Language Modeling and N-Grams

Ling 282/482: Deep Learning for Computational Linguistics

C.M. Downey

Fall 2025

Language Modeling (task)

Language Modeling

Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**

Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**
  - This will **usually** mean predicting the distribution of **words in a sentence**

Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**
  - This will **usually** mean predicting the distribution of **words in a sentence**
- Formally, we want to know $P(w_1, w_2, w_3 \dots w_k)$

Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**
    - This will **usually** mean predicting the distribution of **words in a sentence**
- Formally, we want to know $P(w_1, w_2, w_3 \ldots w_k)$
    - Where $w_{1:k}$ are the **words** of a particular sentence/sequence

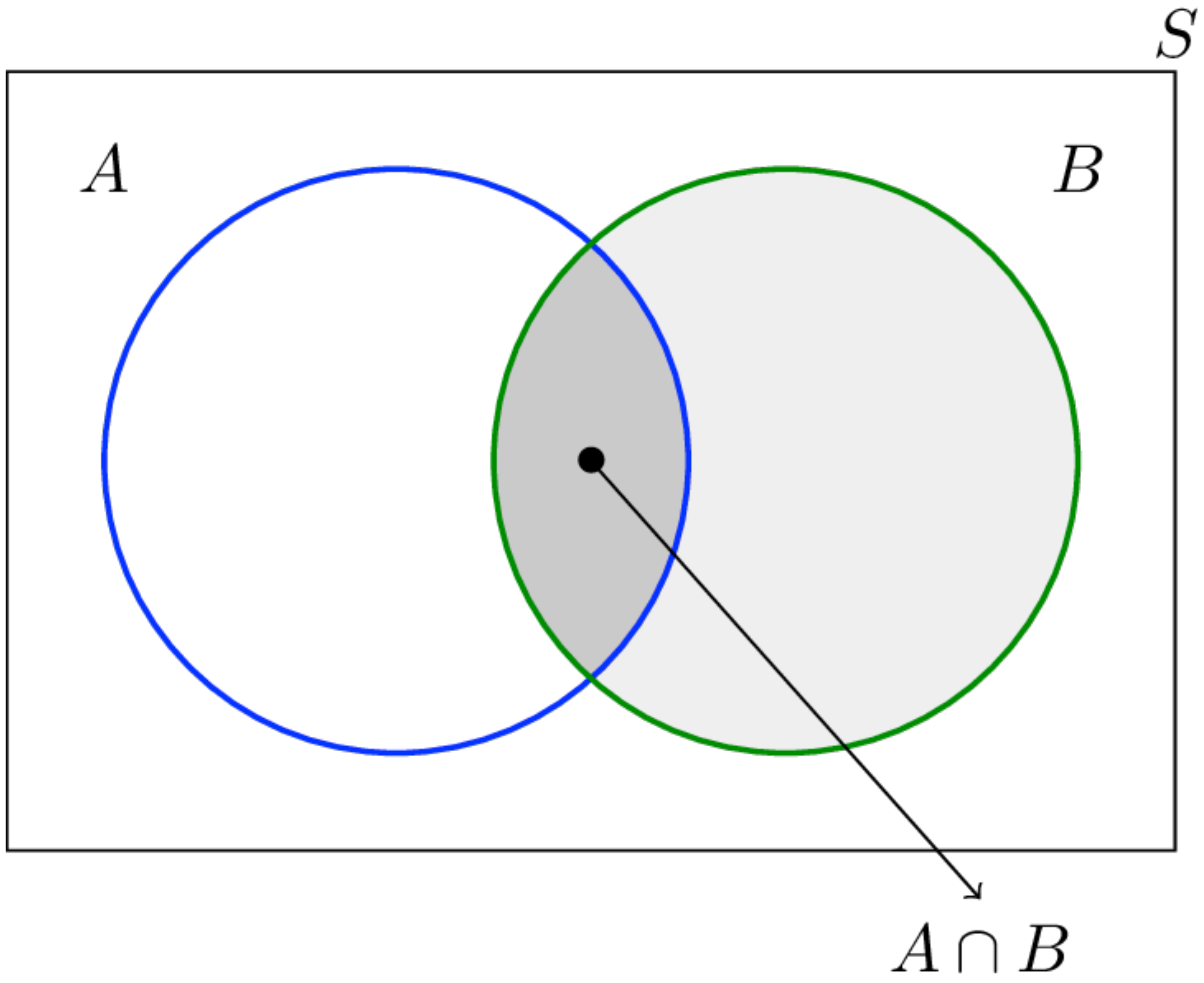
Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**
  - This will **usually** mean predicting the distribution of **words in a sentence**
- Formally, we want to know $P(w_1, w_2, w_3 \ldots w_k)$
  - Where $w_{1:k}$ are the **words** of a particular sentence/sequence
  - This is shorthand for the **joint probability** of seeing these words **together and in this order**

