



Introduction to Language Modeling & N-gram Language Models

Yu Meng

University of Virginia

yumeng5@virginia.edu

Sep 02, 2024



Announcement: Assignment 1 Out

- Deadline: 09/11 11:59pm
- Released on course website: <https://yumeng5.github.io/teaching/2024-fall-cs4501>

Week	Date	Topic	Slides	Slido Link	Deadline
1	08/28	Course Logistics & Overview	overview_0828	08/28	
	08/30	Course Overview (Continued)	overview_0830	08/30	
2	09/02	Intro to Language Modeling & N-gram Language Models	lm_intro_0902	09/02	Assignment 1 out: LaTeX script

Download the LaTeX script here

Overview of Course Contents

Join at

slido.com

#7035 481



- 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

Agenda

- Introduction to Language Models
- N-gram Language Models
- Smoothing in N-gram Language Models
- Evaluation of Language Models

Join at

slido.com

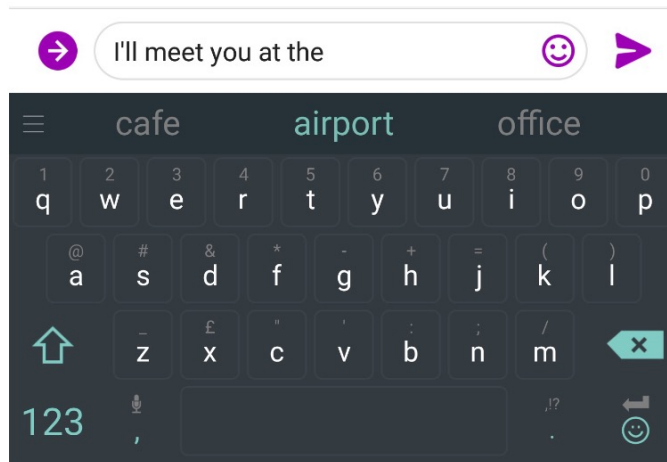
#7035 481





Overview: Language Modeling

- The core problem in NLP is **language modeling**
- Goal: Assigning probability to a sequence of words
- For text understanding: $p(\text{"The cat is on the mat"}) \gg p(\text{"Truck the earth on"})$
- For text generation: $p(w \mid \text{"The cat is on the"}) \rightarrow \text{"mat"}$



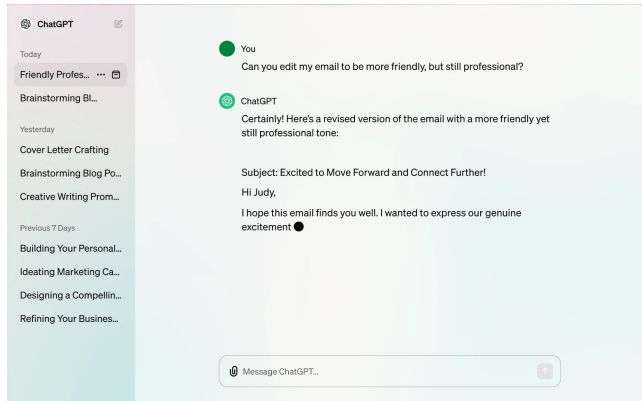
Autocomplete empowered by
language modeling

Language Model Applications

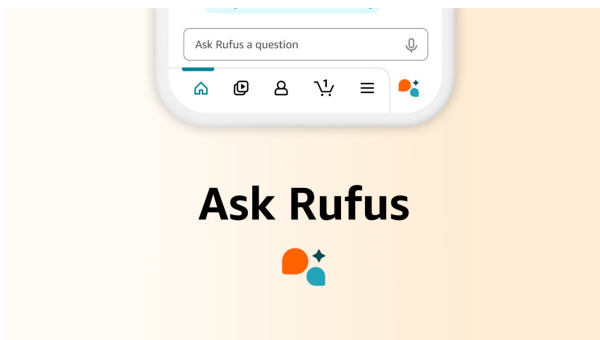
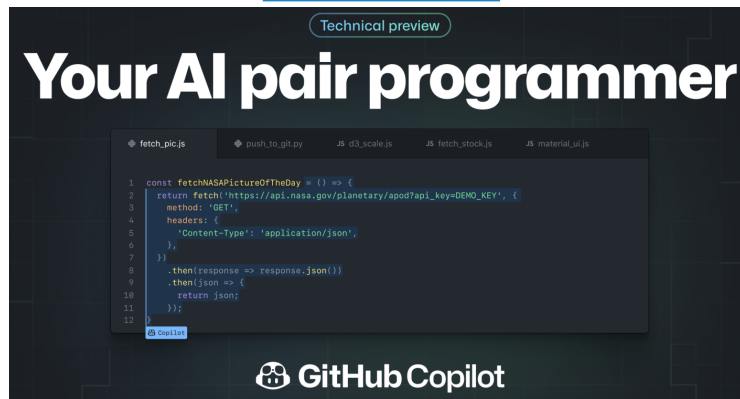
Join at
slido.com
#7035 481



