Lecture 1: Introduction to Introduction to Natural Language Processing

Cornell Bowers C·IS
Computer Science

Course logistics (+more at the end)

- Course website is up! https://www.cs.cornell.edu/courses/cs4740/2026sp/
- Course policies listed on the webpage.
- Up-to-date schedule and slides will always be available on this webpage.

Instructors:

Tanya Goyal

What a time to work in NLP!!

Start of my PhD v/s End of my PhD

Write a story about an alien who wants to return to his home planet.

Radford++ 2018

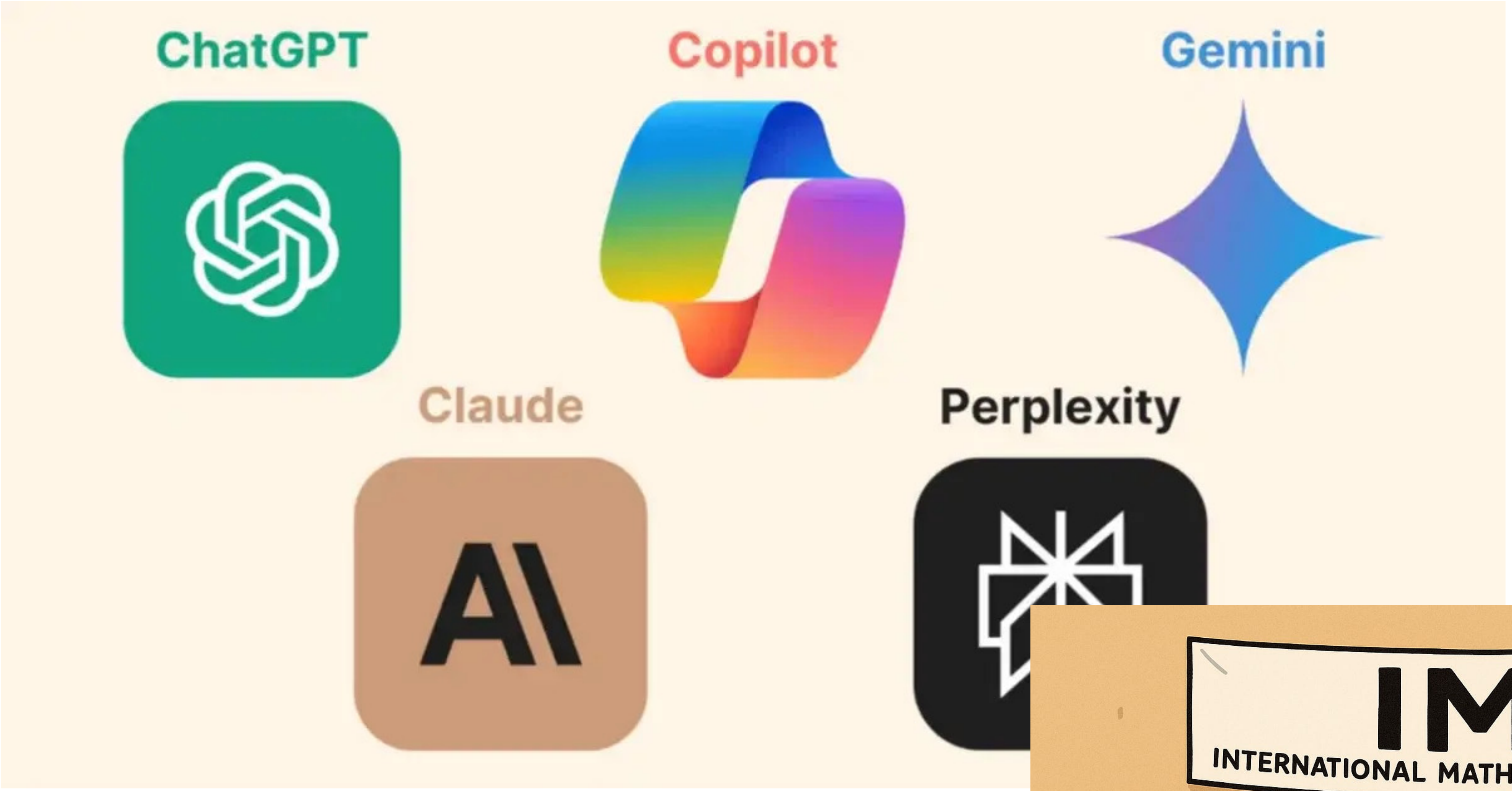
I'm not writing your story! They are already part of this story, but I need to take them a little further, so I can write one for the future book of the watchers saga.

OpenAI's ChatGPT (2022)

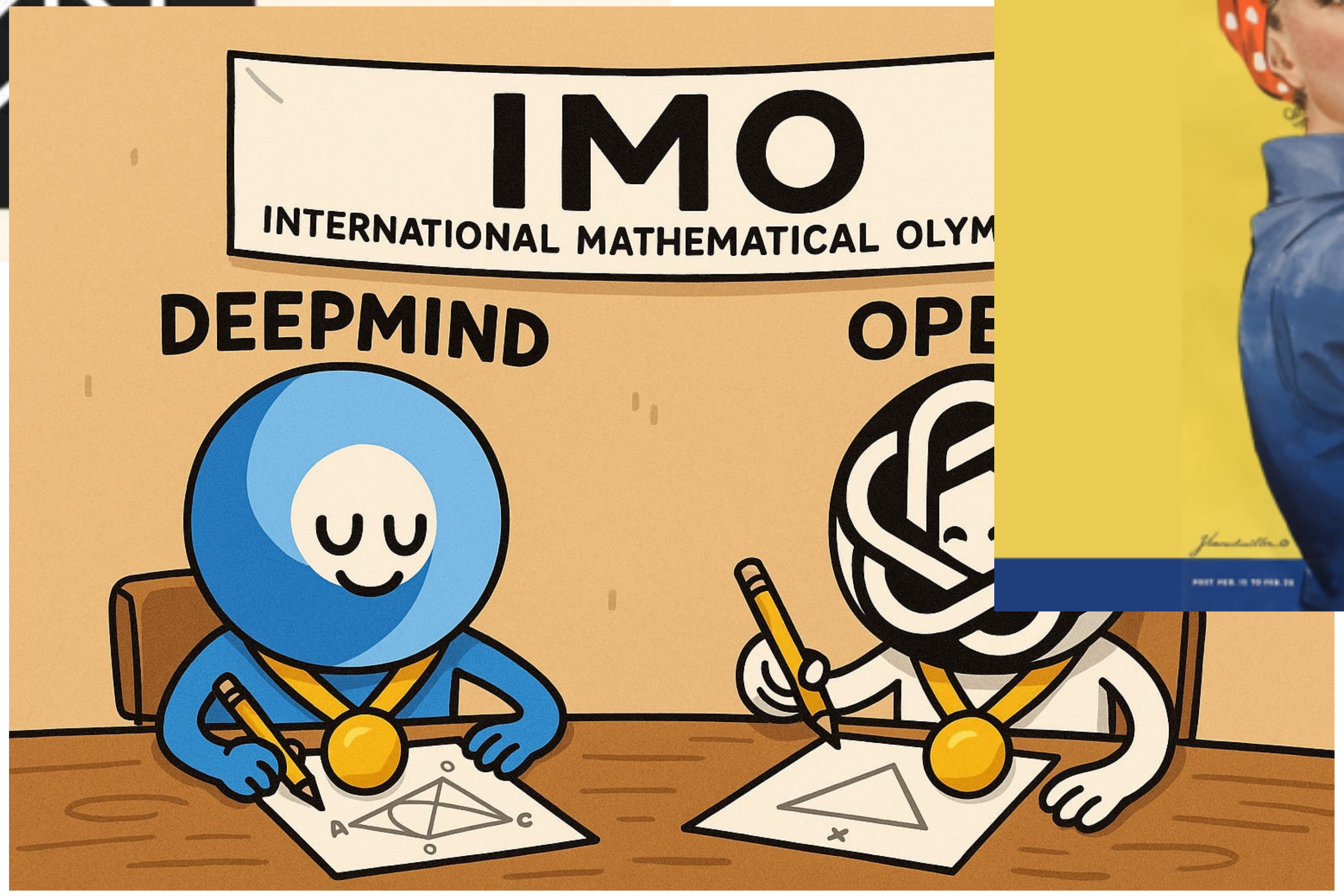
Once upon a time, in a galaxy far, far away, there was an alien named Zor. Zor was a curious and adventurous creature [...] As the years passed, Zor grew more and more homesick [...]

