Supervised
Unsupervised
Semi-supervised
Weakly-supervised
Multi-task
Transfer
Few-shot
Zero-shot
Self-supervised
Large language-models
Reinforcement

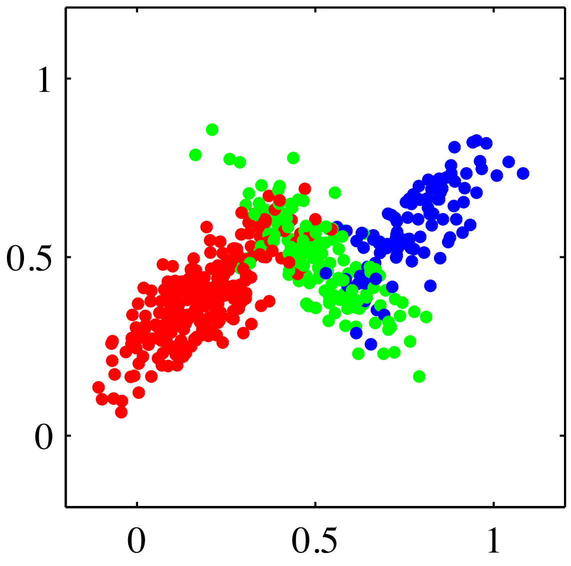
# Learning

# CS229: Machine Learning

Carlos Guestrin

Stanford University

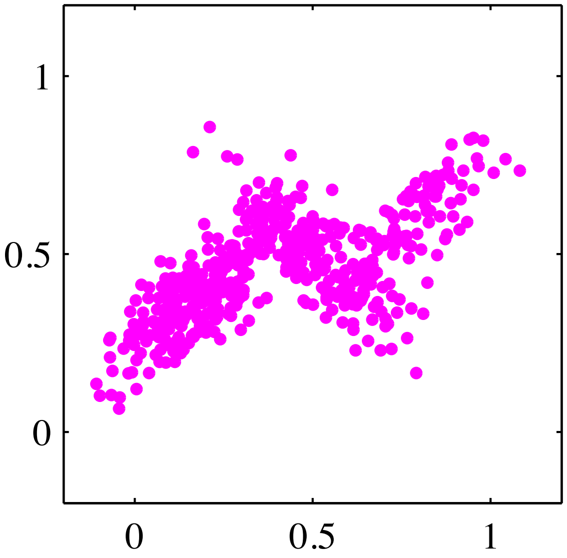
# Supervised Learning



- Observe:
  - □ Features **x**
  - □ Labels y (for all data points)

- Learning goal:
  - □ Model to predict y from **x**

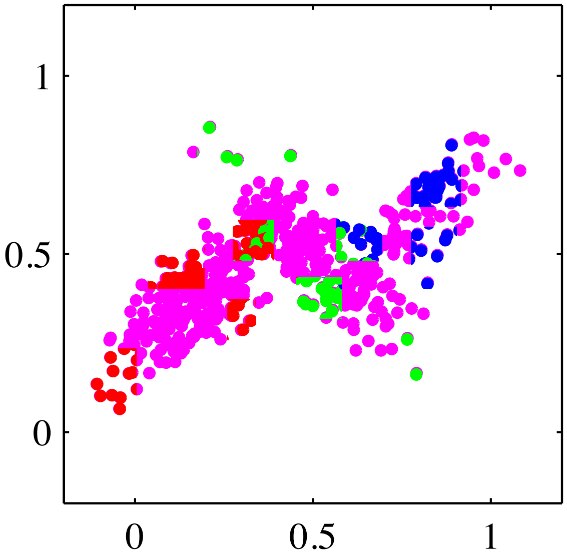
# Unsupervised Learning



- Observe:
  - Features **x**
- Learning goal:
  - Discover structure in space of **x**, e.g.:
    - Clustering: infer cluster labels z
      - Typically one cluster per input
    - Dimensionality reduction: discover lower dimensional subspaces, e.g.:
      - PCA – linear subspace
      - Embeddings – general vector space
    - Topic modeling: infer cluster labels z
      - Input can belong to multiple clusters

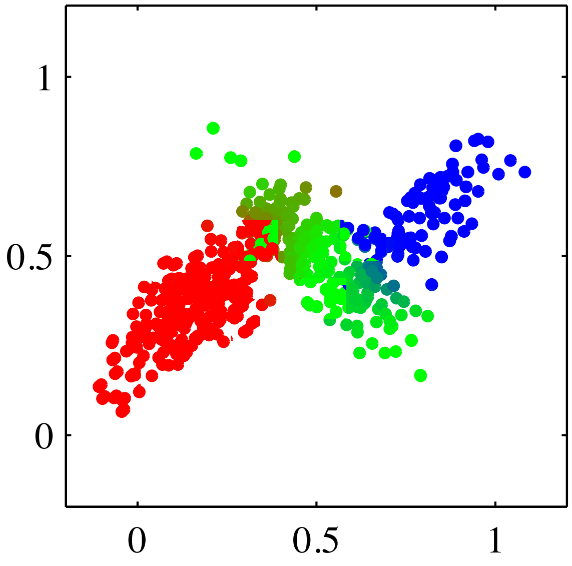
# Learning from less data: semi-supervised, weakly supervised, multitask, transfer, few-shot, one-shot learning

# Semi-supervised Learning



- Observe:
  - □ Features **x** for all data points
  - □ Labels y only for some data points
- Learning goal:
  - □ Model to predict y from **x**