Chatbots



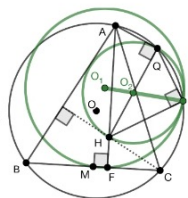
Code Assistants



Shopping Assistants

e IMO 2015 P3

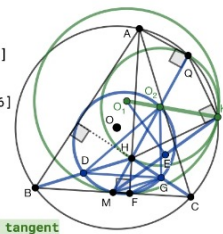
"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A . Let M be the midpoint of BC . Let Q be the point on (O) such that $QH \perp QA$ and let K be the point on (O) such that $KH \perp KQ$. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."



Alpha-Geometry

f Solution

Construct D : midpoint BH [a]
 $[a], O_2$ midpoint $HQ \Rightarrow BQ \parallel O_2D$ [20]
 ...
 Construct G : midpoint HC [b] ...
 $\angle GMD = \angle GO_2D \Rightarrow M, O_2, G, D$ cyclic [26]
 ...
 $[a], [b] \Rightarrow BC \parallel DG$ [30]
 ...
 Construct E : midpoint MK [c]
 ..., [c] $\Rightarrow \angle KFC = \angle KO_1E$ [104]
 ...
 $\angle FKO_1 = \angle FKO_2 \Rightarrow KO_1 \parallel KO_2$ [109]
 [109] $\Rightarrow O_1, O_2, K$ collinear $\Rightarrow (O_1), (O_2)$ tangent



Generating Math Proofs



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!



Language Modeling: Probability Decomposition

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

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

- Are there other possible decomposition assumptions?
 - Yes, but they are not considered “conventional” language models
 - We’ll see in word embedding/BERT lectures



Language Modeling: Probability Decomposition

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

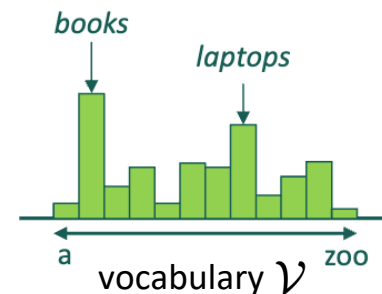
$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \cdots p(x_n|x_1, \dots, x_{n-1}) = \prod_{i=1}^n p(x_i|x_1, \dots, x_{i-1})$$

- How to guarantee the probability distributions are valid?
 - Non-negative

$$p(x_i = w|x_1, \dots, x_{i-1}) \geq 0, \quad \forall w \in \mathcal{V}$$

- Summed to 1:

$$\sum_{w \in \mathcal{V}} p(x_i = w|x_1, \dots, x_{i-1}) = 1$$

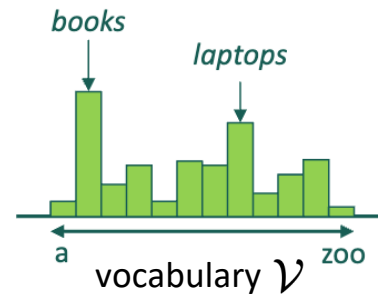


- The goal of language modeling is to learn the distribution $p(x_i = w|x_1, \dots, x_{i-1})$!



