NPTEL ONLINE CERTIFICATION COURSES

DEEP LEARNING FOR NATURAL LANGUAGE PROCESSING

Lecture 03 : N-gram Language Models: Part 1

PROF.PAWAN GOYAL
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

CONCEPTS COVERED

- What is Language Modeling? (LM in LLMs!!)
- N-gram Language Models
- Some Practical Issues

Predicting words

- The water of Walden Pond is beautifully ...

blue
green
clear

*refrigerator
*that

Source: https://web.stanford.edu/~jurafsky/slp3.

Language Models

Systems that can predict upcoming words

- Can assign a probability to each potential next word
- Can assign a probability to a whole sentence

Source: https://web.stanford.edu/~jurafsky/slp3.

Why word prediction?

It's a helpful part of language tasks

- Grammar or spell checking

  Their are two midterms ~~Their~~ There are two midterms

  Everything has improve Everything has ~~improve~~ improved

- Speech recognition

  I will be back soonish I will be bassoon dish

Source: https://web.stanford.edu/~jurafsky/slp3.

Why word prediction?

It's how **large language models (LLMs)** work!

LLMs are **trained** to predict words

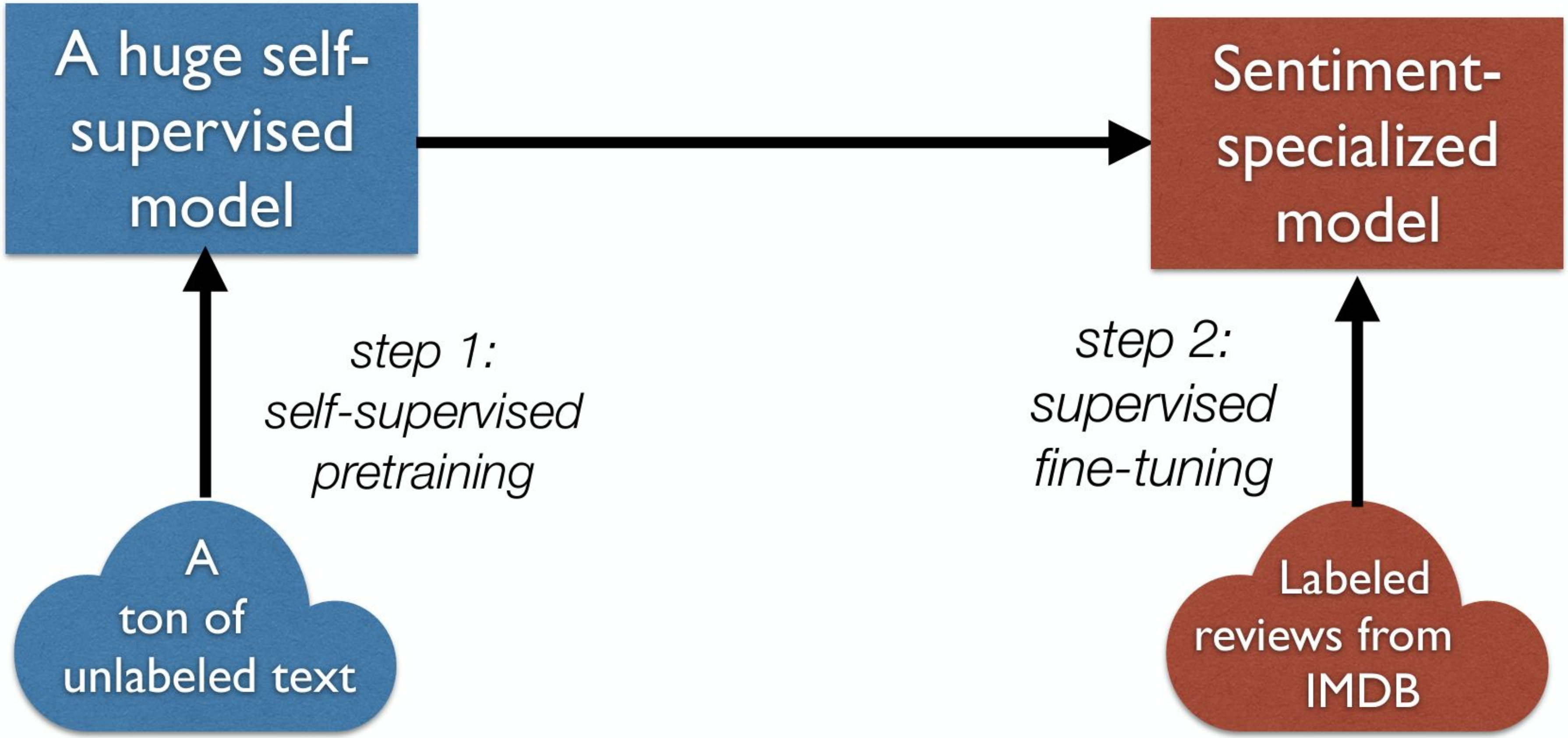
- Left-to-right (autoregressive) LMs learn to predict next word