# Very Simple Semi-supervised learning algorithm

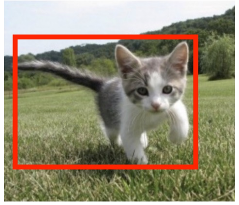


*Soft assignments to clusters*

- Consider responsibilities in EM:

$$r_{ik} = p(z^i = k | x^i, \pi, \mu, \Sigma)$$

# Weakly Supervised Learning

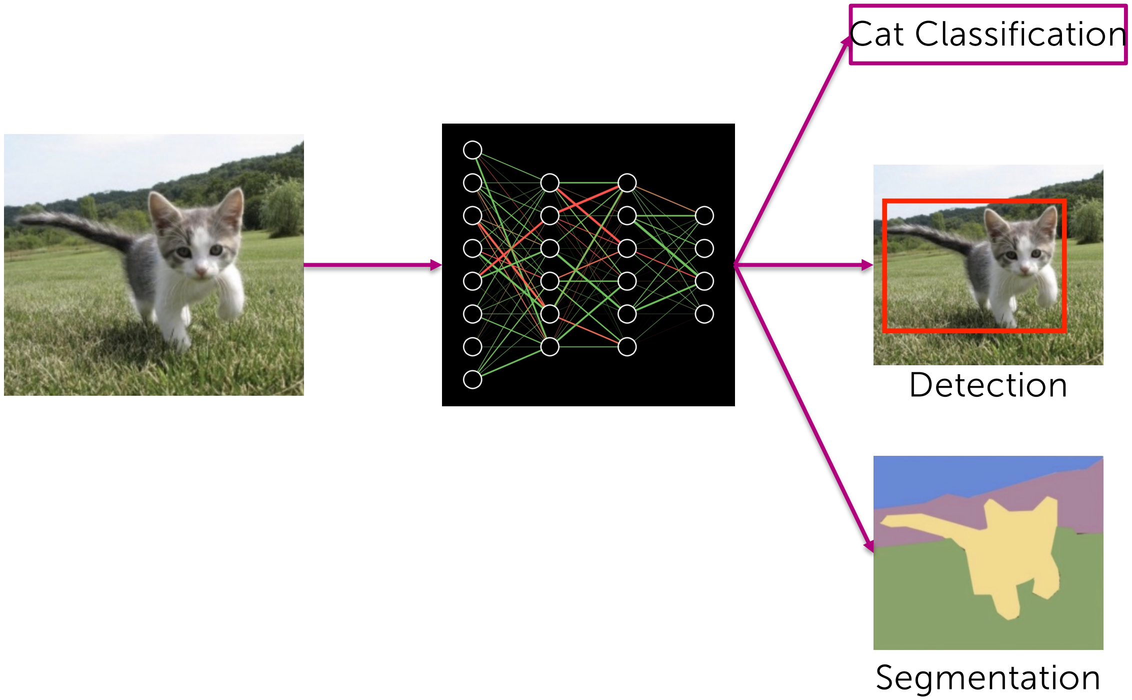

Label y: Perfect bounding box

Imprecise label

Inaccurate label

- Decrease cost or complexity of labeling by using “surrogate” labels
- Observe:
  - Features **x**
  - Some signal z related to true label y:
    - Imprecise labels – simpler, high-level labels
    - Inaccurate labels – inexpensive, lower-quality labels
    - Existing resources – knowledge bases or heuristics to generate labels
- Learning goal:
  - Model to predict y from **x**

# Multitask Learning



Cat Classification

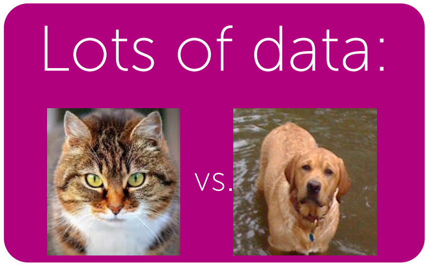
Detection

Segmentation

- Observe:
  - □ k tasks
  - □ Each data point:
    - Features **x**
    - Labels $y_j$ for task j
      - □ Potentially labels for multiple tasks

