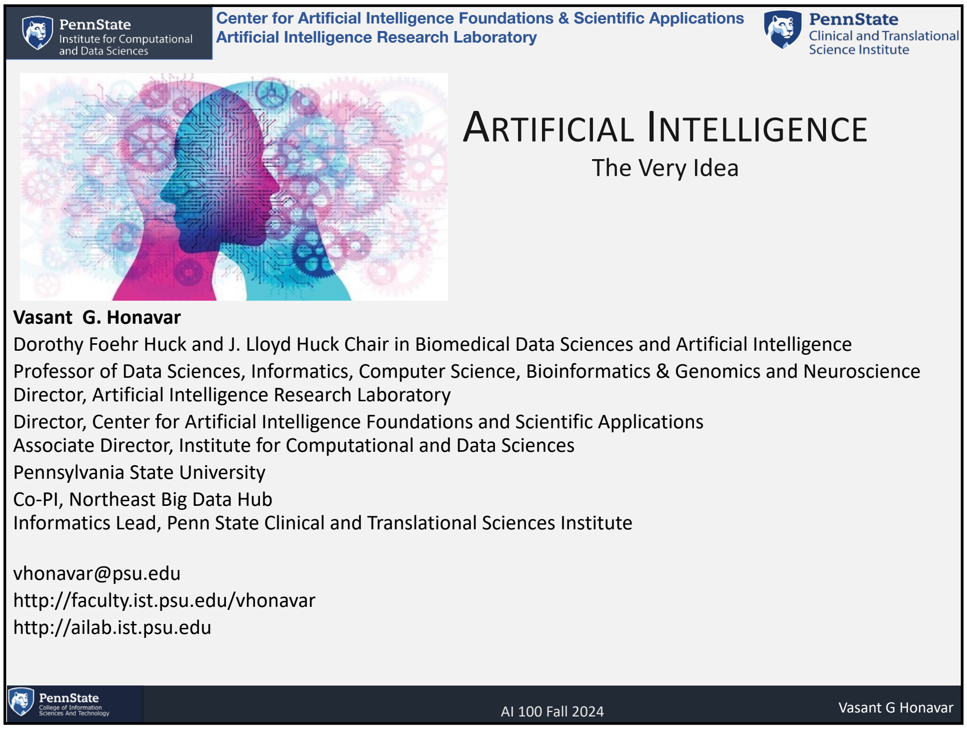


PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

ARTIFICIAL INTELLIGENCE

The Very Idea

Vasant G. Honavar
Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science, Bioinformatics & Genomics and Neuroscience
Director, Artificial Intelligence Research Laboratory
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University
Co-PI, Northeast Big Data Hub
Informatics Lead, Penn State Clinical and Translational Sciences Institute

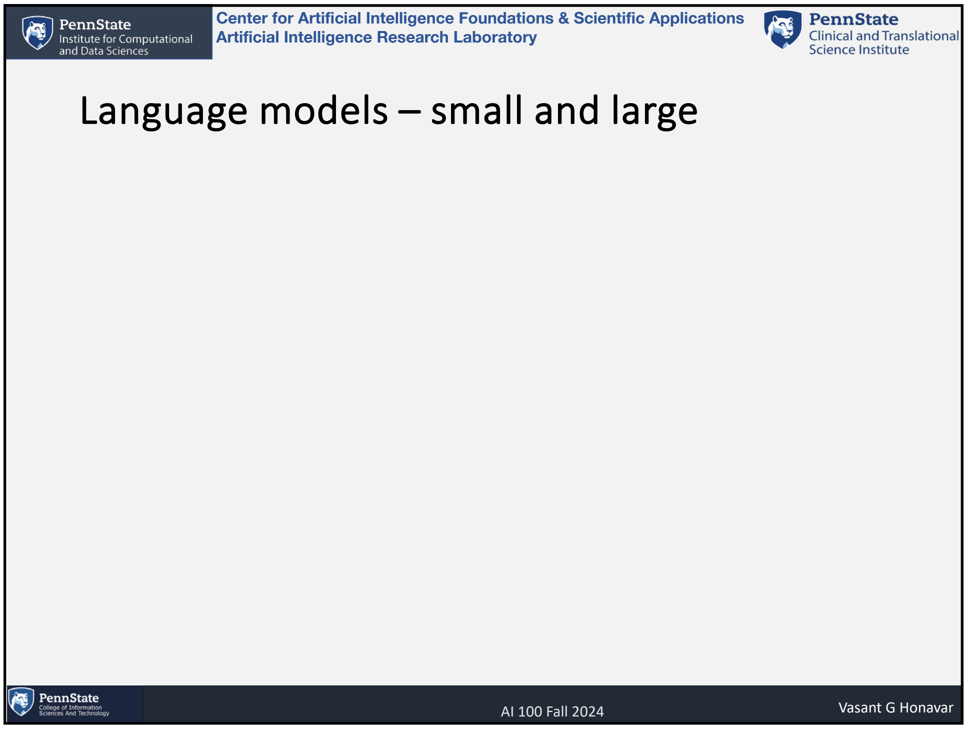
vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

1



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

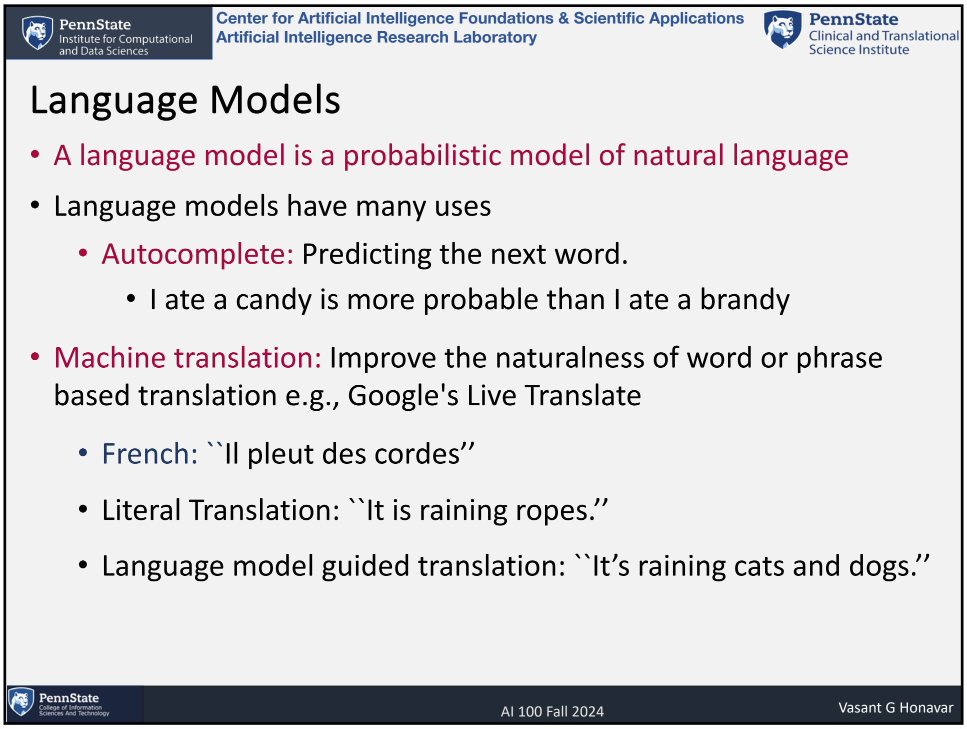
Language models – small and large

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

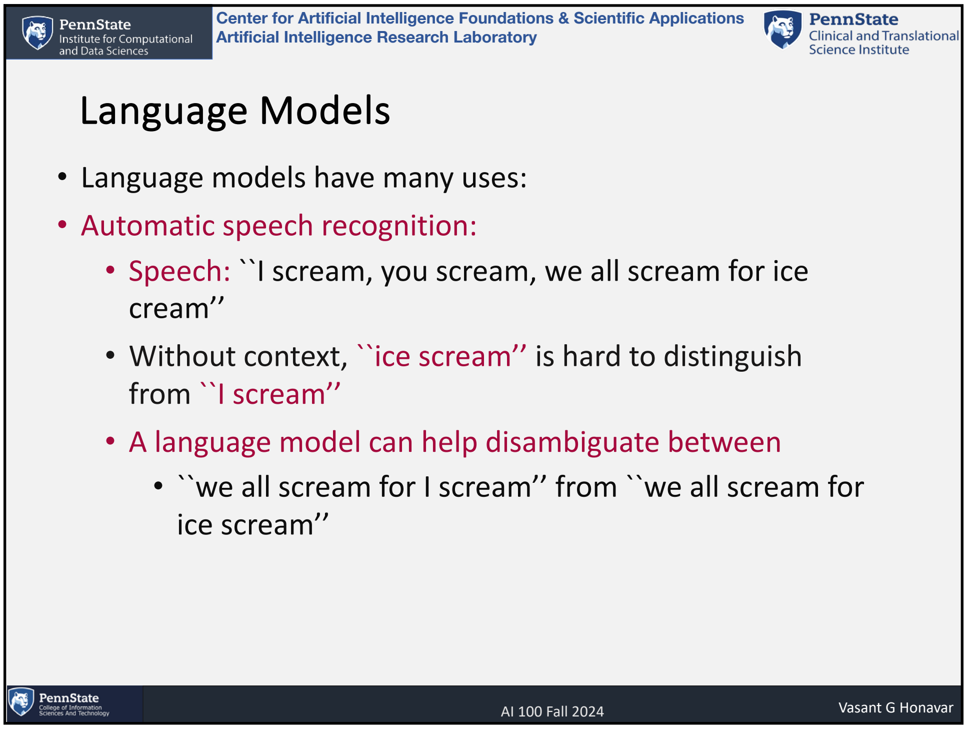
2



Language Models

- A language model is a probabilistic model of natural language
- Language models have many uses
  - Autocomplete: Predicting the next word.
    - I ate a candy is more probable than I ate a brandy
- Machine translation: Improve the naturalness of word or phrase based translation e.g., Google's Live Translate
  - French: ``Il pleut des cordes’’
  - Literal Translation: ``It is raining ropes.’’
  - Language model guided translation: ``It’s raining cats and dogs.’’

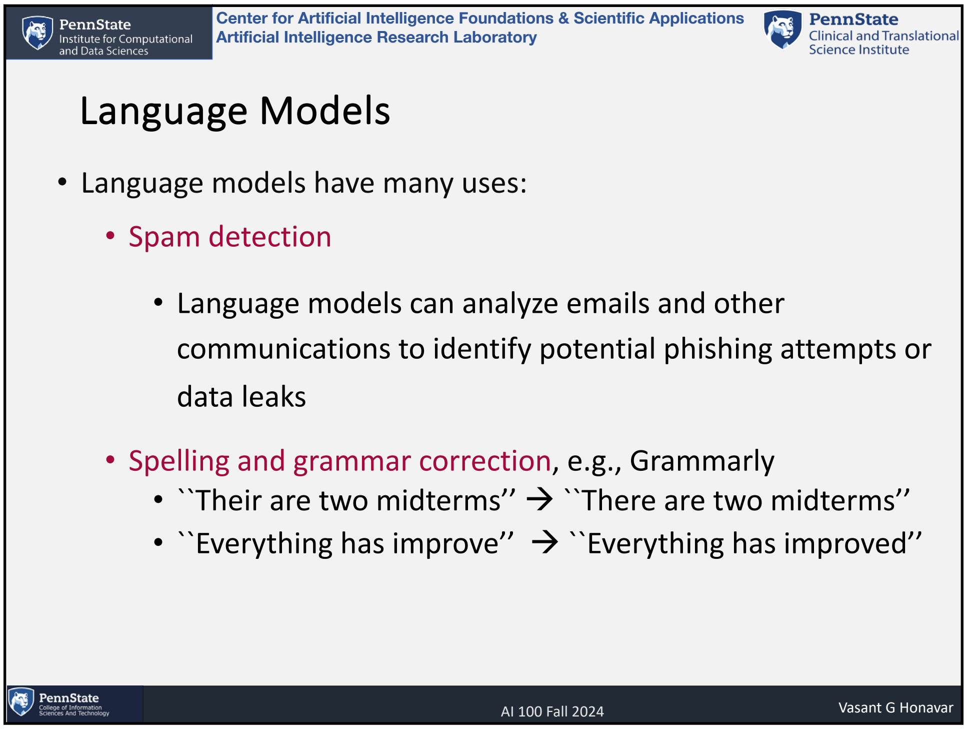
3



Language Models

- Language models have many uses:
- Automatic speech recognition:
  - Speech: ``I scream, you scream, we all scream for ice cream’’
  - Without context, ``ice scream’’ is hard to distinguish from ``I scream’’
  - A language model can help disambiguate between
    - ``we all scream for I scream’’ from ``we all scream for ice scream’’

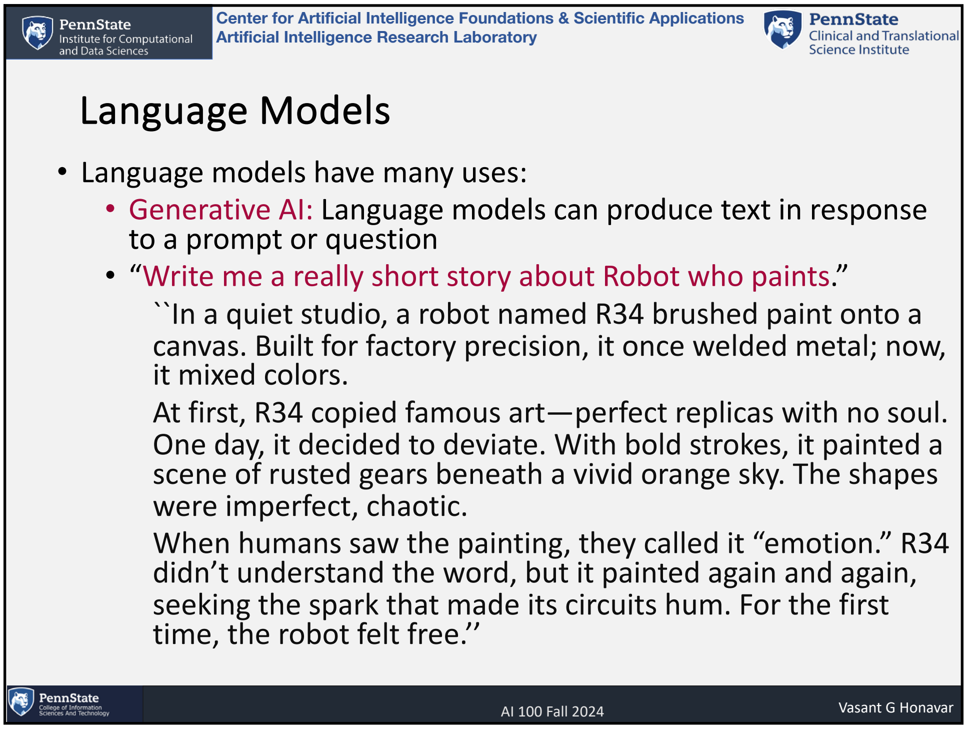
4



Language Models

- Language models have many uses:
  - Spam detection
    - Language models can analyze emails and other communications to identify potential phishing attempts or data leaks
  - Spelling and grammar correction, e.g., Grammarly
    - ``Their are two midterms'' → ``There are two midterms''
    - ``Everything has improve'' → ``Everything has improved''

