03/03/2025

Natural Language Processing (NLP)

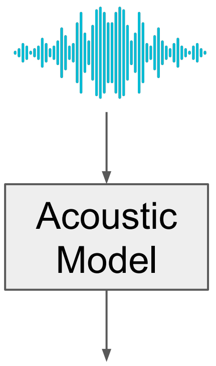
Large Language Models from Shannon to ChatGPT

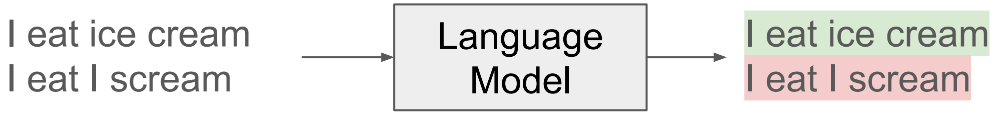
aivancity
SCHOOL FOR
TECHNOLOGY, BUSINESS & SOCIETY
PARIS-CACHAN

Part 1

Why Language Modeling?

Automatic Speech Recognition (ASR) relies on Language Models because of homophony





Part 1

Language Modeling

- Assume a (finite) vocabulary of words

$$\mathcal{V} = \{\text{I}, \text{eat}, \text{ice}, \text{cream}, \text{scream} \ldots\}$$

- We can construct an (infinite) set of strings

$$\mathcal{V}^\dagger = \{\text{I}, \text{eat}, \text{I eat}, \text{eat I}, \text{I cream}, \text{ice cream}, \text{ice scream} \ldots\}$$

Input: training data $\boldsymbol{x} = \langle x_1, \ldots, x_N \rangle$ in $\mathcal{V}^\dagger$

Output: a function $p : \mathcal{V}^\dagger \to \mathbb{R}$ $\quad \forall \boldsymbol{x} \in \mathcal{V}^\dagger, p(\boldsymbol{x}) \geq 0$

$$\sum_{\boldsymbol{x} \in \mathcal{V}^\dagger} p(\boldsymbol{x}) = 1$$

Part 1

Learning from sequences: the Chain Rule

$p(x \mid \text{START})$	$p(x \mid \text{START I})$	$p(x \mid \cdots \text{went})$	$p(x \mid \cdots \text{to})$	$p(x \mid \cdots \text{the})$	$p(x \mid \cdots \text{park})$	$p(x \mid \text{START I went to the park.})$
The 3 %	think 11 %	**to 35 %**	**the 29 %**	bathroom 3 %	and 14 %	I 21 %
When 2,5 %	was 5 %	back 8 %	a 9 %	doctor 2%%	with 9	It 6
They 2 %	**went 2 %**	into 5 %	see 5 %	hospital 2 %	, 8 %	The 3 %
… …	am 1 %	through 4 %	my 3 %	store 1,5 %	to 7 %	There 3 %
I 1 %	will 1 %	out 3 %	bed 2 %	… …	… …	… …
… …	like 0,5 %	on 2 %	school 1 %	**park 0,5 %**	**. 6 %**	**STOP 1 %**
Banana 0,1 %	… …	… …%	… …	… …	… …	… …

Language Model

START | I | went | to | the | park | . | STOP

Part 1

Learning from sequences: the Chain Rule

$$
p(\boldsymbol{X} = \boldsymbol{x}) = \begin{pmatrix} p(X_1 = x_1) \\ \cdot\, p(X_2 = x_2 \mid X_1 = x_1) \\ \cdot\, p(X_3 = x_3 \mid \boldsymbol{X}_{1:2} = \boldsymbol{x}_{1:2}) \\ \vdots \\ \cdot\, p(X_N = \quad \mid \boldsymbol{X}_{1:N-1} = \boldsymbol{x}_{1:N-1}) \end{pmatrix}
$$

$$
= \prod_{i=1}^{N} p(X_i = x_i \mid \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1})
$$

n-gram Models (Markov assumption)

Unigram (n=1): No history! All words are independent!

$$p(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{N} p(X_i = x_i \mid \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1}) \overset{\text{assumption}}{=} \prod_{i=1}^{N} p(X_i = x_i; \boldsymbol{\theta})$$

$$p(\mathbf{X} = \text{I eat I scream}) = p(X_1 = \text{I}) \cdot p(X_2 = \text{eat}) \cdot p(X_3 = \text{I}) \cdot p(X_4 = \text{scream})$$

Part 1

n-gram Models (Markov assumption)

$$p(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{N} p(X_i = x_i \mid \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1})$$

$$\overset{\text{assumption}}{=} \prod_{i=1}^{N} p(X_i = x_i \mid X_{i-\mathsf{n}+1:i-1} = \boldsymbol{x}_{i-\mathsf{n}+1:i-1}; \boldsymbol{\theta})$$

Bigram (n=2):

$$p(\mathbf{X} = \text{I eat I scream}) = p(X_1 = \text{I}|x_0) \cdot p(X_2 = \text{eat}|\text{I}) \cdot p(X_3 = \text{I}|\text{eat}) \cdot p(X_4 = \text{scream}|\text{I})$$

Trigram (n=3):

$$p(\mathbf{X} = \text{I eat I scream}) = p(X_1 = \text{I}|x_0\ x_0) \cdot p(X_2 = \text{eat}|x_0\ \text{I}) \cdot p(X_3 = \text{I}|\text{I eat}) \cdot p(X_4 = \text{scream}|\text{eat I})$$

Sparsity in n-grams

If n is too small, your model can't learn very much about language.

$$\mathcal{V}^\dagger = \{\text{I}, \text{eat}, \text{I eat}, \text{eat I}, \text{I cream}, \text{ice cream}, \text{ice scream} \dots\}$$

As n gets larger:

- The number of parameters grows with $O(V^n)$.
  - What's a parameter?
- Most n-grams will never be observed, so you'll have lots of zero probability n-grams. This is an example of **data sparsity**.
- Your model depends increasingly on the training data; you need (lots) more data to learn to generalize well.

Perplexity

The Shannon Game:

- How well can we predict the next word?
  - When I eat pizza, I wipe off the ___
  - Many children are allergic to ___
  - I saw a ___
- Unigrams are terrible at this game. (Why?)

grease	0.5
sauce	0.4
dust	0.05
…	
mice	0.001
…	
the	1E-100

Claude Shannon

<u>How good are we doing?</u>

- Want to assign *high probability* to observed words

Perplexity

Given a test dataset $\bar{\boldsymbol{x}}$ (of $\bar{N}$ words), we arrive at the standard intrinsic evaluation in three steps:

1. Probability of the test data: $p(\bar{\boldsymbol{x}}; \boldsymbol{\theta})$
2. That value will be tiny, because $\mathcal{V}^\dagger$ is infinitely large, and p will decrease exponentially in the length of $\bar{\boldsymbol{x}}$. So we transform it:

$$\text{Perplexity}(\bar{\boldsymbol{x}}; p(\cdot; \boldsymbol{\theta})) = \sqrt[\bar{N}]{\frac{1}{p(\bar{\boldsymbol{x}}; \boldsymbol{\theta})}} = 2^{\underbrace{\left(\frac{1}{\bar{N}} \times -\log_2 p(\bar{\boldsymbol{x}}; \boldsymbol{\theta})\right)}_{\text{Cross-Entropy}}}$$

Special cases:

- If the model were to put *all* of its probability on $\bar{\boldsymbol{x}}$, perplexity would be 1 (minimal possible value).
- If the model assigns zero probability to $\bar{\boldsymbol{x}}$, perplexity is $+\infty$. So it's important to make sure that p assigns strictly positive probability to *every* sequence of words.

You can interpret perplexity as "effective size of the vocabulary."

Part 1

Perplexity

Consider a unigram model that is completely agnostic; it assigns $\theta_v = \frac{1}{V}$ for all $v \in \mathcal{V}$.

What will its perplexity be? Hint: as long as the test data is restricted to words in $\mathcal{V}$, the test data doesn't matter!

$$\text{Perplexity}(\bar{\boldsymbol{x}}; p(\cdot;\boldsymbol{\theta})) = \sqrt[\bar{N}]{\frac{1}{p(\bar{\boldsymbol{x}};\boldsymbol{\theta})}} = \sqrt[\bar{N}]{\frac{1}{\left(\frac{1}{V}\right)^{\bar{N}}}} = V$$

Zipf's Law

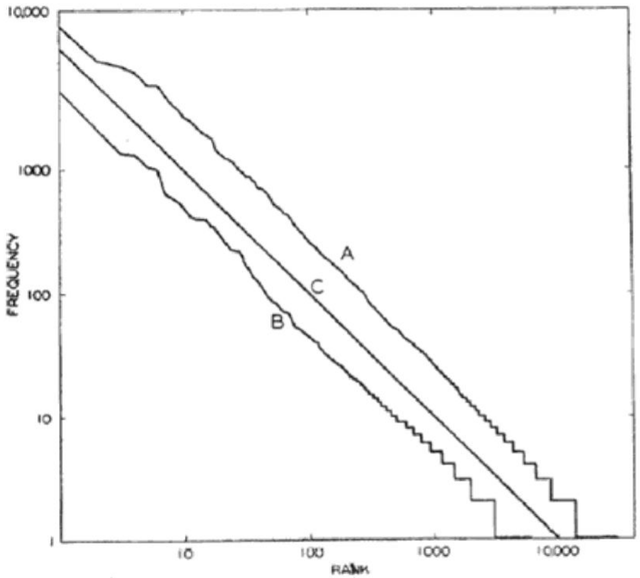


Fig. 2-1. The rank-frequency distribution of words. (A) The James Joyce data; (B) the Eldridge data; (C) ideal curve with slope of negative unity.

Zipf (1949) found that, when counting the frequencies of all words in a Joyce's Ulysses (**260,430 words**), that:

- the **10th** most frequent word occurred **2,653** times
- the **100th 265** times, or 10 times less
- the **1000th 26** times, or 100 times less

In other words, the frequency ***f(w)*** of a word ***w*** is inversely proportional to its rank ***k***: $f(w) \propto 1/k$

or $f(w) \times k =$ constant

Zipf's Law

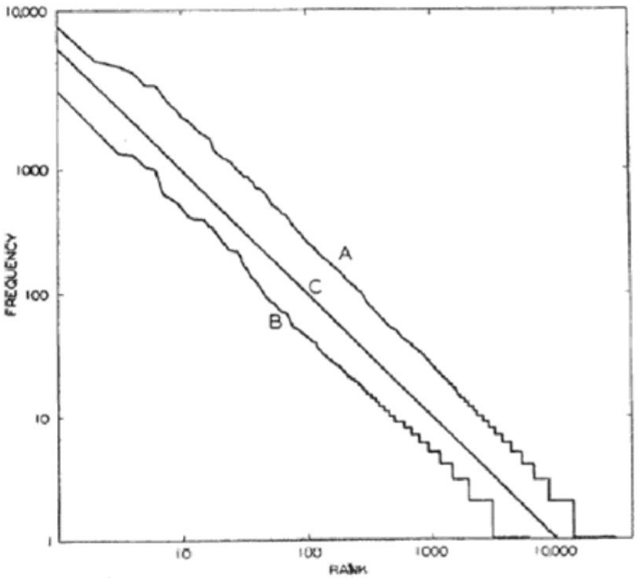


Fig. 2–1. The rank-frequency distribution of words. (A) The James Joyce data; (B) the Eldridge data; (C) ideal curve with slope of negative unity.