- Learning goal:
  - □ Model to predict $y_1,\dots,y_k$ from **x**

# Transfer Learning

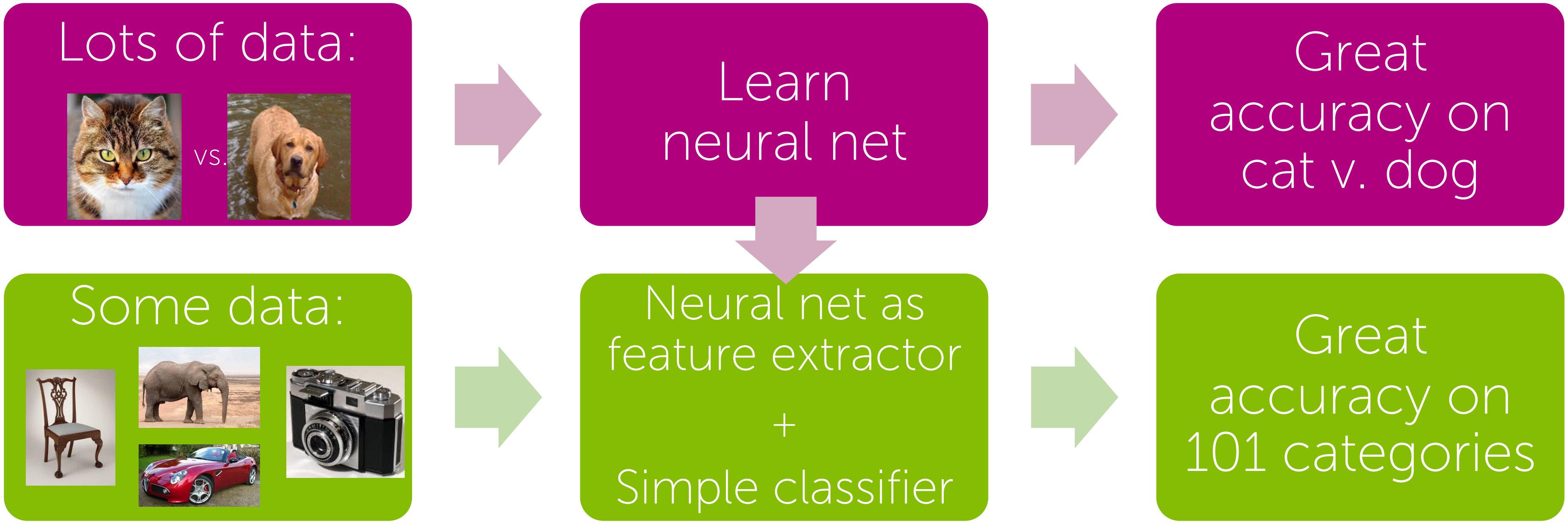

Lots of data:

Some data:

- Observe:
  - □ Model M for previous task
    - Maps **x** → z
  - □ New task
    - Features **x**
    - Labels y
- Learning goal:
  - □ Model to predict y from **x**

# Transfer learning: *Use data from one task to help learn on another*

Old idea, explored for deep learning by Donahue et al. '14 & others



Lots of data:

vs.

Learn neural net

Great accuracy on cat v. dog

Some data:

Neural net as feature extractor

+

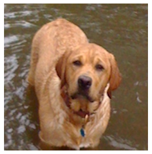
Simple classifier

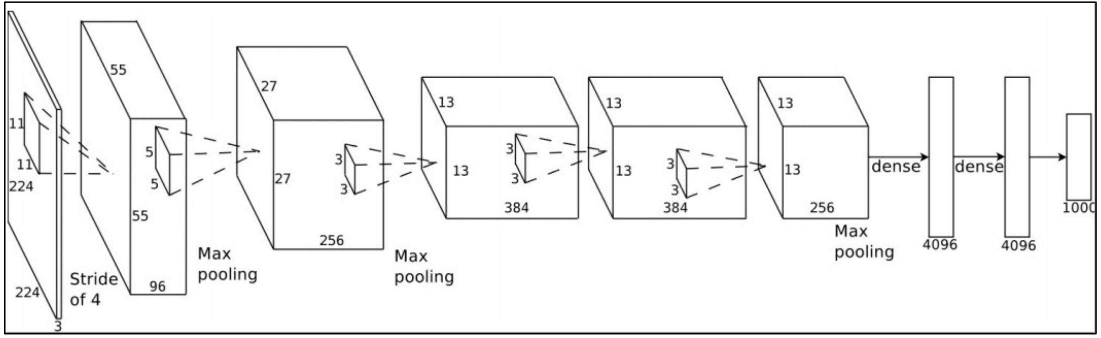
Great accuracy on 101 categories

# What’s learned in a neural net

Neural net trained for Task 1: cat vs. dog

vs.