What a time to work in NLP!!

Language Models Today



- Essentially "chatbots" that we can interact with using a question-answering format.



41%

of all code in 2025

This Course:

- What even are Language Models?
- What was NLP before language models? What worked and didn't work?
- How do we train Language Models?
- How do we specialize Language Models for any task?
- How do we test if our Language Models are any good?

Today

- What is NLP?
- Why is NLP hard? Classical Perspective.
- Language Modeling 101
- Course Outline
- More Administrative Stuff.

What is NLP anyway?

Fundamental Goal: Build technologies to solve tasks requiring a deep understanding of natural language.

What is NLP anyway?

Fundamental Goal: Build technologies to solve tasks requiring a deep understanding of natural language.

Languages we use to communicate with each other.

What is NLP anyway?

Fundamental Goal: Build technologies to solve tasks requiring a deep understanding of natural language.

Languages we use to communicate with each other.

Process/interpret/communicate as well as humans (or better?)

What is NLP anyway?

Fundamental Goal: Build technologies to solve tasks requiring a deep understanding of natural language.

Languages we use to communicate with each other.

Process/interpret/communicate as well as humans (or better?)

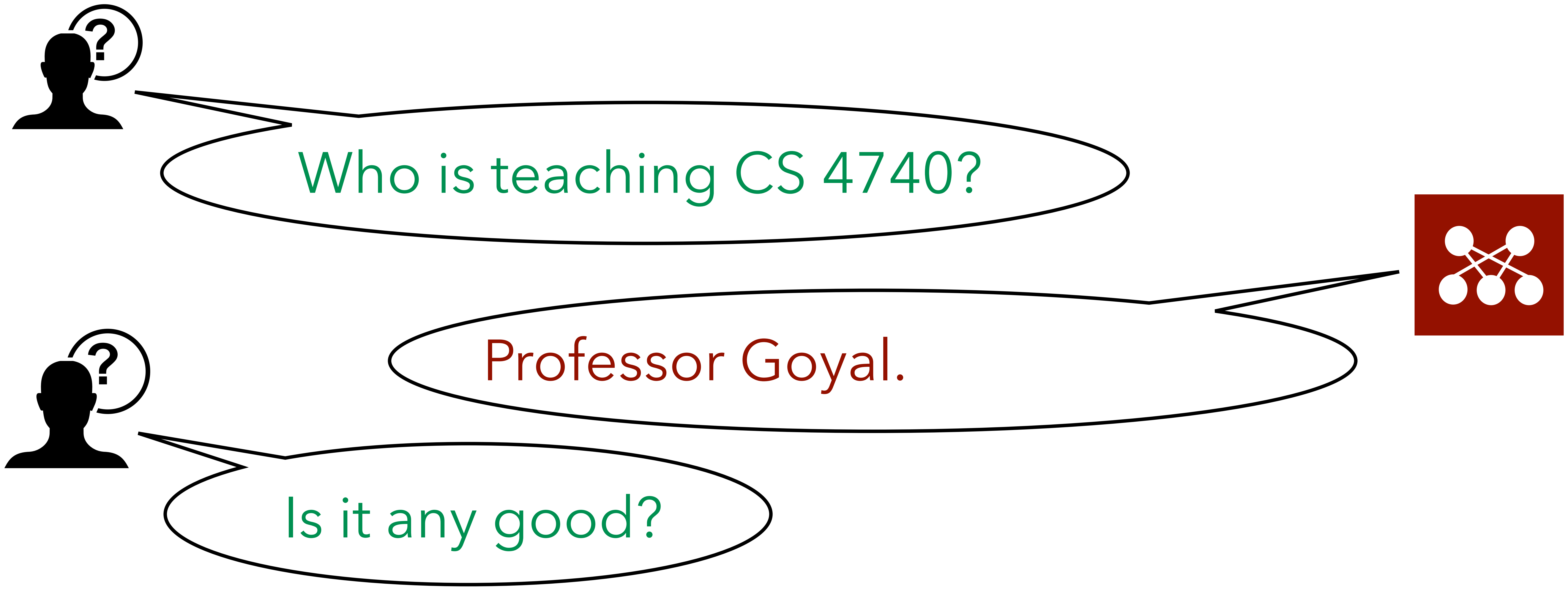
Any task with text inputs and/or outputs is in scope.

NLP tasks

All tasks where either the input X and/or the output Y is text is in scope.

Help us communicate with machines.

E.g. Dialogue systems, question answering, etc.



NLP tasks

All tasks where either the input X and/or the output Y is text is in scope.

Help us transform text.

E.g. Machine translation, grammar correction, summarize etc.

