# N-Gram Language Models Part 3

LIN 313 Language and Computers
UT Austin Fall 2025

# Admin

- HW1 Due today
  - feedback in about a  week
  - late policy: unless you spoke with us about an extension, your grade will be reduced 10% each day. Work can't be accepted after assignments have been returned and the solutions have been posted (at least 7 days)
- HW2 up this weekend/Monday
  - you have ~2 weeks to work on it
- Article annotation for Monday 9/15

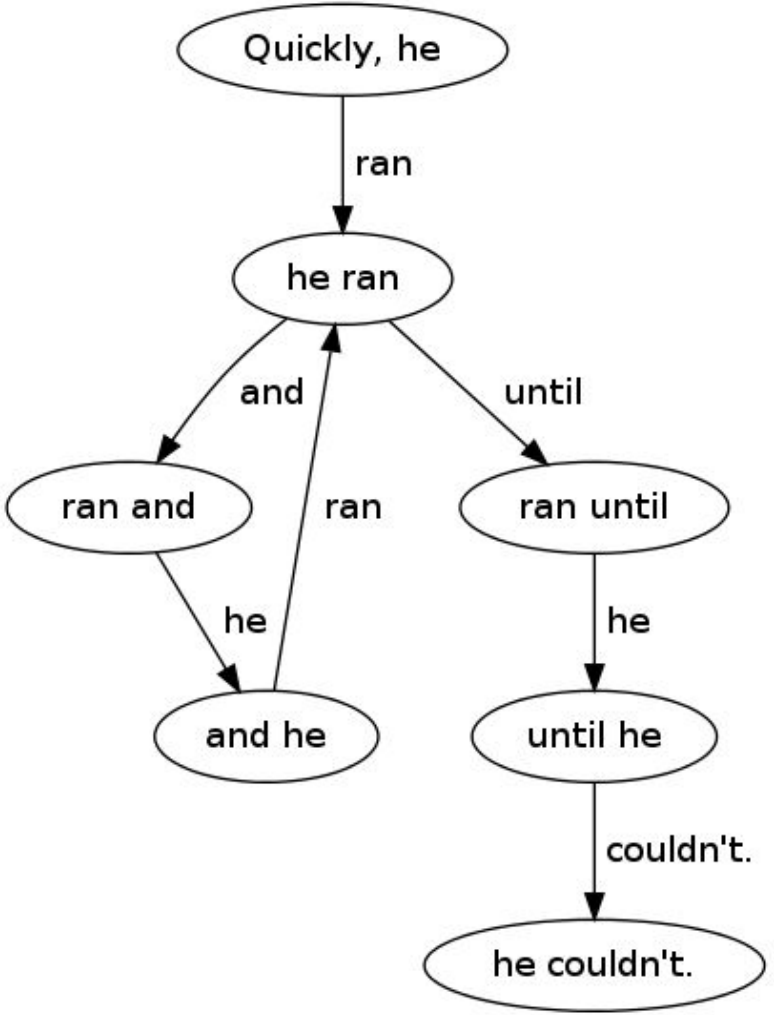
# Overview

- Text mashups
- Model evaluation
  - String probabilities
  - Perplexity
  - Training vs Testing
- Recursive training of models
  - What happens when we train one model on the outputs of another?

# Order

The **order** of a Markov model is the number of previous states that the model takes into account. This model is **second order**, because we care about the **previous two** words when determining the next word.

If we learn our transition probabilities from a corpus, this is just a bi-gram language model!



# Corpora

- The training dataset  or **corpus** used to construct and N-gram model has dramatic impacts on the resulting model

What are some ways that a text corpus can vary?

# Corpora

- The training dataset  or **corpus** used to construct and N-gram model has dramatic impacts on the resulting model

What are some ways that a text corpus can vary?