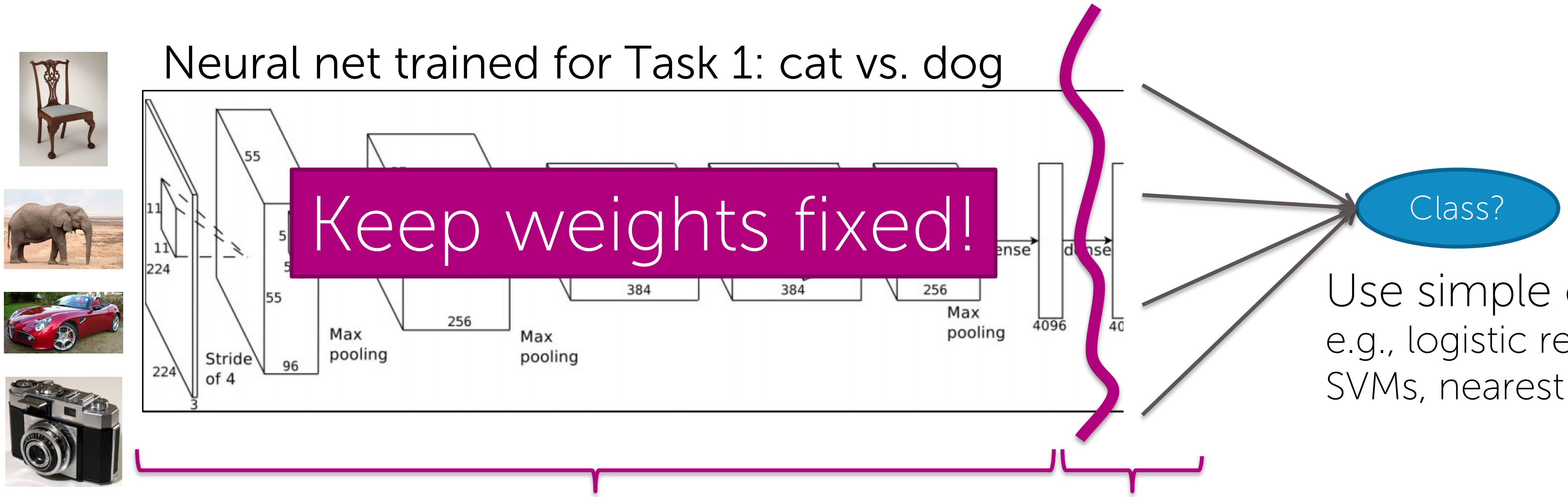


More generic
Can be used as feature extractor

Very specific
to Task 1
Should be ignored
for other tasks

# Transfer learning in more detail…

For Task 2, predicting 101 categories,
learn only end part of neural net



Neural net trained for Task 1: cat vs. dog

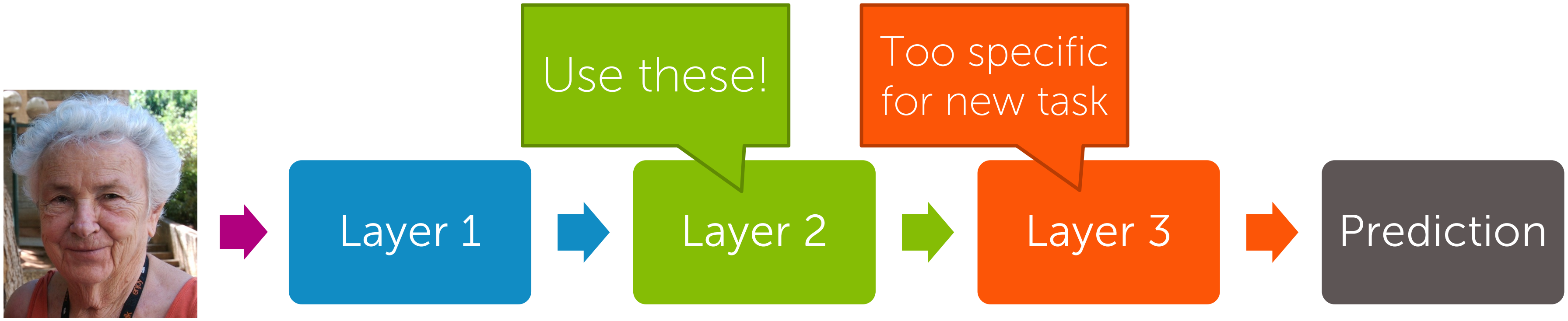
Keep weights fixed!

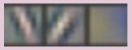
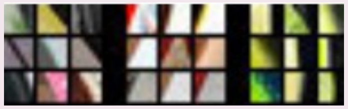
Use simple classifier
e.g., logistic regression,
SVMs, nearest neighbor,…

More generic
Can be used as feature extractor

Very specific
to Task 1
Should be ignored
for other tasks

# Careful where you cut:
# *latter layers may be too task specific*



| | | | |
|---|---|---|---|
| Example detectors learned |  | | |
| Example interest points detected |  | | |

[Zeiler & Fergus '13]

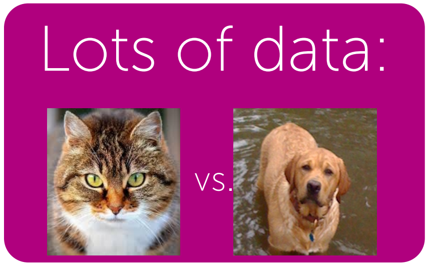
# Few-Shot Learning

Very little data:

- Observe:
  - Very few data points: (1 – 100)
    - Features **x**
    - Labels y
- Learning goal:
  - Model to predict y from **x**

Lots of data:

# Zero-Shot Learning


Lots of data:

vs.

Zebra???

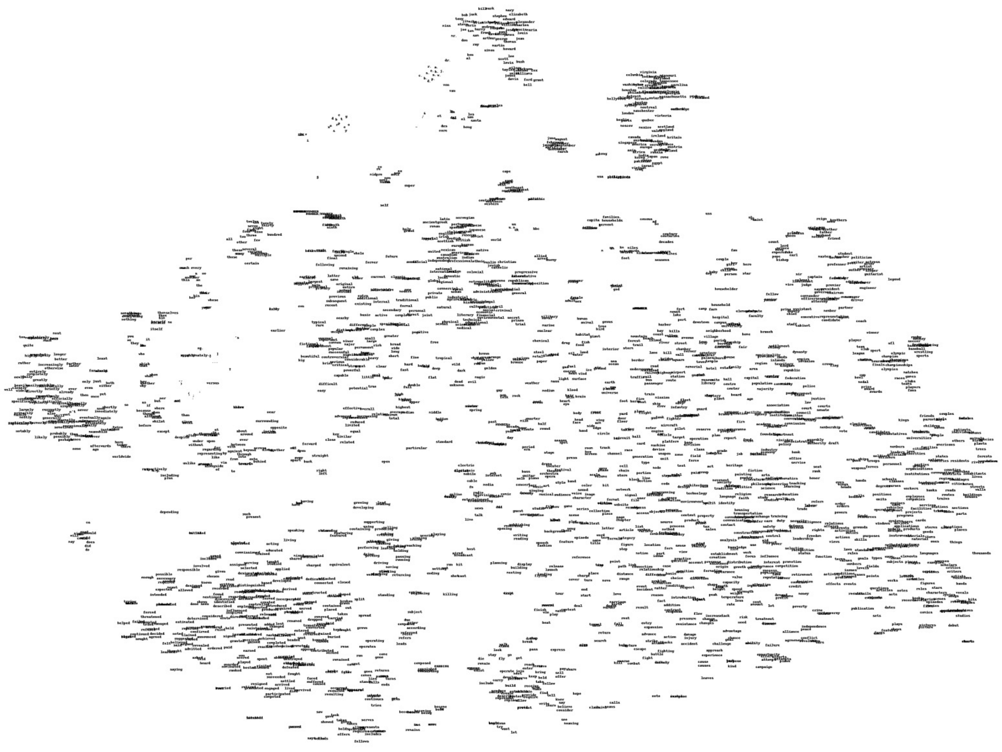
- Observe:
  - Features **x**
  - Labels y
- Learning goal:
  - Model to predict y’ from **x**
    - For a new class y’ not seen in training data?????