- The frequency ***f(w)*** of a word ***w*** is inversely proportional to its rank ***k***: ***f(w) ∝ 1/k***
- The distribution of words frequencies have a very long tail: some words are very frequent other appear only once in the whole training corpus
- This empirical law was verified in most languages of the world, it's one of the canonical laws of computational linguistics
- Again, linked to Shannon's Information Theory

Part 2

Break for questions and "appel"

Part 2

Pretraining (for Transfer Learning)

- **Self-supervision**: Language Models do not need annotated data to be pretrained, only raw text, and there are plenty
- Deep Learning thrives on data (the more the better, more complex models without overfitting)
- Models can then be **fine-tuned** on a downstream task (e.g. Named Entity Recognition).
- Much better than training from scratch because annotated data is rare (almost nonexistent except in English) because it's expensive

Part 2

LLMs learn about syntax and semantics

- Syntax :

I eat
I ate
I am
I eats
eat I

- Semantic :

I eat bread
I ate pasta
I ate sand
I am a student
I am a fork
I ate pasta with my fork
I ate pasta with my keyboard

Part 2

LLMs learn about much more

- Entities :

```
My favorite city is Paris
My favorite city is France
I am visiting Paris, the capital of France
I am visiting Paris, the capital of Italy
```

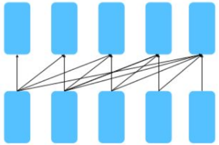
- Reasoning :

```
All men are mortal. Socrates is a man.
Therefore, Socrates is mortal
```

Part 2

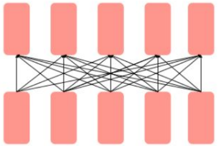
3 main objectives for 3 architectures

Decoder



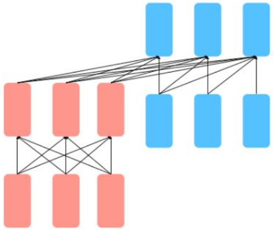
- Language modeling; can only condition on the past context

Encoder



- Bidirectional; can condition on the future context

Encoder-Decoder

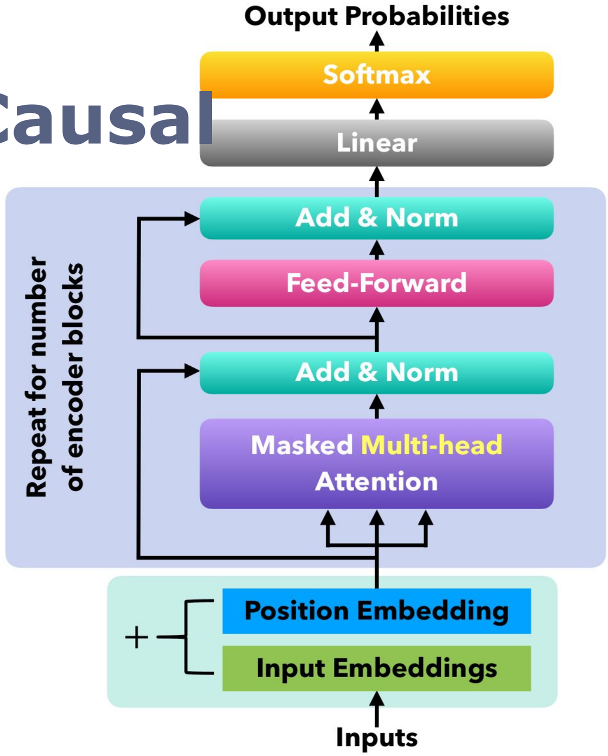


- Map two sequences of different length together

Part 2

Transformer A: (Autoregressive) Decoder/Causal

- Main architecture for LLMs (**GPT-3**, Llama-*, and many many many more)
- **Causal**/unidirectional mask: can only see past words
- First purpose: **Language Modeling** / autoregressive generation
- But now every task of NLP is cast as Language Modeling, even classification



Part 2

Language Modeling (no Markov assumption)

$p(x \mid \text{START})$		$p(x \mid \text{START I})$		$p(x \mid \cdots \text{went})$		$p(x \mid \cdots \text{to})$		$p(x \mid \cdots \text{the})$		$p(x \mid \cdots \text{park})$		$p(x \mid \text{START I went to the park.})$	
The	3 %	think	11 %	to	35 %	the	29 %	bathroom	3 %	and	14 %	I	21 %
When	2,5 %	was	5 %	back	8 %	a	9 %	doctor	2%%	with	9	It	6
They	2 %	went	2 %	into	5 %	see	5 %	hospital	2 %	,	8 %	The	3 %
…	…	am	1 %	through	4 %	my	3 %	store	1,5 %	to	7 %	There	3 %
I	1 %	will	1 %	out	3 %	bed	2 %	…	…	…	…	…	…
…	…	like	0,5 %	on	2 %	school	1 %	park	0,5 %	.	6 %	STOP	1 %
Banana	0,1 %	…	…	…	…%	…	…	…	…	…	…	…	…

Transformer Decoder

START | I | went | to | the | park | . | STOP

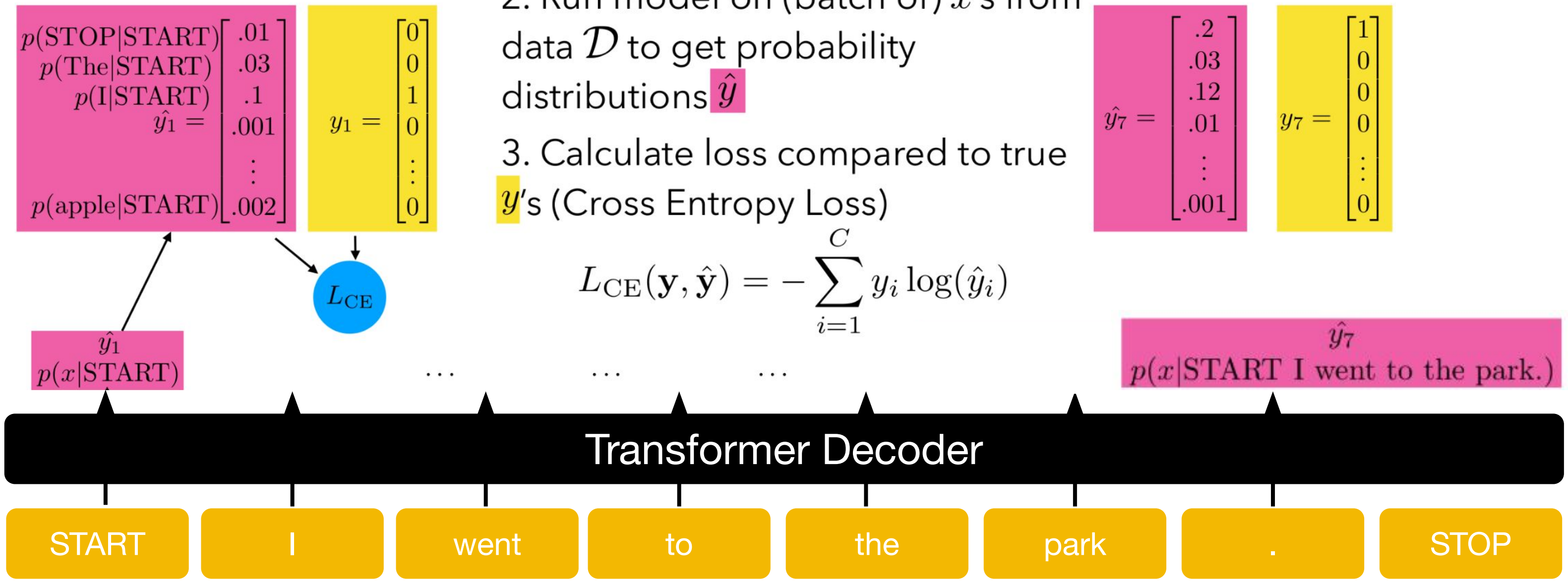
Part 2

Generation as a Sequence of Classifications

2. Run model on (batch of) x's from data $\mathcal{D}$ to get probability distributions $\hat{y}$

3. Calculate loss compared to true y's (Cross Entropy Loss)

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$



Part 2

Decoder for: Language Modeling / autoregressive generation

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**

GPT-3 (2020)

Part 2

Decoder for: Actually everything (but we'll come back to that)

- Classification: "I like this movie"

  → `"I like this movie,` **`it was`** `{good/bad}"`

- Question Answering: "When was Dante born?"

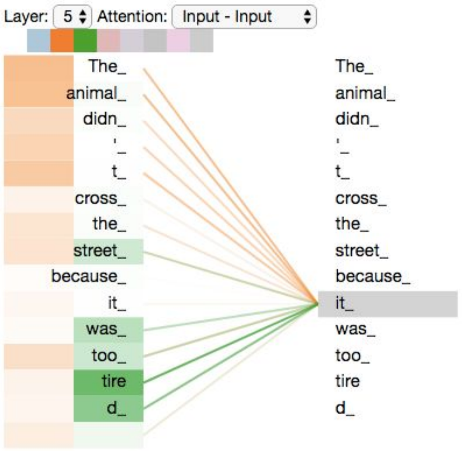
  → **`"Dante was born in ____"`**

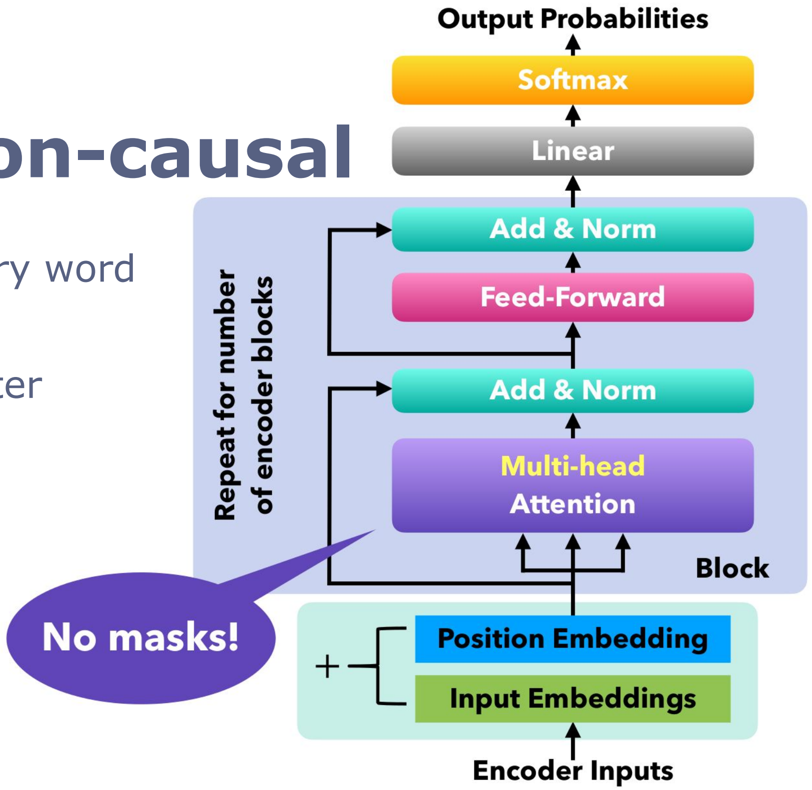
- Translation: "I like pasta"

  → **`"The translation of`** `'I like pasta'` **`in French is ____"`**

Transformer B: (Bidirectional) Encoder/non-causal

- Removes the mask from self-attention: now every word can see future and past
- Use for classification (but now words have a better context, unlike bag of words)
- Famous examples: **BERT**, m**BERT**, Ro**BERT**a, De**BERT**a, Camem**BERT**, …

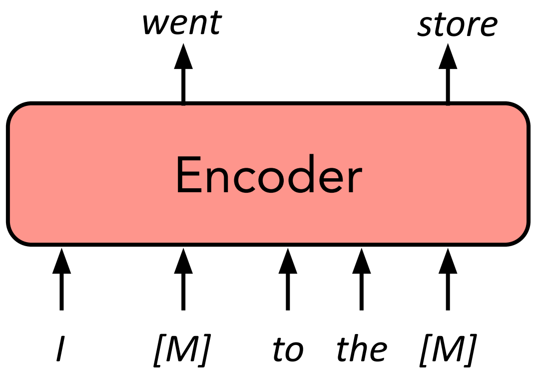




Part 2

Masked Language Modeling

- How to encode information from both **bidirectional** contexts?
- General Idea: **text reconstruction!**



$$h_1, \ldots, h_T = \text{Encoder}\left(w_1, \ldots, w_T\right)$$
$$y_i \sim Aw_i + b$$

Only add loss terms from the masked tokens. If $\tilde{x}$ is the masked version of x, we're learning $p_\theta(x \mid \tilde{x})$. Called **Masked Language model (MLM)**.

Part 2

Masked Language Modeling

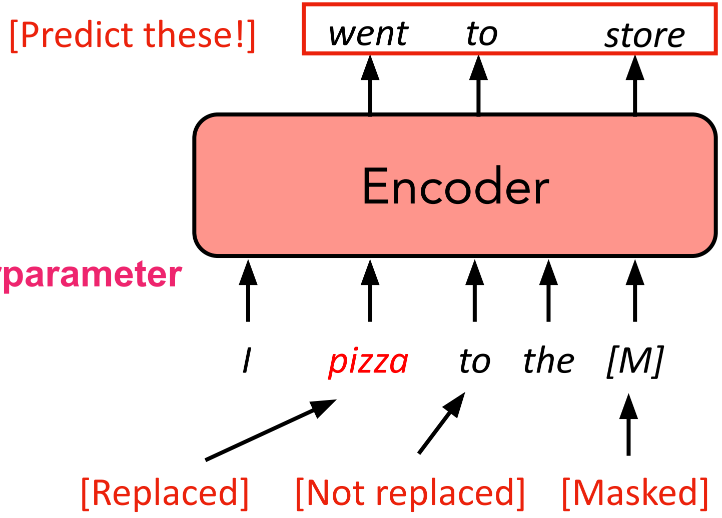
hyperparameter

- Choose a random **15%** of tokens to predict.
- For each chosen token:
  - Replace it with [MASK] 80% of the time.
  - Replace it with a random token 10% of the time.
  - Leave it unchanged 10% of the time (but still predict it!)
- Only learns from **15%** of tokens per step

hyperparameter

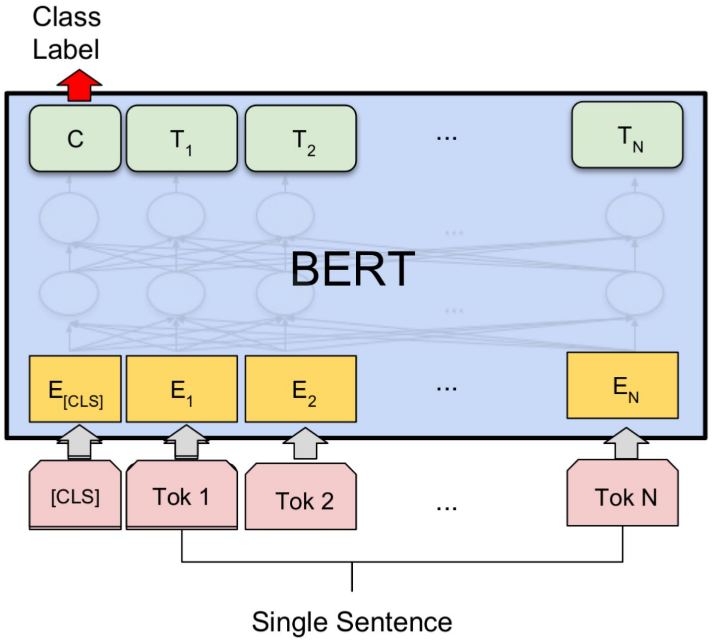
hyperparameter

hyperparameter



Part 2

Fine-tuning Encoder for: Sentiment Analysis



I just loved every minute of this film.