LLMs **generate** text by predicting words

- By predicting the next word over and over again

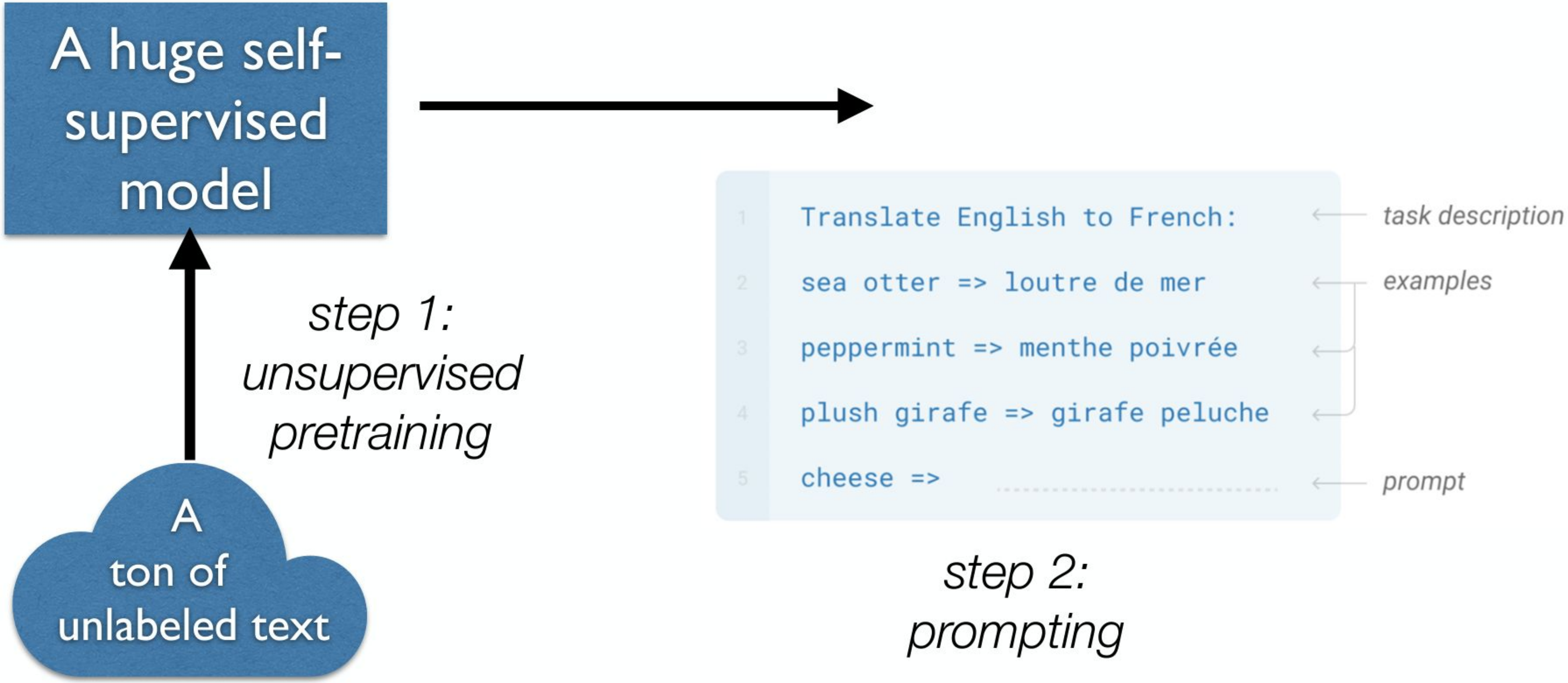
Source: https://web.stanford.edu/~jurafsky/slp3.