जाने-माने वैज्ञानिक सिवान के. को भारतीय अंतरिक्ष
अनुसंधान संगठन (इसरो) का अध्यक्ष नियुक्त किया गया है।

Translate

New Delhi: Noted scientist Sivan K was appointed Chairman of the Indian Space Research Organisation on Wednesday.

NLP tasks

All tasks where either the input X and/or the output Y is text is in scope.

Help us understand and analyze text corpora or language.

E.g. syntactic analysis, text classification, topic modeling etc.

"I absolutely loved waiting three hours in line for the worst meal of my life."

$\xrightarrow{\text{Sentiment Classification}}$ Negative

NLP tasks

All tasks where either the input X and/or the output Y is text is in scope.

Help us understand and analyze text corpora or language.

E.g. syntactic analysis, text classification, topic modeling etc.

"What do Vegans do in their Spare Time? Latent Interest Detection in Multi-Community Networks", Hessel et al., 2015

	Top Interests	Latent Interests
Vegans	diet, food, cooking, animal, flora	Anarchism, yoga, VegRecipes, Feminism, bicycling, [...]

Why is NLP hard? Ambiguity

"John went to the bank."

Two different meanings of the word bank.

Why is NLP hard? Ambiguity

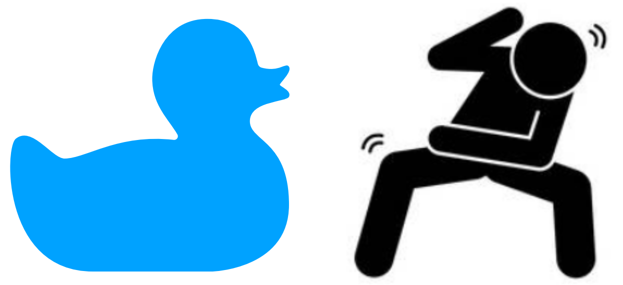
"Retrieve all the local patient files."

Retrieve all the local patient files.

Retrieve all the local patient files.

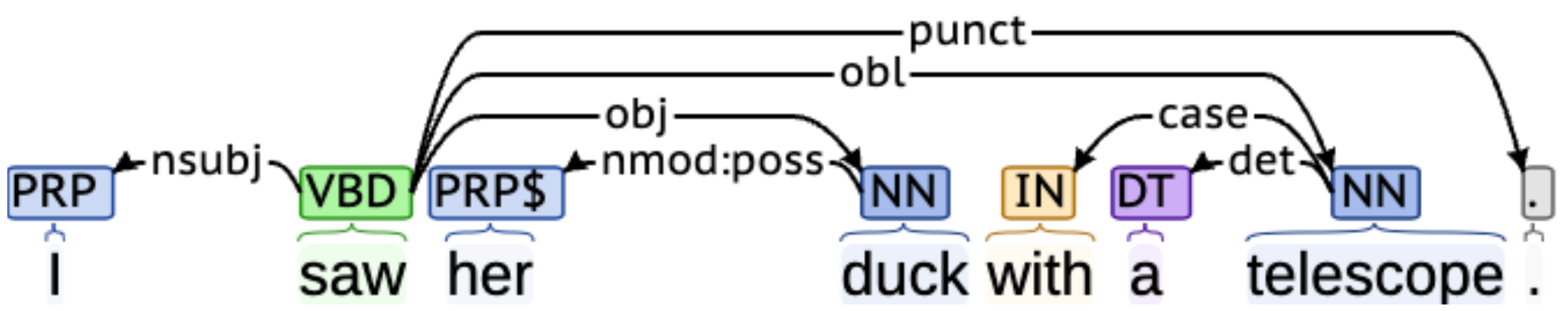
Syntactic ambiguity: what modifies what?

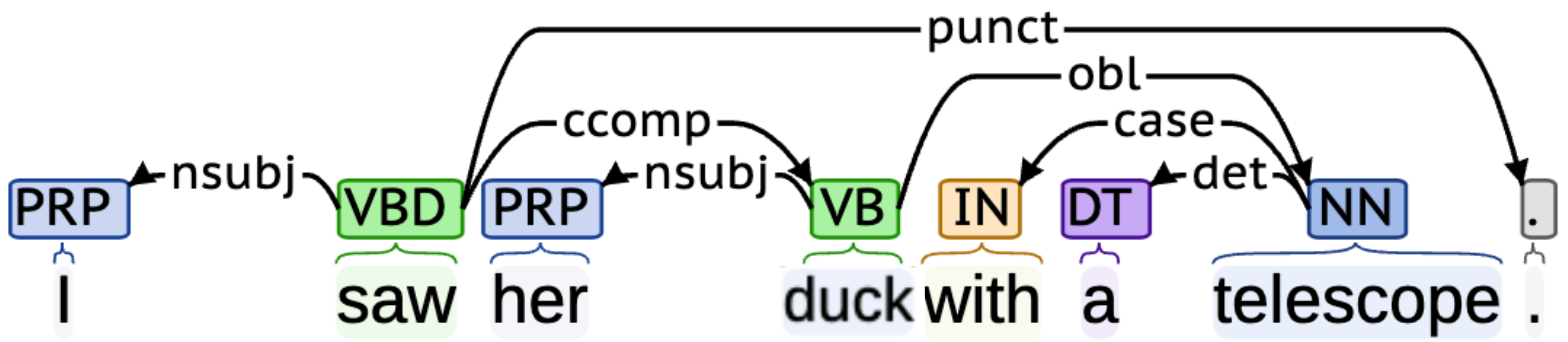
Why is NLP hard? Ambiguity

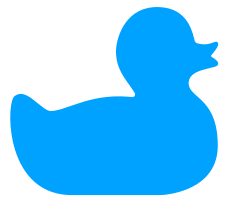
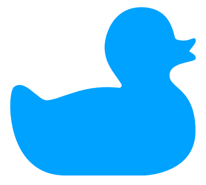


"I saw her duck with a telescope."

How many possible interpretations of this can you think of?





- I used a telescope to see her duck 
- I used a telescope to see her duck 
- I saw her  who had a telescope.
- I saw her  with a telescope in hand.

Why is NLP hard? Ambiguity

- Cases that are easy for humans can be ambiguous for models.

Susan

Susan knows all about Ann's personal problems because she is nosy.