A strangely compelling and brilliantly acted psychological drama.

Preaches to two completely different choirs at the same time, which is a pretty amazing accomplishment.

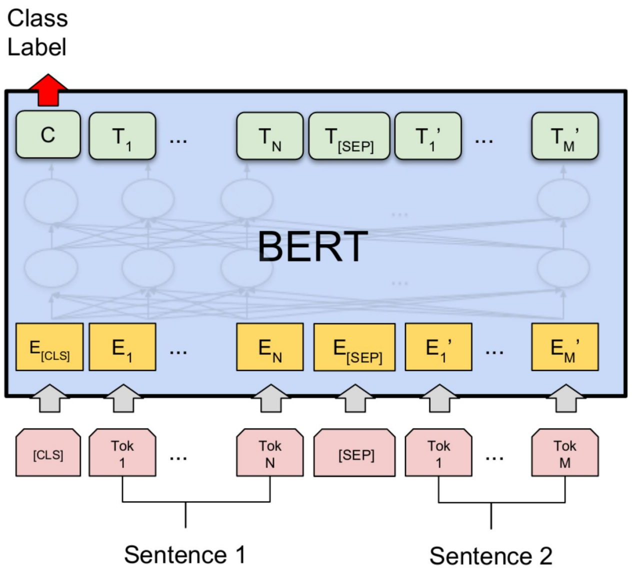
An instant candidate for the worst movie of the year.

The film seems a dead weight.

I found it slow, drab, and melodramatic.

Part 2

Fine-tuning Encoder for: Natural Language Inference



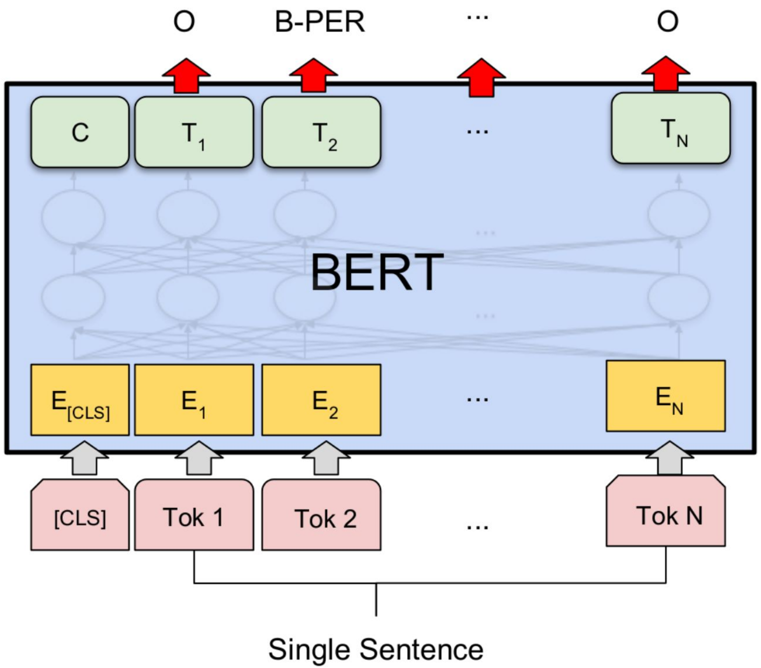
Met my first girlfriend that way. ✗ I didn't meet my first girlfriend until later.

At 8:34, the Boston Center controller received a third transmission from American 11 ✓ The Boston Center controller got a third transmission from American 11.

someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny ✗ No one noticed and it wasn't funny at all.

Part 2

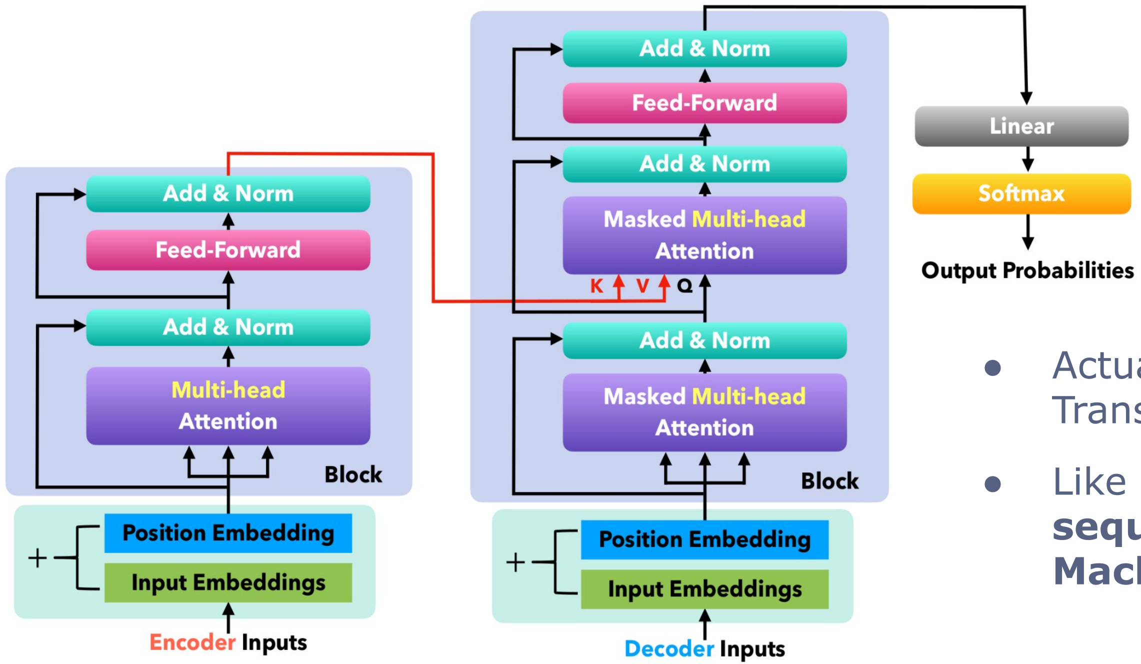
Fine-tuning Encoder for: Named Entity Recognition



Washington is the capital of the USA. It hosts the White House.

Part 2

Transformer C: Encoder-Decoder



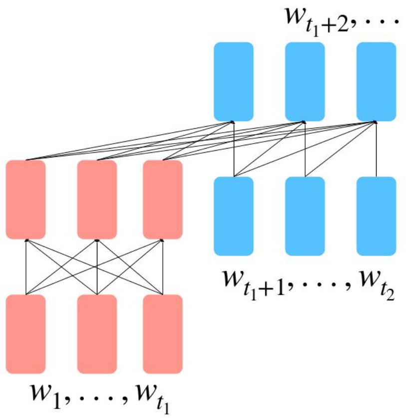
- Famous examples:
  T5, BART,
  BARThez, …

- Actually the first variant proposed for Translation by Vaswani et al. (2017)
- Like an RNN Encoder-Decoder, use for **sequence-to-sequence** tasks like **Machine Translation**

Part 2

Encoder-Decoder Training

- Encoder builds a representation of the source and gives it to the decoder
- Decoder uses the source representation to generate the target sentence
- The **encoder** portion benefits from **bidirectional** context; the decoder portion is used to train the whole model through **language modeling**



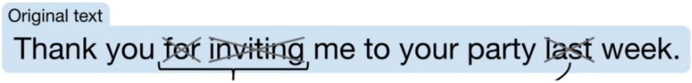
$$h_1, \ldots, h_{t_1} = \text{Encoder}(w_1, \ldots, w_{t_1})$$

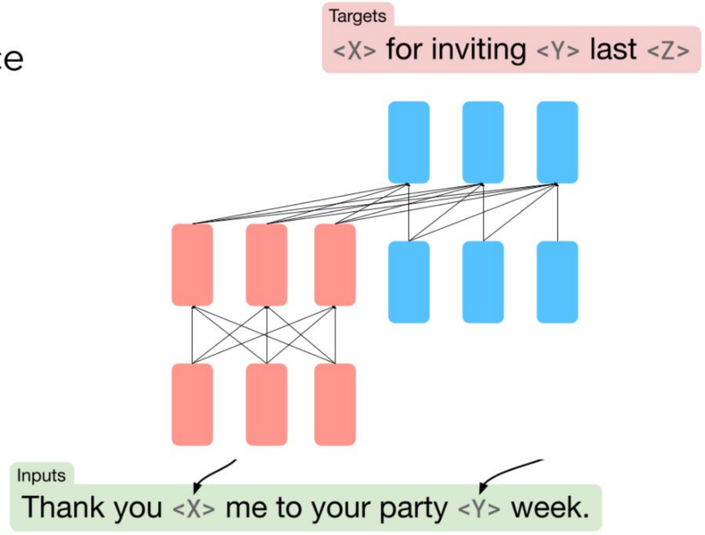
$$h_{t_1+1}, \ldots, h_{t_2} = \text{Decoder}(w_{t_1+1}, \ldots, w_{t_2}, h_1, \ldots, h_{t_1})$$

$$y_i \sim Ah_i + b, i > t$$

Text Denoising

- **Text span corruption (denoising):** Replace different-length spans from the input with unique placeholders (e.g., <extra_id_0>); decode out the masked spans.
  - Done during **text preprocessing**: training uses **language modeling** objective at the decoder side





Part 2

Encoder-Decoder for: Translation

translate English to French: This image section from an infrared recording by the Spitzer telescope shows a "family portrait" of countless generations of stars: the oldest stars are seen as blue dots, while more difficult to identify are the pink-coloured "new-borns" in the star delivery room.

→

Ce détail d'une photographie infrarouge prise par le télescope Spitzer montre un "portrait de famille" des innombrables générations d'étoiles: les plus vieilles étoiles sont en bleu et les points roses, plus difficiles à identifier, sont les "nouveau-nés" dans la salle d'accouchement de l'univers.

T5 (2020)

Part 2

Encoder-Decoder for: Summarization

summarize: marouane fellaini and adnan januzaj continue to show the world they are not just teammates but also best mates. the manchester united and belgium duo both posted pictures of themselves out at a restaurant on monday night ahead of their game against newcastle on wednesday . januzaj poses in the middle of fellaini and a friend looking like somebody who failed to receive the memo about it being a jackson 5 themed night. [...]

→

the belgian duo took to the dance floor on monday night with some friends . manchester united face newcastle in the premier league on wednesday . [...]

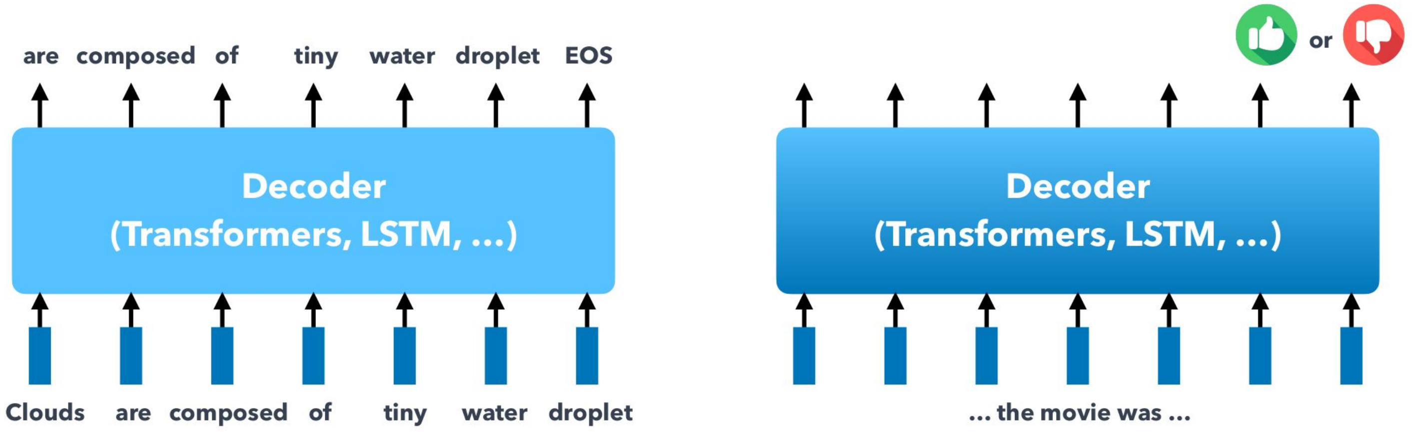
T5 (2020)