Pretrain-then-finetune paradigm



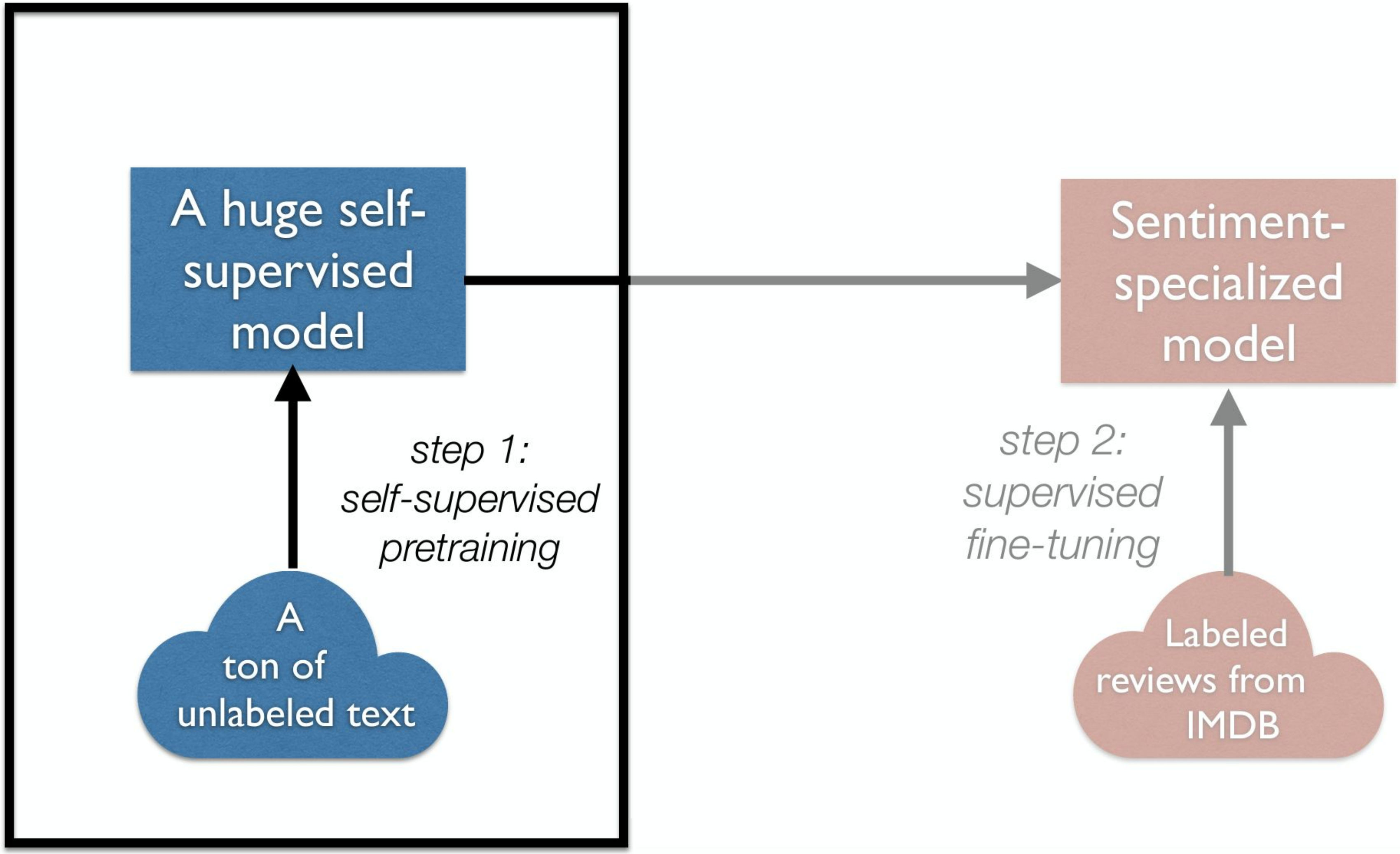
Source: https://people.cs.umass.edu/~miyyer/cs685