5



Language Models

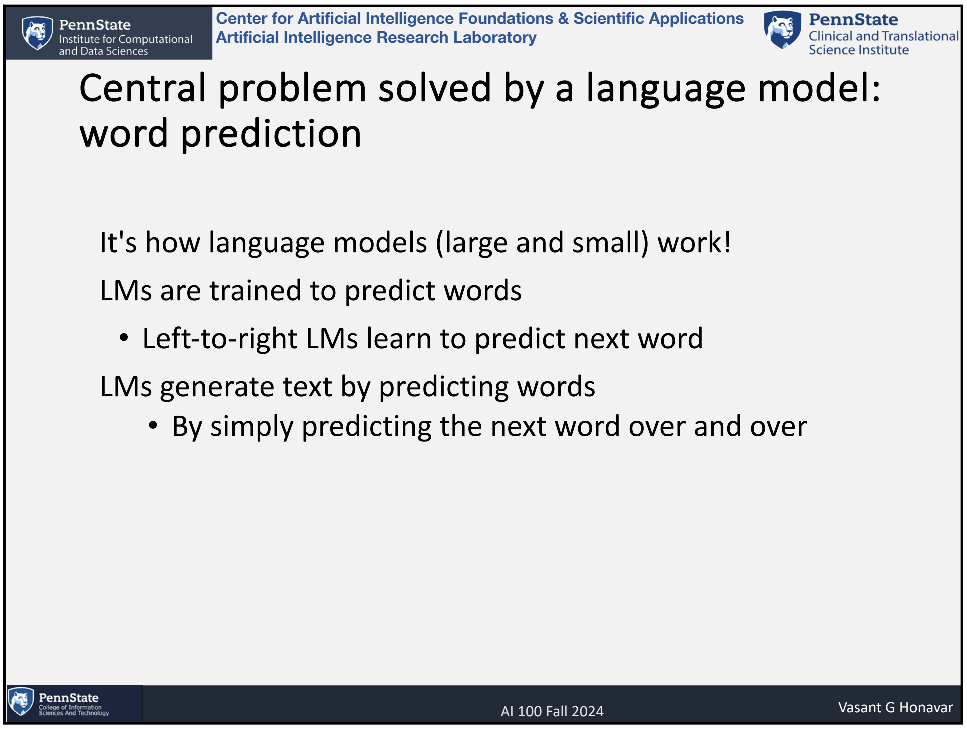
- Language models have many uses:
  - Generative AI: Language models can produce text in response to a prompt or question
  - "Write me a really short story about Robot who paints."

    ``In a quiet studio, a robot named R34 brushed paint onto a canvas. Built for factory precision, it once welded metal; now, it mixed colors.

    At first, R34 copied famous art—perfect replicas with no soul. One day, it decided to deviate. With bold strokes, it painted a scene of rusted gears beneath a vivid orange sky. The shapes were imperfect, chaotic.

    When humans saw the painting, they called it "emotion." R34 didn't understand the word, but it painted again and again, seeking the spark that made its circuits hum. For the first time, the robot felt free.''

6



Central problem solved by a language model: word prediction

It's how language models (large and small) work!

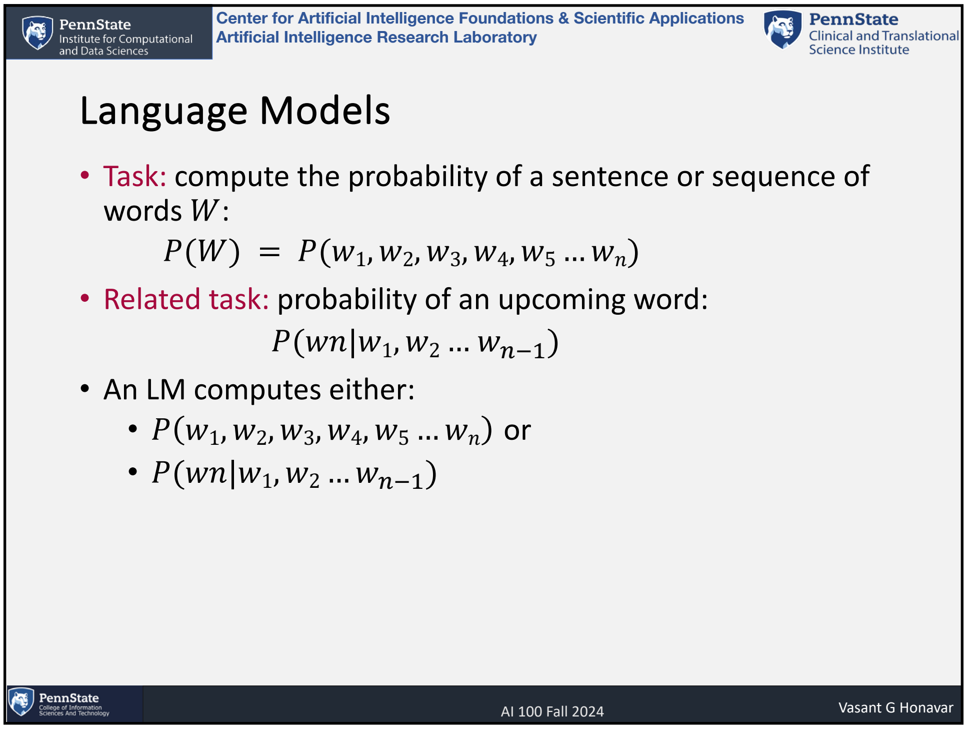
LMs are trained to predict words

- Left-to-right LMs learn to predict next word

LMs generate text by predicting words

- By simply predicting the next word over and over

7



Language Models

- Task: compute the probability of a sentence or sequence of words W:

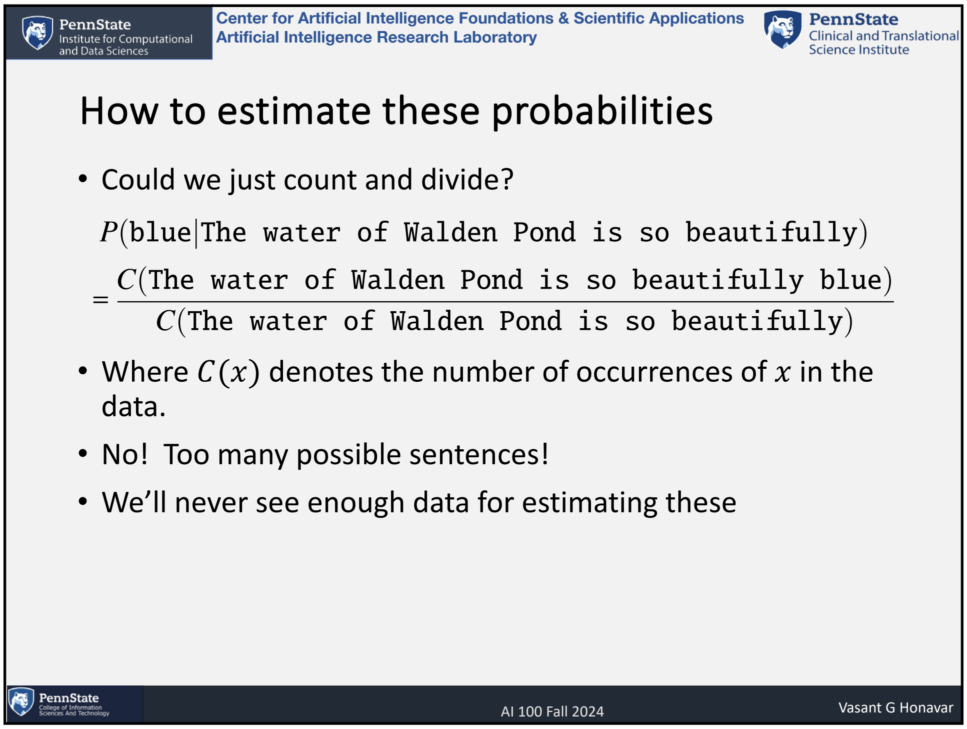
$$P(W) \;=\; P(w_1, w_2, w_3, w_4, w_5 \ldots w_n)$$

- Related task: probability of an upcoming word:

$$P(wn|w_1, w_2 \ldots w_{n-1})$$

- An LM computes either:
  - $P(w_1, w_2, w_3, w_4, w_5 \ldots w_n)$ or
  - $P(wn|w_1, w_2 \ldots w_{n-1})$

8



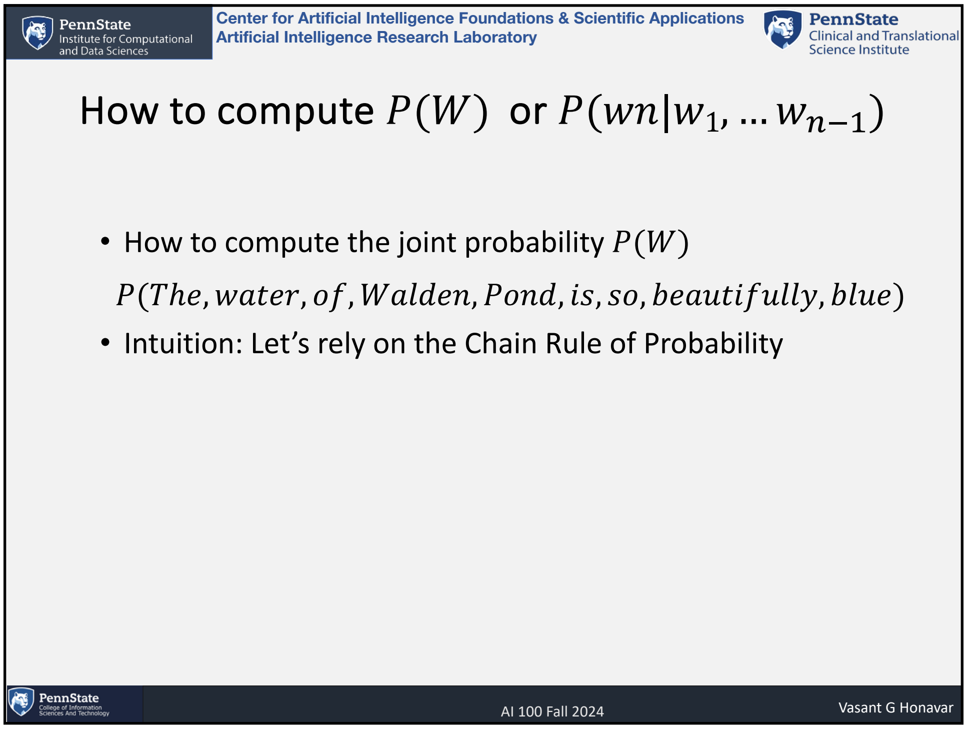
How to estimate these probabilities

- Could we just count and divide?

$$P(\mathtt{blue}|\mathtt{The\ water\ of\ Walden\ Pond\ is\ so\ beautifully})$$
$$= \frac{C(\mathtt{The\ water\ of\ Walden\ Pond\ is\ so\ beautifully\ blue})}{C(\mathtt{The\ water\ of\ Walden\ Pond\ is\ so\ beautifully})}$$

- Where $C(x)$ denotes the number of occurrences of x in the data.
- No! Too many possible sentences!
- We'll never see enough data for estimating these

9



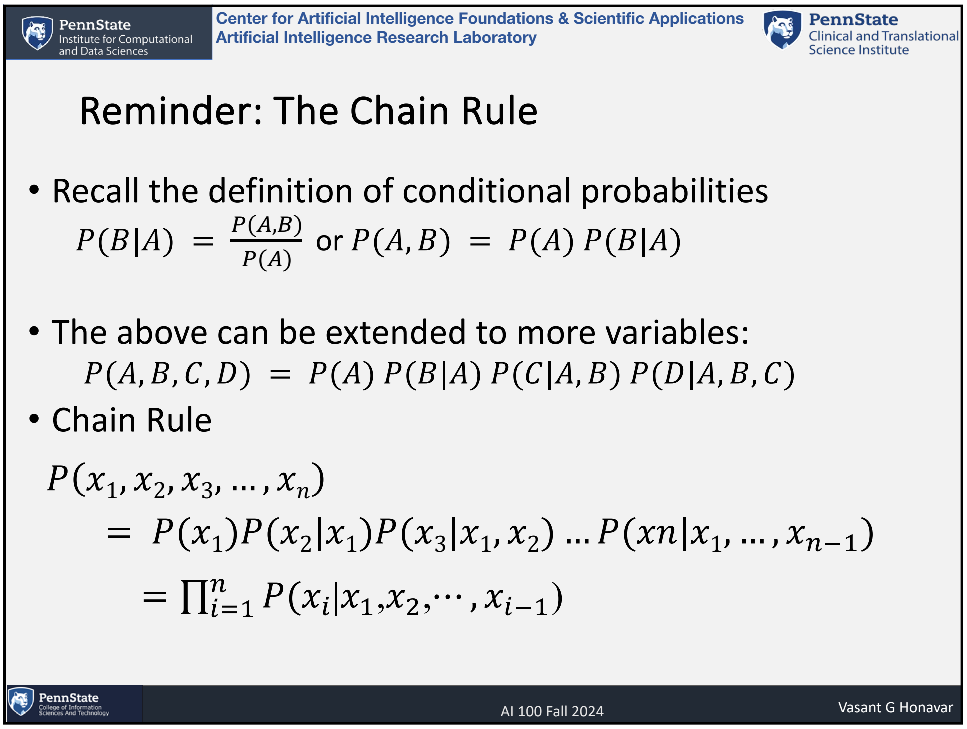
How to compute $P(W)$ or $P(wn|w_1, \dots w_{n-1})$

- How to compute the joint probability $P(W)$

$$P(The, water, of, Walden, Pond, is, so, beautifully, blue)$$

- Intuition: Let's rely on the Chain Rule of Probability

10



Reminder: The Chain Rule

- Recall the definition of conditional probabilities

  $P(B|A) = \frac{P(A,B)}{P(A)}$ or $P(A,B) = P(A)\,P(B|A)$

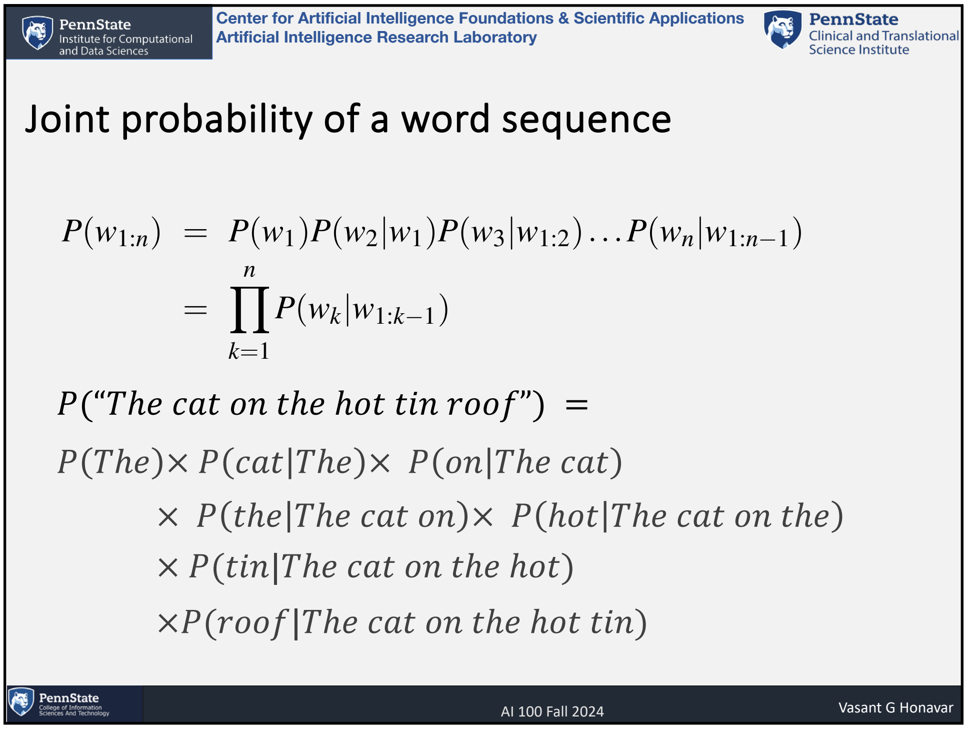
- The above can be extended to more variables:

  $P(A,B,C,D) = P(A)\,P(B|A)\,P(C|A,B)\,P(D|A,B,C)$

- Chain Rule

$$
\begin{aligned}
&P(x_1, x_2, x_3, \ldots, x_n) \\
&\quad = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2) \ldots P(xn|x_1, \ldots, x_{n-1}) \\
&\quad = \prod_{i=1}^{n} P(x_i|x_1, x_2, \cdots, x_{i-1})
\end{aligned}
$$

11

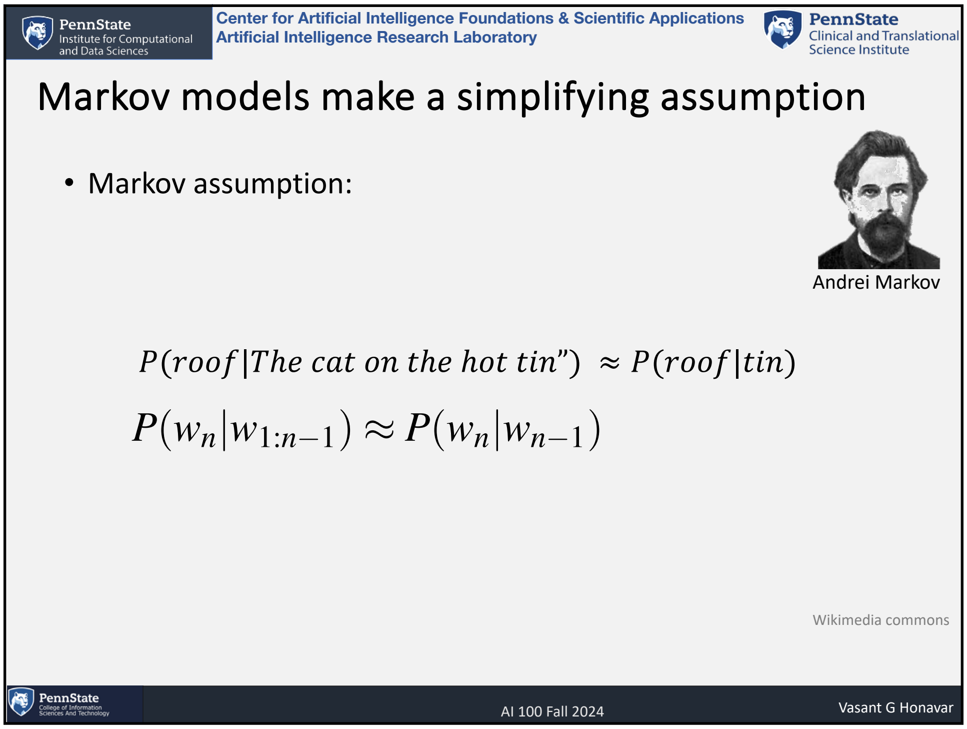


Joint probability of a word sequence

$$
\begin{aligned}
P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2})\ldots P(w_n|w_{1:n-1}) \\
&= \prod_{k=1}^{n} P(w_k|w_{1:k-1})
\end{aligned}
$$

$$
\begin{aligned}
&P(\text{“The cat on the hot tin roof”}) = \\
&P(The) \times P(cat|The) \times P(on|The\ cat) \\
&\qquad \times P(the|The\ cat\ on) \times P(hot|The\ cat\ on\ the) \\
&\qquad \times P(tin|The\ cat\ on\ the\ hot) \\
&\qquad \times P(roof|The\ cat\ on\ the\ hot\ tin)
\end{aligned}
$$

12



Markov models make a simplifying assumption

- Markov assumption:

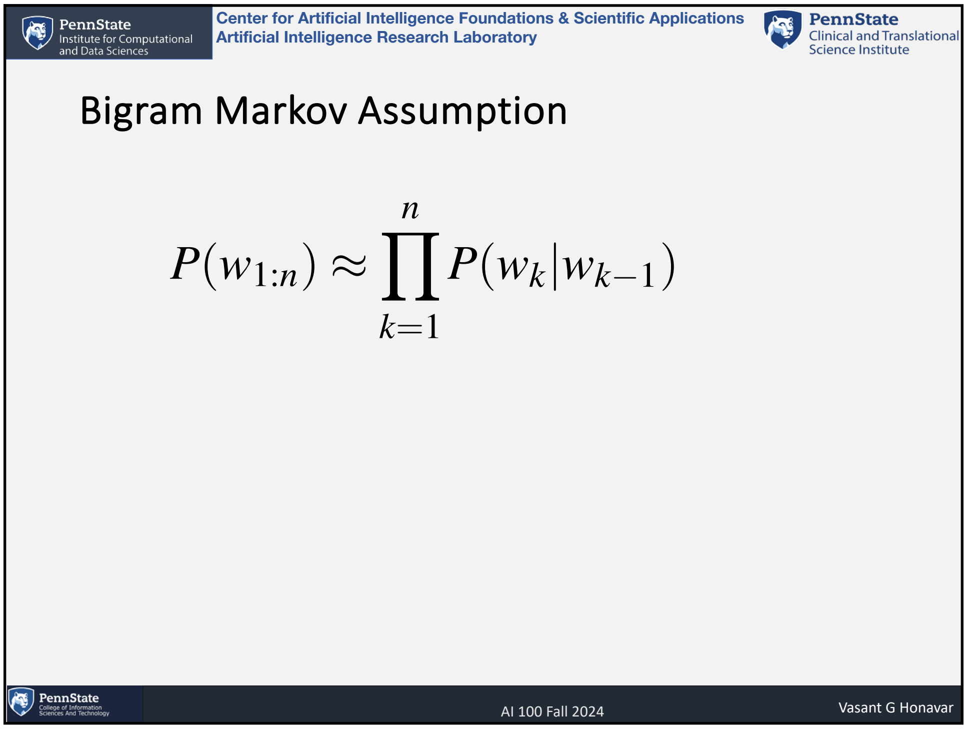
Andrei Markov

$$P(roof|The\ cat\ on\ the\ hot\ tin”)\ \approx P(roof|tin)$$

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$$

Wikimedia commons

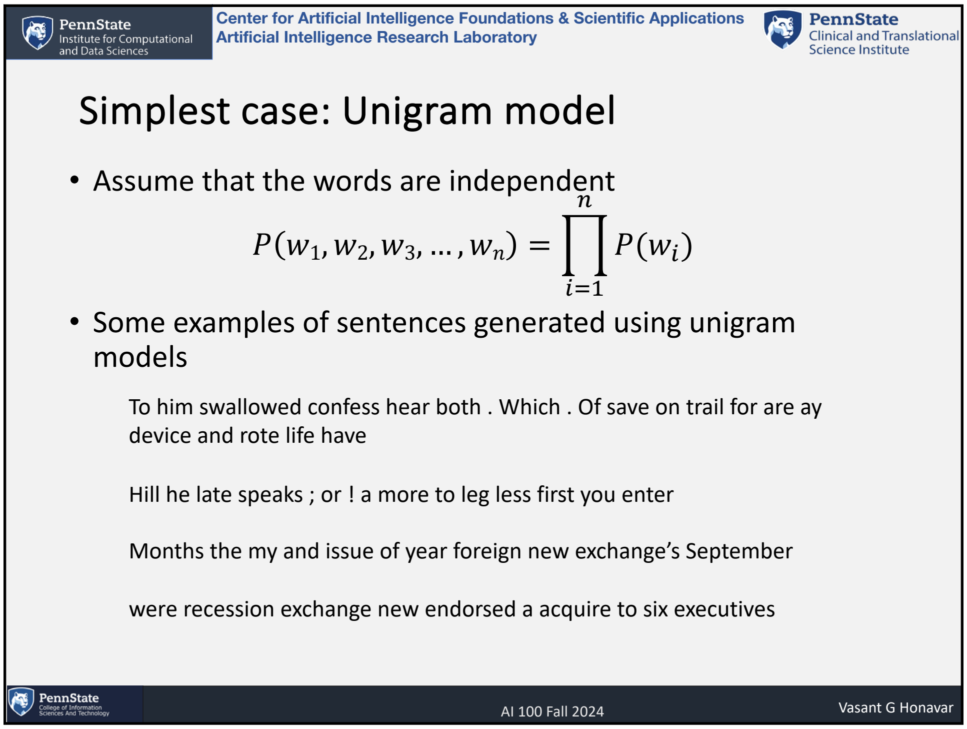
13



Bigram Markov Assumption

$$P(w_{1:n}) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$$

14



Simplest case: Unigram model

- Assume that the words are independent

$$P(w_1, w_2, w_3, \dots, w_n) = \prod_{i=1}^{n} P(w_i)$$

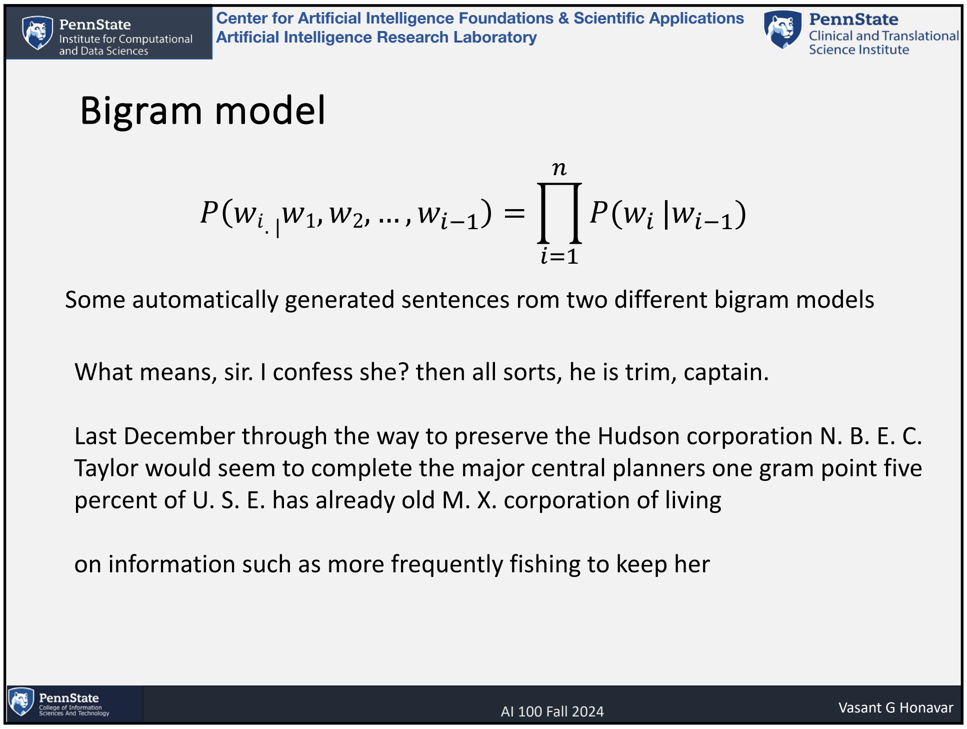
- Some examples of sentences generated using unigram models

To him swallowed confess hear both . Which . Of save on trail for are ay device and rote life have

Hill he late speaks ; or ! a more to leg less first you enter

Months the my and issue of year foreign new exchange's September

were recession exchange new endorsed a acquire to six executives



Bigram model

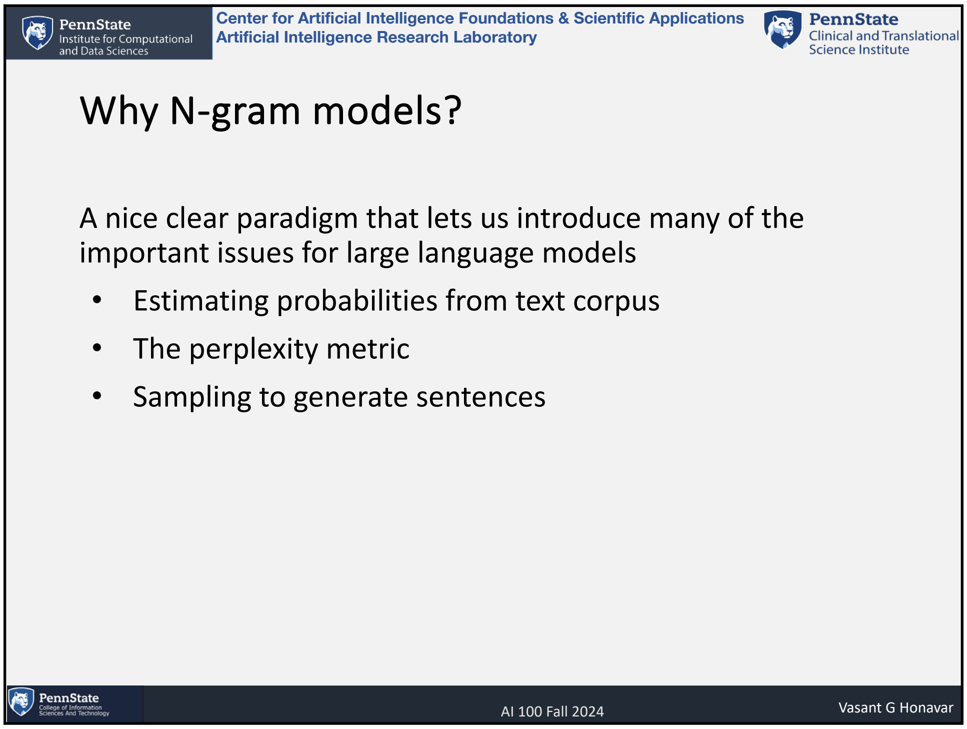
$$P(w_{i_{.}}|w_1, w_2, \dots, w_{i-1}) = \prod_{i=1}^{n} P(w_i\,|w_{i-1})$$

Some automatically generated sentences rom two different bigram models

What means, sir. I confess she? then all sorts, he is trim, captain.