Ann

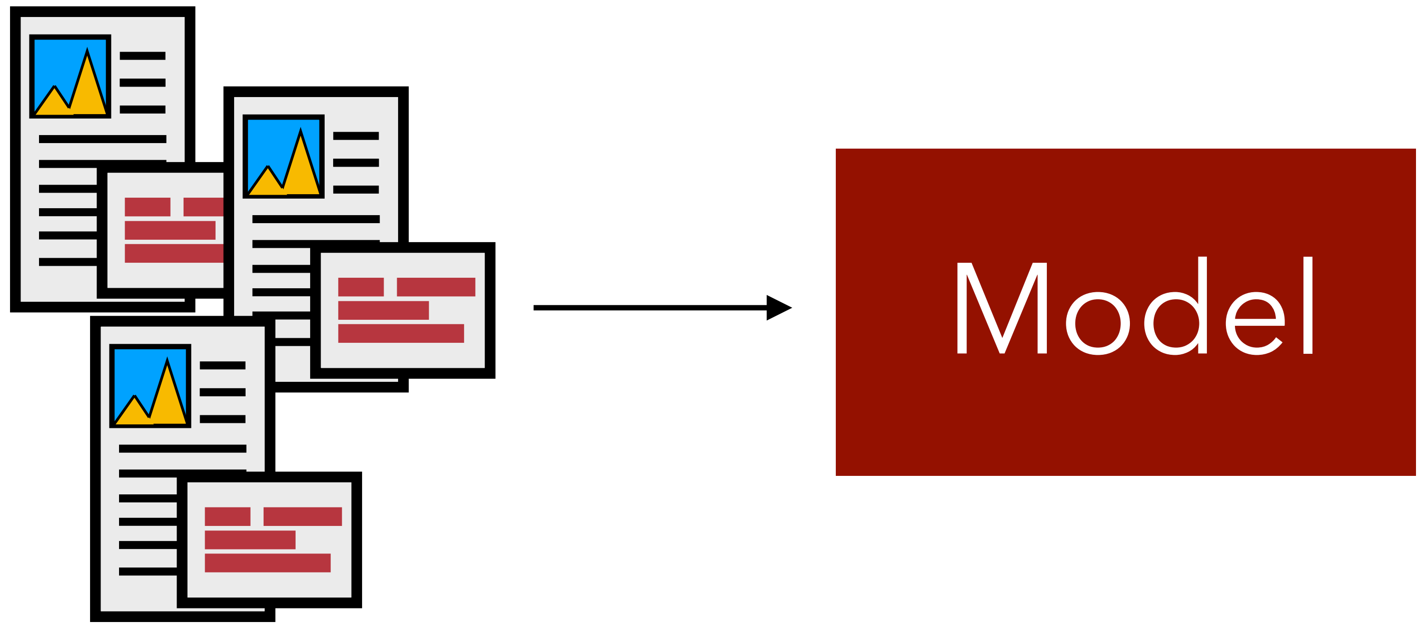
Susan knows all about Ann's personal problems because she is indiscreet.

Coreference resolution: Who is she?

- Easy for humans to resolve given the context, but difficult for statistical models. Why?

Why is NLP hard? Data

- NLP models learn from data.



- Impossible to include data points corresponding every possible linguistic phenomenon and edge case in this data.
- Models can struggle to learn rare phenomenon.
- This is true even for large language models today that are trained on terabytes of data.

NLP in Today's World

- Shift from **classical NLP**
  - We trained one model for each task (separate model for summarization, separate model for parsing, for QA).
  - We trained on task-specific data
- to **language modeling**
  - We **one-size-fits-all** models trained to "model" language.
  - The same models are expected to be good at all tasks – coding, math, writing, translation, etc.

[what are some ways in which you have used ChatGPT?]

Is NLP now solved with ChatGPT et al.?

NLP is not "solved"

- Errors you have noticed with ChatGPT/ Claude?
  - Is it always factually correct?

Generate a biography for Claire Cardie.

OpenAI's ChatGPT

Claire Cardie is a computer scientist and professor […] Cardie earned her Ph.D. in computer science from the University of Pennsylvania, where she developed a strong foundation […]

NLP is not "solved"

- Errors you have noticed with ChatGPT/ Claude?
  - Is it always factually correct?
  - Does it always give up-to-date information?

Who is the current president of United States?

OpenAI's ChatGPT

The current President of the United States is Joe Biden. He has been in office since January 20, 2021.

NLP is not "solved"

- Errors you have noticed with ChatGPT/Claude?
  - Is it always factually correct?
  - Does it always give up-to-date information?
  - What about our favorite parsing examples?

Generate the dependency parse of "Susan knows all about Ann's personal problems because she is indiscreet."

OpenAI's ChatGPT

[…] "she" is the subject of the subordinate clause, referring back to Susan […]

NLP is not "solved"

- Errors you have noticed with ChatGPT/Claude?
  - Is it always factually correct?
  - Does it always give up-to-date information?
  - What about our favorite parsing examples?
  - +reasoning, coding, creative writing, etc.

Today

- What is NLP?
- Why is NLP hard? Classical Perspective.
- **Language Modeling 101**
- Course Outline
- More Administrative Stuff.

What is a Language Model?

- A model that computes a probability distribution over **any** sequence of words:

$$P(w_1 w_2 w_3 \dots w_n)$$

legacy example from Cornell NLP course.

e.g.

$P(\text{Mayenne ate my shoes today.}) = 10^{-12}$

$P(\text{Mayenne ate my}) = 10^{-9}$

$P(\text{I ate dinner in Collegetown.}) = 2 \times 10^{-10}$

$P(\text{Collegetown Bagels slaps.}) = 10^{-14}$

Q: Why would we ever want to do this?