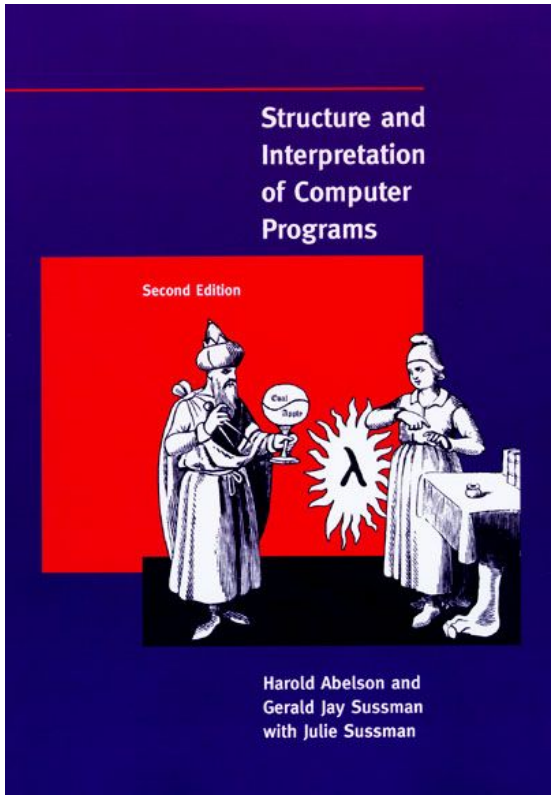
Well, what are the ways that language can vary?

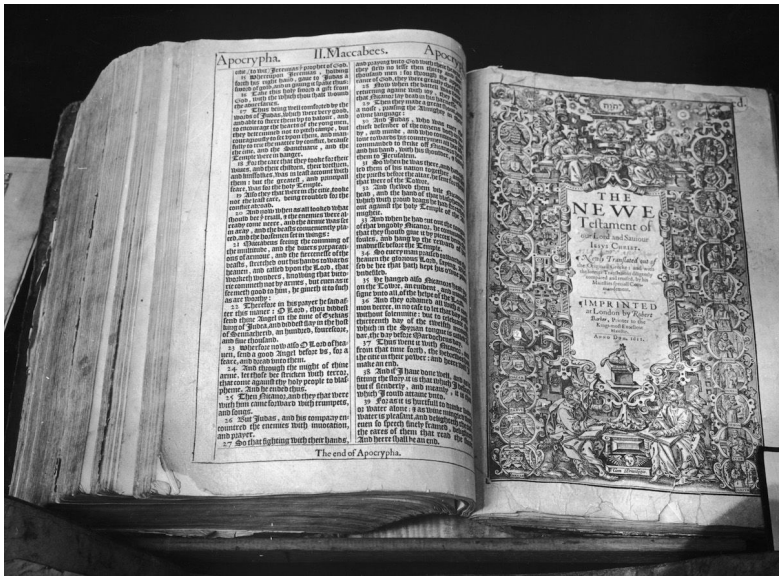
- genre
- register (formal, transactional, intimate, bureaucratic)
- dialect (African American Englishes, Global Englishes, code switching)
- historical era
- ….

# N-gram Mashups

1. Navigate to https://gchronis.github.io/n-gram-models.html (linked from the schedule)
2. Choose a corpus
    a. project gutenberg has .txt files of expired copyright books https://www.gutenberg.org/
    b. your own writing!
    c. sometimes you can get lucky searching the internet
        i. e.g. searching 'kendrick lamar lyrics .txt' → https://github.com/pajose/Kendrick-Lamar-AI/blob/master/kendrick_lamar_lyrics.txt
    d. kaggle and huggingface are good sources (usually need an account)
3. Generate an N-gram model
    a. experiment with the settings. Try a bigram, trigram, and 4gram
4. Use two different texts to create a mashup
    a. ensure that they are about the same length
    b. you can 'weight' the smaller text by pasting it in multiple times
5. Enter one of the generations you like into Instapoll.