Pretrain-then-Prompt paradigm



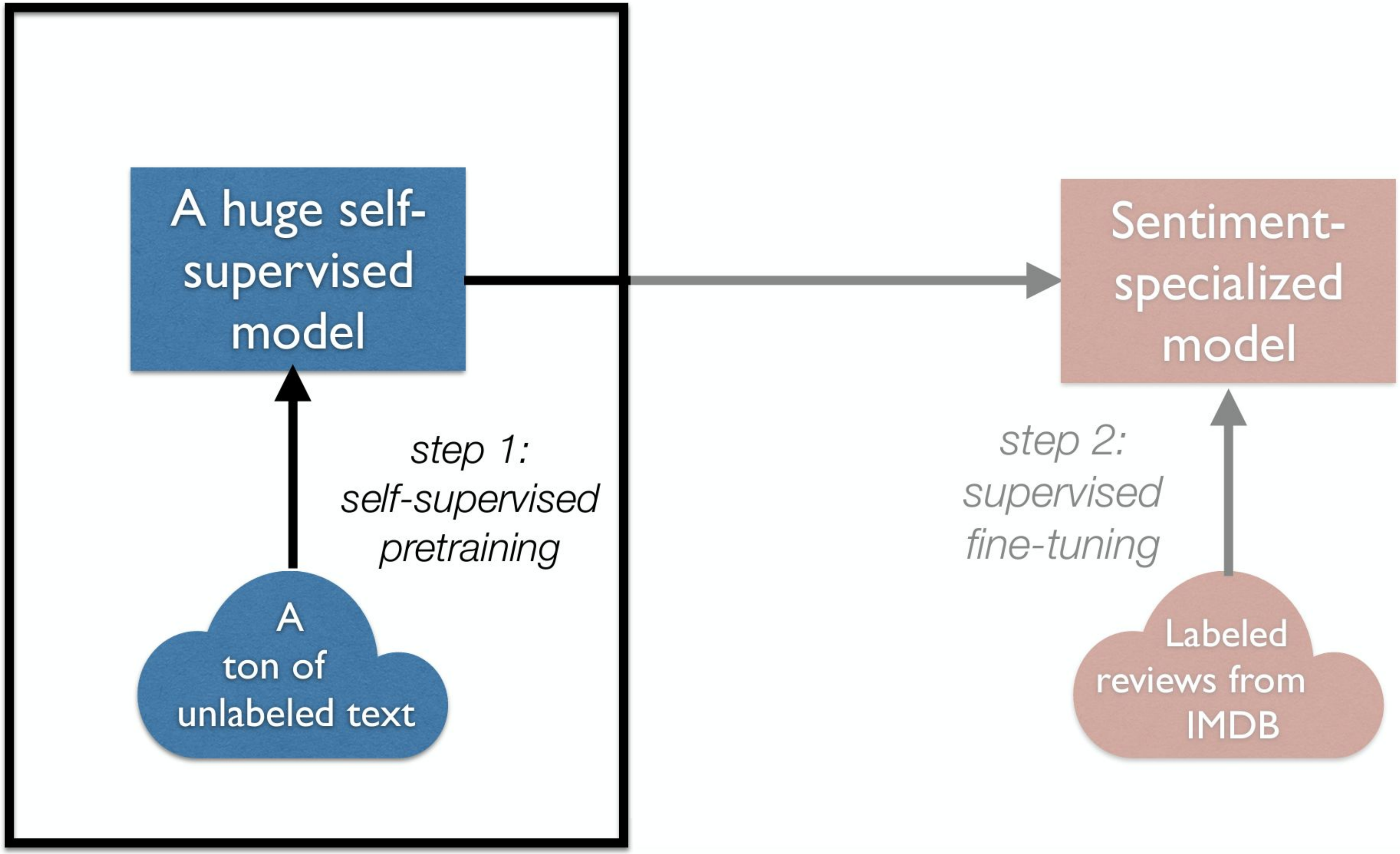
Source: https://people.cs.umass.edu/~miyyer/cs685

Language modeling forms the core of most self-supervised NLP approaches



Source: https://people.cs.umass.edu/~miyyer/cs685

Language modeling forms the core of most self-supervised NLP approaches



Source: https://people.cs.umass.edu/~miyyer/cs685

Language Modeling: More Formally

- **Goal:** Compute the probability of a sentence or sequence of words:

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

- **Related Task:** probability of an upcoming word:

$$P(w_4 | w_1, w_2, w_3)$$

- A model that computes either of these is called a **language model**

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

- 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

Source: https://web.stanford.edu/~jurafsky/slp3.