Language Models' Use

- Grammar Error Correction

  $P($ *You're nice.* $)$ $>>$ $P($ *Your nice.* $)$

- Automatic Speech Recognition (ASR)
  - **Input**: Audio, **Output**: Text

  $P($ *I saw a van* $)$ $>>>>$ $P($ *Eyes awe of an* $)$

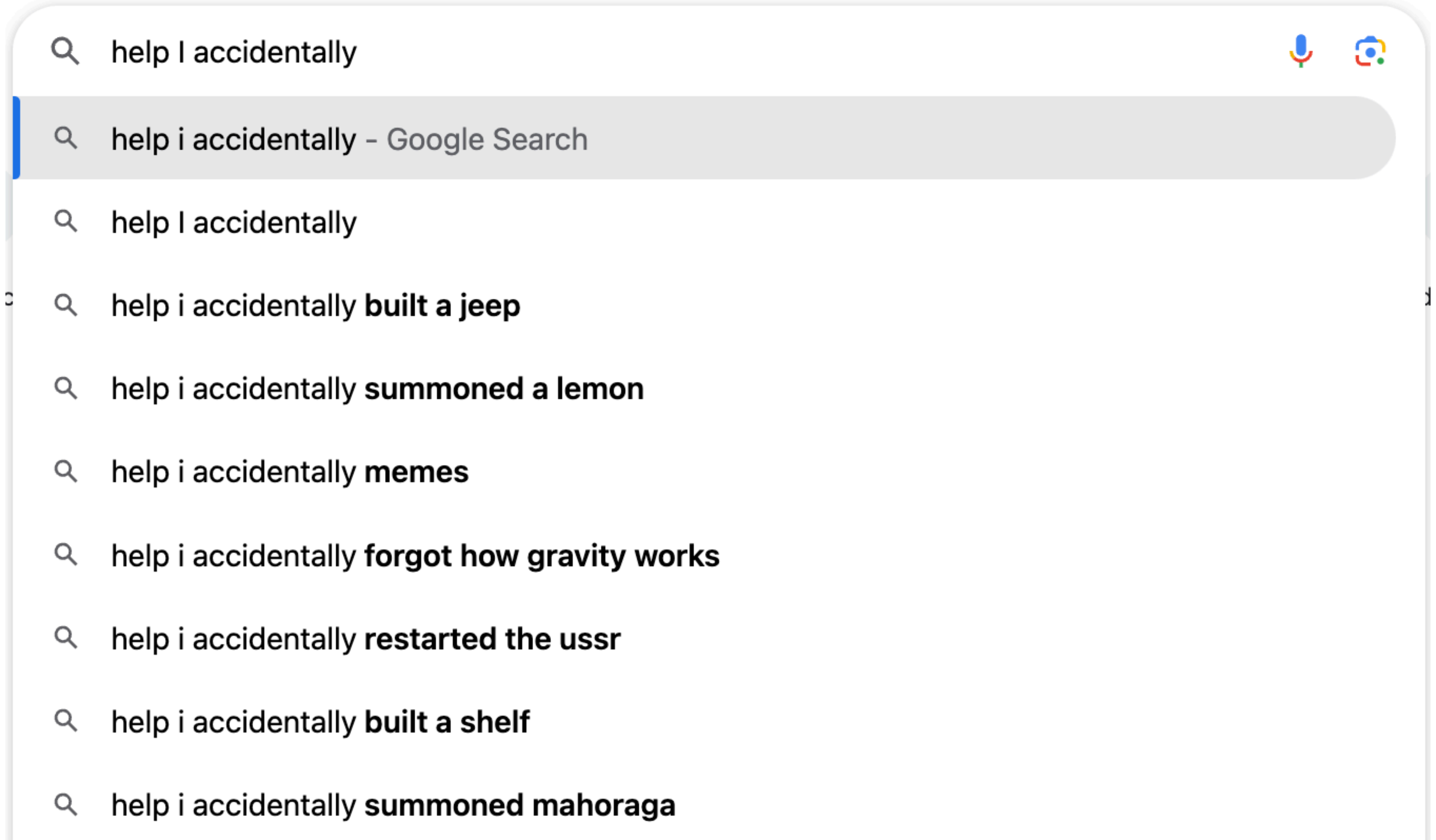
What else?

Credit: *Yoav Artzi's LM-Class*

Language Models' Use

Where else are language models used?

Google



Language Models can be powerful

FEBRUARY 14, 2019

Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization — all without task-specific training.

VIEW CODE

READ PAPER

READ MORE

If any language task can be described as a text-to-text problem…

Sentiment Analysis:

What is the sentiment of I loved the movie? Very positive.

Language Models can be powerful

FEBRUARY 14, 2019

Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization — all without task-specific training.

VIEW CODE

READ PAPER

READ MORE

If any language task can be described as a text-to-text problem…

Machine Translation:

What is the translation of "J'aime Lucy" in English? I love Lucy.

…then conceptually, we can solve it by just generating the answer as a continuation of a "prompt"

It would need to be a very powerful LM though!

Language Modeling Problem

- Let $\mathscr{V}$ be a finite vocabulary of words.

$$\mathscr{V} = \{ \text{the, a, man, telescope, Madrid, two, } \ldots \}$$

- We can construct (infinite) word sequences $\mathbf{w}$

$$\mathscr{V}^{\dagger} = \{ \text{the, a, the a, the fan, the man, the man with a telescope} \}$$

- **Given**: a dataset of **M** sentences $\mathscr{D} = \{\mathbf{w}\}_{i=1}^{M}$
- **Goal/ Output**: estimate a probability distribution $P(\mathbf{w}) \geq 0$ over **all** word sequences $\mathbf{w} \in \mathscr{V}^{+}$.

Terminology: the ambiguous term "word"

- We will often need to distinguish (the counting of)
  - **word types**
    - *Unique* words. This is a finite set, which we will pre-determine as our vocabulary or lexicon $\mathscr{V}$.
  - **word tokens**
    - Instantiations of items in the vocab in the "running" text or sequences.

Example: All for one and one for all .

- **Word tokens: 8** (if we include punctuations in our lexicon)
- **Word types:** 6 (if we assume capitalization is a distinguisher) or 5 (if capitalization differences are ignored)

Language Modeling Problem

- **Given**: a dataset of **M** sentences $\mathscr{D} = \{\mathbf{w}\}_{i=1}^{M}$
- **Goal/ Output**: estimate a probability distribution $P(\mathbf{w})$ over **all** word sequences $\mathbf{w} \in \mathscr{V}^{+}$.

- Probabilities should broadly indicate plausibility of sentences:
  - P(I saw a van) > P(eyes awe of an)
  - Not *only* grammaticality: P(artichokes intimidate zippers) ~0
  - Plausibility depends on the context.

Language Modeling Problem

- **Given**: a dataset of **M** sentences $\mathscr{D} = \{\mathbf{w}\}_{i=1}^{M}$
- **Goal/ Output**: estimate a probability distribution $P(\mathbf{w})$ over **all** word sequences $\mathbf{w} \in \mathscr{V}^{+}$.

So, how do we estimate $P(\mathbf{w})$?

Näive option: compute the empirical distribution over the training data:

$$P(\mathbf{w}) = \frac{c(\mathbf{w})}{\text{Total number of sequences}}$$

Problem?

There can be valid $\mathbf{w}$ that are not seen in this training dataset. Naive option will assign 0 probabilities to these. We will never have enough data that all valid sequences are seen.