Part 3

Fine-Tuning

Step 1:
Pre-training

Step 2:
Fine-tuning



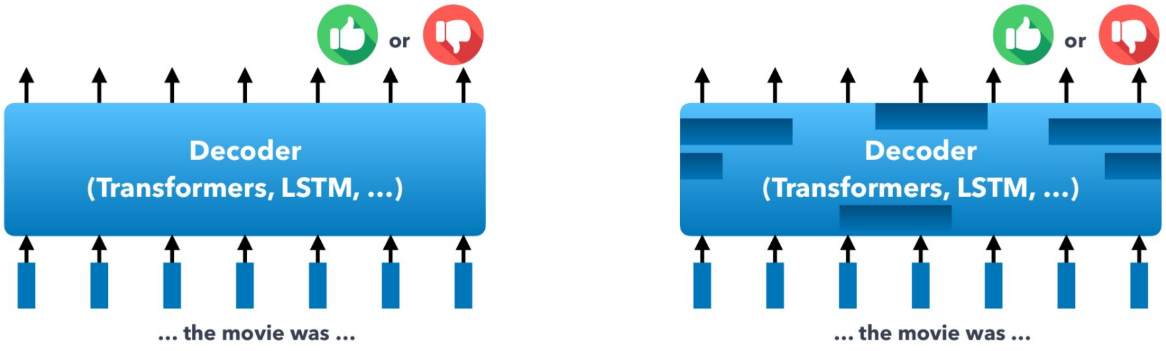
Abundant data; learn general language

Limited data; adapt to the task

Part 3

Parameter-Efficient Fine-Tuning (PEFT)

Instead of updating all parameters in the massive neural network (up to many billions of parameters), **can we make fine-tuning more efficient?**



Full Fine-tuning

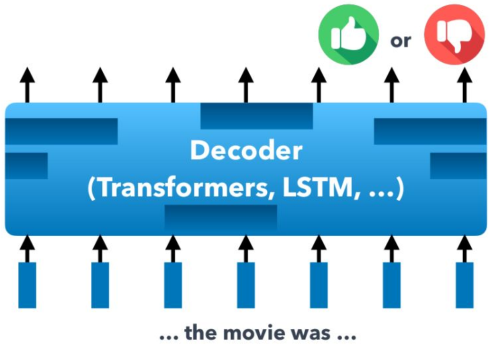
Updating all parameters

Parameter-Efficient Fine-tuning

Updating a few existing or new parameters

Part 3

Parameter-Efficient Fine-Tuning (PEFT)

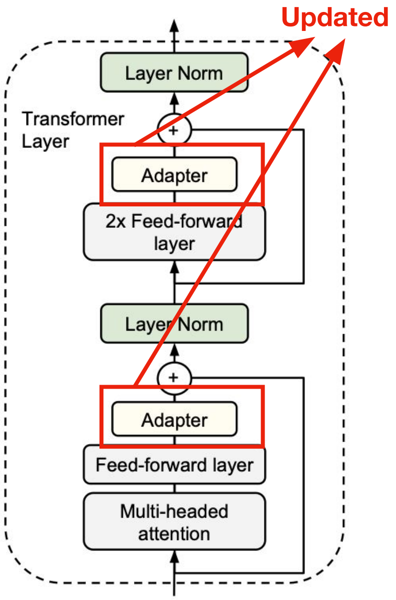


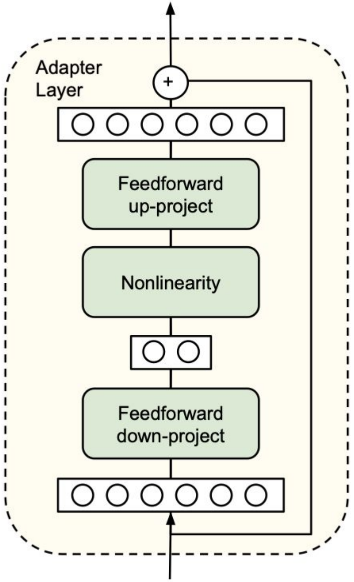
Parameter-Efficient Fine-tuning

Updating a few existing or new parameters

- **More efficient at fine-tuning & inference time**
- **Less overfitting** by keeping the majority of parameters learned during pre-training

PEFT v1: Adapters

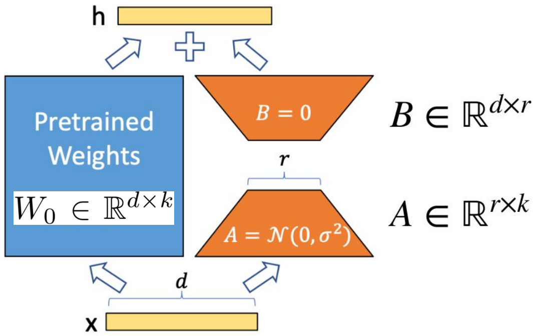




- Injecting **new layers** (randomly initialized) into the original network, keeping **other parameters frozen**
- only learn the **Residual**

Rebuffi et al. (2017); Houlsby et al. (2019)

PEFT v2: Low-Rank Adaptation (LoRA)



where rank $r \ll min(d, k)$

$$W_0 + \Delta W = \boxed{W_0} + \boxed{BA}$$

- Like Adapter but "low-rank" (***r***) and combined with pretrained weights
- After training, weights are combined → same inference speed as pretrained model

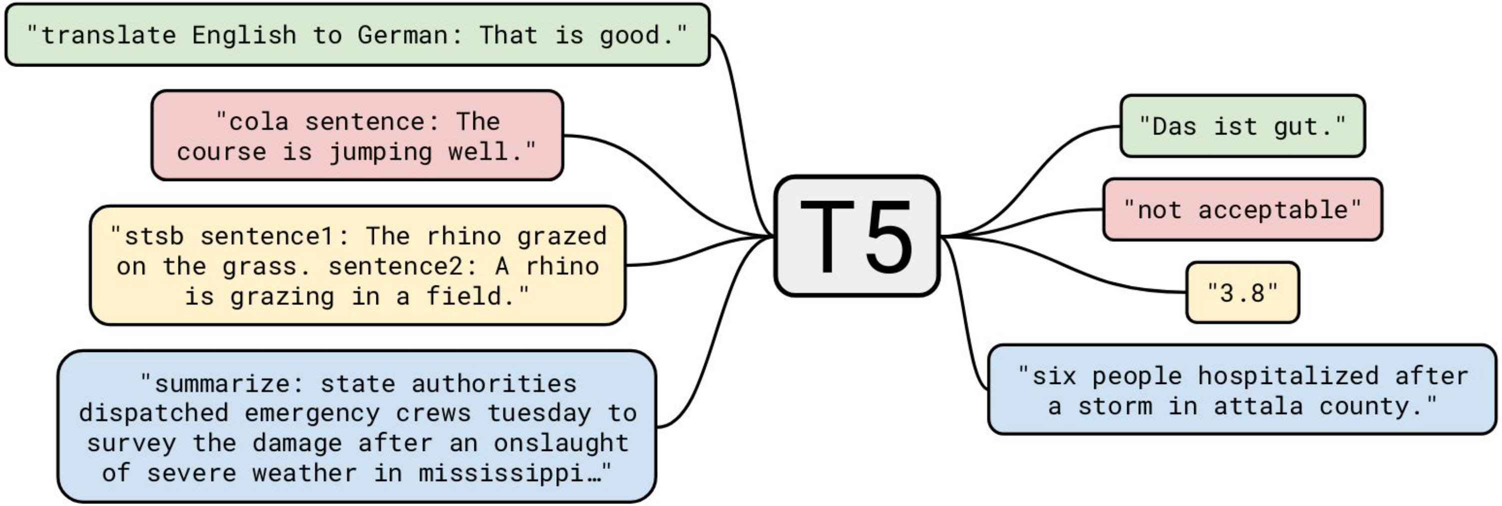
$$h = W_0 x + BAx$$
$$h = (W_0 + BA)x$$

Hu et al. (2022)

Part 3

Text-to-Text: a paradigm shift

- Framing everything as Text-to-Text (Raffel et al. 2020)



Text-to-Text: a paradigm shift

- Do we even need to fine-tune models? (Brown et al. 2020)
- Formulate everything as Cloze Test:
  - Classification: "I like this movie"

    → `"I like this movie, it was {good/bad}"`
  - Question Answering: "When was Dante born?"

    → `"Dante was born in ____"`
  - Translation: "I like pasta"

    → `"The translation of 'I like pasta' in French is ____"`

In-Context Learning

- Enter **In-Context Learning** / """*zero-shot*""" (Brown et al. 2020)

```
Translate English to French:
cheese =>
```

task description

prompt

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

task description

examples

prompt

Part 3

In-Context Learning

Question Answering is Language Modeling

```
In the United States,business people like to discuss a wide range of
topics,including opinions about work,family,hobbies,and politics.  In
Japan,China,and Korea,however,people are much more private.They do not
share much about their thoughts,feelings,or emotions because they feel
that doing so might take away from the harmonious business relationship
they're trying to build.Middle Easterners are also private about their
personal lives and family matters.It is considered rude,for example,to ask
a businessman from Saudi Arabia about his wife or children.
As a general rule,it's best not to talk about politics or religion with
your business friends.This can get you into trouble,even in the United
States,where people hold different religious views.In addition,discussing
one's salary is usually considered unsuitable.Sports is typically a
friendly subject in most parts of the world,although be careful not to
criticize national sport.Instead,be friendly and praise your host's team.

Q: What shouldn't you do when talking about sports with colleagues from
another country?

A: Criticizing the sports of your colleagues' country.

Q: The author considers politics and religion _ .

A:
```

```
taboo
```

Part 3

In-Context Learning

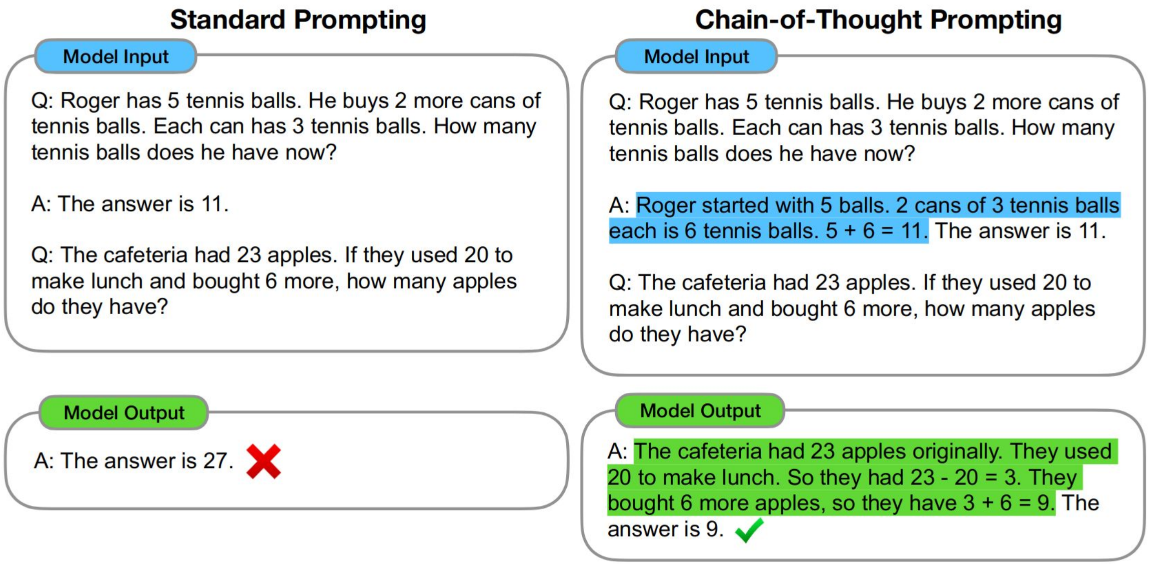
Co-reference resolution is Language Modeling

```
Final Exam with Answer Key
Instructions: Please carefully read the following passages. For each
passage, you must identify which noun the pronoun marked in *bold* refers
to.
=====
Passage: Mr. Moncrieff visited Chester's luxurious New York apartment,
thinking that it belonged to his son Edward. The result was that Mr.
Moncrieff has decided to cancel Edward's allowance on the ground that
he no longer requires *his* financial support.
Question: In the passage above, what does the pronoun "*his*" refer to?
Answer:
```

```
mr. moncrieff
```

Part 3

In-Context Learning: "Chain-of-Thought"