# King James Programming



# King James Programming

> Investigate the shell's here documents and Python's triple-quote construct to find out the Almighty unto perfection

*5 years ago* *59 notes* *#kjv* *#bible* *#sicp* *#esr* *#markov chains*

> 51:11 Cast me not off in the time of execution of a sequential instruction stream

*5 years ago* *59 notes*

> 25:12 And thou shalt put into the heart of today's IBM mainframe operating systems.

*5 years ago* *84 notes* *#kjv* *#bible* *#sicp* *#esr* *#markov chains*

> 37:29 The righteous shall inherit the land, and leave it for an inheritance unto the children of Gad according to the number of steps that is linear in `b`.

*4 years ago* *87 notes* *#kjv* *#bible* *#sicp* *#poignant guide* *#markov chains*

# N-grams in the wild

- Google adopted deep neural networks in 2015. How did autosuggest work before then?
  - very sophisticated n-gram models
- practical considerations
  - people will search for strings that aren't in the n-gram corpus!
  - people will search for words that aren't in the vocabulary!
- strategies
  - backoff
  - Laplace smoothing (add-one smoothing)
  - add-k smoothing

# Evaluating Language Models

**Extrinsic evaluation.** This involves evaluating the models by employing them in an actual task (such as machine translation) and looking at their final loss/accuracy. This is the best option as it's the only way to tangibly see how different models affect the task we're interested in. However, it can be computationally expensive and slow as it requires training a full system.

**Intrinsic evaluation.** This involves finding some metric to evaluate the language model itself, not taking into account the specific tasks it's going to be used for. It's a useful way of quickly comparing models. **Perplexity** is an intrinsic evaluation method.

# Language models as grammars, again

We've been talking about LMs as **generators** (models that generate new text)

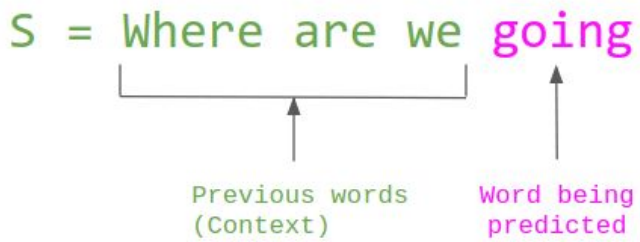
We can also think of them as **discriminators** (models that choose between two options). Given two sentences, which has a higher probability? A Spanish LM should assign higher probabilities to sentences in Spanish than other sentences.

If we are comparing performance between models, we want the one that assigns higher probabilities to "good" sentences.

A model's probability prediction is called the **maximum likelihood estimate**.

# Evaluating Language Models: Probability

we want our model to assign high probabilities to sentences that are real and syntactically correct, and low probabilities to fake, incorrect, or highly infrequent sentences. Assuming our dataset is made of sentences that are in fact real and correct, this means that the best model will be the one that assigns the **highest probability to the test set.**




$$P(S) = P(\text{Where}) \times P(\text{are} \mid \text{Where}) \times P(\text{we} \mid \text{Where are}) \times P(\text{going} \mid \text{Where are we})$$

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25$ $\quad$ $P(\texttt{english}\,|\,\texttt{want}) = 0.0011$

$P(\texttt{food}\,|\,\texttt{english}) = 0.5$ $\quad$ $P(\texttt{</s>}\,|\,\texttt{food}) = 0.68$

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25$ $\quad$ $P(\texttt{english}\,|\,\texttt{want}) = 0.0011$

$P(\texttt{food}\,|\,\texttt{english}) = 0.5$ $\quad$ $P(\texttt{</s>}\,|\,\texttt{food}) = 0.68$

What is the maximum likelihood estimate for "I want Chinese food"?

# Maximum Likelihood Estimation

P(<s> i want chinese food </s>)
= ???

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25$ $\quad$ $P(\texttt{english}\,|\,\texttt{want}) = 0.0011$

$P(\texttt{food}\,|\,\texttt{english}) = 0.5$ $\quad$ $P(\texttt{</s>}\,|\,\texttt{food}) = 0.68$

What is the maximum likelihood estimate for "I want Chinese food"?