"Imagine a small blue chair sitting quietly next to a window on a rainy afternoon"

Language Modeling Problem

First, let's decompose $P(\mathbf{w})$

$$P(\mathbf{w}_1^n) = P(w_1 w_2 w_3 \ldots w_n)$$

Shorthand for

$w_1 w_2 \ldots w_n$

applying chain rule

$$= P(w_1)\ P(w_2 \,|\, w_1)\ P(w_3 \,|\, w_2 w_1) \ldots P(w_n \,|\, w_1 \ldots w_{n-1})$$

assumption: probability of a word depends on previous words only

$$= \prod_{i=1}^{n} P(w_i \,|\, w_1 \ldots w_{i-1})$$

$$P(\text{I saw a man}) = P(\text{I})\ P(\text{saw} \,|\, \text{I})\ P(\text{a} \,|\, \text{I saw})\ P(\text{man} \,|\, \text{I saw a})$$

Language Modeling Problem

$$P(\mathbf{w}_1^n) = P(w_1 w_2 w_3 \ldots w_n) = \prod_{i=1}^{n} P(w_i \,|\, w_1 \ldots w_{i-1})$$

$$= P(w_1)\, P(w_2 \,|\, w_1)\, P(w_3 \,|\, w_2 w_1) \ldots \boxed{P(w_n \,|\, w_1 \ldots w_{n-1})}$$

Can we now use count based estimates?

"Imagine a small blue chair sitting quietly next to a window on a rainy afternoon"

No, if a test sentence $\mathbf{w}_1^n$ is unseen in the training data, this will again be zero!

Language Modeling Problem

$$P(\mathbf{w}_1^n) = P(w_1 w_2 w_3 \ldots w_n) = \prod_{i=1}^{n} P(w_i \,|\, w_1 \ldots w_{i-1})$$

Key idea: Markov Assumption: Probability of each word in a sequence only depends on a fixed number of previous words

Unigram Model $\rightarrow P(w_i \,|\, w_1 \ldots w_{i-1}) := P(w_i)$

Bigram Model $\rightarrow P(w_i \,|\, w_1 \ldots w_{i-1}) := P(w_i \,|\, w_{i-1})$

Trigram Model $\rightarrow P(w_i \,|\, w_1 \ldots w_{i-1}) := P(w_i \,|\, w_{i-2} w_{i-1})$

N-gram language models: Probability of each word depends on N-1 previous words.

$$:= \prod_{i=1}^{n} P(w_i \,|\, w_{i-k+1} \ldots w_{i-1})$$

N-Gram Language Model Example

P(lost | Not all those who wander are)

According to our various models, that probability is equal to …

Unigram Model: P(lost)

Bigram Model: P(lost | are)

Trigram Model: P(lost | wander are)

Sequence Probabilities w/ Bi-gram model

- **Goal**: Compute $P(w_1 w_2 \ldots w_n)$, **with implicit** $w_o = < s >$

$$P(\mathbf{w}_1^n) = P(w_1)\ P(w_2 | w_1)\ P(w_3 | w_2 w_1) \ldots P(w_n | w_1 \ldots w_{n-1})$$

$$= P(w_1)\ P(w_2 | w_1)\ P(w_3 | w_2) \ldots P(w_n | w_{n-1})$$

$$= P(w_1 | < s >)\ P(w_2 | w_1)\ P(w_3 | w_2) \ldots P(w_n | w_{n-1})$$

$$= \prod_i^n P(w_i | w_{i-1})$$

One way to "learn" an n-gram model

- **"Raw"** count approach
  - Estimate Bi-gram probability by $P(w_i \mid w_{i-1}) = \dfrac{\text{Count}(w_{i-1}w_i)}{\text{Count}(w_{i-1})}$
  - Trigram??
  - Unigram??

General case for an N-gram language model?

$$P(w_i \mid \mathbf{w}_{i-N+1}^{i-1}) = \frac{\text{Count}(\mathbf{w}_{i-N+1}^{i})}{\text{Count}(\mathbf{w}_{i-N+1}^{i-1})}$$

One way to "learn" an n-gram model

- **"Raw"** count approach
  - Estimate Bi-gram probability by $P(w_i \mid w_{i-1}) = \dfrac{\text{Count}(w_{i-1}w_i)}{\text{Count}(w_{i-1})}$
  - Trigram??

General case for an N-gram language model?

$$P(w_i \mid \mathbf{w}_{i-N+1}^{i-1}) = \frac{\text{Count}(\mathbf{w}_{i-N+1}^{i})}{\text{Count}(\mathbf{w}_{i-N+1}^{i-1})}$$

These are called the models' parameters.

Let's see an example

Training Data:

> **<s> I get what I eat and I eat what I get </s>**

Goal: Learn the parameters of a bigram language model.

<s> I	1
I get	2
get what	1
what I	2
I eat	2
eat and	1
and I	1
eat what	1
get </s>	1

<s>	1
I	4
get	2
what	2
eat	2
and	1
</s>	1

Applying the bigram model

Training Data:

<s> I get what I eat and I eat what I get </s>

Test Example: P (<s> I get what)

<s> I	1
I get	2
get what	1
what I	2
I eat	2
eat and	1
and I	1
eat what	1
get </s>	1

<s>	1
I	4
get	2
what	2
eat	2
and	1
</s>	1

Applying the bigram model

Training Data:

<s> I get what I eat and I eat what I get </s>

<s> I	1
I get	2
get what	1
what I	2
I eat	2
eat and	1
and I	1
eat what	1
get </s>	1

<s>	1
I	4
get	2
what	2
eat	2
and	1
</s>	1

Another note about a different sequence:

P(I get what I get .) will NOT be 0, even though it isn't in the data!

The model does generalize to(some) unseen sequences.

But **unseen bigrams** will cause a sequence to be assigned probability 0.

E.g. P(<s> eat and see) = 0

Sparsity Problem!

Generating Text Using a Language Model!

- **In addition to assigning a probability distribution to some sentence, we can also generate/decode a sentence!**
- How do we generate using a sentence using a Bi-gram language model?

N-gram Models on Shakespeare

- **Corpus statistics**
  - 884,647 tokens, vocabulary size of =29,066
  - Shakespeare produced 300,000 bigram types out of = 844M possible bigrams
    - So 99.96% of the possible bigrams were never seen (have zero entries in the table)

N-gram Models on Shakespeare

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

N-gram Language Models

- How should we choose N?

Because it was a sunny day, I should take a ___________.

Suppose N=2:

P (raincoat | Because it was a sunny day, I should take a) = P (raincoat | a)

P (hat | Because it was a sunny day, I should take a) = P (hat | a)

Suppose N=3:

P (raincoat | Because it was a sunny day, I should take a) = P (raincoat | take a)

P (hat | Because it was a sunny day, I should take a) = P (hat | take a)

N-gram Language Models

- How should we choose N?

rainy
Because it was a sunny day, I should take a ___________.