Language Modeling

- As seen in the course intro, a Language Model makes **probabilistic predictions** about **symbols in a sequence**
    - This will **usually** mean predicting the distribution of **words in a sentence**
- Formally, we want to know $P(w_1, w_2, w_3 \ldots w_k)$
    - Where $w_{1:k}$ are the **words** of a particular sentence/sequence
    - This is shorthand for the **joint probability** of seeing these words **together and in this order**
    - This is a simplification of notation, as you might notice if taking LING 214. More precise would be something like $P(X_1 = w_1, X_2 = w_2 \ldots X_k = w_k)$

Briefly: Conditional Probability



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

Chain Rule of Probability

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**
  - $P(x_1 \wedge x_2 \ldots \wedge x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1 \wedge x_2) \ldots P(x_k \mid x_{1:k-1})$

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**
  - $P(x_1 \wedge x_2 \ldots \wedge x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1 \wedge x_2)\ldots P(x_k \mid x_{1:k-1})$
  - Key idea: $P(x_1 \wedge x_2) = P(x_1)P(x_2 \mid x_1) = P(x_2)P(x_1 \mid x_2)$

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**
  - $P(x_1 \wedge x_2 \ldots \wedge x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1 \wedge x_2) \ldots P(x_k \mid x_{1:k-1})$
  - Key idea: $P(x_1 \wedge x_2) = P(x_1)P(x_2 \mid x_1) = P(x_2)P(x_1 \mid x_2)$
  - This is a **recursive** definition: to add another variable, you add another **conditional probability**

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**
  - $P(x_1 \wedge x_2 \ldots \wedge x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1 \wedge x_2) \ldots P(x_k \mid x_{1:k-1})$
  - Key idea: $P(x_1 \wedge x_2) = P(x_1)P(x_2 \mid x_1) = P(x_2)P(x_1 \mid x_2)$
  - This is a **recursive** definition: to add another variable, you add another **conditional probability**
  - The Chain Rule does **NOT** say anything about order

Chain Rule of Probability

- In general, true joint probabilities like $P(w_{1:n})$ are **very hard to compute!**
  - There are all kinds of **complex relationships** between words in a sequence
- The **Chain Rule** (of Probability) lets us break joint probabilities into **conditional probabilities**
  - $P(x_1 \wedge x_2 \ldots \wedge x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1 \wedge x_2) \ldots P(x_k \mid x_{1:k-1})$
  - Key idea: $P(x_1 \wedge x_2) = P(x_1)P(x_2 \mid x_1) = P(x_2)P(x_1 \mid x_2)$
  - This is a **recursive** definition: to add another variable, you add another **conditional probability**
  - The Chain Rule does **NOT** say anything about order
- For Language Modeling: what is the probability of the next word **given the previous words**

Chain Rule Example

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\mathit{The}, \mathit{calico}, \mathit{cat}, \mathit{sits})$?

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\textit{The}, \textit{calico}, \textit{cat}, \textit{sits})$?
  - **Decompose** using the Chain Rule

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\textit{The}, \textit{calico}, \textit{cat}, \textit{sits})$?
  - **Decompose** using the Chain Rule
  - $P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{The calico}) \cdot P(\textit{sits} \mid \textit{The calico cat})$

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\textit{The}, \textit{calico}, \textit{cat}, \textit{sits})$?
  - **Decompose** using the Chain Rule
  - $P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{The calico}) \cdot P(\textit{sits} \mid \textit{The calico cat})$
- What does this buy us?