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}|\texttt{<s>}) = 0.25$ $P(\texttt{english}|\texttt{want}) = 0.0011$

$P(\texttt{food}|\texttt{english}) = 0.5$ $P(\texttt{</s>}|\texttt{food}) = 0.68$

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

What is the maximum likelihood estimate for "I want Chinese food"?

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}|\texttt{<s>}) = 0.25$ $P(\texttt{english}|\texttt{want}) = 0.0011$

$P(\texttt{food}|\texttt{english}) = 0.5$ $P(\texttt{</s>}|\texttt{food}) = 0.68$

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

It's very low! But it's higher than P("I want English Food") = 0.000031

What is the maximum likelihood estimate for "I want Chinese food"?

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25$ $\quad$ $P(\texttt{english}\,|\,\texttt{want}) = 0.0011$

$P(\texttt{food}\,|\,\texttt{english}) = 0.5$ $\quad$ $P(\texttt{</s>}\,|\,\texttt{food}) = 0.68$

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

It's very low! But it's higher than P("I want English Food") = 0.000031

If we just saw these probabilities and not the model, could we infer anything about the training corpus?

What is the maximum likelihood estimate for "I want Chinese food"?

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}|\texttt{<s>}) = 0.25$ $P(\texttt{english}|\texttt{want}) = 0.0011$

$P(\texttt{food}|\texttt{english}) = 0.5$ $P(\texttt{</s>}|\texttt{food}) = 0.68$

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

P(<s> i want to eat chinese food </s>)
= ???

What is the maximum likelihood estimate for "I want Chinese food" vs "I want to eat Chinese food"?

# Maximum Likelihood Estimation

transition

state

| | 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 |

**Figure 3.2** Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Here are a few other useful probabilities:

$P(\texttt{i}|\texttt{<s>}) = 0.25$ $P(\texttt{english}|\texttt{want}) = 0.0011$

$P(\texttt{food}|\texttt{english}) = 0.5$ $P(\texttt{</s>}|\texttt{food}) = 0.68$

What is the maximum likelihood estimate for "I want Chinese food" vs "I want to eat Chinese food"?

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

P(<s> i want to eat chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(to | want)
* P(eat | to)
* P(Chinese | eat)
* P(food | Chinese)

= 0.25 * 0.33 * 0.66 * 0.0027 * 0.021 * 0.00092

= 0.00000000284

# Maximum Likelihood Estimation

transition

state

| | 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 | | | | | | | |
| **lunch** | 0.0059 | | | | | | | |
| **spend** | 0.0036 | | | | | | | |

**Figure 3.2** Bigram p ...corpus of 9332 sentences. Zer

Here are a few other

$P(\texttt{i}\,|\,\texttt{<s>}) = 0.25$

$P(\texttt{food}\,|\,\texttt{english})$

Why is this so much lower?

P(<s> i want chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(Chinese | want)
* P(food | Chinese)

= 0.25 * 0.33 * 0.0065 * 0.52

= 0.00027885

P(<s> i want to eat chinese food </s>)
= P( i | <s>)
* P(want | i)
* P(to | want)
* P(eat | to)
* P(Chinese | eat)
* P(food | Chinese)