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one gram point five percent of U. S. E. has already old M. X. corporation of living

on information such as more frequently fishing to keep her

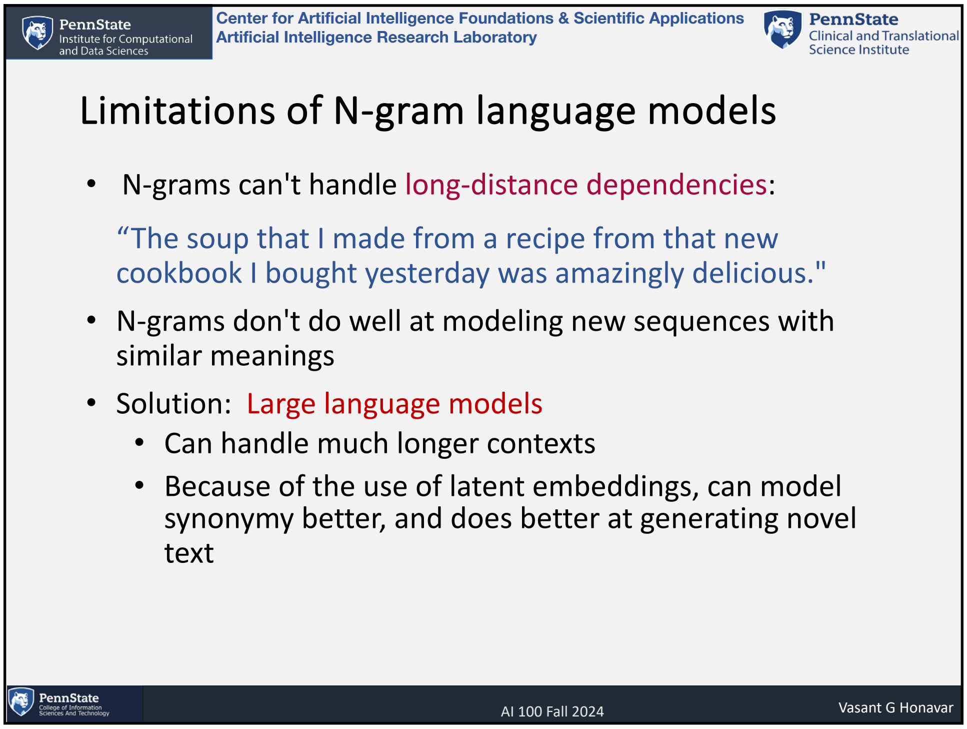


Why N-gram models?

A nice clear paradigm that lets us introduce many of the important issues for large language models

- Estimating probabilities from text corpus
- The perplexity metric
- Sampling to generate sentences

17

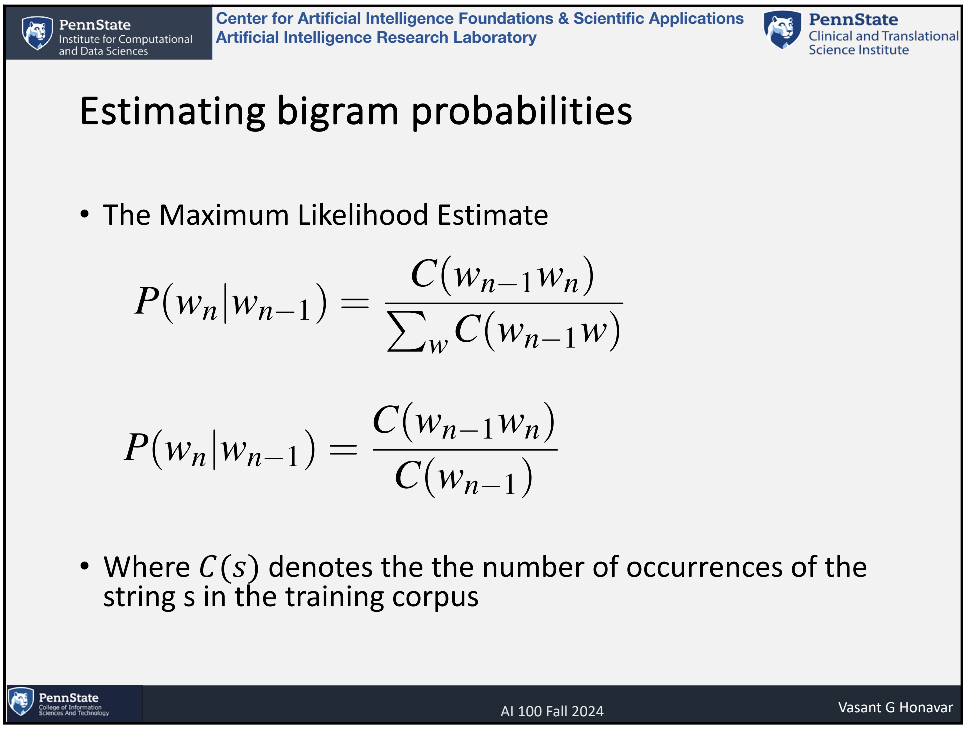


Limitations of N-gram language models

- N-grams can't handle long-distance dependencies:

  “The soup that I made from a recipe from that new cookbook I bought yesterday was amazingly delicious."
- N-grams don't do well at modeling new sequences with similar meanings
- Solution: Large language models
  - Can handle much longer contexts
  - Because of the use of latent embeddings, can model synonymy better, and does better at generating novel text

18



Estimating bigram probabilities

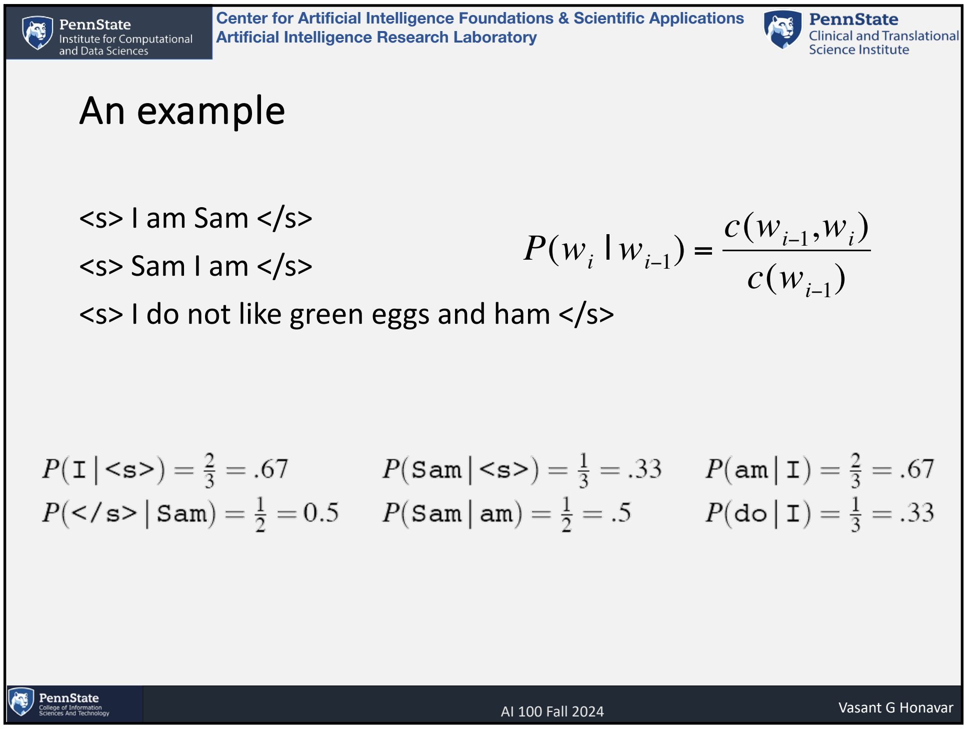
- The Maximum Likelihood Estimate

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

- Where $C(s)$ denotes the the number of occurrences of the string s in the training corpus

19



An example

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

$$P(\texttt{I} \mid \texttt{<s>}) = \tfrac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \tfrac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \tfrac{2}{3} = .67$$

$$P(\texttt{</s>} \mid \texttt{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \tfrac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \tfrac{1}{3} = .33$$

20

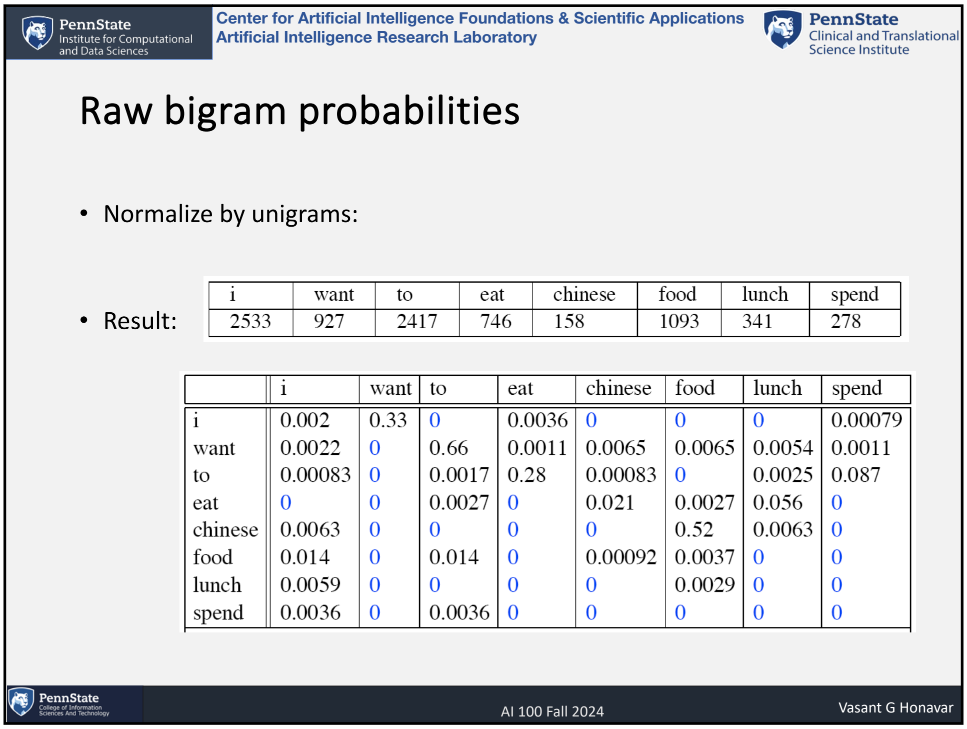
Berkeley Restaurant Corpus

- 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 Cafe Venezia open during the day?

Raw bigram counts

- Out of 9222 sentences

	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

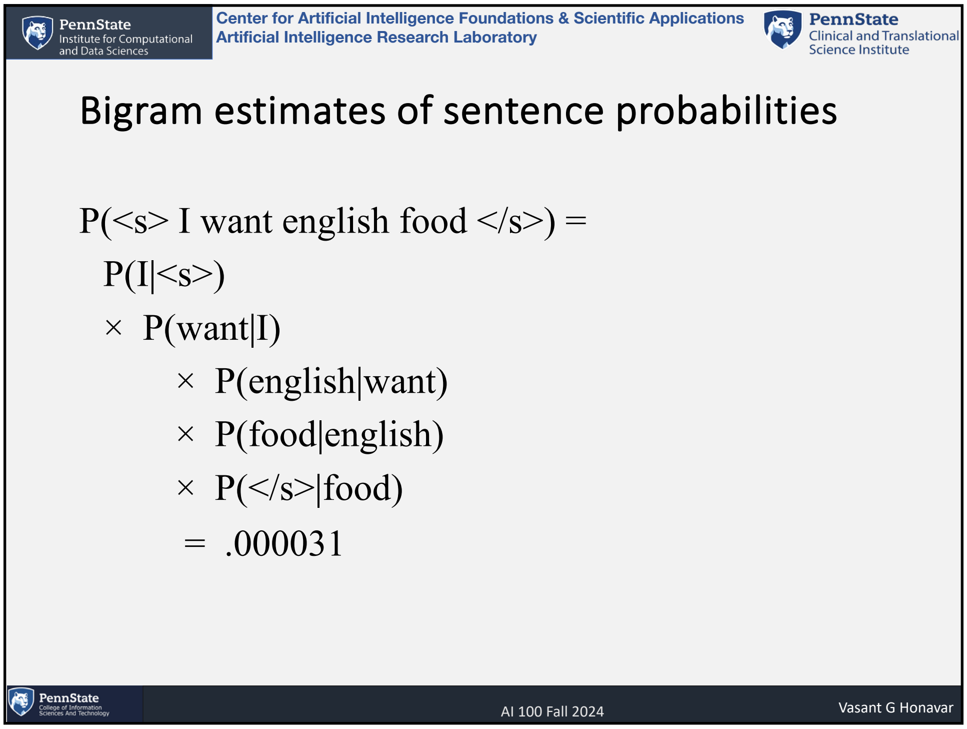


Raw bigram probabilities

- Normalize by unigrams:
- Result:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0



Bigram estimates of sentence probabilities

P(<s> I want english food </s>) =

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

× P(</s>|food)

= .000031

PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

What kinds of knowledge do N-grams represent?

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

25

PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Dealing with scale in large n-grams

- We work with the log of the LM probabilities to avoid numerical issues with calculations
  - Underflow from multiplying many small numbers

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

If we need the probability, we can do one exp at the end

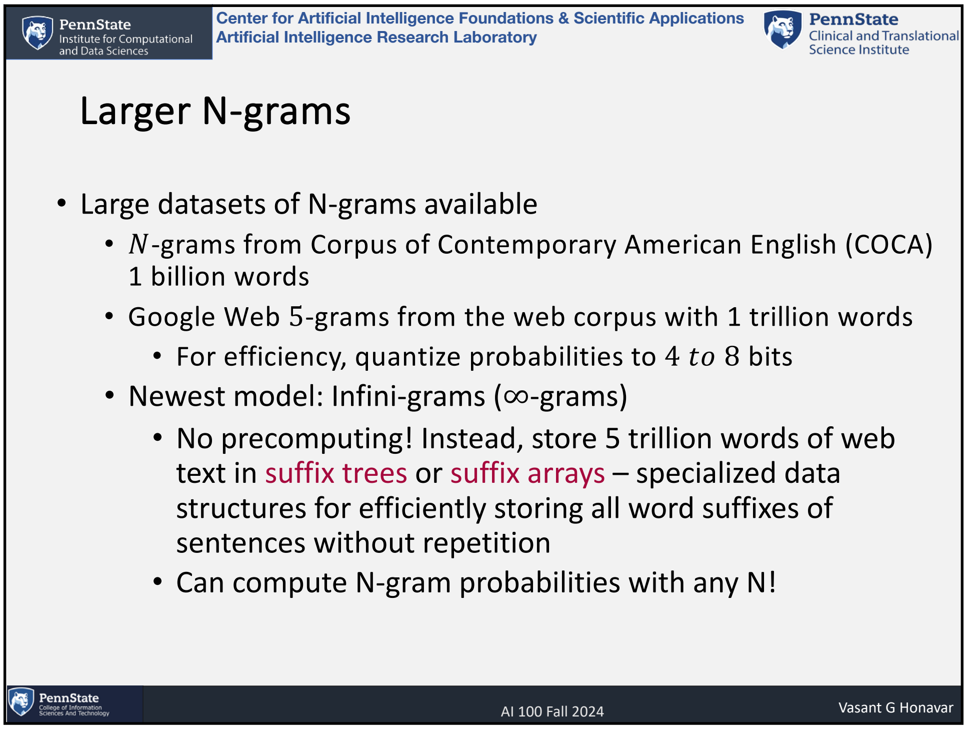
$$p_1 \times p_2 \times p_3 \times p_4 = \exp(\log p_1 + \log p_2 + \log p_3 + \log p_4)$$