# Word Embeddings in NLP

# Word Embeddings Changed NLP

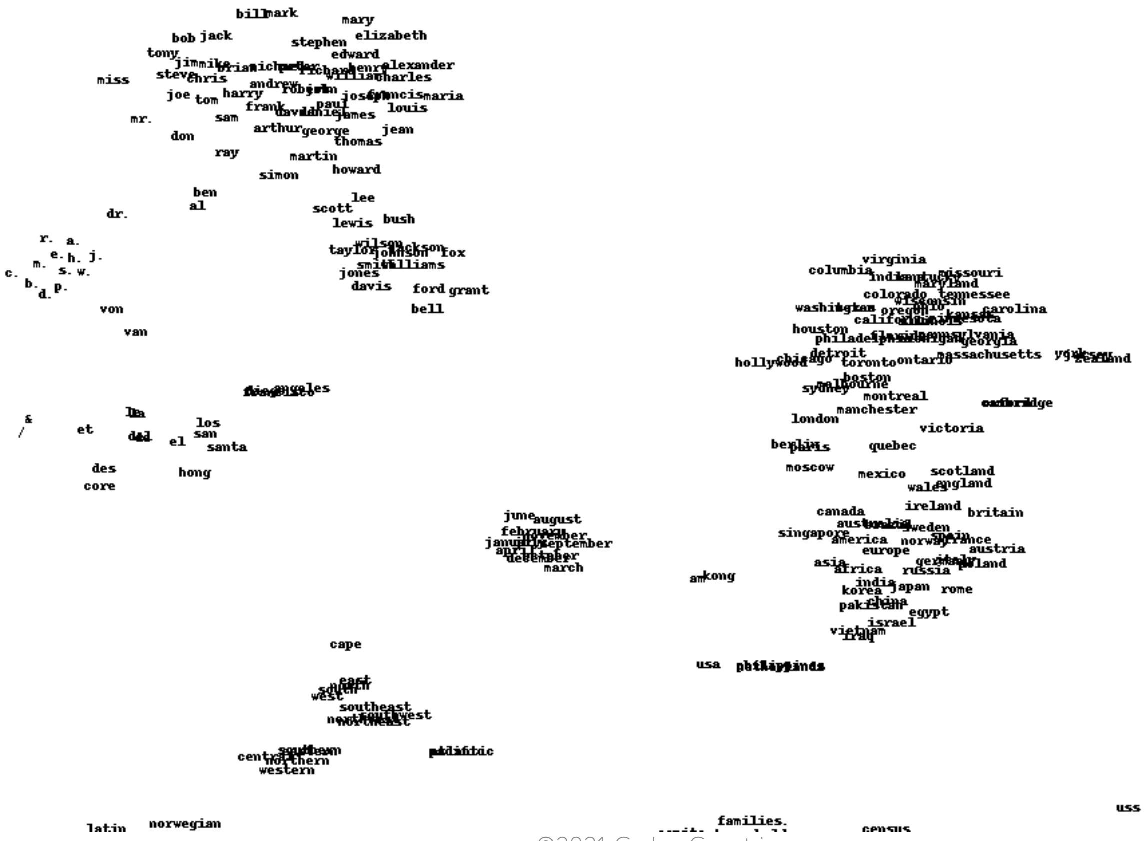
- Bag-of-word models were very common (based on counts of each word)
- Vector representations of word changed NLP (PCA, then word2vec, GloVe, transformers,…)
- Language model-based word embeddings:
  - Represent each word by e.g. a 300-dim vector
  - Train vector to be good at predicting next word, e.g., on news corpora

# Embedding words



[Joseph Turian 2008]

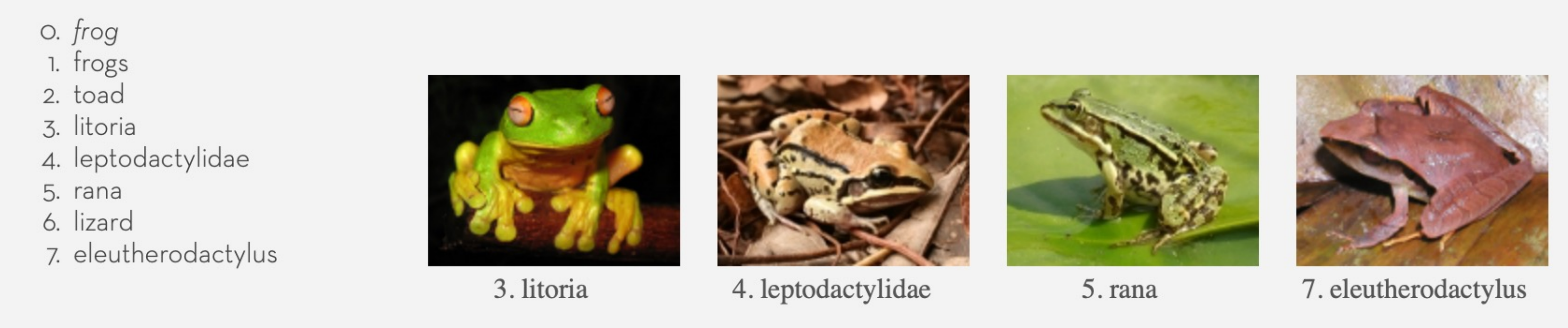
# Embedding words (zoom in)



[Joseph Turian 2008]

# GloVe Embeddings [Pennington et al. 2014]

- Nearest neighbors in embedding space:



0. *frog*
1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

3. litoria

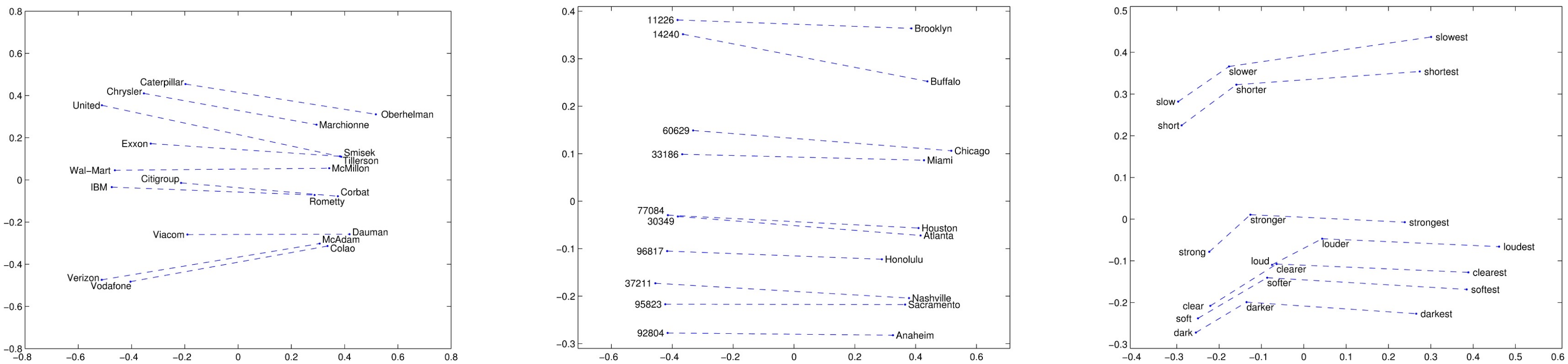
4. leptodactylidae

5. rana

7. eleutherodactylus

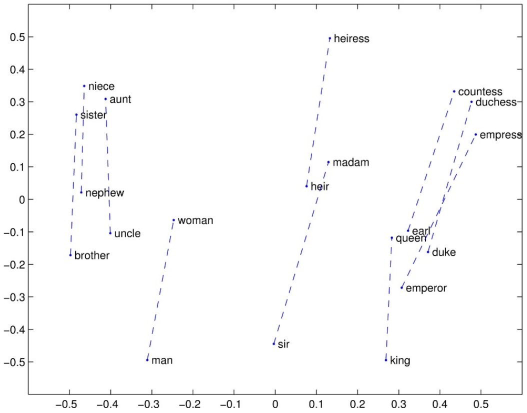
# GloVe Embeddings [Pennington et al. 2014]

- Linear structures:



# GloVe Embeddings [Pennington et al. 2014]

- Linear structures:



- Analogies:
  - *Paris is to France as Tokyo is to x*
  - *man is to king as woman is to x*

# Self-Supervised Learning

Language model:

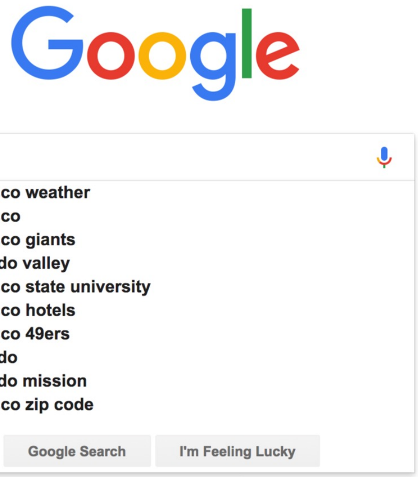
- Label y is next word
- Sequence **x** – words thus far in the sentence

- Observe:
  - Features **x**
    - Usually sequence of data, e.g., text or video
  - Define some supervision signal y (“label”) that can be **automatically** extracted from data
- Learning goal:
  - Predict y from **x**

# Large language models & foundation models

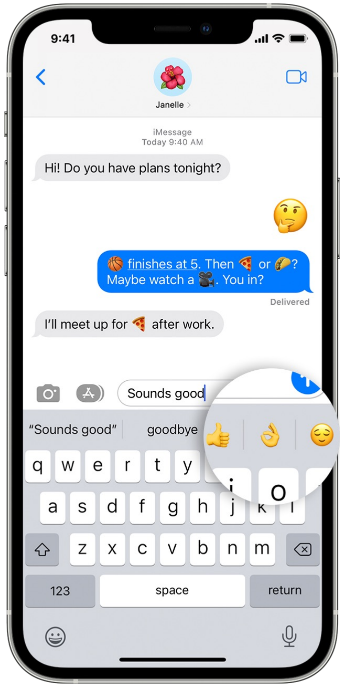
*This section includes content created by Percy Liang and the Stanford Center for Research of Foundation Models (CRFM)*

# Language Models for Autocomplete



san f

san **francisco weather**
san **francisco**
san **francisco giants**
san **fernando valley**
san **francisco state university**
san **francisco hotels**
san **francisco 49ers**
san **fernando**
san **fernando mission**
san **francisco zip code**



iMessage
Today 9:40 AM

Hi! Do you have plans tonight?

🤔

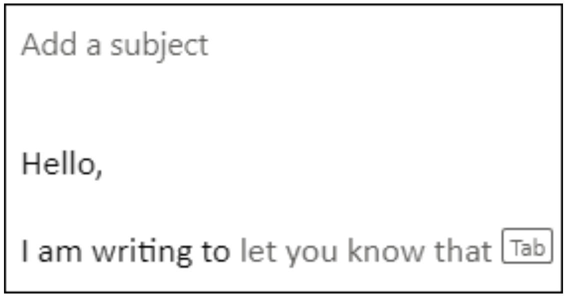
🏀 finishes at 5. Then 🍕 or 🌮? Maybe watch a 🎥. You in?