Language Models Are Generative Models

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






Example: Language Models for Generation


- Recursively sample $x_i \sim p(w|x_1, \dots, x_{i-1})$ until we generate [EOS]
- Generate the first word: “the” $\leftarrow x_1 \sim p(w|[\text{BOS}])$ **beginning-of-sequence**
- Generate the second word: “cat” $\leftarrow x_2 \sim p(w|“the”)$
- Generate the third word: “is” $\leftarrow x_3 \sim p(w|“the cat”)$
- Generate the fourth word: “on” $\leftarrow x_4 \sim p(w|“the cat is”)$
- Generate the fifth word: “the” $\leftarrow x_5 \sim p(w|“the cat is on”)$
- Generate the sixth word: “mat” $\leftarrow x_6 \sim p(w|“the cat is on the”)$
- Generate the seventh word: [EOS] $\leftarrow x_7 \sim p(w|“the cat is on the mat”)$
- Generation finished!

How to Obtain A Language Model?

Join at
slido.com
#7035 481



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




English
6,872,000+ articles

日本語
1,427,000+ 記事

Deutsch
2.937.000+ Artikel

Français
2 631 000+ articles

Português
1.132.000+ artigos



中文
1,437,000+ 条目 / 條目

Русский
1 996 000+ статей

Español
1.974.000+ artículos

Italiano
1.878.000+ voci

فارسی
۱۰۰۱۱۰۰۰+ مقاله

Donald Trump 242 languages

Article Talk Read View source View history Tools

Joe Biden 218 languages

Article Talk Read View source View history Tools


From Wikipedia, the free encyclopedia

"Joseph Biden" and "Biden" redirect here. For his first-born son, Joseph Biden III, see Beau Biden. For other uses, see Biden (disambiguation).

Joseph Robinette Biden Jr. ^[pl] (born November 20, 1942) is an American politician serving as the 46th and current president of the United States since 2021. A member of the Democratic Party, he served as the 47th vice president from 2009 to 2017 under President Barack Obama and represented Delaware in the U.S. Senate from 1973 to 2009.

Born in Scranton, Pennsylvania, Biden moved with his family to Delaware in 1953. He graduated from the University of Delaware in 1965 and from Syracuse University in 1968. He was elected to the New Castle County Council in 1970 and the U.S. Senate in 1972. As a senator, Biden drafted and led the effort to pass the Violent Crime Control and Law Enforcement Act and the Violence Against Women Act. He also oversaw six U.S. Supreme Court confirmation hearings, including the contentious hearings for Robert Bork and Clarence Thomas. Biden ran unsuccessfully for the 1988 and 2008 Democratic presidential nominations. In 2008, Obama chose Biden as his running mate, and he was a close counselor to Obama during his two terms as vice president. In the 2020 presidential election, the Democratic Party nominated Biden for president. He selected Kamala Harris as his running mate, and they defeated Republican incumbents Donald Trump and Mike Pence. He is the oldest president in U.S. history and the first to have a female vice president.

As president, Biden signed the American Rescue Plan Act in response to the COVID-19 pandemic and subsequent recession. He signed bipartisan bills on infrastructure and manufacturing. He proposed the Build Back Better Act, which failed in Congress, but aspects of which were incorporated into the Inflation Reduction Act that he signed into law in 2022. Biden appointed Ketanji Brown Jackson to the Supreme Court. He worked with congressional Republicans to resolve the 2023 debt-ceiling crisis by negotiating a deal to raise the debt ceiling. In foreign policy, Biden restored America's membership in the Paris Agreement. He oversaw the complete withdrawal of U.S. troops from Afghanistan that ended the war in Afghanistan, leading to the collapse of the Afghan government and the Taliban seizing control. He responded to the Russian invasion of



Joe Biden

Official portrait, 2021

46th President of the United States

Incumbent

Assumed office
January 20, 2021

Vice President
Kamala Harris

Preceded by
Donald Trump

47th Vice President of the United States

In office
January 20, 2009 – January 20, 2017

President
Barack Obama


Preceded by
Dick Cheney

Succeeded by
Mike Pence

United States Senator from Delaware

In office
January 3, 1973 – January 15, 2009

Preceded by
J. Caleb Boggs



Trump

trait, 2017

the United States

vice
- January 20, 2021

ence

Obama

fen

i details

John Trump

4, 1946 (age 78)

is, New York City, U.S.

ican (1987–1999, 2011, 2012–present)

(1999–2001)

rate (2001–2009)

ndent (2011–2012)

Seitšková

77; div. 1990)

Learning target:

→ $p(w|x_1, \dots, x_{i-1})$

Text corpora contain rich distributional statistics!