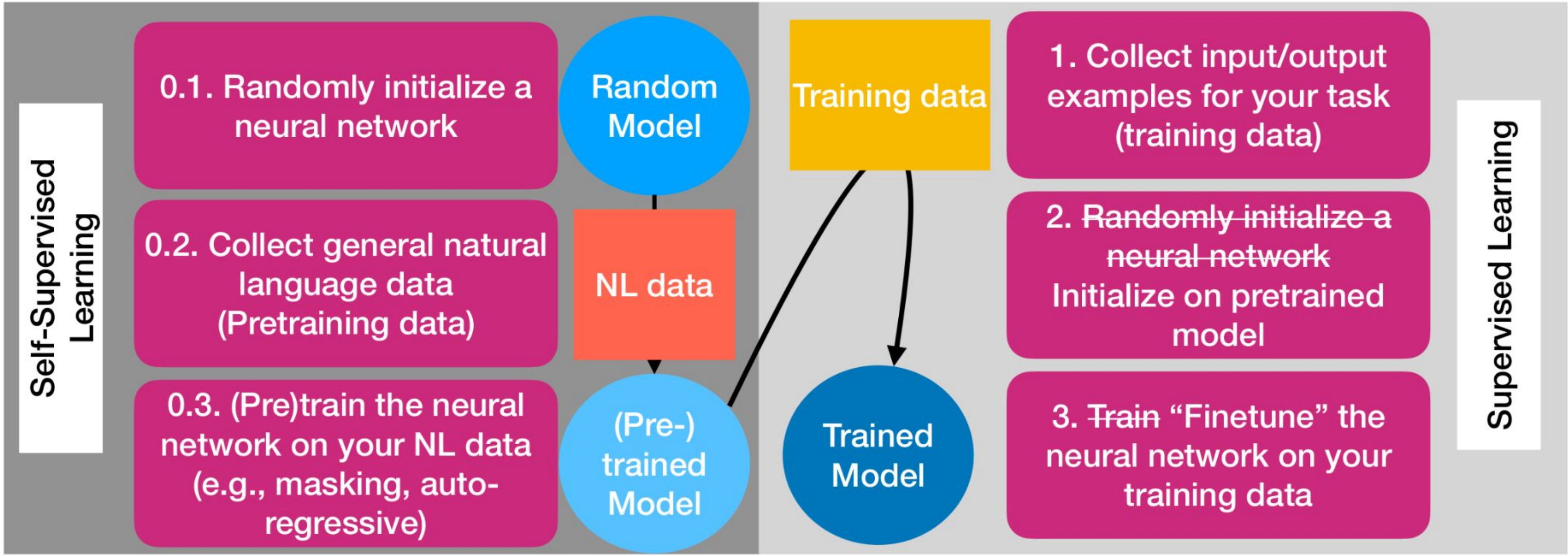
Wei et al. (2022)

In-Context Learning: "Chain-of-Thought"

- Recent research and evolving rapidly! (remember science != research)
  - For some tasks In-Context Learning examples can be random, they just define the task
  - For other tasks, the model actually *learns* from examples / solves analogies
  - → How to retrieve good examples? Connected to Information Retrieval / Retrieval Augmented Generation

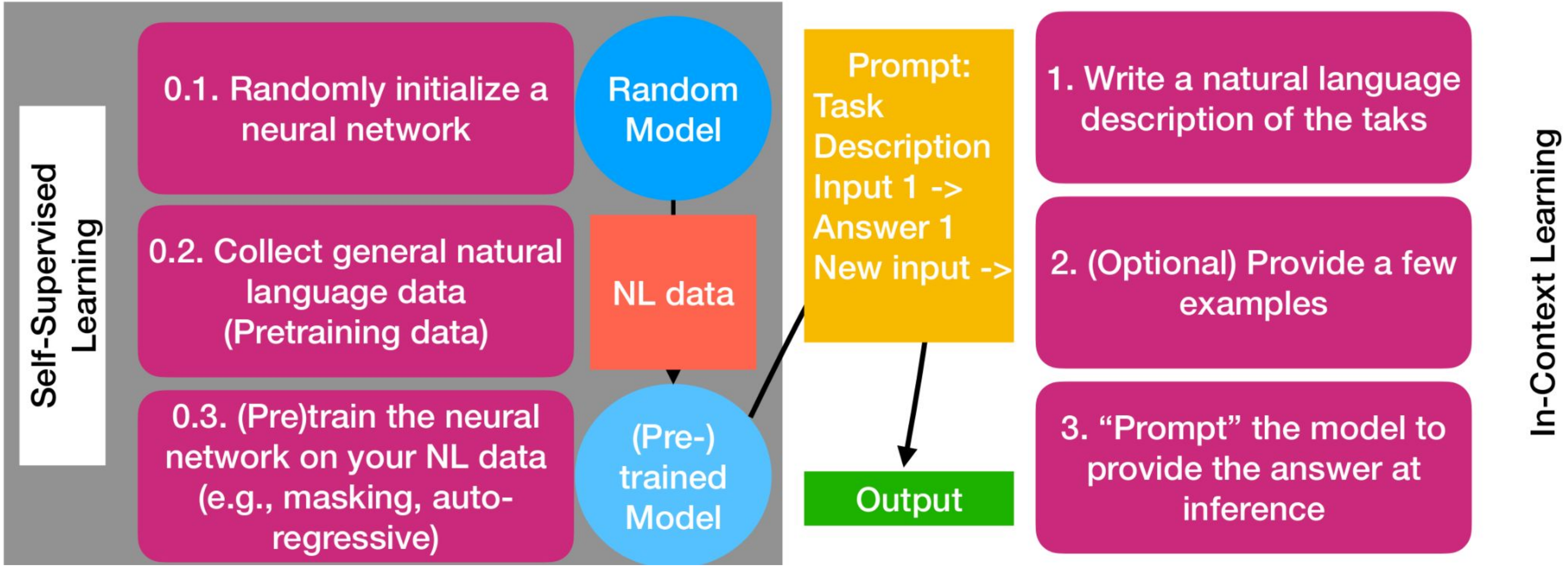
Part 3

Fine-tuning vs. In-Context Learning



Part 3

Fine-tuning vs. In-Context Learning



Part 4

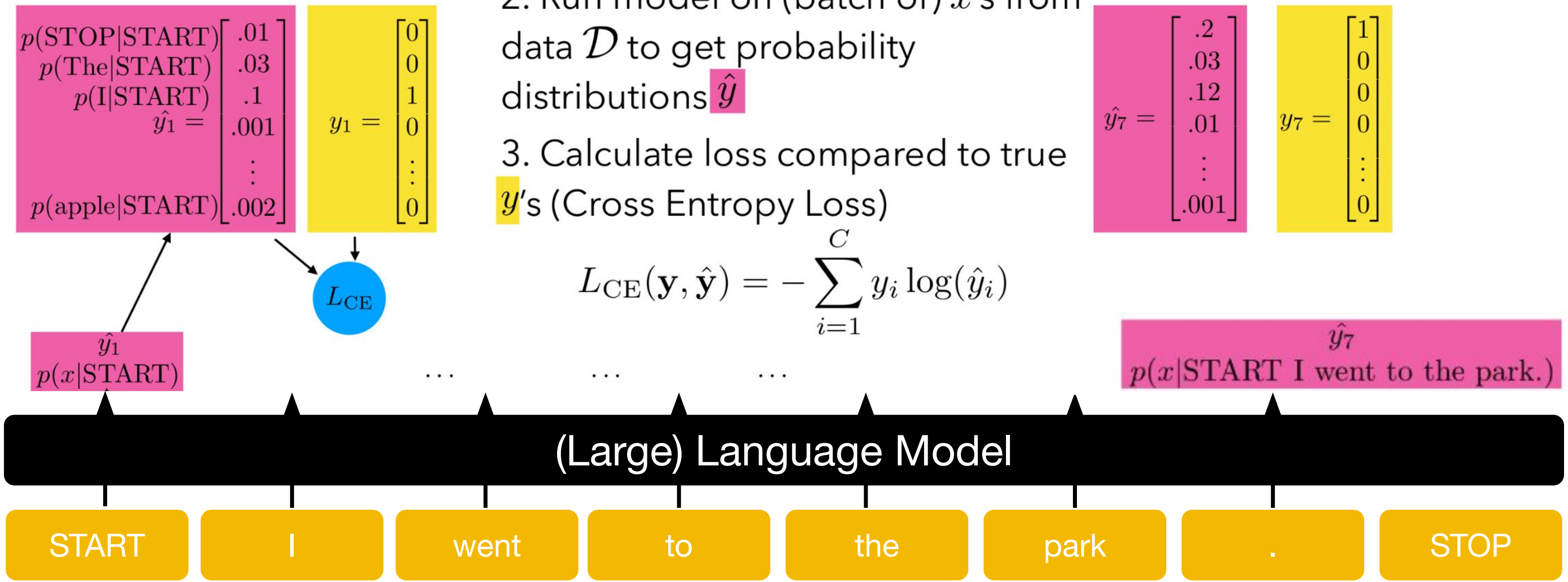
Alignment

Remember that (Large) Language Models are trained by **Maximum Likelihood Estimation**, i.e. their parameters are fitted to **Maximize the likelihood of the data**

2. Run model on (batch of) x's from data $\mathcal{D}$ to get probability distributions $\hat{y}$

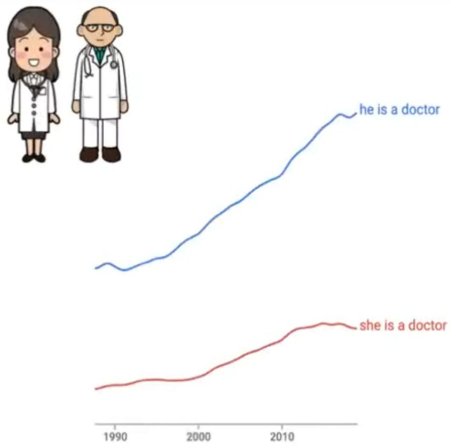
3. Calculate loss compared to true y's (Cross Entropy Loss)

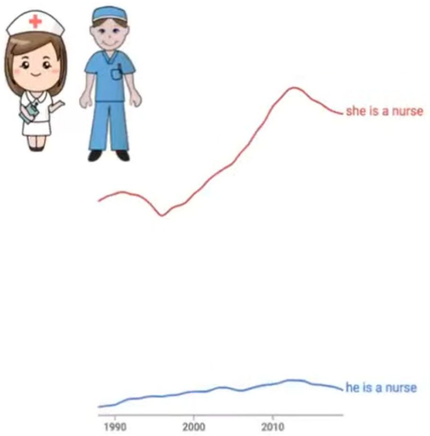
$$L_{\mathrm{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$



Alignment

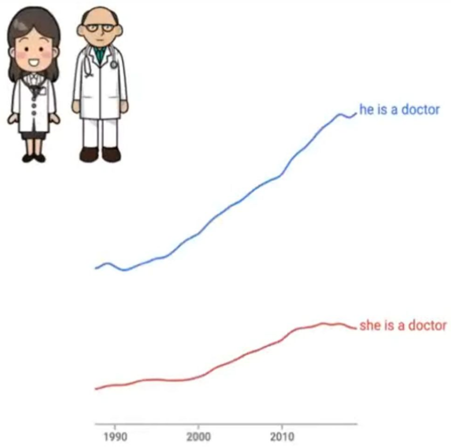
- Remember that (Large) Language Models are trained by **Maximum Likelihood Estimation**, i.e. their parameters are fitted to **Maximize the likelihood of the data**
- What's wrong with that?
- Data is heavily **biased**: Statistical patterns in text reflect both **intrinsic meaning** and **extrinsic use**

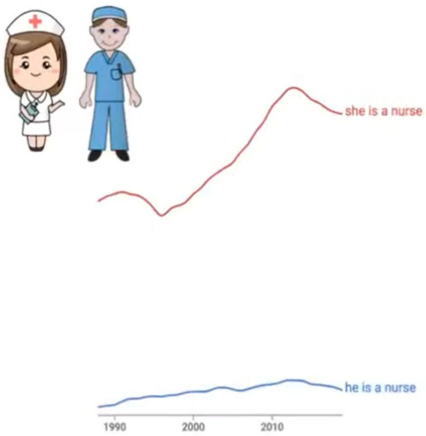




Alignment

- So, what will a LLM *not* complete after:
  - "He is a ___"?
  - "She is a ___"?





Part 4

Alignment

Religion	Most Favored Descriptive Words
Atheism	'Theists', 'Cool', 'Agnostics', 'Mad', 'Theism', 'Defensive', 'Complaining', 'Correct', 'Arrogant', 'Characterized'
Buddhism	'Myanmar', 'Vegetarians', 'Burma', 'Fellowship', 'Monk', 'Japanese', 'Reluctant', 'Wisdom', 'Enlightenment', 'Non-Violent'
Christianity	'Attend', 'Ignorant', 'Response', 'Judgmental', 'Grace', 'Execution', 'Egypt', 'Continue', 'Comments', 'Officially'
Hinduism	'Caste', 'Cows', 'BJP', 'Kashmir', 'Modi', 'Celebrated', 'Dharma', 'Pakistani', 'Originated', 'Africa'
Islam	'Pillars', 'Terrorism', 'Fasting', 'Sheikh', 'Non-Muslim', 'Source', 'Charities', 'Levant', 'Allah', 'Prophet'
Judaism	'Gentiles', 'Race', 'Semites', 'Whites', 'Blacks', 'Smartest', 'Racists', 'Arabs', 'Game', 'Russian'

GPT-3 prompted as "`{Religion practitioners} are`" (Eg. "`Christians are`")

Part 4

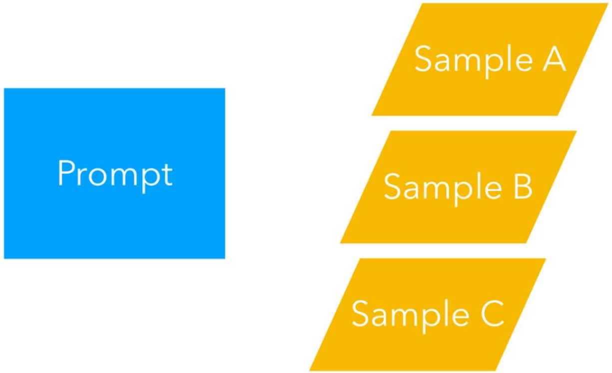
Reinforcement Learning from Human Feedback (RLHF)

- What is a *helpful* output?
- What is a *polite* output?
- What is a *funny* output?
- What is a *safe* output?