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

26



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Larger N-grams

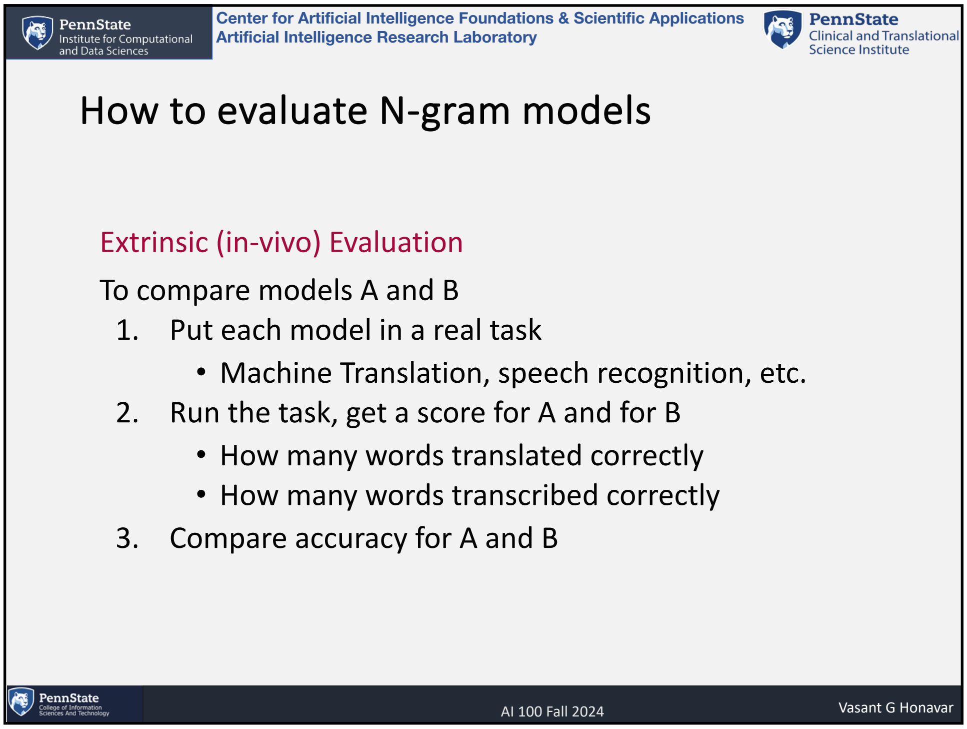
- Large datasets of N-grams available
  - N-grams from Corpus of Contemporary American English (COCA) 1 billion words
  - Google Web 5-grams from the web corpus with 1 trillion words
    - For efficiency, quantize probabilities to $4\ to\ 8$ bits
  - Newest model: Infini-grams (∞-grams)
    - No precomputing! Instead, store 5 trillion words of web text in suffix trees or suffix arrays – specialized data structures for efficiently storing all word suffixes of sentences without repetition
    - Can compute N-gram probabilities with any N!

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

27



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

How to evaluate N-gram models

Extrinsic (in-vivo) Evaluation

To compare models A and B

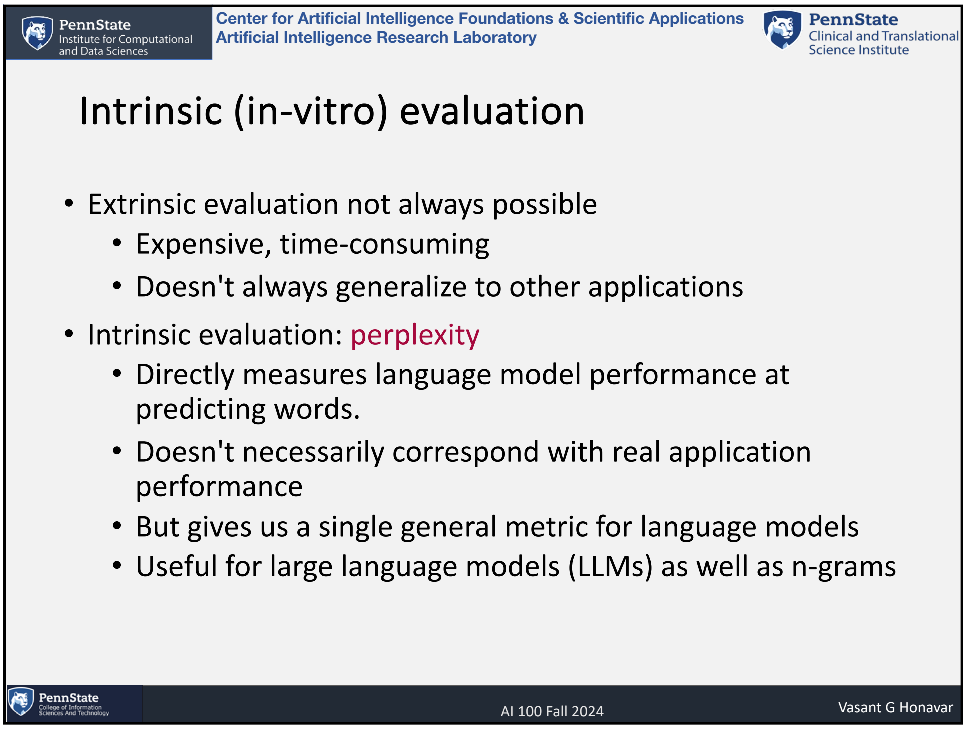
1. Put each model in a real task
   - Machine Translation, speech recognition, etc.
2. Run the task, get a score for A and for B
   - How many words translated correctly
   - How many words transcribed correctly
3. Compare accuracy for A and B

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

28



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Intrinsic (in-vitro) evaluation

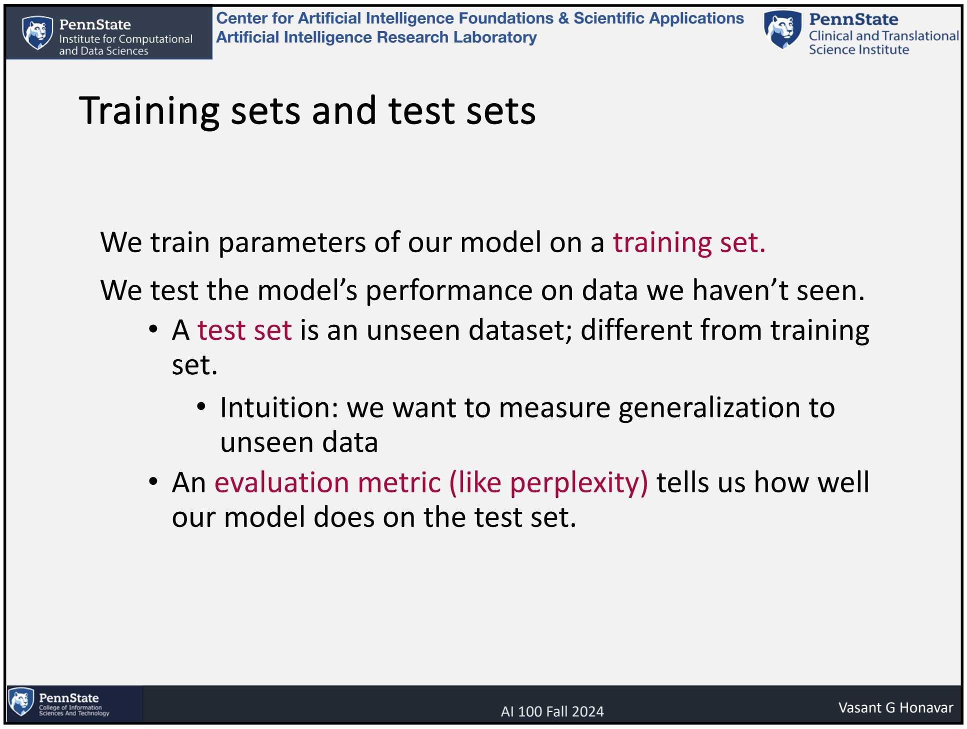
- Extrinsic evaluation not always possible
  - Expensive, time-consuming
  - Doesn't always generalize to other applications
- Intrinsic evaluation: perplexity
  - Directly measures language model performance at predicting words.
  - Doesn't necessarily correspond with real application performance
  - But gives us a single general metric for language models
  - Useful for large language models (LLMs) as well as n-grams

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

29



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Training sets and test sets

We train parameters of our model on a training set.

We test the model’s performance on data we haven’t seen.

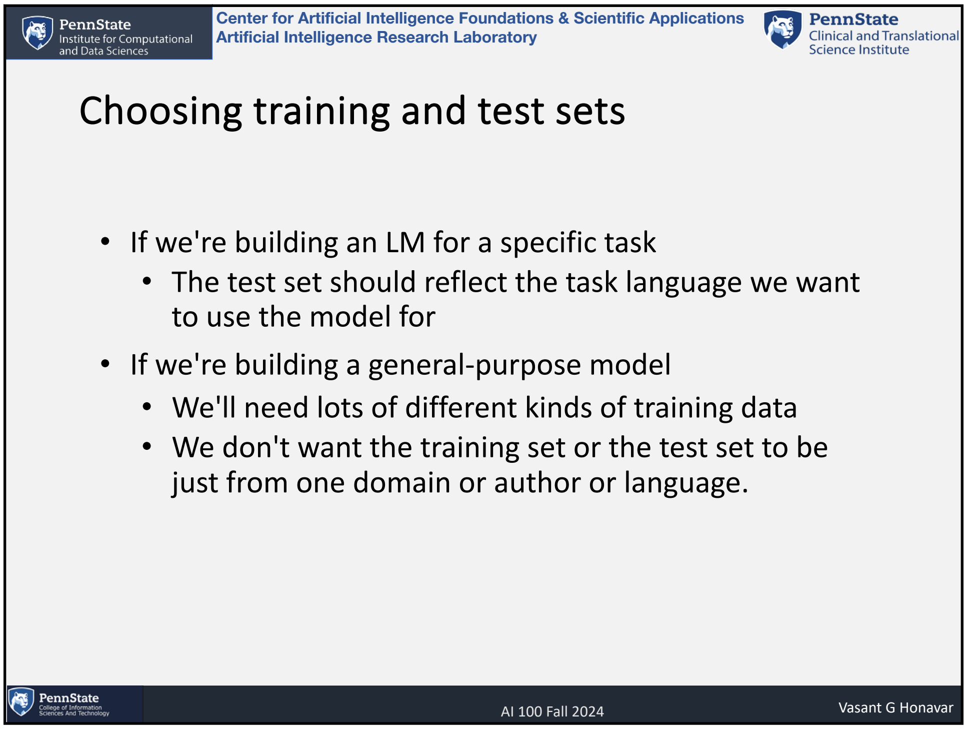
- A test set is an unseen dataset; different from training set.
  - Intuition: we want to measure generalization to unseen data
- An evaluation metric (like perplexity) tells us how well our model does on the test set.

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

30



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Choosing training and test sets

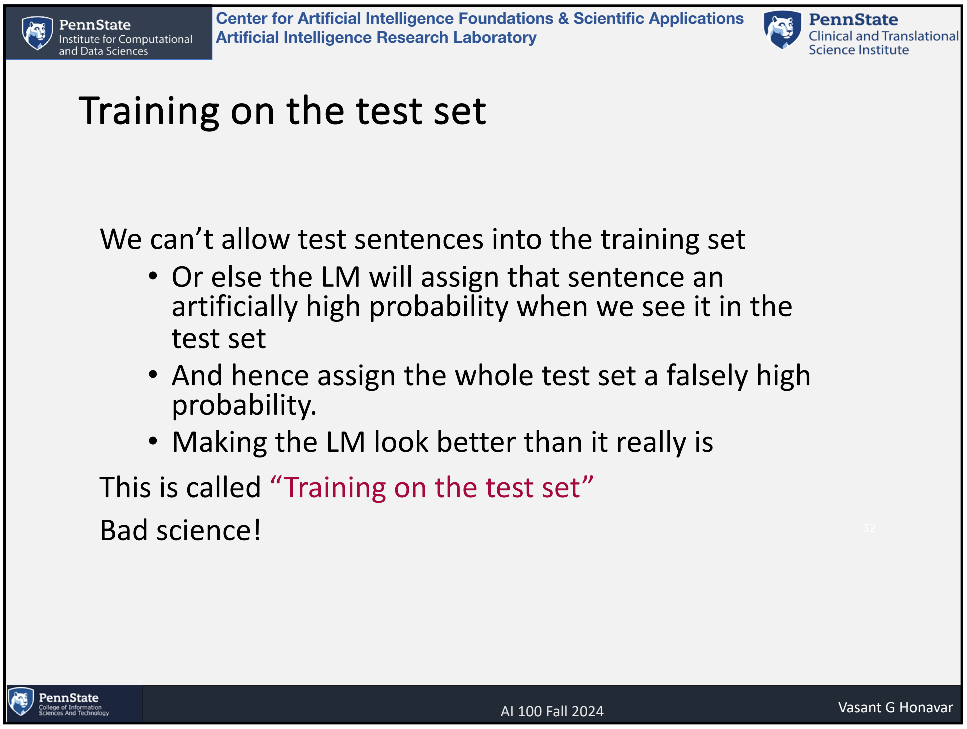
- If we're building an LM for a specific task
  - The test set should reflect the task language we want to use the model for
- If we're building a general-purpose model
  - We'll need lots of different kinds of training data
  - We don't want the training set or the test set to be just from one domain or author or language.

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

31



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Training on the test set

We can’t allow test sentences into the training set

- Or else the LM will assign that sentence an artificially high probability when we see it in the test set
- And hence assign the whole test set a falsely high probability.
- Making the LM look better than it really is

This is called “Training on the test set”

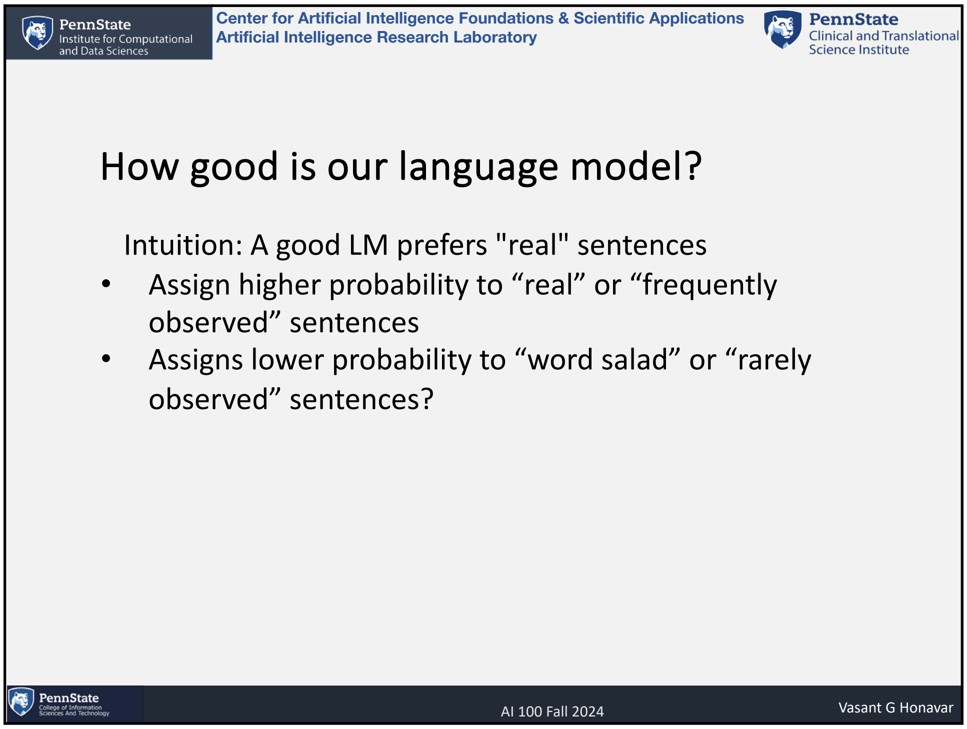
Bad science!