Reminder: The Chain Rule

- Recall the definition of conditional probabilities

  $P(B|A) = P(A,B)/P(A)$ Rewriting: $P(A,B) = P(A)\,P(B|A)$

- More variables:

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

- The Chain Rule in General

$$P(x_1,x_2,x_3,\ldots,x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1,\ldots,x_{n-1})$$

Source: https://web.stanford.edu/~jurafsky/slp3.

The Chain Rule applied to compute joint probability of words in sentence

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

P(“The water of Walden Pond is so beautifully blue”) =

P(The) × P(water|The) × P(of|The water)

× P(Walden|The water of) ×

P(Pond|The water of Walden) × ….

Source: https://web.stanford.edu/~jurafsky/slp3.

How to estimate these probabilities

- Could we just count and divide?

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

$$=$$

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

- *We'll never see enough data for estimating these!!*

Source: https://web.stanford.edu/~jurafsky/slp3.

Markov Assumption

Andrei Markov

- Simplifying assumption:

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

$$\approx \quad P(\texttt{blue}|\texttt{beautifully})$$

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

Source: https://web.stanford.edu/~jurafsky/slp3.

Wikimedia commons

Bigram Markov Assumption

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

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

More generally, we approximate each component in the product

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

Source: https://web.stanford.edu/~jurafsky/slp3.

Simplest case: Unigram model

$$P(w_1 w_2 \dots \ w_n) \approx \prod_i P(w_i)$$

Some automatically generated sentences from two different 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

Source: https://web.stanford.edu/~jurafsky/slp3.

Bigram model

$$P(w_i \mid w_1 w_2 \ldots\ w_{i-1}) \approx P(w_i \mid w_{i-1})$$

Some automatically generated sentences from two different unigram models

```
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.

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
```

Source: https://web.stanford.edu/~jurafsky/slp3.

Approximating Shakespeare

Unigram
To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
Every enter now severally so, let
Hill he late speaks; or! a more to leg less first you enter
Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like
Bigram
What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?
Trigram
Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
Quadrigram
King Henry.What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
Will you not tell me who I am?
It cannot be but so.
Indeed the short and the long. Marry, 'tis a noble Lepidus.

Source: https://web.stanford.edu/~jurafsky/slp3.

Problems with N-gram models

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

"**The soups** that I made from that new cookbook I bought yesterday **were** amazingly delicious."

- N-grams don't do well at modeling new sequences with similar meanings

The solution: **Large language models**

- can handle much longer contexts
  (because of using embedding spaces)
- can model synonymy better

Source: https://web.stanford.edu/~jurafsky/slp3.

Why N-gram models?

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

- **training** and **test** sets
- the **perplexity** metric
- **sampling** to generate sentences
- ideas like **interpolation** and **backoff**

Source: https://web.stanford.edu/~jurafsky/slp3.

Estimating n-gram probabilities

Maximum Likelihood Estimate

Value that makes the observed data the "most probable"

$$P(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

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

Estimating n-gram probabilities: an Example

Given a corpus C, the bigram probability of "paper | question" is 0.3 and the count of occurrences of the word "question" is 600. What will be the frequency of the pair (question, paper) in the corpus C?

P (paper | question) = freq(question, paper)/freq(question)

freq (question, paper) = 180

An Example

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

<s>I am here </s>
<s>who am I </s>
<s>I would like to know </s>

Estimating bigrams

P(I|<s>) = 2/3
P(</s>|here) =1
P(would | I) = 1/3
P(here | am) = 1/2
P(know | like) = 0

Computing Sentence Probabilities

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

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

Practical Issues

Everything in log space

- Avoids underflow
- Adding is faster than multiplying

$$log(p_1 \times p_2 \times p_3 \times p_4) = logp_1 + logp_2 + logp_3 + logp_4$$

REFERENCES

- Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released August 20, 2024. [Chapter 3]

THANK YOU