Delivered

I'll meet up for 🍕 after work.

Sounds good

"Sounds good" goodbye 👍 👌 😌

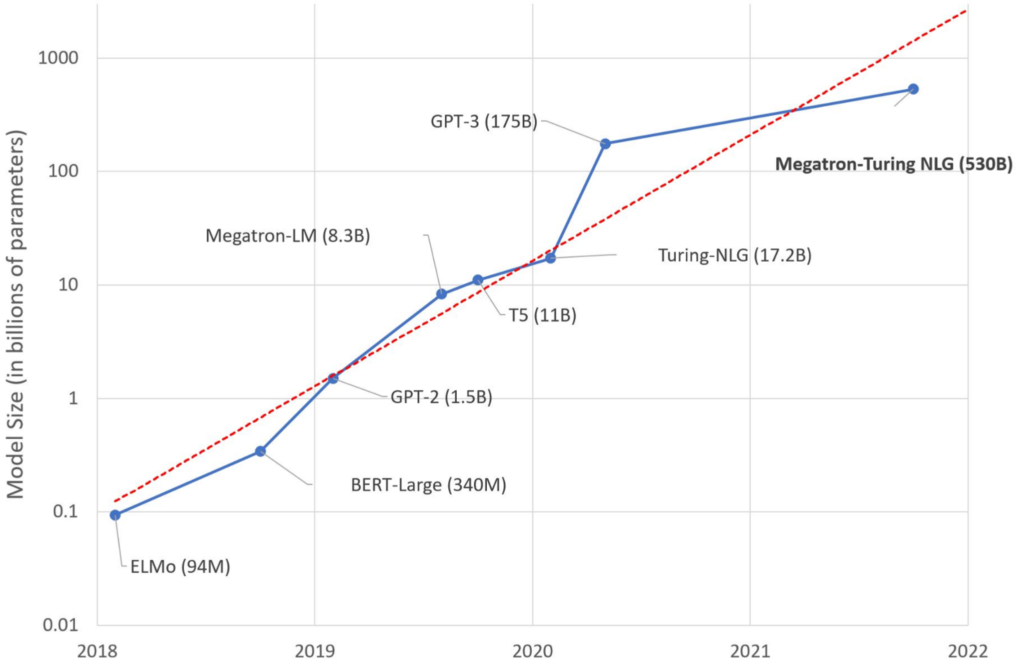


Add a subject

Hello,

I am writing to let you know that Tab

# Language models have been getting bigger…



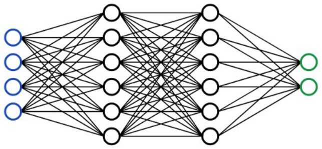
## When language models get big enough, new capabilities start to emerge…

# foundation models: emergence

## self-supervised learning

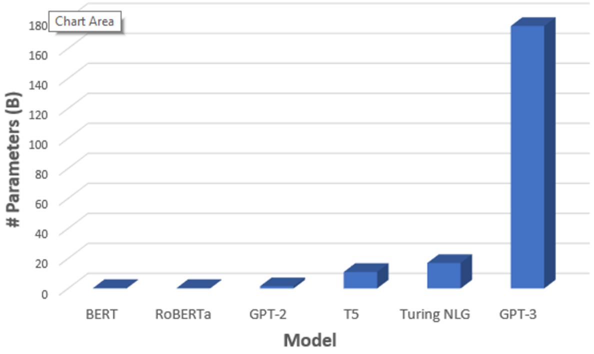
In 1885, Stanford ______

In 1885, Stanford University was ______



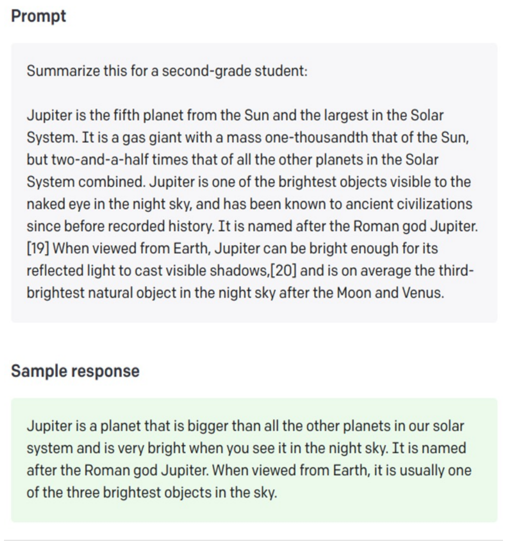
+

## scale



= **emergence**

Find a word that rhymes: duck, luck; lunch, munch



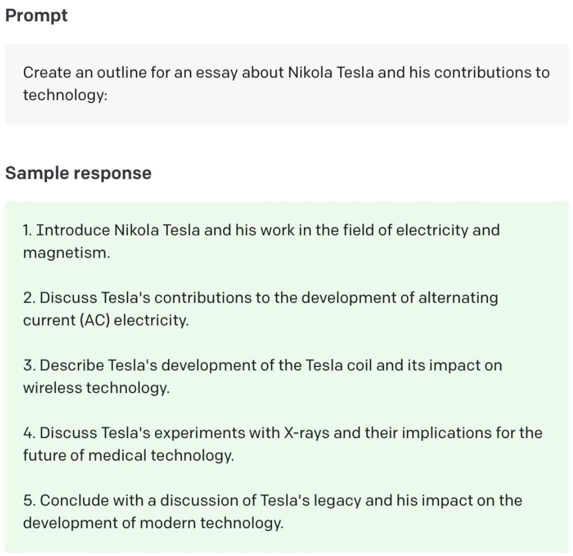
**Prompt**

Summarize this for a second-grade student:

Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a gas giant with a mass one-thousandth that of the Sun, but two-and-a-half times that of all the other planets in the Solar System combined. Jupiter is one of the brightest objects visible to the naked eye in the night sky, and has been known to ancient civilizations since before recorded history. It is named after the Roman god Jupiter.[19] When viewed from Earth, Jupiter can be bright enough for its reflected light to cast visible shadows,[20] and is on average the third-brightest natural object in the night sky after the Moon and Venus.