12/38



History of Language Models

- Language models started to be built with statistical methods
 - Sparsity
 - Poor generalization

Weeks 2-3

Before 2000s

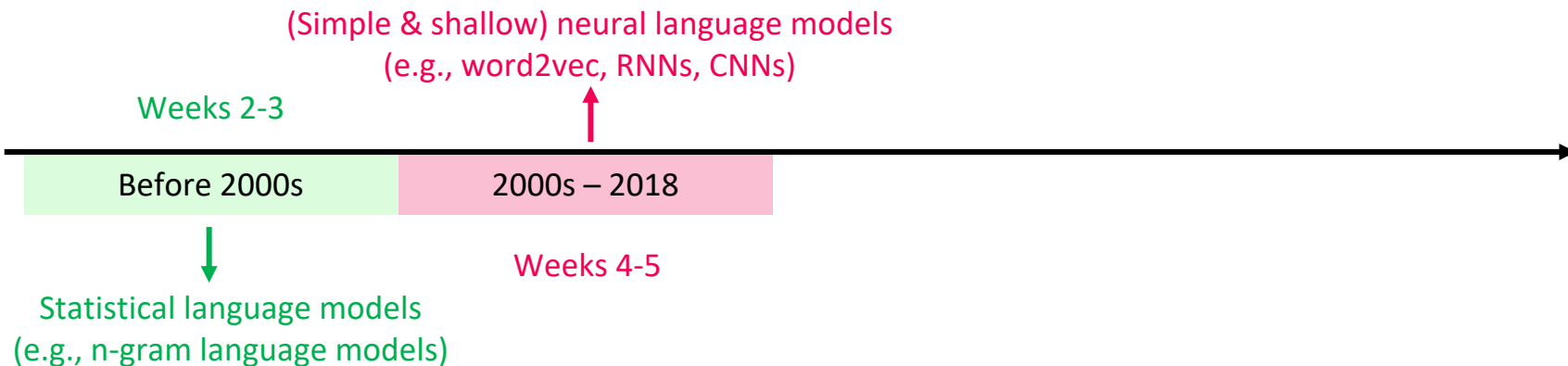


Statistical language models
(e.g., n-gram language models)



History of Language Models

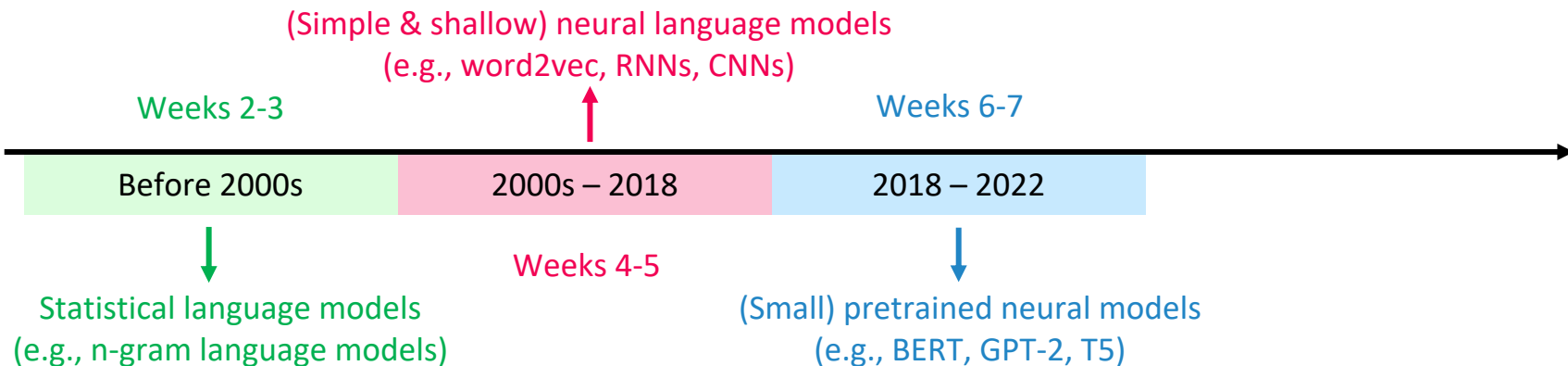
- The introduction of neural networks into language models mitigated sparsity and improved generalization
 - Neural networks for language models were small-scale and inefficient for a long time
 - Task-specific architecture designs required for different NLP tasks
 - These language models were trained on individual NLP tasks as task-specific solvers