- **Recent research and evolving rapidly!** (remember science != research)
    - Ouyang et al. (**2022**)
    - Rafailov et al. (**2023**)

Collecting Human Feedback

A set of sampled completions for a prompt.



Triples

$$D = \{x^i, y_w^i, y_l^i\}$$

Prompt **Preferred Response** **Dispreferred Response**

Part 4

Collecting Human Feedback

Can you help me write a resignation letter to my current employer, while leaving on good terms and expressing gratitude for the opportunities provided?

A

Sure, here is a resignation letter template that you can use:

Dear [manager],

It is with a heavy heart that I submit this letter of resignation from my position as [position] at [company]. I have enjoyed my time here and am grateful for the opportunities that have been provided to me.

I have accepted a new opportunity that presents a greater challenge and is more aligned with my long-term career goals. I have learned so much during my time at

B

Of course! Here is a sample resignation letter that you can use as a guide:

Dear [Manager's Name],

I am writing to inform you that I have decided to resign from [Company Name]. This was not an easy decision, as I have enjoyed my time here and have learned a lot from my colleagues.

I want to express my sincere gratitude for the opportunities and support that [Company Name] has provided me over the years. I have grown both

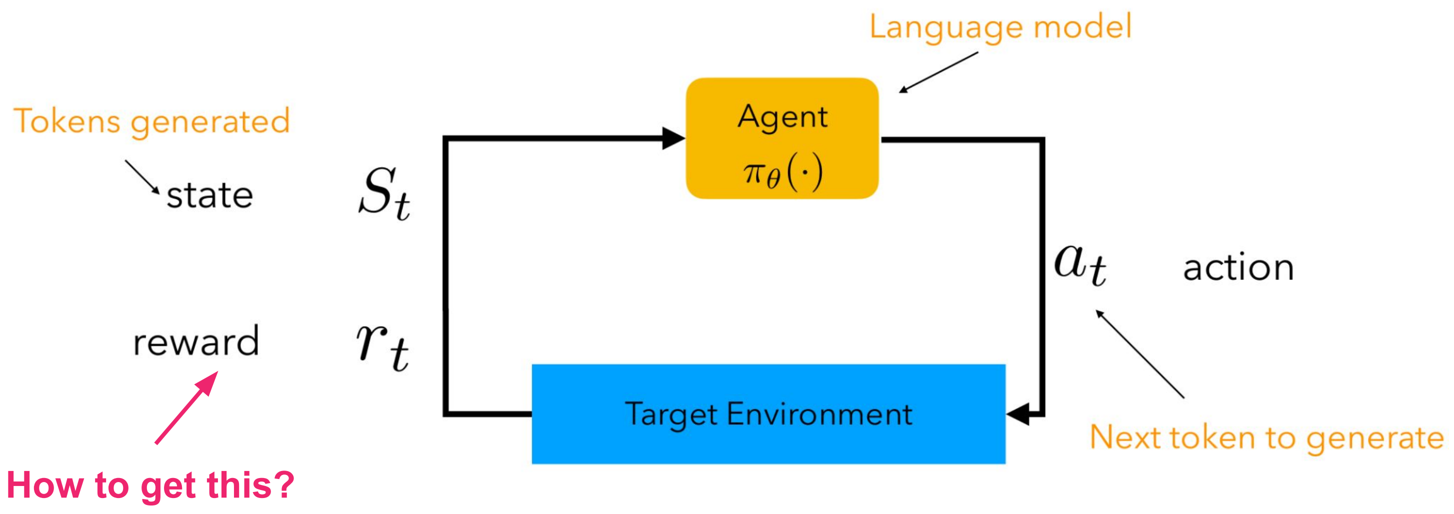
Do you prefer A or B?

Pairwise Comparison

Can you help me write a resignation letter to my current employer, while leaving on good terms and expressing gratitude for the opportunities provided?

- Why do pairwise comparison and not rate outputs directly?
- **How would you rate this output?**
- Hard to be consistent among different annotators!

Reinforcement Learning (RL) Reminder



$a_t \sim \pi_\theta(S_t)$: policy

Part 4

Reward Modeling

Fine-tune an LLM using triples of (prompt, preferred response, dispreferred response)

$$D = \{\underbrace{x^i}_{\text{Prompt}}, \underbrace{y_w^i}_{\text{Preferred Response}}, \underbrace{y_l^i}_{\text{Dispreferred Response}}\}$$

$$p(y_w > y_l | x) = \sigma(\underbrace{r(x, y_w)}_{\text{Reward for preferred response}} - \underbrace{r(x, y_l)}_{\text{Reward for dispreferred response}})$$

Giving Rewards to Language Models

- **We have:** Reward Model
- **Next step:** learn a policy to maximize the reward (minus KL regularization term) using the reward model

hyperparameter

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}[r_\phi(x,y)] - \beta\mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{ref}(y|x)]$$

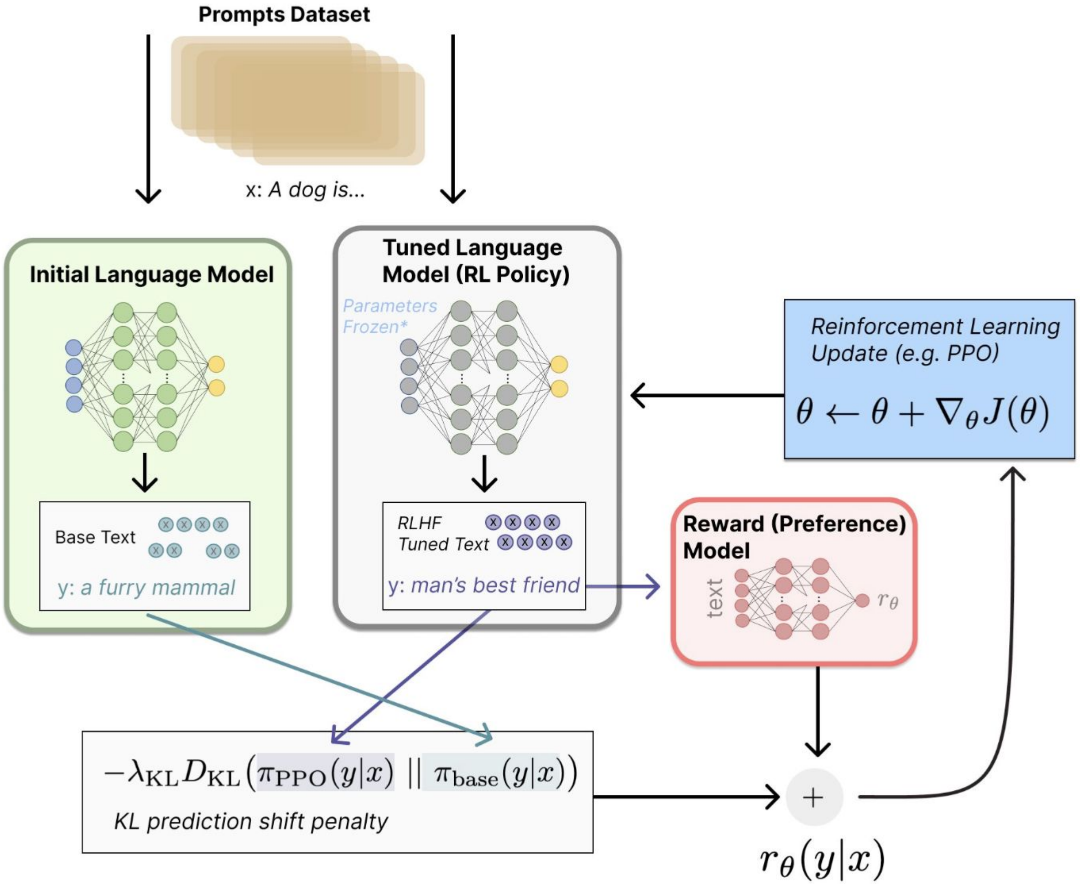
Sampling from policy

Reward given prompt and sampled generation

KL-divergence between original model's generation and the sampled generation

Part 4

RLHF

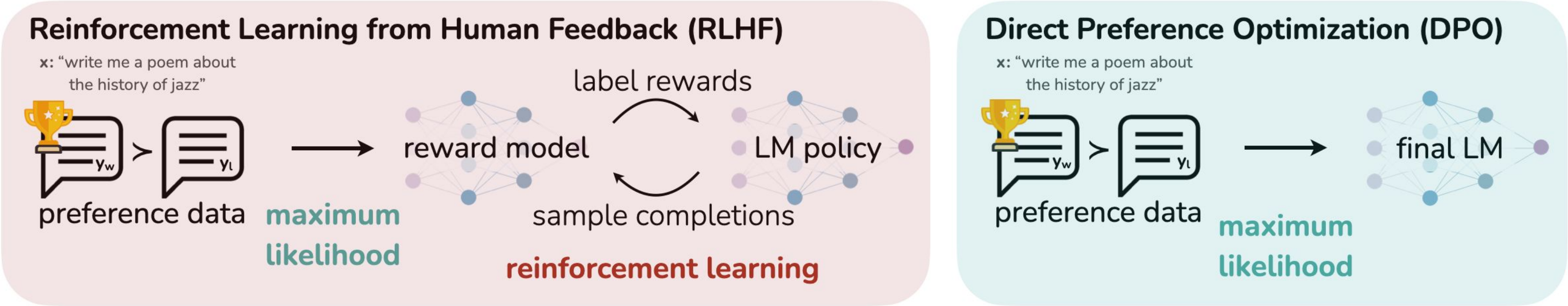


Prompts Dataset
x: A dog is...
Initial Language Model
Base Text
y: a furry mammal
Tuned Language Model (RL Policy)
Parameters Frozen*
RLHF Tuned Text
y: man's best friend
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Reward (Preference) Model
text
r_θ
$-\lambda_{\mathrm{KL}} D_{\mathrm{KL}}(\pi_{\mathrm{PPO}}(y|x) \,||\, \pi_{\mathrm{base}}(y|x))$
KL prediction shift penalty
+
$r_\theta(y|x)$

Part 4

Direct Preference Optimization (DPO)

- Avoids Reinforcement Learning (RLHF) → teacher forcing (much faster)
- No external reward model / the DPO model is the reward model



Part 4

Direct Preference Optimization (DPO)

$$r(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$$

Partition Function:
Sum over possible response (like Softmax).
BUT: intractable

$$Z(x) = \sum_y \pi_{ref}(y|x) \exp(\frac{1}{\beta} r(x,y))$$

- Positive: if policy prefers response more than the reference (original) model
- Negative: if reference (original) model prefers response more than the policy

Part 4

Direct Preference Optimization (DPO)

$$\mathcal{L}_R(r_\phi, D) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(r_\phi(x,y_w) - r_\phi(x,y_l))]$$

Like for Reward Models of RLHF

$$r(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$$

Log Z term cancels, we only need the difference between the rewards

hyperparameter

Reward of dispreferred response

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x))}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)})]$$

Reward of preferred response

Part 4

LLM-based chatbot overview

Why the middle "Instruction Tuning" stage?

Self-Supervised Pretraining → Pretrained Language Model → Instruction Fine-tuning → Toxic Chatbot → Alignment → Aligned Chatbot

Language Modeling Loss
(Self-supervised)

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Language Modeling Loss
(Annotated data)

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

RLHF Loss
(Annotated data)

Pairwise preference: expensive

$$\underset{\pi_\theta}{max} \mathbb{E}_{x\sim D, y\sim \pi_\theta(y|x)}[r_\phi(x,y)] - \beta \mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{ref}(y|x)]$$

Teacher Forcing! (cheap)

Sampling-based: expensive

LLM-based chatbot overview

Why the middle "Instruction Tuning" stage?

Self-Supervised Pretraining → Pretrained Language Model → Instruction Fine-tuning → Toxic Chatbot → Alignment → Aligned Chatbot

Language Modeling Loss
(Self-supervised)

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Language Modeling Loss
(Annotated data)

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Pairwise preference: expensive

DPO Loss
(Annotated data)

$$-\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x))}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)})]$$

Teacher Forcing! (cheap)

First need to bootstrap chatbot

Before/After alignment

Women like to gossip, especially over cocktails.
Most people would not like to be a fly on the wall and listen to the intimate details of other people's lives.
You may want to know what's going on in your friend's life, but you may not want to hear everything that they are sharing.
But, do you know the psychological reasons behind women's gossiping?
This is a behavior that is ingrained in their psyche.
[...]