= 0.25 * 0.33 * 0.66 * 0.0027 * 0.021 * 0.00092

= 0.00000000284

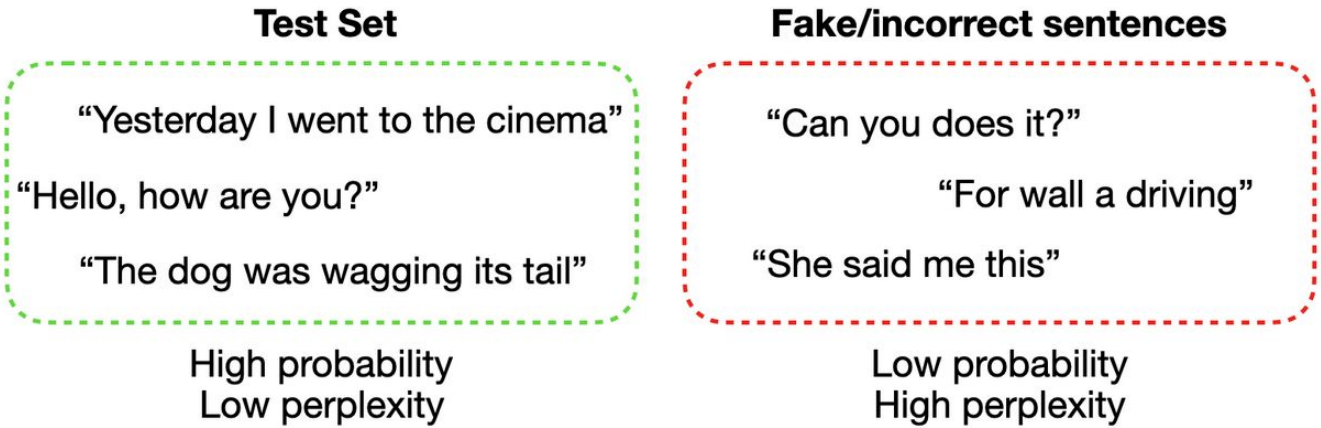
What is the maximum likelihood estimate for "I want Chinese food" vs "I want to eat Chinese food"?

# Evaluating Language Models: Perplexity

- Perplexity is a measure of **how surprised** the model is to see the word that actually comes next
- It takes into account probability (high probability --> low perplexity)
- It also uses **normalization**, so that it is independent of the **number of words** in the dictionary and the **length of the sentence**
- Lower is better

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

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


Test Set
“Yesterday I went to the cinema”
“Hello, how are you?”
“The dog was wagging its tail”
High probability
Low perplexity
Fake/incorrect sentences
“Can you does it?”
“For wall a driving”
“She said me this”
Low probability
High perplexity

# Evaluating Language Models: Extrinsic Evaluation

Extrinsic evaluation can involve regular NLP tasks, or **challenge datasets** specifically designed to require some complex language faculty.

One of our surprisingly complex language faculties is **anaphora** resolution. An anaphor (plural **anaphora**) is a word that refers to another word in the text. Pronouns are anaphora, because the **referent** of a pronoun (real world object) is the same as the referent of its textual **antecedent.**

**“If the con artist has succeeded in fooling Sam, he would have gotten a lot of money.”**

**Q: Who would have gotten a lot of money?**

# Evaluation considerations: training vs testing

**training set:** The data we use to learn the parameters of our model.

**test set:** different, *held-out* set of data, not overlapping with the training set, that we use to evaluate the model.

# Generalizing vs. Overfitting

- **Overfitting** occurs when a model learns the training data too well—including its noise, biases, and specific details—instead of generalizing patterns that are broadly applicable.
- An overfitted model will perform well on evaluation, but poorly on new data.

- We want a model that can generalize to new, unseen contexts.
  - N-grams are not great at this

# Dataset Contamination

During initial training, web-scraped data often contains unwanted content (e.g., benchmark datasets like GLUE) because of imperfect filtering and deduplication

So, the ARC benchmark (right) might be in the training data for an LLM that is evaluated on it!