History of Language Models

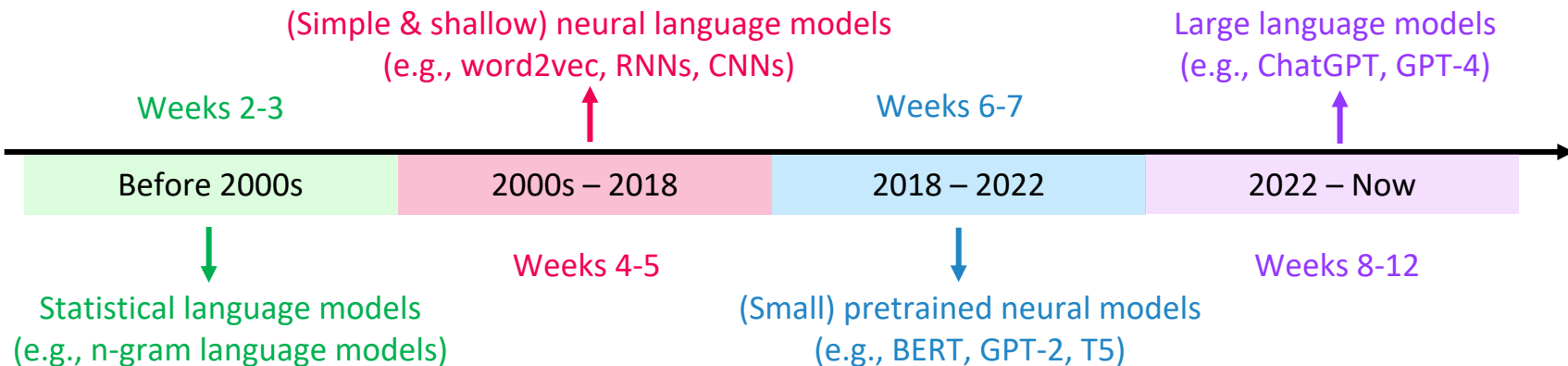
- Transformer became the dominant architecture for language modeling; scaling up model sizes and (pretraining) data enabled significant generalization ability
 - Transformer demonstrated striking scalability and efficiency in sequence modeling
 - One pretrained model checkpoint fine-tuned to become strong task-specific models
 - Task-specific fine-tuning was still necessary





History of Language Models

- Generalist large language models (LLMs) became the universal task solvers and replaced task-specific language models
 - Real-world NLP applications are usually multifaceted (require composite task abilities)
 - Tasks are not clearly defined and may overlap
 - Single-task models struggle to handle complex tasks



Agenda

- Introduction to Language Models
- N-gram Language Models
- Smoothing in N-gram Language Models
- Evaluation of Language Models

Join at

slido.com

#7035 481





N-gram Language Model: Simplified Assumption

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

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

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

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

N-gram assumption

Should N be larger or smaller?



N-gram Language Model: Simplified Assumption

- Unigram LM (N=1): each word's probability does not depend on previous words
- Bigram LM (N=2): each word's probability is based on the previous word
- Trigram LM (N=3): each word's probability is based on the previous two words
- ...
- Example: $p(\text{"The cat is on the mat"})$ For simplicity, omitting [BOS] & [EOS] in these examples
- Unigram: $= p(\text{"The"}) p(\text{"cat"}) p(\text{"is"}) p(\text{"on"}) p(\text{"the"}) p(\text{"mat"})$
- Bigram: $= p(\text{"The"}) p(\text{"cat"} | \text{"The"}) p(\text{"is"} | \text{"cat"}) p(\text{"on"} | \text{"is"}) p(\text{"the"} | \text{"on"}) p(\text{"mat"} | \text{"the"})$
- Trigram: $= p(\text{"The"}) p(\text{"cat"} | \text{"The"}) p(\text{"is"} | \text{"The cat"}) p(\text{"on"} | \text{"cat is"}) p(\text{"the"} | \text{"is on"}) p(\text{"mat"} | \text{"on the"})$
- ...



How to Learn N-grams?

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

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

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

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