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\textit{The}, \textit{calico}, \textit{cat}, \textit{sits})$?
  - **Decompose** using the Chain Rule
  - $P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{The calico}) \cdot P(\textit{sits} \mid \textit{The calico cat})$
- What does this buy us?
  - Breaks a hard problem into **tractable sub-problems**

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\textit{The}, \textit{calico}, \textit{cat}, \textit{sits})$?
  - **Decompose** using the Chain Rule
  - $P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{The calico}) \cdot P(\textit{sits} \mid \textit{The calico cat})$
- What does this buy us?
  - Breaks a hard problem into **tractable sub-problems**
  - Allows the model to **make predictions word-by-word**

Chain Rule Example

- Let's apply the chain rule to a sentence: "The calico cat sits"
- What is the **joint probability** $P(\mathit{The}, \mathit{calico}, \mathit{cat}, \mathit{sits})$?
  - **Decompose** using the Chain Rule
  - $P(\mathit{The}) \cdot P(\mathit{calico} \mid \mathit{The}) \cdot P(\mathit{cat} \mid \mathit{The\ calico}) \cdot P(\mathit{sits} \mid \mathit{The\ calico\ cat})$
- What does this buy us?
  - Breaks a hard problem into **tractable sub-problems**
  - Allows the model to **make predictions word-by-word**
- But how do we get these **conditional probabilities?**

Counting-based Language Models

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

Counting-based Language Models

- **Given some prefix of words**, how do we know the **probability that a specific word will follow?** (see example below)

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

Counting-based Language Models

- **Given some prefix of words**, how do we know the **probability that a specific word will follow?** (see example below)
- Naive solution: **count** the number of times it occurs in your data!
  - $C(\dots)$ indicates the count of the string
  - Equation below gives the **conditional probability** of the following word

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

Counting-based Language Models

- **Given some prefix of words**, how do we know the **probability that a specific word will follow?** (see example below)
- Naive solution: **count** the number of times it occurs in your data!
  - $C(\ldots)$ indicates the count of the string
  - Equation below gives the **conditional probability** of the following word
- **But...** how many times can we actually expect to encounter that prefix?
  - Not many! This probably gives a **poor estimation** of the conditional prob.

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

N-Grams

Motivation

Motivation

- Problem: as sequences get **long**, they also get **incredibly rare**
  - Makes it **intractable to observe/count** all of the words that can come next
  - The "probability" of **most possible sequences** will be zero!

Motivation

- Problem: as sequences get **long**, they also get **incredibly rare**
  - Makes it **intractable to observe/count** all of the words that can come next
  - The "probability" of **most possible sequences** will be zero!
- Solution: make **simplifying assumptions** about model

Motivation

- Problem: as sequences get **long**, they also get **incredibly rare**
  - Makes it **intractable to observe/count** all of the words that can come next
  - The "probability" of **most possible sequences** will be zero!
- Solution: make **simplifying assumptions** about model
- **Markov assumption:** probabilities are only conditioned on a **finite history**
  - i.e. **don't look too far** into the past

Motivation

- Problem: as sequences get **long**, they also get **incredibly rare**
  - Makes it **intractable to observe/count** all of the words that can come next
  - The "probability" of **most possible sequences** will be zero!
- Solution: make **simplifying assumptions** about model
- **Markov assumption:** probabilities are only conditioned on a **finite history**
  - i.e. **don't look too far** into the past
- For LMs, often called the **n-gram assumption**
  - Only use **a few previous words** to predict the current word!

Definition

Definition

- n-gram: a **specific sequence of n words** (or other symbols)
  - n-gram probability: the **conditional probability** of the **n^{th} word** given the **previous n-1 words**

Definition

- n-gram: a **specific sequence of n words** (or other symbols)
  - n-gram probability: the **conditional probability** of the **n^{th} word** given the **previous n-1 words**
- **Bigram** ("2-gram"): sequence of **two words**, e.g. $P(cat \mid calico)$

Definition

- n-gram: a **specific sequence of n words** (or other symbols)
  - n-gram probability: the **conditional probability** of the $\mathbf{n^{th}}$ **word** given the **previous n-1 words**
- **Bigram** ("2-gram"): sequence of **two words**, e.g. $P(cat \mid calico)$
- **Trigram** ("3-gram"): sequence of **three words**, e.g. $P(cat \mid the\ calico)$

Definition

- n-gram: a **specific sequence of n words** (or other symbols)
  - n-gram probability: the **conditional probability** of the **nth word** given the **previous n-1 words**
- **Bigram** ("2-gram"): sequence of **two words**, e.g. $P(cat \mid calico)$
- **Trigram** ("3-gram"): sequence of **three words**, e.g. $P(cat \mid the\ calico)$
- Any other size would be called 4-gram, 5-gram, etc.