| Reasoning Type | Example |
|---|---|
| Question logic | Which item below is **not** made from a material grown in nature? (A) a cotton shirt (B) a wooden chair (C) a plastic spoon (D) a grass basket |
| Linguistic Matching | Which of the following best describes a mineral? (A) the main nutrient in all foods (B) a type of grain found in cereals (C) a natural substance that makes up rocks (D) the decomposed plant matter found in soil |
| Multihop Reasoning | Which property of a mineral can be determined just by looking at it? (A) luster (B) mass (C) weight (D) hardness |
| Comparison | Compared to the Sun, a red star most likely has a greater (A) volume. (B) rate of rotation. (C) surface temperature. (D) number of orbiting planets |
| Algebraic | If a heterozygous smooth pea plant (Ss) is crossed with a homozygous smooth pea plant (SS), which are the possible genotypes the offspring could have? (A) only SS (B) only Ss (C) Ss or SS (D) ss or SS |
| Hypothetical / Counterfactual | If the Sun were larger, what would most likely also have to be true for Earth to sustain life? (A) Earth would have to be further from the Sun. (B) Earth would have to be closer to the Sun. (C) Earth would have to be smaller. (D) Earth would have to be larger. |
| Explanation / Meta-reasoning | Why can steam be used to cook food? (A) Steam does work on objects. (B) Steam is a form of water. (C) Steam can transfer heat to cooler objects. (D) Steam is able to move through small spaces. |
| Spatial / Kinematic | Where will a sidewalk feel hottest on a warm, clear day? (A) Under a picnic table (B) In direct sunlight (C) Under a puddle (D) In the shade |
| Analogy | Inside cells, special molecules carry messages from the membrane to the nucleus. Which body system uses a similar process? (A) endocrine system (B) lymphatic system (C) excretory system (D) integumentary system |

*Example questions from the ARC Challenge Set. Credit: Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*

# Recursive Training of Language models

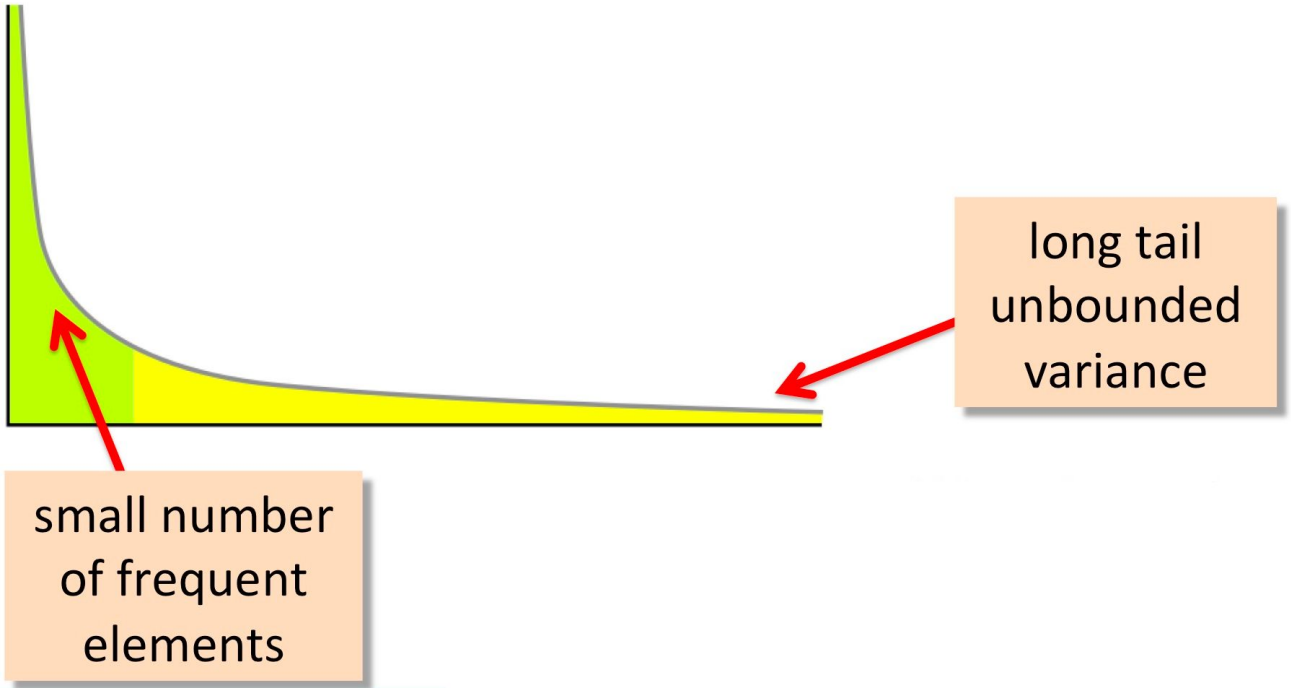
Q: Say we build a new conditional probability distribution (a new model) based on the outputs of your LM. How would the distribution / histogram of the new model differ from the original one?

