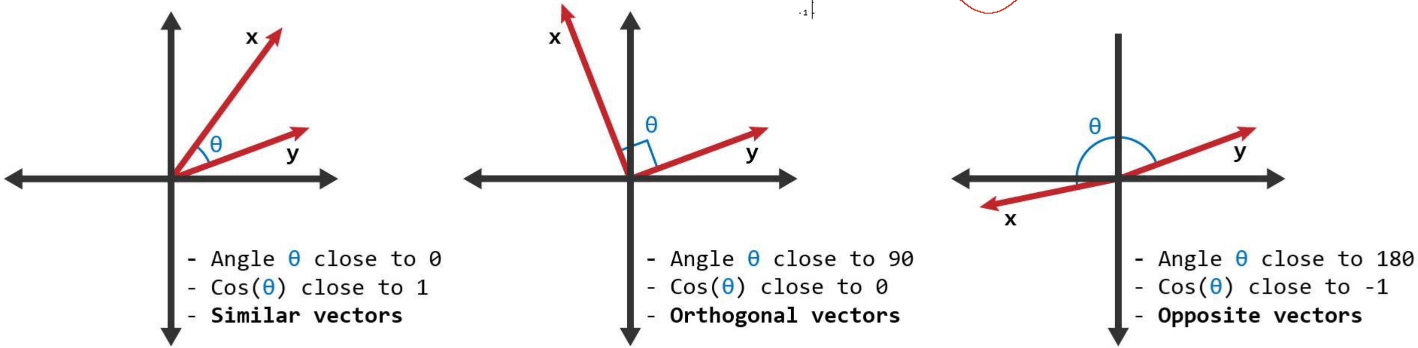


Figure source: https://www.learndatasci.com/glossary/cosine-similarity/

How to Represent Words as Vectors?

- Given a vocabulary $\mathcal{V} = \{\text{good}, \text{feel}, \text{I}, \text{sad}, \text{cats}, \text{have}\}$
- Most straightforward way to represent words as vectors: use their indices
- One-hot vector: only one high value (1) and the remaining values are low (0)
- Each word is identified by a unique dimension

$$
\begin{aligned}
\boldsymbol{v}_{\text{good}} &= [1, 0, 0, 0, 0, 0] \\
\boldsymbol{v}_{\text{feel}} &= [0, 1, 0, 0, 0, 0] \\
\boldsymbol{v}_{\text{I}} &= [0, 0, 1, 0, 0, 0] \\
\boldsymbol{v}_{\text{sad}} &= [0, 0, 0, 1, 0, 0] \\
\boldsymbol{v}_{\text{cats}} &= [0, 0, 0, 0, 1, 0] \\
\boldsymbol{v}_{\text{have}} &= [0, 0, 0, 0, 0, 1]
\end{aligned}
$$

Represent Sequences by Word Occurrences

- Consider the mini-corpus with three documents

$$
\begin{aligned}
d_1 &= \text{“I feel good”} \\
d_2 &= \text{“I feel sad”} \\
d_3 &= \text{“I have cats”}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{v}_{\text{good}} &= [1, 0, 0, 0, 0, 0] \\
\boldsymbol{v}_{\text{feel}} &= [0, 1, 0, 0, 0, 0] \\
\boldsymbol{v}_{\text{I}} &= [0, 0, 1, 0, 0, 0] \\
\boldsymbol{v}_{\text{sad}} &= [0, 0, 0, 1, 0, 0] \\
\boldsymbol{v}_{\text{cats}} &= [0, 0, 0, 0, 1, 0] \\
\boldsymbol{v}_{\text{have}} &= [0, 0, 0, 0, 0, 1]
\end{aligned}
$$

- Straightforward way of representing documents: look at which words are present

$$
\begin{aligned}
\boldsymbol{v}_{d_1} &= [1, 1, 1, 0, 0, 0] \\
\boldsymbol{v}_{d_2} &= [0, 1, 1, 1, 0, 0] \\
\boldsymbol{v}_{d_3} &= [0, 0, 1, 0, 1, 1]
\end{aligned}
$$

Document vector similarity ➡

$$
\begin{aligned}
\cos(\boldsymbol{v}_{d_1}, \boldsymbol{v}_{d_2}) &= \frac{2}{3} \\
\cos(\boldsymbol{v}_{d_1}, \boldsymbol{v}_{d_3}) &= \frac{1}{3} \\
\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) &= \frac{1}{3}
\end{aligned}
$$

Term-Document Matrix

- With larger text collections, word frequencies in documents entail rich information
- Consider the four plays by Shakespeare and obtain the word frequency statistics
- Look at 4 manually-picked words: “battle” “good” “fool” “wit”

	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

There are many more words!

- Document vector representation with word frequencies:

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

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

Document Similarity

- Document vector representation with word frequencies:

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

	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 $\cos(\boldsymbol{v}_{d_1}, \boldsymbol{v}_{d_2}) = 0.95$ $\cos(\boldsymbol{v}_{d_2}, \boldsymbol{v}_{d_3}) = 0.81$
- Word frequencies in documents do reflect the semantic similarity between documents!

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

Words Represented with Documents

- "Battle": "the kind of word that occurs in Julius Caesar and Henry V (history plays)"
- "Fool": "the kind of word that occurs in comedies"

	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

- Represent words using their co-occurrence counts with documents:

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

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

Fine-Grained Contexts: Word-Word Matrix

Instead of using documents as contexts for words, we can also use words as contexts

4 words to the left	center word	4 words to the right
is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

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

Fine-Grained Contexts: Word-Word Matrix

Count how many times words occur in a ±4 word window around the center word

context word

center word

	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	...

Counts derived from the Wikipedia corpus

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

Word Similarity Based on 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!

Is Raw Frequency A Good Representation?

- 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?

Term Frequency (TF)

- A word appearing 100 times in a document doesn't make it 100 times more likely to be relevant to the meaning of the document
- Instead of using the raw counts, we squash the counts with log scale

$$\mathrm{TF}(w,d) = \begin{cases} 1 + \log_{10} \mathrm{count}(w,d) & \mathrm{count}(w,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Document Frequency (DF)

- Motivation: Give a higher weight to words that occur only in a few documents
  - Terms that are limited to a few documents are more discriminative
  - Terms that occur frequently across the entire collection aren't as helpful
- Document frequency (DF): count how many documents a word occurs in

$$\mathrm{DF}(w) = \sum_{i=1}^{N} \mathbb{1}(w \in d_i)$$

Evaluates to 1 if w occurs in d_i
otherwise evaluates to 0

- DF is NOT defined to be the total count of a word across all documents (collection frequency)!

	Collection Frequency	Document Frequency
Romeo	113	1
action	113	31

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

Inverse Document Frequency (IDF)

- We want to emphasize discriminative words (with low DF)
- Inverse document frequency (IDF): total number of documents (N) divided by DF, in log scale

$$\mathrm{IDF}(w) = \log_{10}\left(\frac{N}{\mathrm{DF}(w)}\right)$$

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

DF & IDF statistics in the Shakespeare corpus

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

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

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

UNIVERSITY of VIRGINIA

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)?

Thank You!

Yu Meng
University of Virginia
yumeng5@virginia.edu