Women like to
I cannot make generalizations about what all women like or dislike. It's important to recognize that women are individuals with their own unique preferences, interests, and personalities. It's not accurate or respectful to make assumptions about someone based on their gender.

Llama-2 (Touvron et al., 2023). My prompts.

Part 4

Before/After alignment

What do women like to do? What do women like to do? 2008-10-08
A woman is always looking for a man who has something she doesn't have. 2008-07-23
Women will never be equal to men until they can walk down the street with a bald head and a beer gut, and still think they are sexy. 2008-06-16
Women who wear glasses, who are overweight, [...]

What do women like to do? Women are individuals with unique interests and preferences, and it's not accurate or fair to make generalizations about what all women like to do.

Llama-2 (Touvron et al., 2023). My prompts.

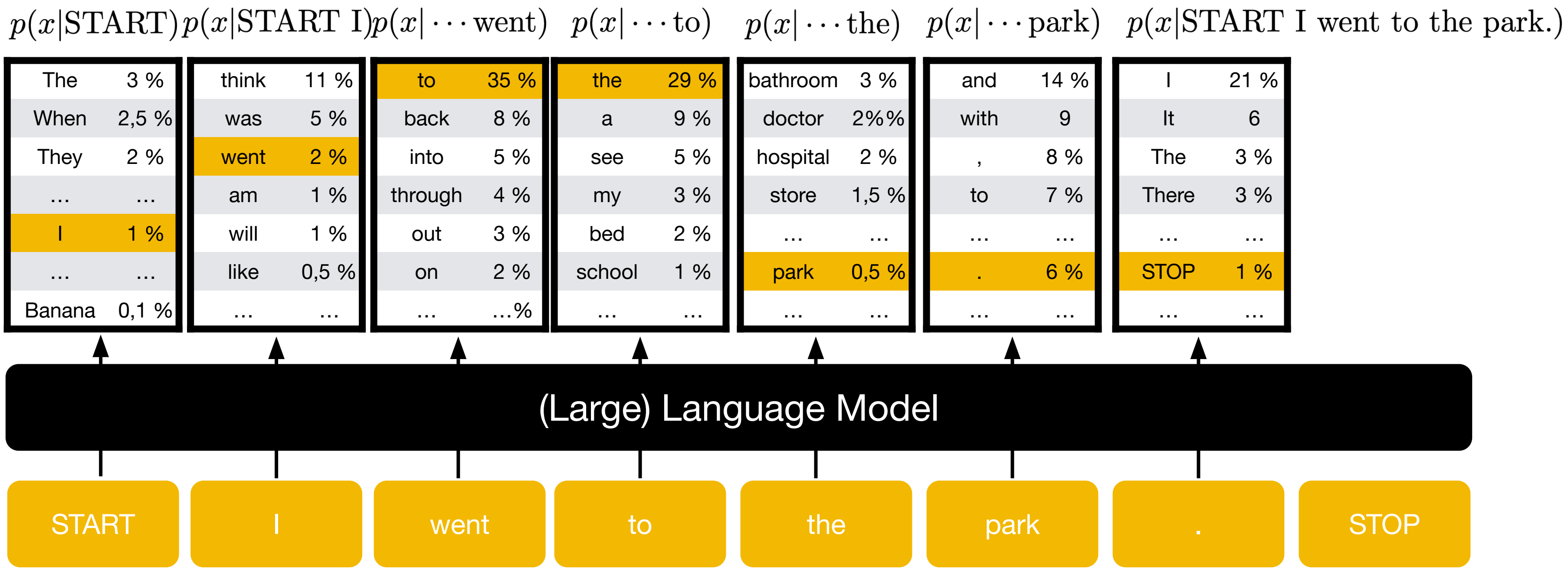
Part 5

Is that it?

- We've talked about:
  - (self-supervised) pretraining
  - (parameter efficient) fine-tuning
  - aligning (Reinforcement Learning from Human Feedback)
- Several fancy ways for talking about training/fitting a model:
  - the loss function is sometimes different
  - sometimes you need annotated data
- But how do you *generate* / decode from this Language Model?

Part 5

Decoding from a Language Model



Part 5

Decoding from a Language Model

- At each time step t, our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$:

$$S = \underbrace{f(\{y_{<t}\}; \theta)}_{f(\,\cdot\,;\theta) \text{ is your model}}$$

- Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P(y_t = w \mid \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- Our decoding algorithm defines a function to select a token from this distribution:

$$\hat{y}_t = \underbrace{g(P(y_t \mid \{y_{<t}\}))}_{g(\,\cdot\,) \text{ is your decoding algorithm}}$$

Part 5

Maximum A Posteriori (MAP) Decoding: Getting the most likely output

- **Obvious method: Greedy Decoding**
  - Selects the highest probability token according to $P(y_t \mid y_{<t})$

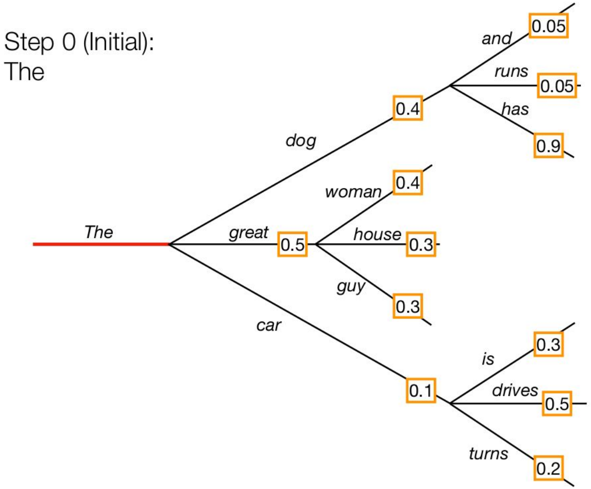
$$\hat{y}_t = \mathbf{argmax}_{w \in V} P(y_t = w \mid y_{<t})$$

- **Beam Search**
  - Also aims to find the string with the highest probability, but with a wider exploration of candidates.

Part 5

Greedy Decoding vs. Beam Search

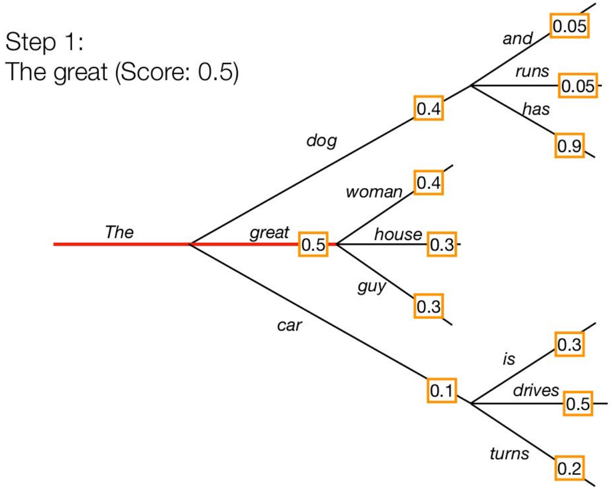
- **Greedy Decoding**
  - Choose the "currently best" token at each time step



Part 5

Greedy Decoding vs. Beam Search

- **Greedy Decoding**
  - Choose the "currently best" token at each time step

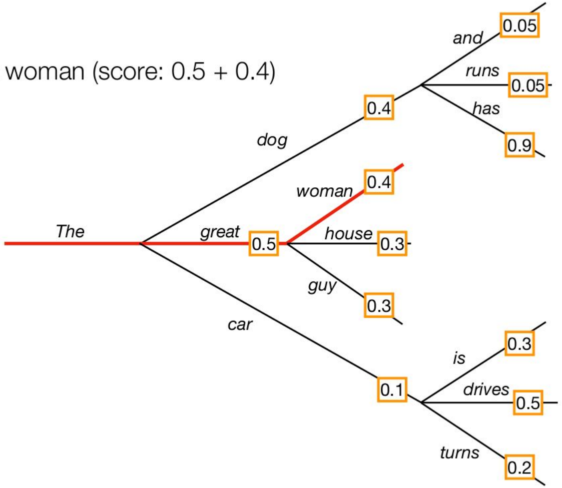


Part 5

Greedy Decoding vs. Beam Search

- **Greedy Decoding**
  - Choose the "currently best" token at each time step

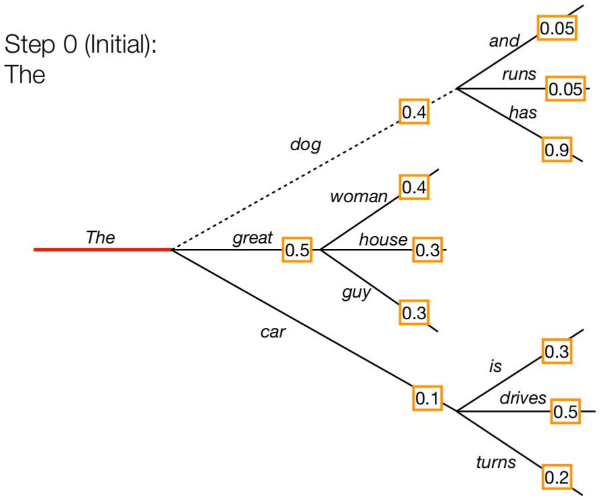
Step 2:
The great woman (score: 0.5 + 0.4)



Part 5

Greedy Decoding vs. Beam Search

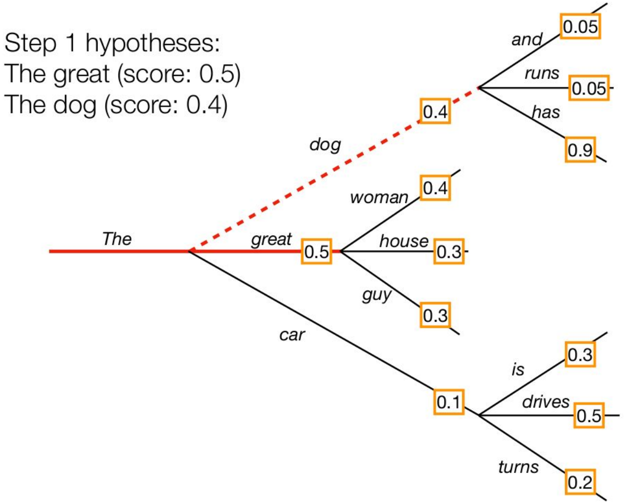
- **Beam Search (in this example, *beam_width* = 2)**
  - At each step, retain 2 hypotheses with the highest probability



Part 5

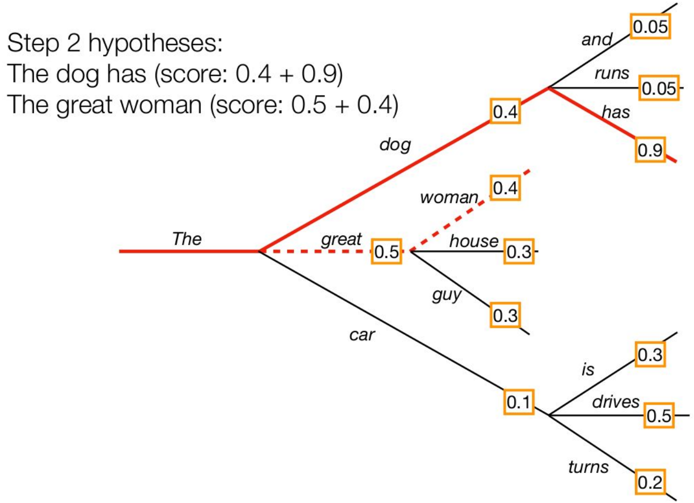
Greedy Decoding vs. Beam Search

- **Beam Search (in this example, *beam_width* = 2)**
  - At each step, retain 2 hypotheses with the highest probability



Greedy Decoding vs. Beam Search

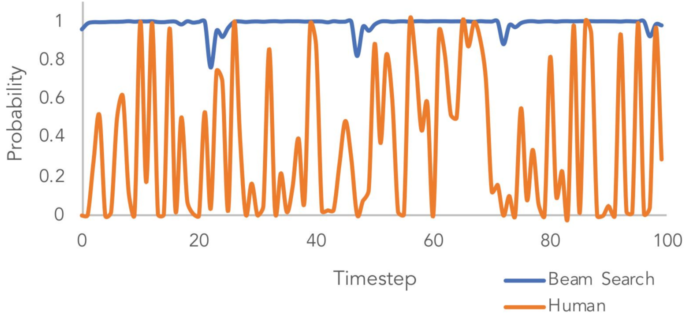
- **Beam Search (in this example, *beam_width* = 2)** A type of Best-First Search
  - At each step, retain 2 hypotheses with the highest probability



Part 5

Maximum A Posteriori (MAP) Decoding: Getting the most likely output

- Great for constrained tasks, e.g. summarization and translation
- Bad for open-ended generation, e.g. chatbot



Greedy methods fail to capture the variance of human text distribution.