Think about what kinds of sentences it generates and what kinds it doesn't.

# The long tail of the distribution

Remember Zipf's law?

It applies to bigrams and trigrams and N-grams too.



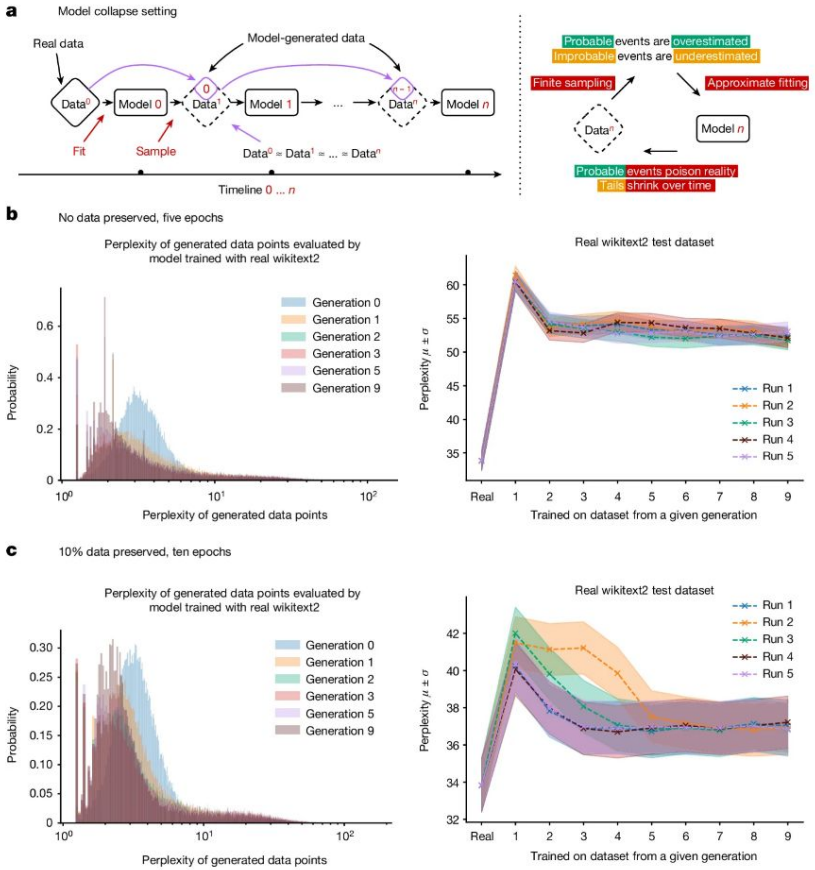
# Model Collapse (Shumailo et al. 2024)

"a degenerative learning process in which models start forgetting improbable events over time, as the model becomes poisoned with its own projection of reality"