Definition

- n-gram: a **specific sequence of n words** (or other symbols)
  - n-gram probability: the **conditional probability** of the **nth word** given the **previous n-1 words**
- **Bigram** ("2-gram"): sequence of **two words**, e.g. $P(cat \mid calico)$
- **Trigram** ("3-gram"): sequence of **three words**, e.g. $P(cat \mid the\ calico)$
- Any other size would be called 4-gram, 5-gram, etc.
- The **value of n** is left as an **engineering choice**
  - **However,** for large n we run into the **exact same rarity problem** as before

Bigram Approximation

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$
- Plugging this into our calico example:

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$
- Plugging this into our calico example:
  - $P(\textit{sits} \mid \textit{The calico cat}) \approx P(\textit{sits} \mid \textit{cat})$

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \,|\, w_{1:i-1}) \approx P(w_i \,|\, w_{i-1})$
- Plugging this into our calico example:
  - $P(\textit{sits} \,|\, \textit{The calico cat}) \approx P(\textit{sits} \,|\, \textit{cat})$
  - $P(\textit{The calico cat sits}) \approx P(\textit{The}) \cdot P(\textit{calico} \,|\, \textit{The}) \cdot P(\textit{cat} \,|\, \textit{calico}) \cdot P(\textit{sits} \,|\, \textit{cat})$

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$
- Plugging this into our calico example:
  - $P(\textit{sits} \mid \textit{The calico cat}) \approx P(\textit{sits} \mid \textit{cat})$
  - $P(\textit{The calico cat sits}) \approx P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{calico}) \cdot P(\textit{sits} \mid \textit{cat})$
- Okay... but how do we get these **bigram probabilities?**

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$
- Plugging this into our calico example:
  - $P(\textit{sits} \mid \textit{The calico cat}) \approx P(\textit{sits} \mid \textit{cat})$
  - $P(\textit{The calico cat sits}) \approx P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{calico}) \cdot P(\textit{sits} \mid \textit{cat})$
- Okay... but how do we get these **bigram probabilities?**
  - Simple: we'll bring back **counting occurrences!**

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

Bigram Approximation

- If using bigrams, we are **approximating the true probability** as follows:
  - $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$
- Plugging this into our calico example:
  - $P(\textit{sits} \mid \textit{The calico cat}) \approx P(\textit{sits} \mid \textit{cat})$
  - $P(\textit{The calico cat sits}) \approx P(\textit{The}) \cdot P(\textit{calico} \mid \textit{The}) \cdot P(\textit{cat} \mid \textit{calico}) \cdot P(\textit{sits} \mid \textit{cat})$
- Okay... but how do we get these **bigram probabilities?**
  - Simple: we'll bring back **counting occurrences!**
  - The prob. of **all possible following** w_i will **sum to 1.0**

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

Practice Example

- What are all the **bigram probabilities** in the following dataset?
  - (<s> and </s> are special symbols meaning **beginning/end of sequence**)

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

Practice Example

- What are all the **bigram probabilities** in the following dataset?
  - (<s> and </s> are special symbols meaning **beginning/end of sequence**)

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

$$P(\texttt{I}|\texttt{<s>}) = \frac{2}{3} = 0.67 \qquad P(\texttt{Sam}|\texttt{<s>}) = \frac{1}{3} = 0.33 \qquad P(\texttt{am}|\texttt{I}) = \frac{2}{3} = 0.67$$

$$P(\texttt{</s>}|\texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam}|\texttt{am}) = \frac{1}{2} = 0.5 \qquad P(\texttt{do}|\texttt{I}) = \frac{1}{3} = 0.33$$

Real 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

Figure 3.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray. Each cell shows the count of the column label word following the row label word. Thus the cell in row **i** and column **want** means that **want** followed **i** 827 times in the corpus.

Real Bigram Probabilities

	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.