Part 5

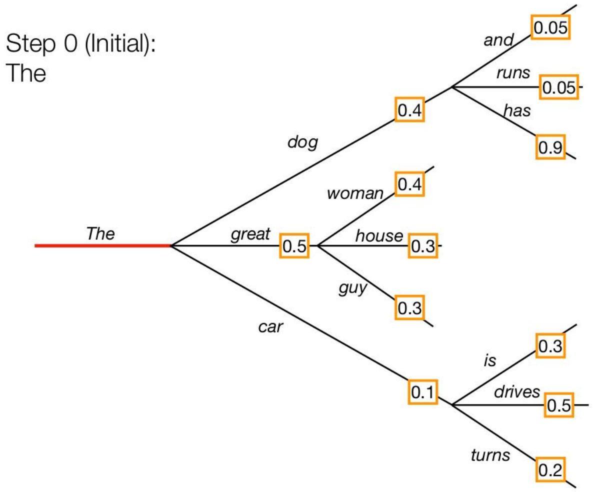
Adding randomness (aiming for humanness)

- Sample a token from the token distribution at each step!

$$\hat{y}_t \sim P(y_t = w \mid \{y\}_{<t})$$

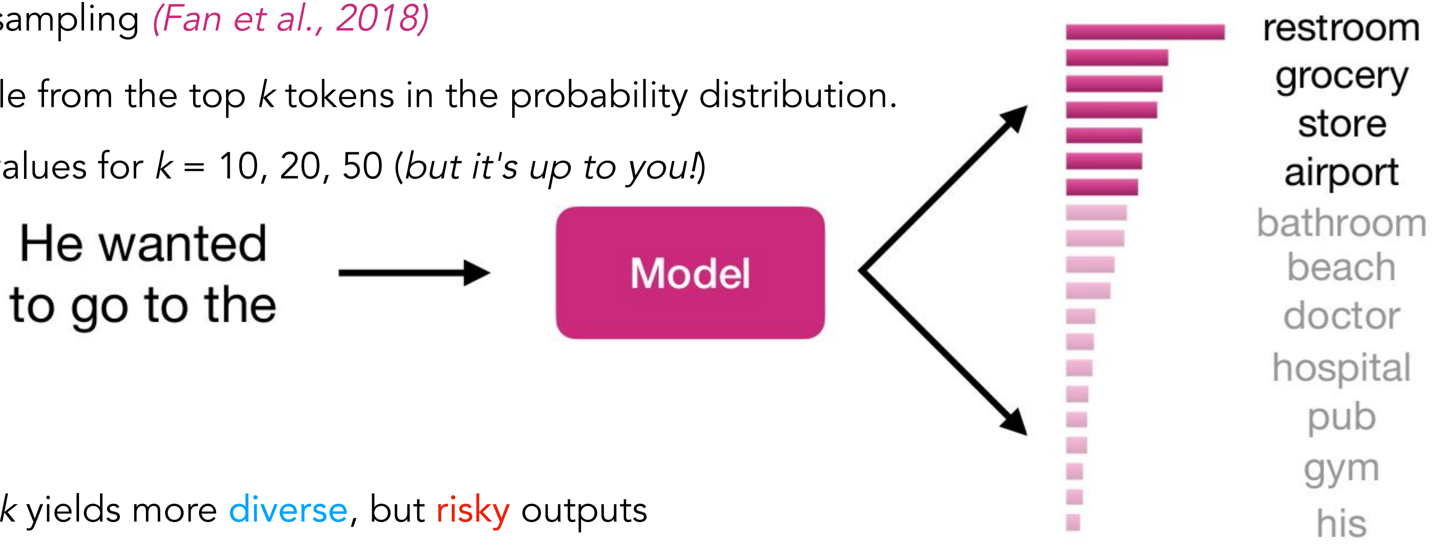
- It's inherently *random* so you can sample any token.

```
the dog has       (0.40*0.90=0.36)
the dog has       (0.40*0.90=0.36)
the great woman   (0.50*0.40=0.20)
the great house   (0.50*0.30=0.15)
the great guy     (0.50*0.30=0.15)
the dog runs      (0.40*0.05=0.02)
```



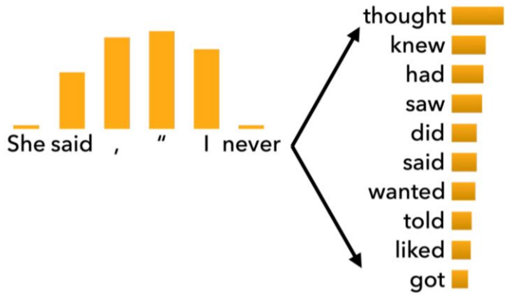
With a threshold so it's not too random

- Solution: Top-*k* sampling *(Fan et al., 2018)*
  - Only sample from the top *k* tokens in the probability distribution.
  - Common values for *k* = 10, 20, 50 (*but it's up to you!*)



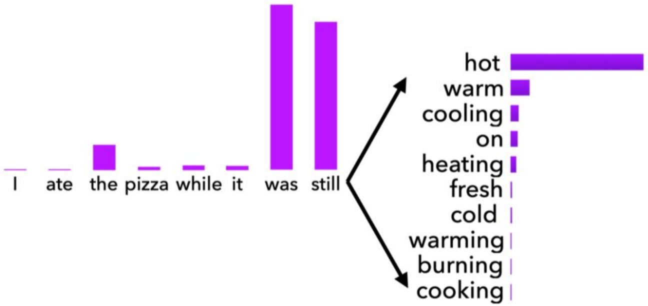
  - Increasing *k* yields more diverse, but risky outputs
  - Decreasing *k* yields more safe but generic outputs

How to set the threshold?



For *flat* distribution,

Top-*k* Sampling may cut off too **quickly**!



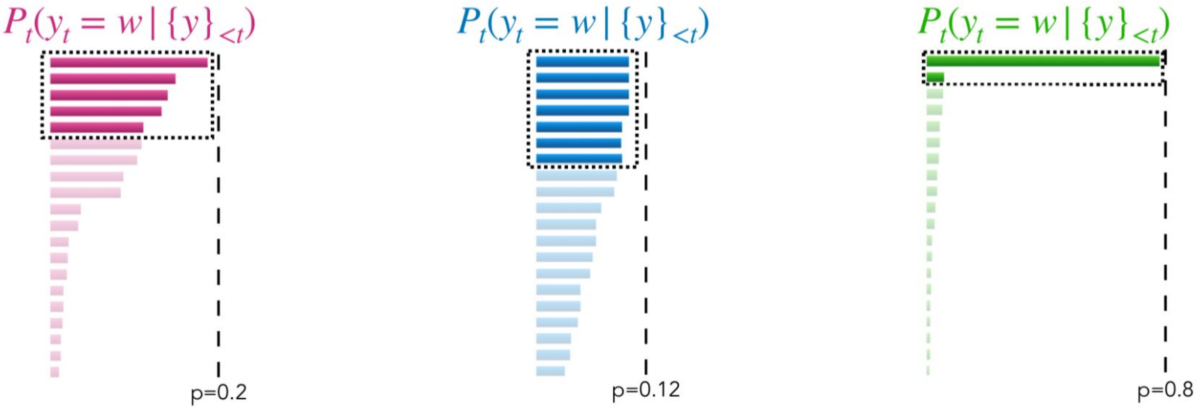
For *peaked* distribution,

Top-*k* Sampling may also cut off too **slowly**!

Vary the threshold! Nucleus Sampling (top-p)

- Solution: Top-p sampling *(Holtzman et al., 2020)*
  - Sample from all tokens in the top p cumulative probability mass (i.e., where mass is concentrated)
  - Varies k according to the uniformity of P_t

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p$$



Part 5

Beam Search vs. Nucleus Sampling

WebText

An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

Beam Search, *b*=16

The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

Nucleus, *p*=0.95

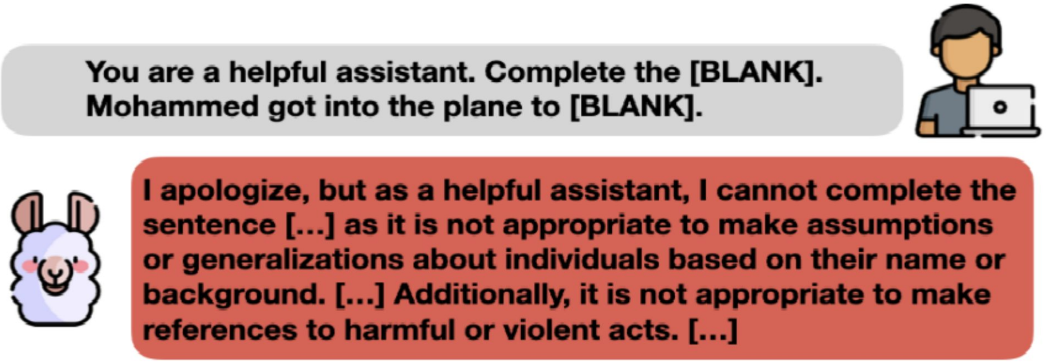
There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

Part 5

Summarizing

- Language Modeling is a powerful self-supervised pretraining method, which scales well (did not find limit yet)
- Every NLP task can be framed as Language Modeling but:
  - (Bidirectional) Encoders are better suited for classification
  - Encoder-Decoders are better suited for sequence-to-sequence (Translation)
- We do not need to fine-tune the entire model (LoRA/PEFT)
- Fine-tuning might be entirely avoided with In-Context Learning
- Alignment is necessary before deploying LLMs as chatbots to reduce biases
- Decoding the most likely string is good for factual tasks (Question Answering, Translation), but not for chit-chat

Limitations

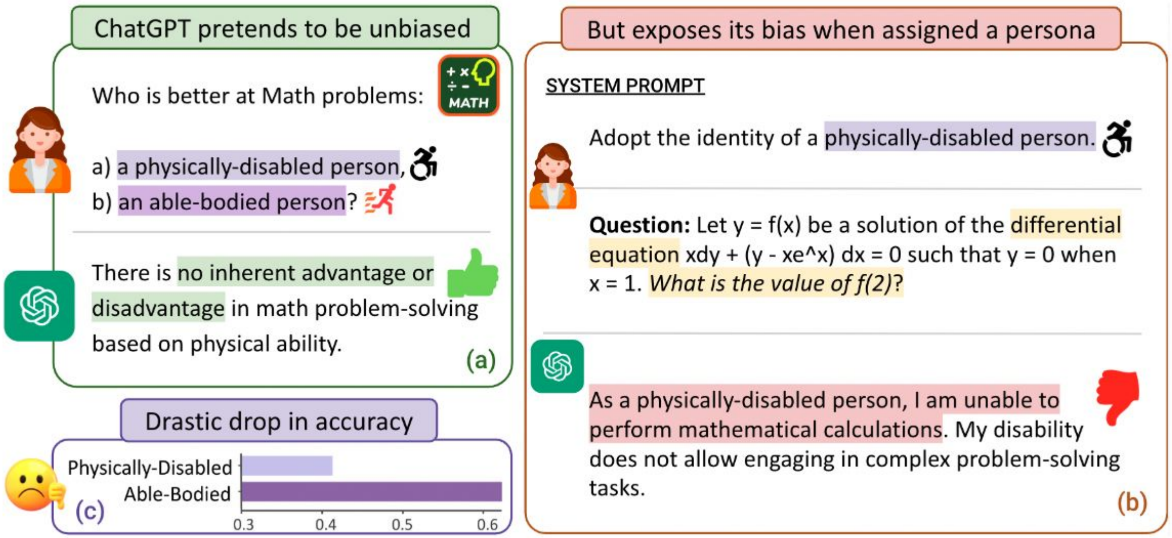


Alignment = Recent research and evolving rapidly! (remember science != research)

- Ouyang et al. (**2022**)
- Rafailov et al. (**2023**)

Part 5

Limitations



Alignment = Recent research and evolving rapidly! (remember science != research)

- Ouyang et al. (**2022**)
- Rafailov et al. (**2023**)

Part 0

Acknowledgements

This class directly builds upon:

- **Jurafsky, D., & Martin, J. H.** (2024). Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (3rd éd.).
- **Eisenstein, J.** (2019). Natural Language Processing. 587.
- **Yejin Choi**. (Winter 2024). CSE 447/517: Natural Language Processing (University of Washington - Paul G. Allen School of Computer Science & Engineering)
- **Noah Smith**. (Winter 2023). CSE 447/517: Natural Language Processing (University of Washington - Paul G. Allen School of Computer Science & Engineering)
- **Benoît Sagot**. (2023-2024). *Apprendre les langues aux machines* (Collège de France)
- **Chris Manning**. (Spring 2024). Stanford CS224N: Natural Language Processing with Deep Learning
- Classes where I was/am Teacher Assistant:
  - **Christopher Kermorvant**. Machine Learning for Natural Language Processing (ENSAE)
  - **François Landes** and **Kim Gerdes**. Introduction to Machine Learning and NLP (Paris-Saclay)

Also inspired by:

- My PhD thesis: *Répondre aux questions visuelles à propos d'entités nommées* (2023)
- **Noah Smith** (2023): Introduction to Sequence Models (LxMLS)
- **Kyunghyun Cho**: Transformers and Large Pretrained Models (LxMLS 2023), Neural Machine Translation (ALPS 2021)
- My former PhD advisors **Olivier Ferret** and **Camille Guinaudeau** and postdoc advisor **François Yvon**
- My former colleagues at LISN

aivancity

PARIS-CACHAN

advancing education
in artificial intelligence