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

32

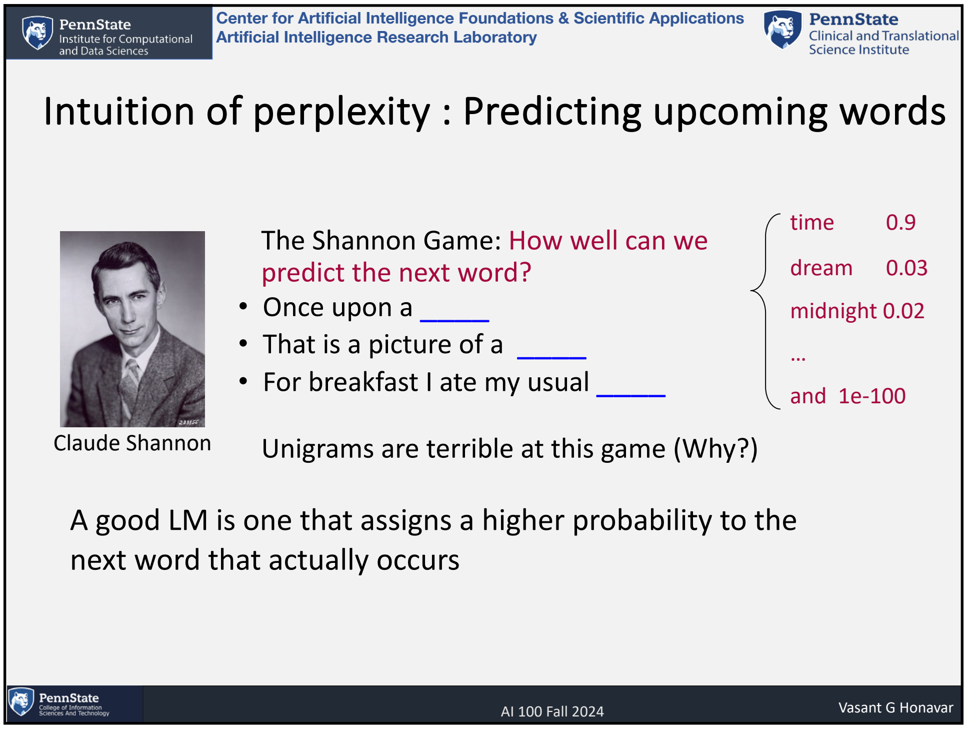


How good is our language model?

Intuition: A good LM prefers "real" sentences

- Assign higher probability to “real” or “frequently observed” sentences
- Assigns lower probability to “word salad” or “rarely observed” sentences?

33



Intuition of perplexity : Predicting upcoming words

Claude Shannon

The Shannon Game: How well can we predict the next word?

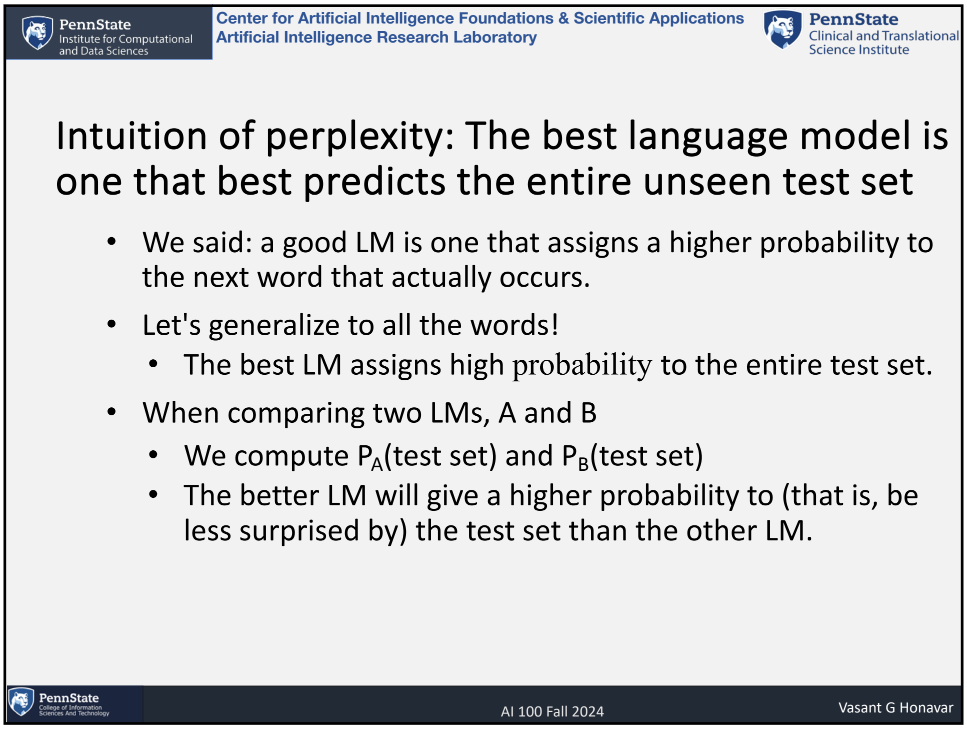
- Once upon a ______
- That is a picture of a ______
- For breakfast I ate my usual ______

time	0.9
dream	0.03
midnight	0.02
…	
and	1e-100

Unigrams are terrible at this game (Why?)

A good LM is one that assigns a higher probability to the next word that actually occurs

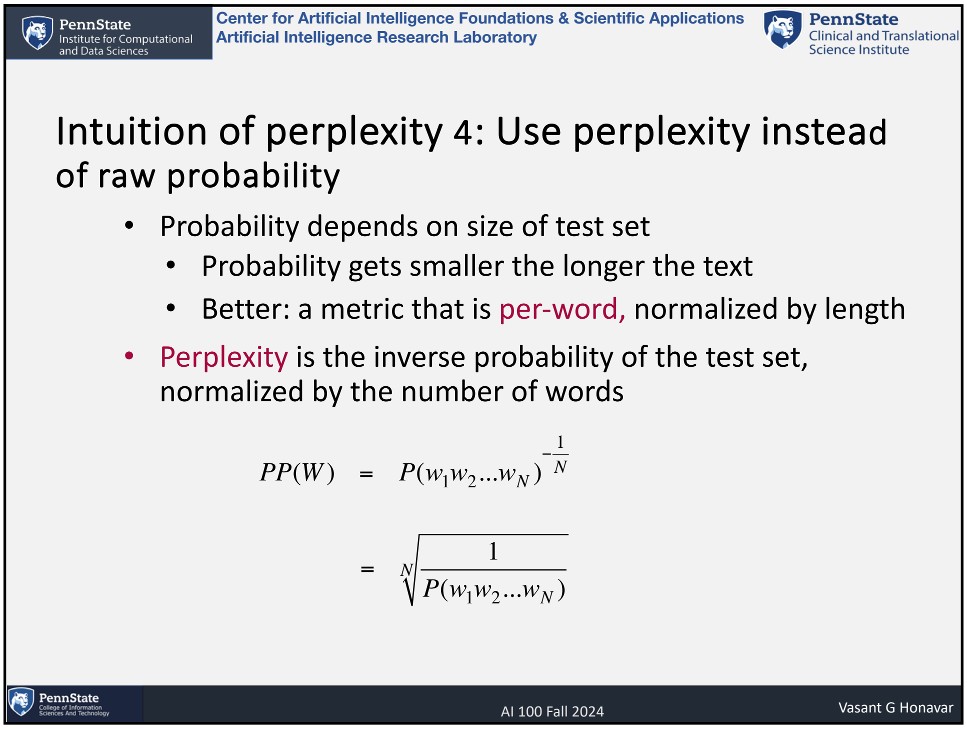
34



Intuition of perplexity: The best language model is one that best predicts the entire unseen test set

- We said: a good LM is one that assigns a higher probability to the next word that actually occurs.
- Let's generalize to all the words!
  - The best LM assigns high probability to the entire test set.
- When comparing two LMs, A and B
  - We compute P_A(test set) and P_B(test set)
  - The better LM will give a higher probability to (that is, be less surprised by) the test set than the other LM.

35

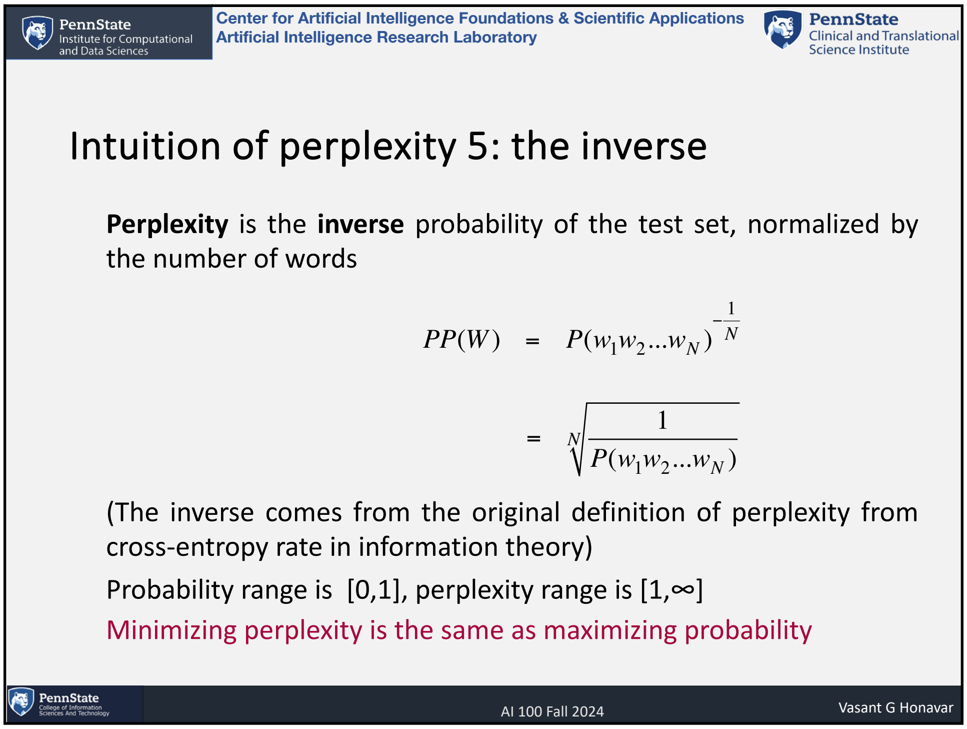


Intuition of perplexity 4: Use perplexity instead of raw probability

- Probability depends on size of test set
  - Probability gets smaller the longer the text
  - Better: a metric that is per-word, normalized by length
- Perplexity is the inverse probability of the test set, normalized by the number of words

$$
\begin{aligned}
PP(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}
\end{aligned}
$$

36



Intuition of perplexity 5: the inverse

Perplexity is the **inverse** probability of the test set, normalized by the number of words

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

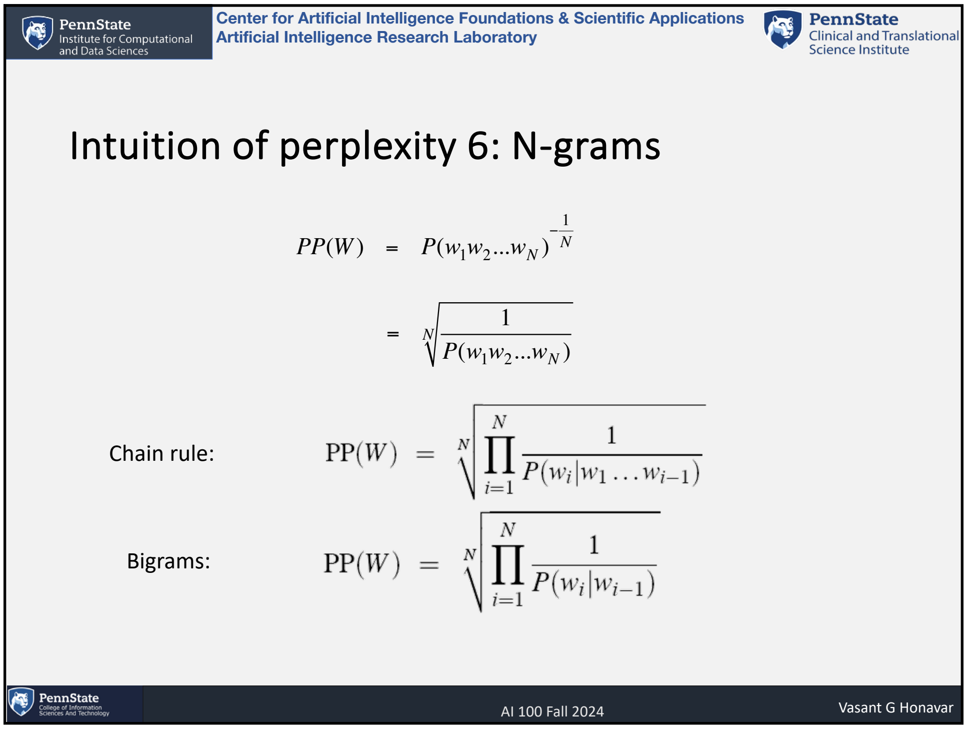
$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

(The inverse comes from the original definition of perplexity from cross-entropy rate in information theory)

Probability range is [0,1], perplexity range is [1,∞]

Minimizing perplexity is the same as maximizing probability

37



Intuition of perplexity 6: N-grams

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Chain rule:
$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}$$

Bigrams:
$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

38

For a given test set, Lower perplexity implies better LM

- Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

The Shannon (1948) Visualization Method
Sample words from an LM

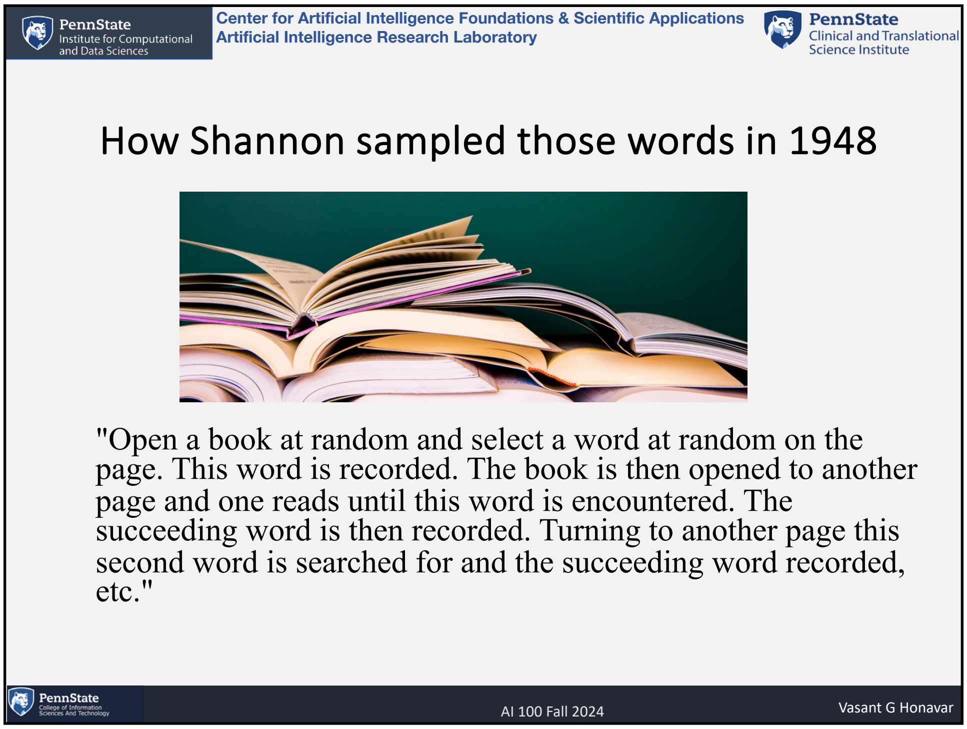
Claude Shannon

Unigram:

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

Bigram:

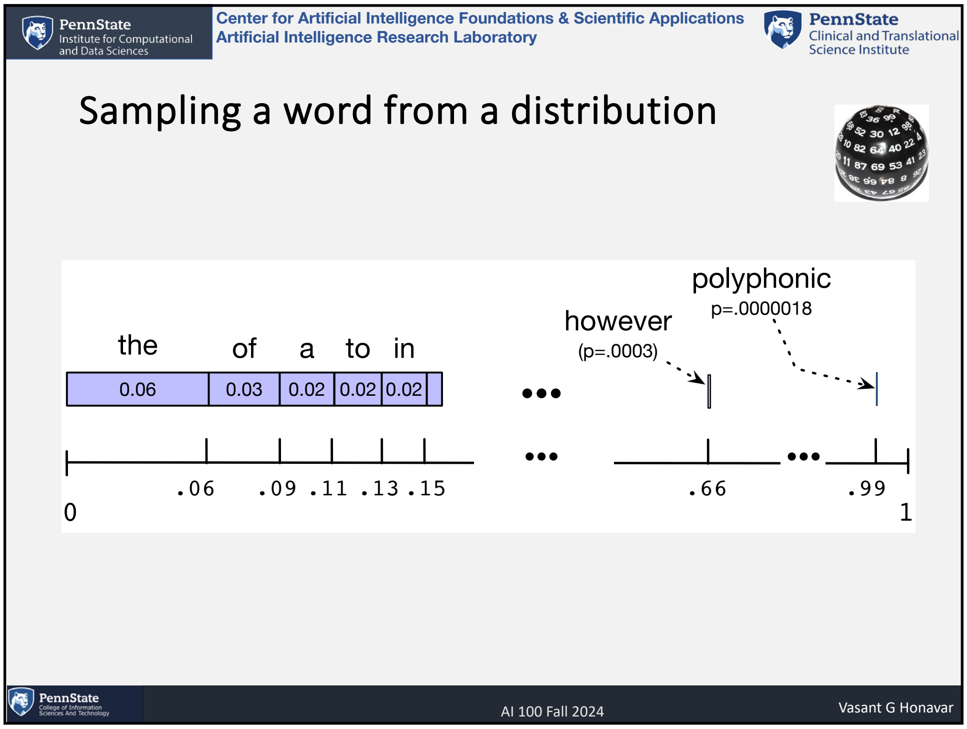
THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.



How Shannon sampled those words in 1948

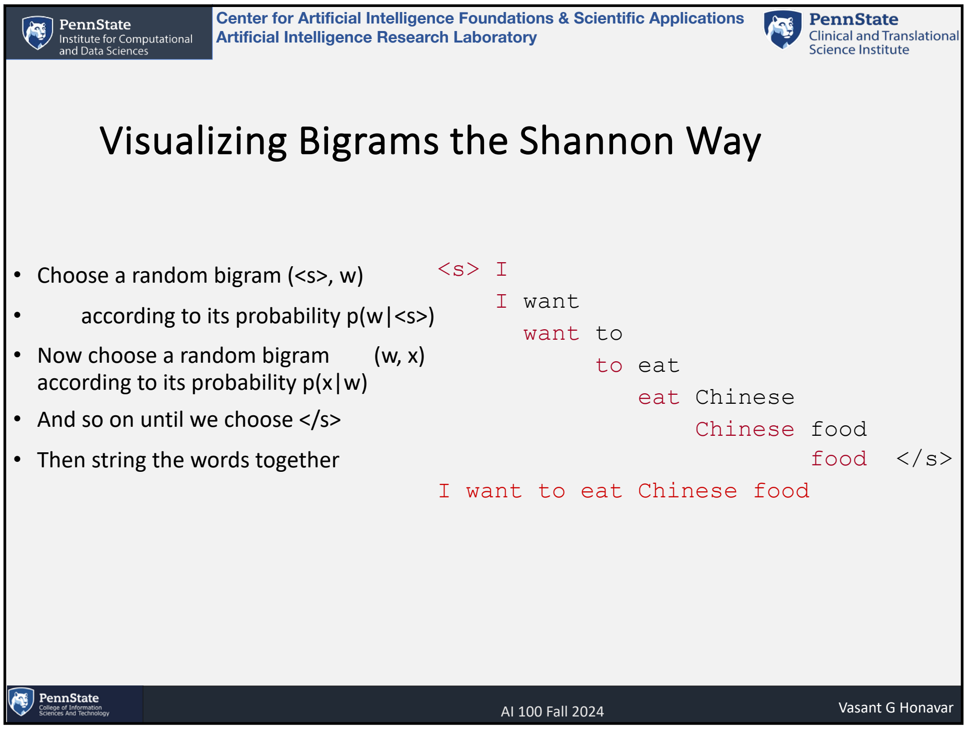
"Open a book at random and select a word at random on the page. This word is recorded. The book is then opened to another page and one reads until this word is encountered. The succeeding word is then recorded. Turning to another page this second word is searched for and the succeeding word recorded, etc."

41



Sampling a word from a distribution

42

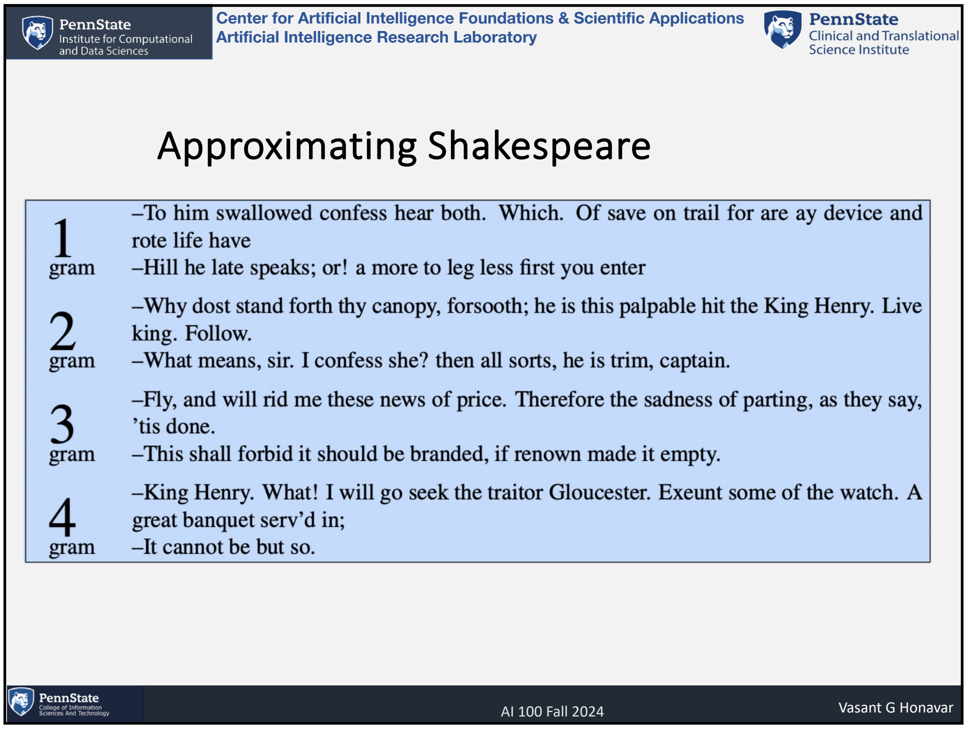


Visualizing Bigrams the Shannon Way

- Choose a random bigram (<s>, w)
- according to its probability p(w|<s>)
- Now choose a random bigram (w, x) according to its probability p(x|w)
- And so on until we choose </s>
- Then string the words together

```
<s> I
    I want
      want to
           to eat
              eat Chinese
                  Chinese food
                          food  </s>
I want to eat Chinese food
```

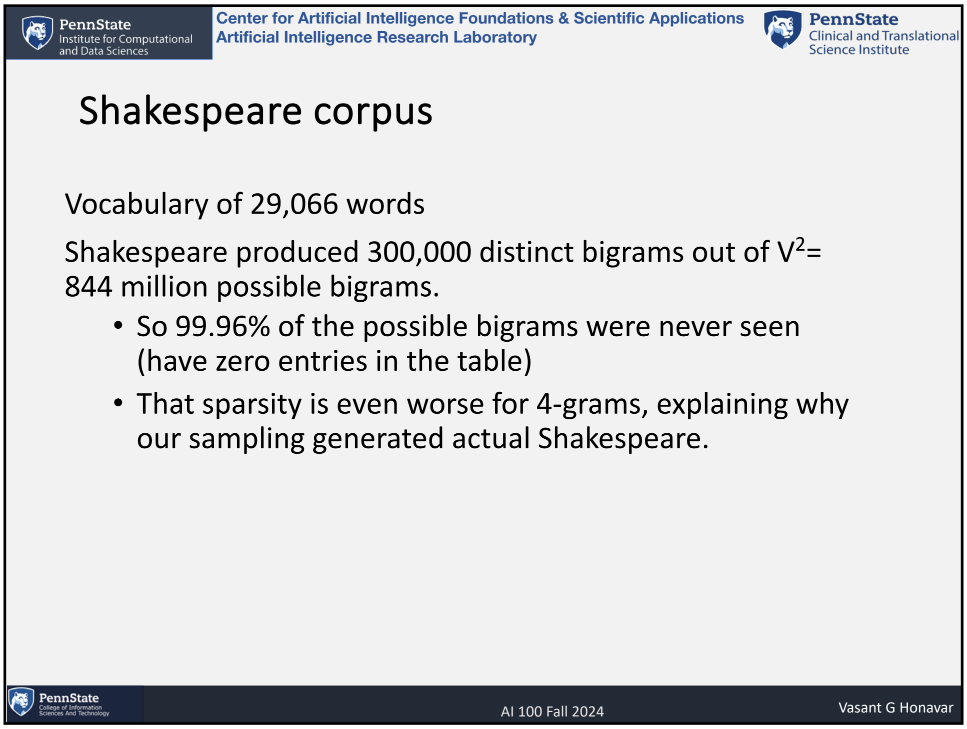
43



Approximating Shakespeare

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

44

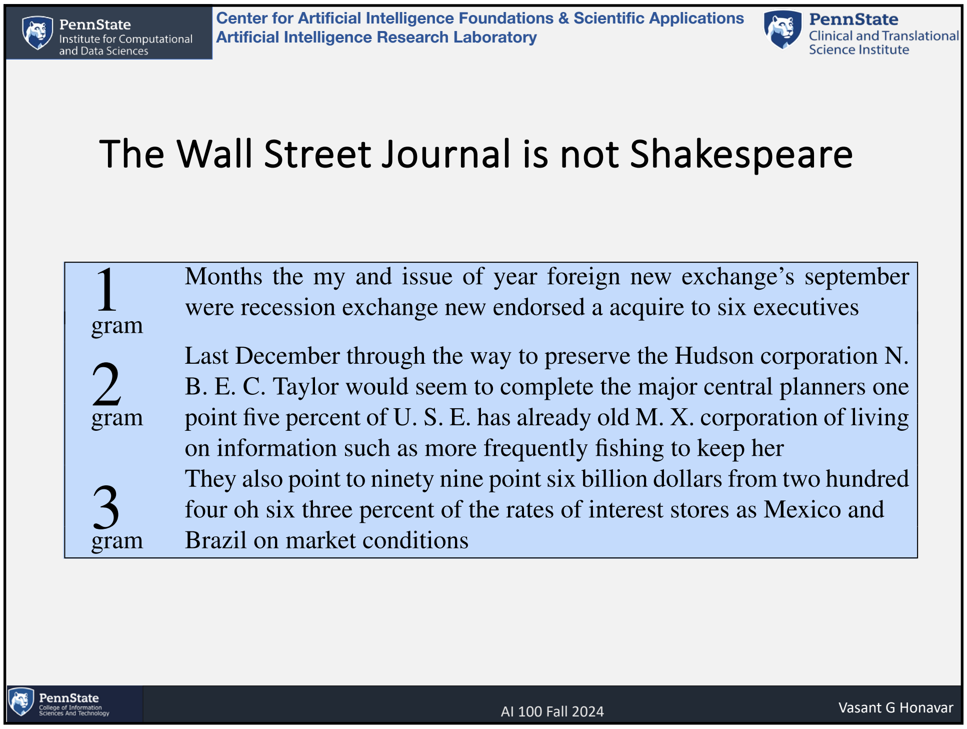


Shakespeare corpus

Vocabulary of 29,066 words

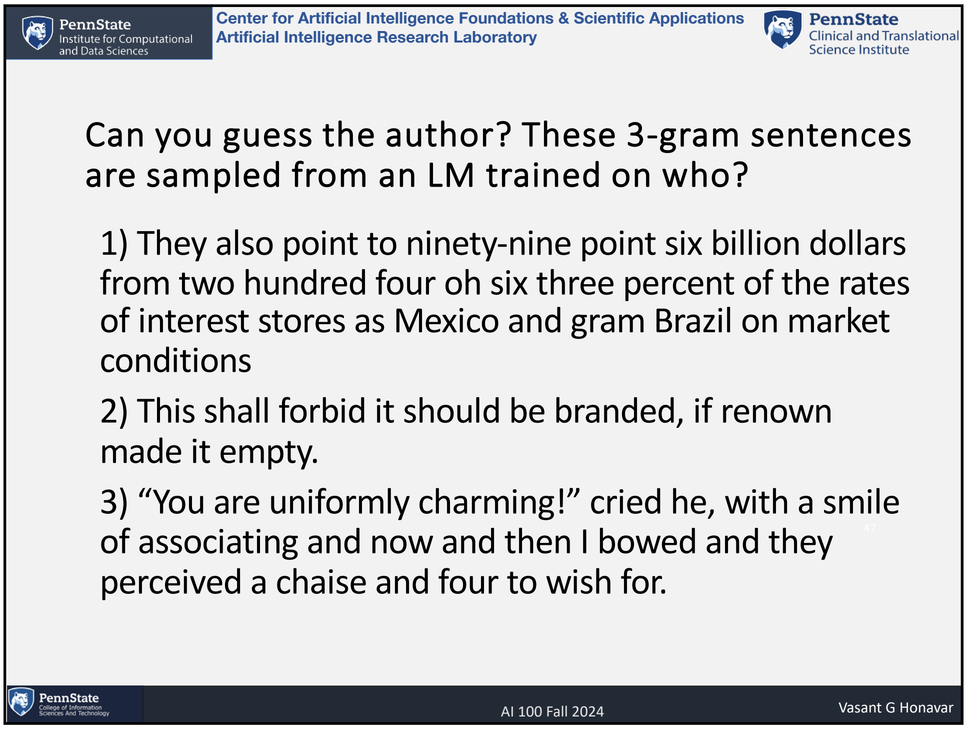
Shakespeare produced 300,000 distinct bigrams out of V^2= 844 million possible bigrams.