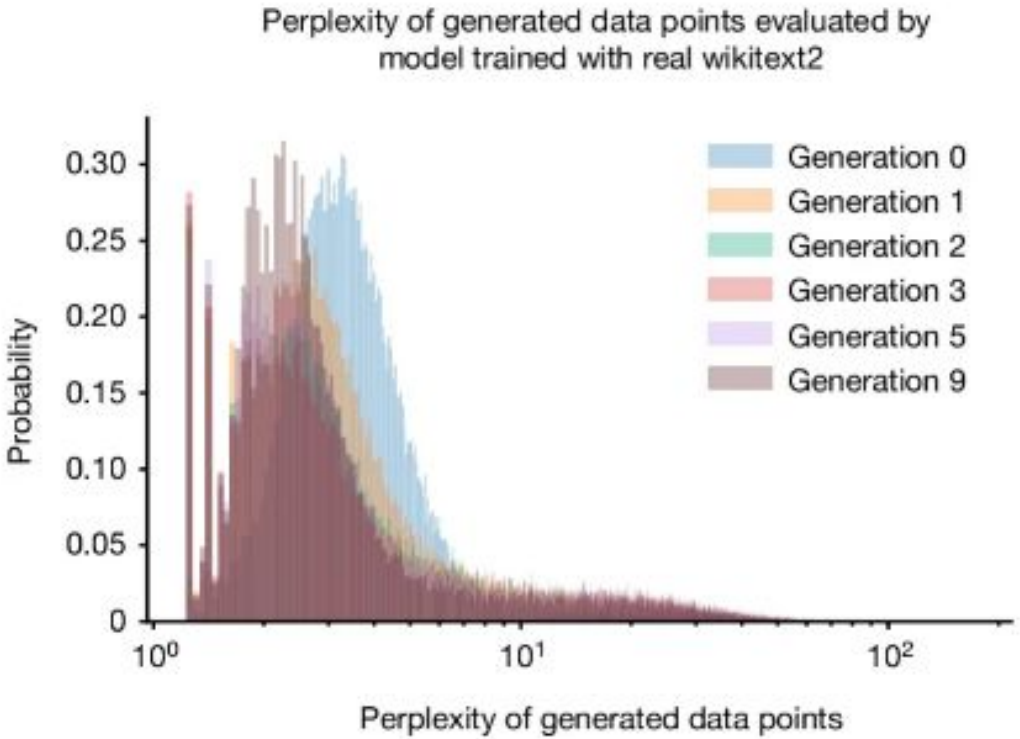


**Fig. 1: The high-level description of the feedback mechanism in the learning process.**

From: AI models collapse when trained on recursively generated data



The model starts producing more text that is highly probable in the dataset



# The Chain Rule of Probability

- A way of relating joint probabilities to conditional probabilities

$$P(X_1 \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_{1:2}) \ldots P(X_n|X_{1:n-1})$$
$$= \prod_{k=1}^{n} P(X_k|X_{1:k-1}) \quad (3.3)$$

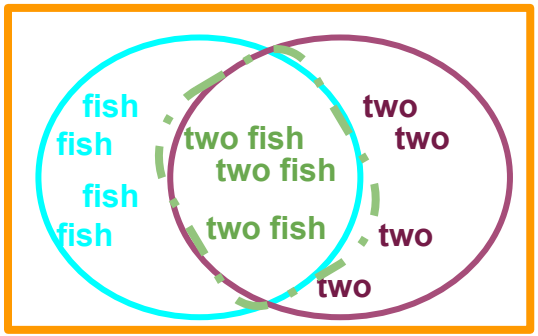
Applying the chain rule to words, we get

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \ldots P(w_n|w_{1:n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_{1:k-1}) \quad (3.4)$$

# Review Conditional vs Joint Probability

joint probability:

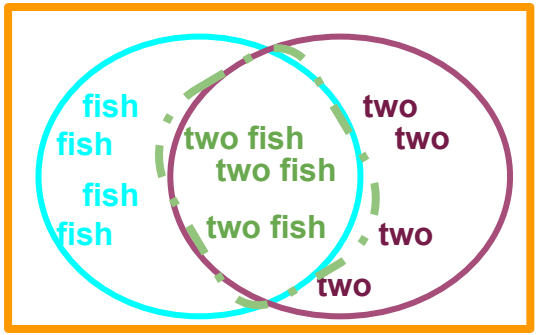
P ("two" ∩ "fish" )




P ( fish ∩ two ) = ——

conditional probability:

P ( "fish" | "two" )




P ( fish | two ) = ——