Suppose N=2:

P (raincoat | Because it was a sunny day, I should take a) = P (raincoat | a)

P (hat | Because it was a sunny day, I should take a) = P (hat | a)

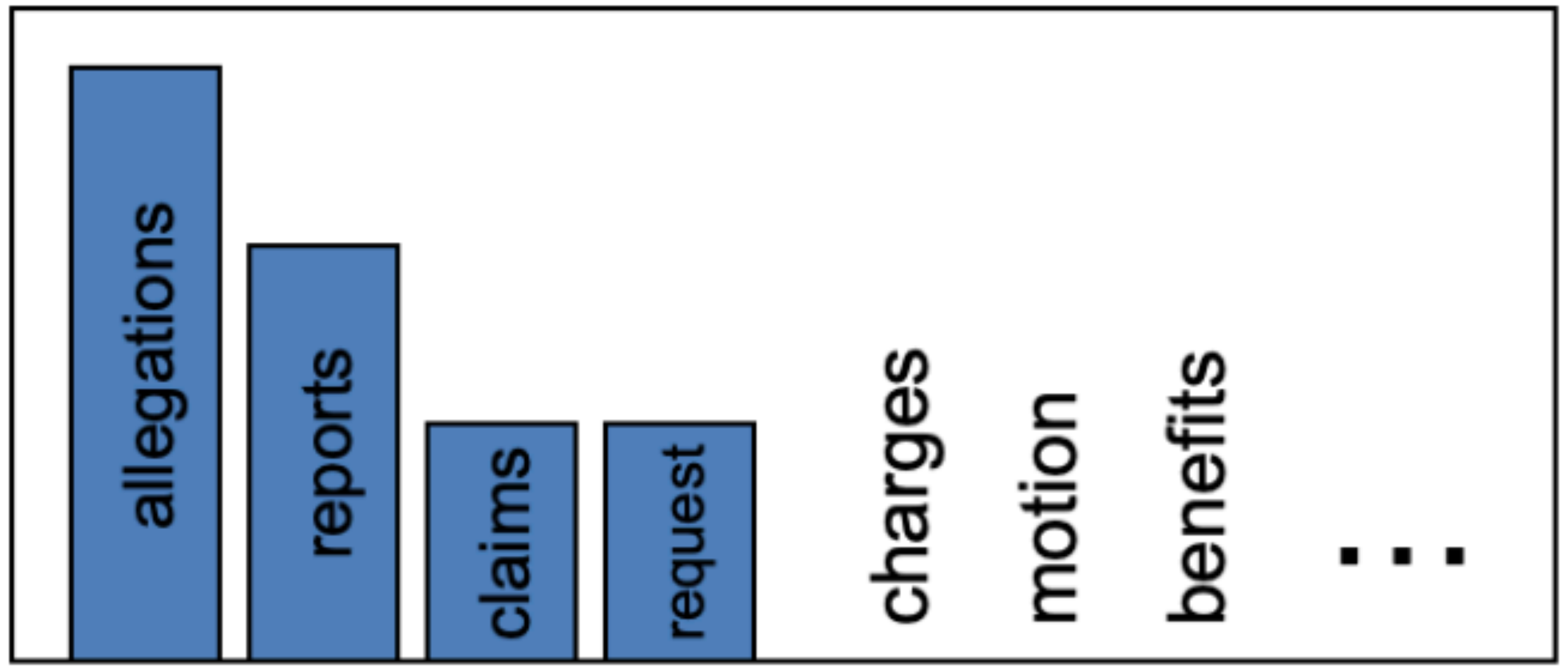
Suppose N=3:

P (raincoat | Because it was a sunny day, I should take a) = P (raincoat | take a)

P (hat | Because it was a sunny day, I should take a) = P (hat | take a)

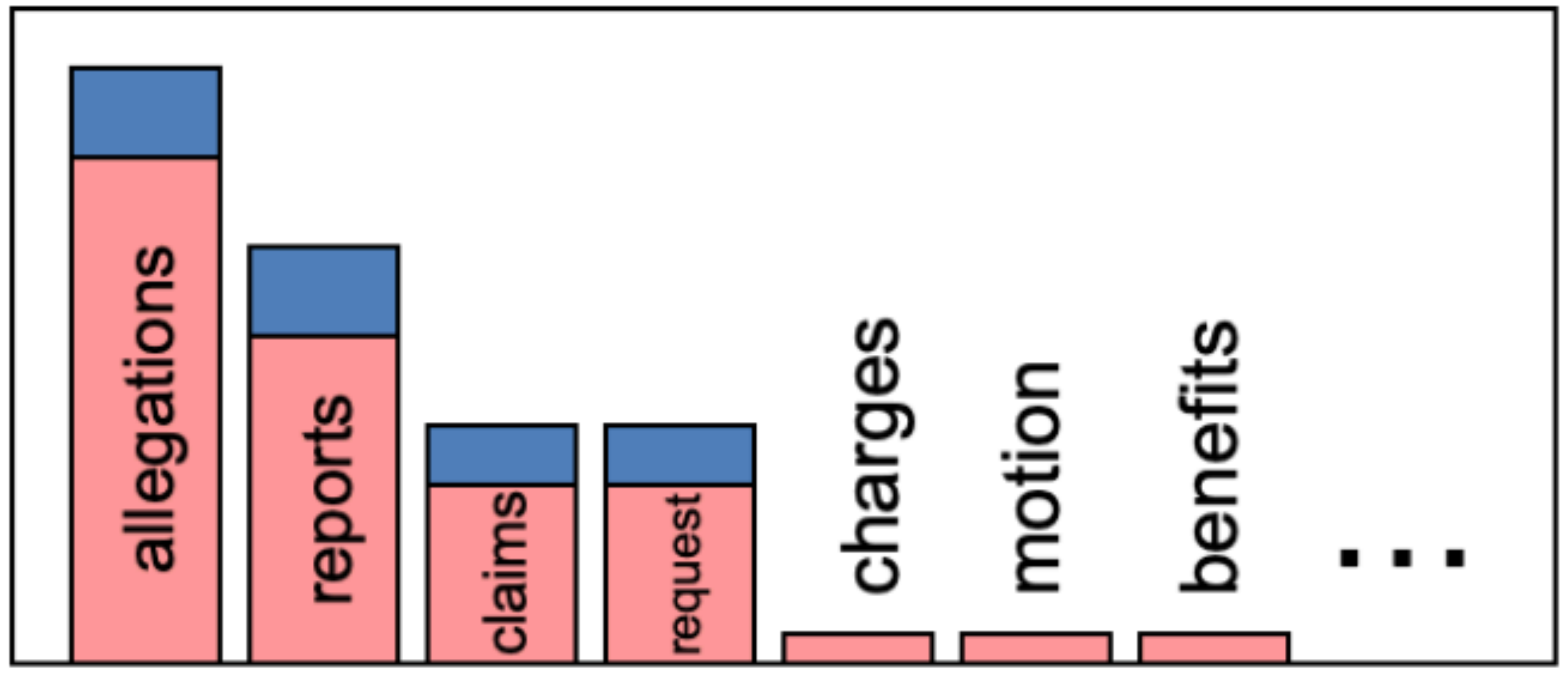
How do we fix this sparsity issue in LMs?