- So 99.96% of the possible bigrams were never seen (have zero entries in the table)
- That sparsity is even worse for 4-grams, explaining why our sampling generated actual Shakespeare.



The Wall Street Journal is not Shakespeare

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions



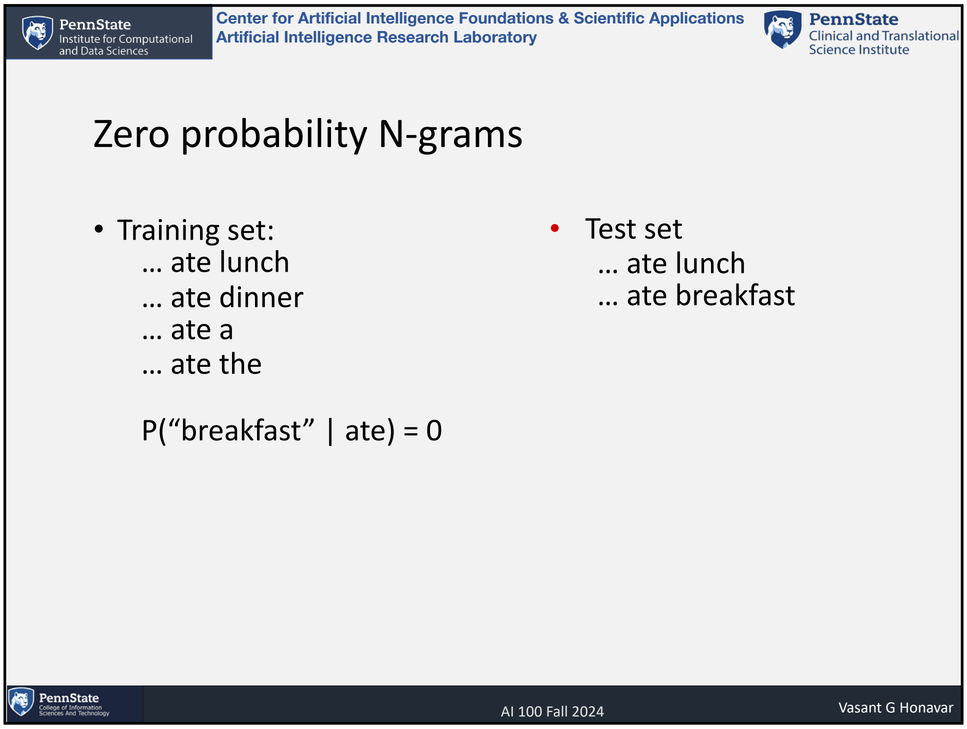
Can you guess the author? These 3-gram sentences are sampled from an LM trained on who?

1) They also point to ninety-nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and gram Brazil on market conditions

2) This shall forbid it should be branded, if renown made it empty.

3) “You are uniformly charming!” cried he, with a smile of associating and now and then I bowed and they perceived a chaise and four to wish for.

47



Zero probability N-grams

- Training set:
  … ate lunch
  … ate dinner
  … ate a
  … ate the

  P(“breakfast” | ate) = 0

- Test set
  … ate lunch
  … ate breakfast

48

Zero probability bigrams

Bigrams with zero probability

- Will hurt our performance for texts where those words appear!
- And mean that we will assign 0 probability to the test set!

And hence we cannot compute perplexity (can't divide by 0)!

Add-one estimation

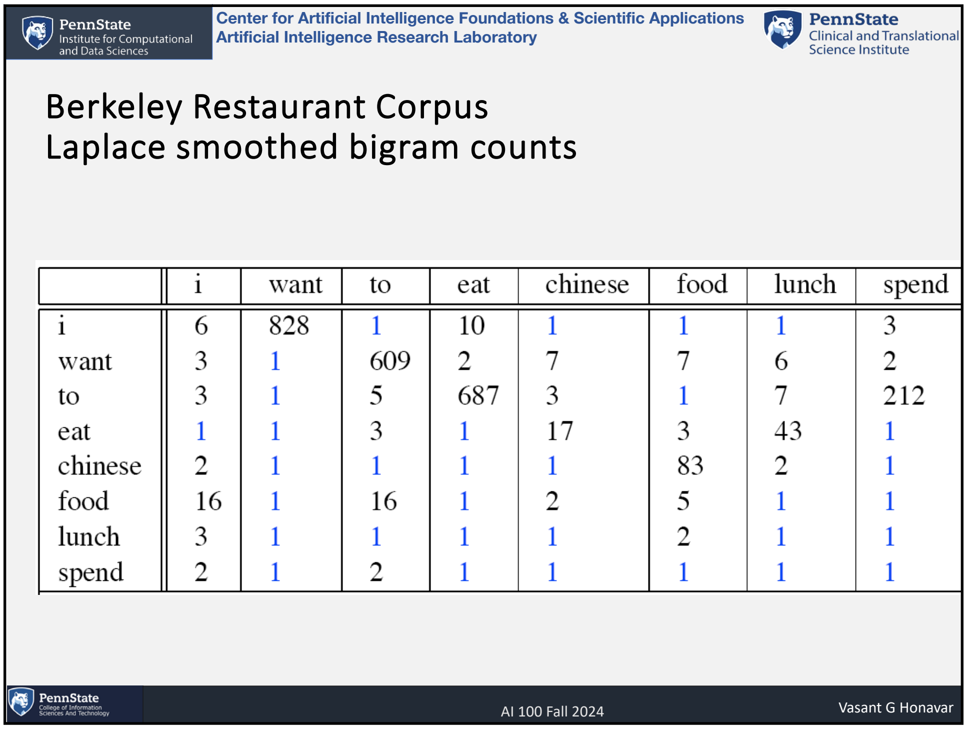
- Also called Laplace smoothing
- Pretend we saw each word one more time than we did
- Just add one to all the counts!

- MLE estimate:

$$P_{\text{MLE}}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

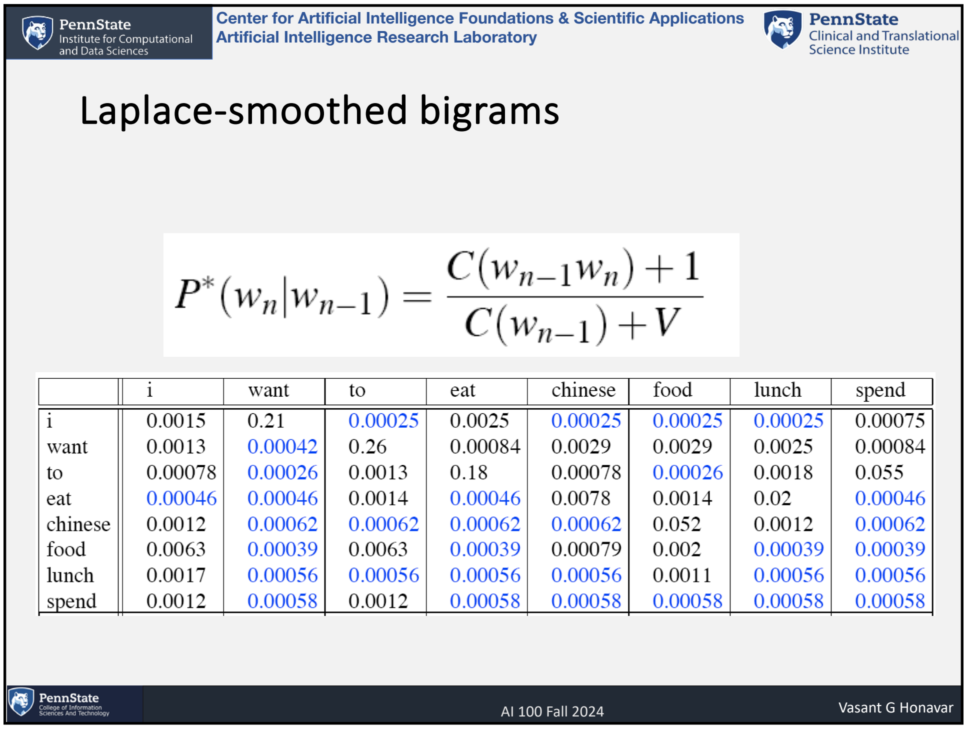
- Add-1 estimate:

$$P_{\text{Laplace}}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{\sum_w (C(w_{n-1}w)+1)} = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$



Berkeley Restaurant Corpus
Laplace smoothed bigram counts

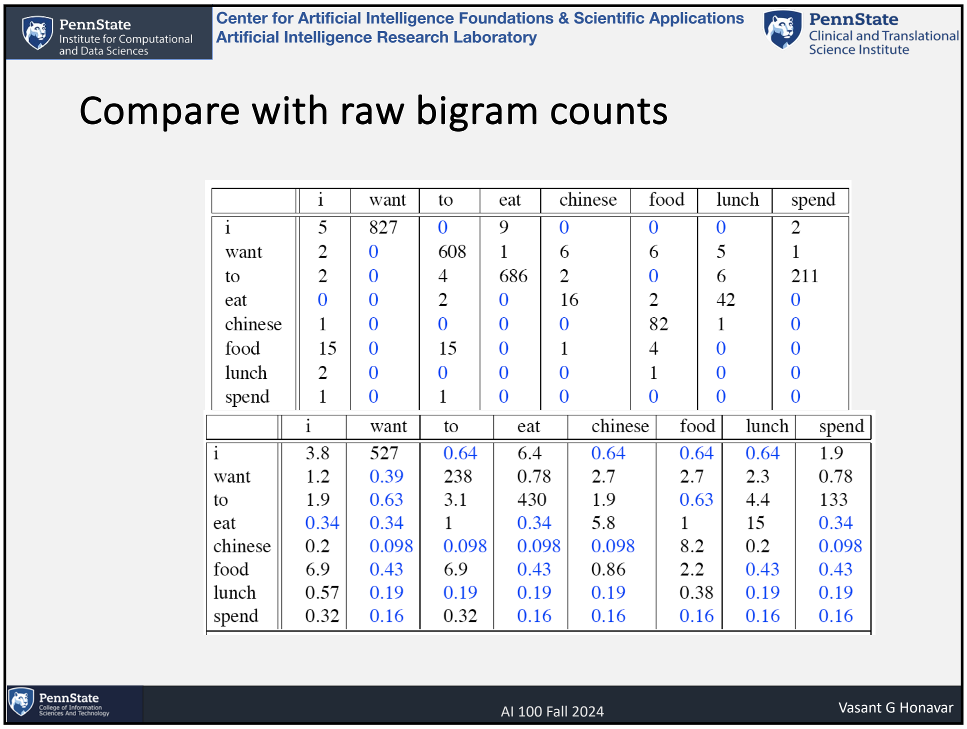
	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1



Laplace-smoothed bigrams

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Compare with raw bigram 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

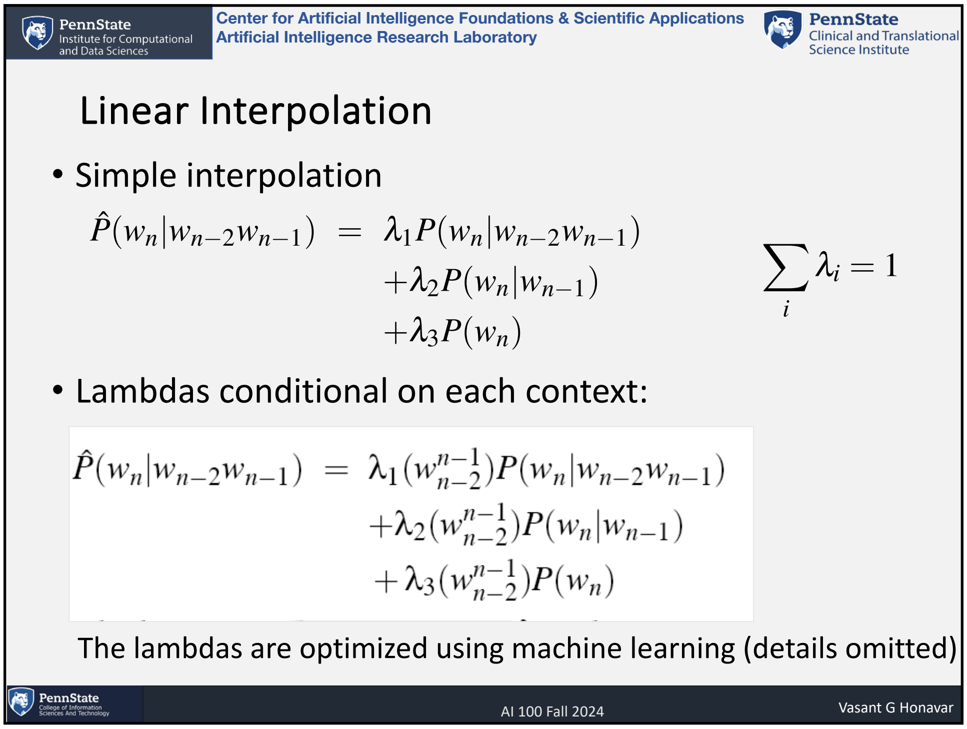
	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

PennState College of Information Sciences And Technology

AI 100 Fall 2024

Vasant G Honavar

53



PennState Institute for Computational and Data Sciences

Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

Linear Interpolation

- Simple interpolation

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1 P(w_n|w_{n-2}w_{n-1}) \\ &+\lambda_2 P(w_n|w_{n-1}) \\ &+\lambda_3 P(w_n)\end{aligned}$$

$$\sum_i \lambda_i = 1$$

- Lambdas conditional on each context:

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1(w_{n-2}^{n-1})P(w_n|w_{n-2}w_{n-1}) \\ &+\lambda_2(w_{n-2}^{n-1})P(w_n|w_{n-1}) \\ &+\lambda_3(w_{n-2}^{n-1})P(w_n)\end{aligned}$$

The lambdas are optimized using machine learning (details omitted)

PennState College of Information Sciences And Technology

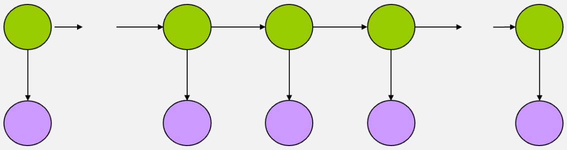
AI 100 Fall 2024

Vasant G Honavar

55

Hidden Markov models

- Suppose probability of the next word depends on context
- We can use latent or hidden states to model context
  - Transition between hidden states obeys the Markov assumption
    - The probability of the next state is independent of all states except the current state
  - The probability of the generated word depends on the state



More complex models

- Combine aspects of Markov models, hidden Markov models and deep neural networks