- A single n-gram with zero probability –> probability of the entire sequence is 0.
- Goal: Estimating statistics from sparse data.
- Idea: **Steal** some probability mass from seen data.



P(w | denied the)
3 allegations
2 reports
1 claims
1 request

7 total



P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other

7 total

Smoothing

- **Add-one smoothing**
- Pretend we saw each word one more time that we did (even unseen ones). For 2-gram:

$$P_{MLE} = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})} \rightarrow P_{MLEAdd-1} = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + |\mathcal{V}|}$$

- Called Laplace Smoothing.
- Can be generalized to Add-K

$$P_{MLEAdd-K} = \frac{c(w_{i-1}, w_i) + K}{c(w_{i-1}) + K.|\mathcal{V}|}$$

Berkeley Restaurant Corpus

Raw counts: 9222 sentences

- Bigrams

	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

- Unigram

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

Berkeley Restaurant Corpus

Bi-gram probabilities

$$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_i w_{i-1})}{c(w_{i-1})}$$

	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

Berkeley Restaurant Corpus

Smoothed counts (Add-1)

	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

Berkeley Restaurant Corpus

Smoothed bigram probs (Add-1)

$$P_{MLEAdd-1} = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + |\mathscr{V}|}$$

	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

Other smoothing options

- **Back-off smoothing:** use lower-order n-gram
  - For tri-gram, use tri-gram if you have good evidence, otherwise use bi-gram, otherwise unigram
- **Linear interpolation**: mix lower-order n-grams
  - For tri-gram, mix with with bi-gram and unigram probabilities

$$P_\lambda(x_i \mid x_{i-1}, x_{i-2}) = \lambda_3 p_{\text{MLE}}(x_i \mid x_{i-1}, x_{i-2}) + \lambda_2 p_{\text{MLE}}(x_i \mid x_{i-1}) + \lambda_1 p_{\text{MLE}}(x_i)$$

$$\sum \lambda_i = 1$$

Outline of this course

Basic Goals

- **We want to learn about the building blocks for large language models (LLMs) like GPTs, Claude, LLaMA, etc.**
- We will build towards this through the course.
- By the end of the course, you will have:
  - Gained insight into how LLMs are basically trained and why they work better than previous approaches.
  - Able to use standard libraries NLP researchers use.
  - Be able to read and understand (most) papers published in NLP conferences.

“Paradigm” Shifts

- **Modeling**: Rule-based systems → Statistical Methods → Neural Methods (FFNNs → RNNs → Transformers)
- **Task-specific** models → **Generic** models
- **Data:** labeled data → more general use of unlabeled data

Course Outline

- **Classical NLP** (2 weeks) → N-gram language modeling, classification, word embeddings.
- **Neural NLP Foundations** (4 weeks) → Feedforward Neural Networks, RNNs.
- **Modern NLP Foundations** (5 weeks) → Transformer models, Pre-training, Post-training.
- **LLM++** (3 weeks) → LLM+Factuality, LLM+Retrieval, LLM+Efficiency

Understand basic building blocks of chatbots like GPTs, LLaMAs.

More cutting edge augmentations to vanilla LLMs.

Administrivia (the boring stuff, as promised)

Prerequisites

- Strong programming skills. Three semesters of programming classes are strongly recommended (e.g., completion of CS3110).
- Python experience.
- Comfort with elementary probability.
- Clear understanding of matrix and vector operations.
- Familiarity with differentiation.

Resources

- Up-to-date syllabus, slides, and other course material will always be available on the course website at: https://www.cs.cornell.edu/courses/cs4740/2026sp/
- You do not need to buy any textbook for this course. We will follow *Jurafsky and Martin, Speech and Language Processing, 3rd edition (draft).* Free online version is available online.
- You will use modern LLM APIs (e.g. for ChatGPT, LLaMA) for latter assignments. This *might* incur a cost of $5-10 if you have already exhausted your free quota.

Coursework and grading

- Homework Assignments (60%)
  - Review assignment / HW0 → **0%**
  - 4 Full homework assignments → **60%** (Can be done in pairs (strongly recommended))
    - **5 slip days** to use throughout the course for *these* 4 HW assignments. Max of 2 slip days/hw.
- Exams (40%)
  - Midterm (**20%**) and Final (**20%**)
  - To receive a C- or above in the course, students must receive at least a C- on both exams.

- We will **not** curve grades, use "strict 90/80/70" grade cutoffs. You are not competing with each other.

Coursework and grading

- Homework Assignments (60%)
  - Review assignment / HW0 → **0%**
  - 4 Full homework assignments → **60**
    - **5 slip days** to use throughout the days/hw.
- Exams (40%)
  - Midterm (**20%**) and Final (**20%**)
  - To receive a C- or above in the cours
- We will **not** curve grades, use "stric with each other.

This will be released **today** on the course website.

Designed to test pre-requisite knowledge. Should not take more than 2 hours.

Talk to course staff if you find yourself struggling with a majority of the questions.

Teaching Staff

- **Instructors:** Claire Cardie, Tanya Goyal
- **TAs:** Wayne Chen, Son Tran, Chengyu Huang, Aileen Huang, Anand Bannerji, Andrew Hu, Jeffrey Huang, Frank Yang, Jay Talwar, Brianna Liu, Deniz Boloni-Turgut, Yunoo Kim, Mahitha Penmetsa

Communication with Staff

- Homework / grading / lecture questions → Ed
- Private inquiry (e.g. health issue requiring accommodations) → Email **both** instructors.

- Office hours listed on the course website. (This statement will be true tonight)
  - Instructor office hours start this week.
  - TA office hours start next week. Times will be listed on the course webpage. **There will be TA office hours every weekday.**

Waitlist

- Refer to the CS enrollment and waitlist information page here: https://www.cs.cornell.edu/courseinfo/enrollment
- You do not need to contact the professors or course staff. We are not handling the waitlist.
- If you face issues with registering or joining the waitlist, please file a ticket using the link in the above webpage.

Final words…

- This is the **most** exciting time to be working in NLP.
- Look out for HW0 to be released **today** on gradescope.
- Slide Acks: Earlier versions of this course offerings including materials from Marten van Schijndel, Lillian Lee, Claire Cardie, Yoav Artzi's LM-class.