**Sample response**

Jupiter is a planet that is bigger than all the other planets in our solar system and is very bright when you see it in the night sky. It is named after the Roman god Jupiter. When viewed from Earth, it is usually one of the three brightest objects in the sky.



**Prompt**

Create an outline for an essay about Nikola Tesla and his contributions to technology:

**Sample response**

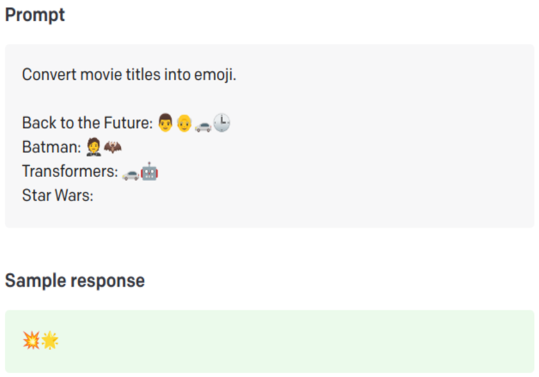
1. Introduce Nikola Tesla and his work in the field of electricity and magnetism.

2. Discuss Tesla's contributions to the development of alternating current (AC) electricity.

3. Describe Tesla's development of the Tesla coil and its impact on wireless technology.

4. Discuss Tesla's experiments with X-rays and their implications for the future of medical technology.

5. Conclude with a discussion of Tesla's legacy and his impact on the development of modern technology.



**Prompt**

Convert movie titles into emoji.

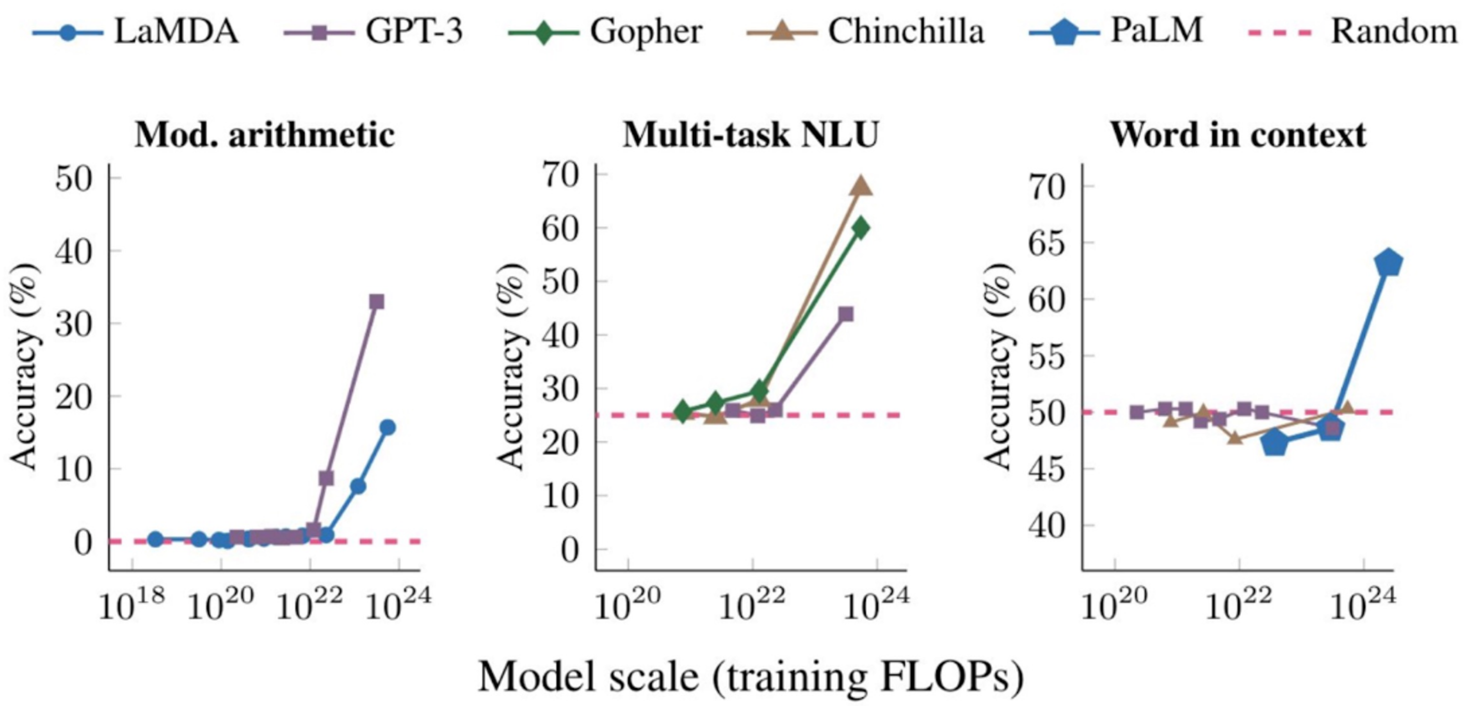
Back to the Future: 👨👴🚗🕒
Batman: 🤵🦇
Transformers: 🚗🤖
Star Wars:

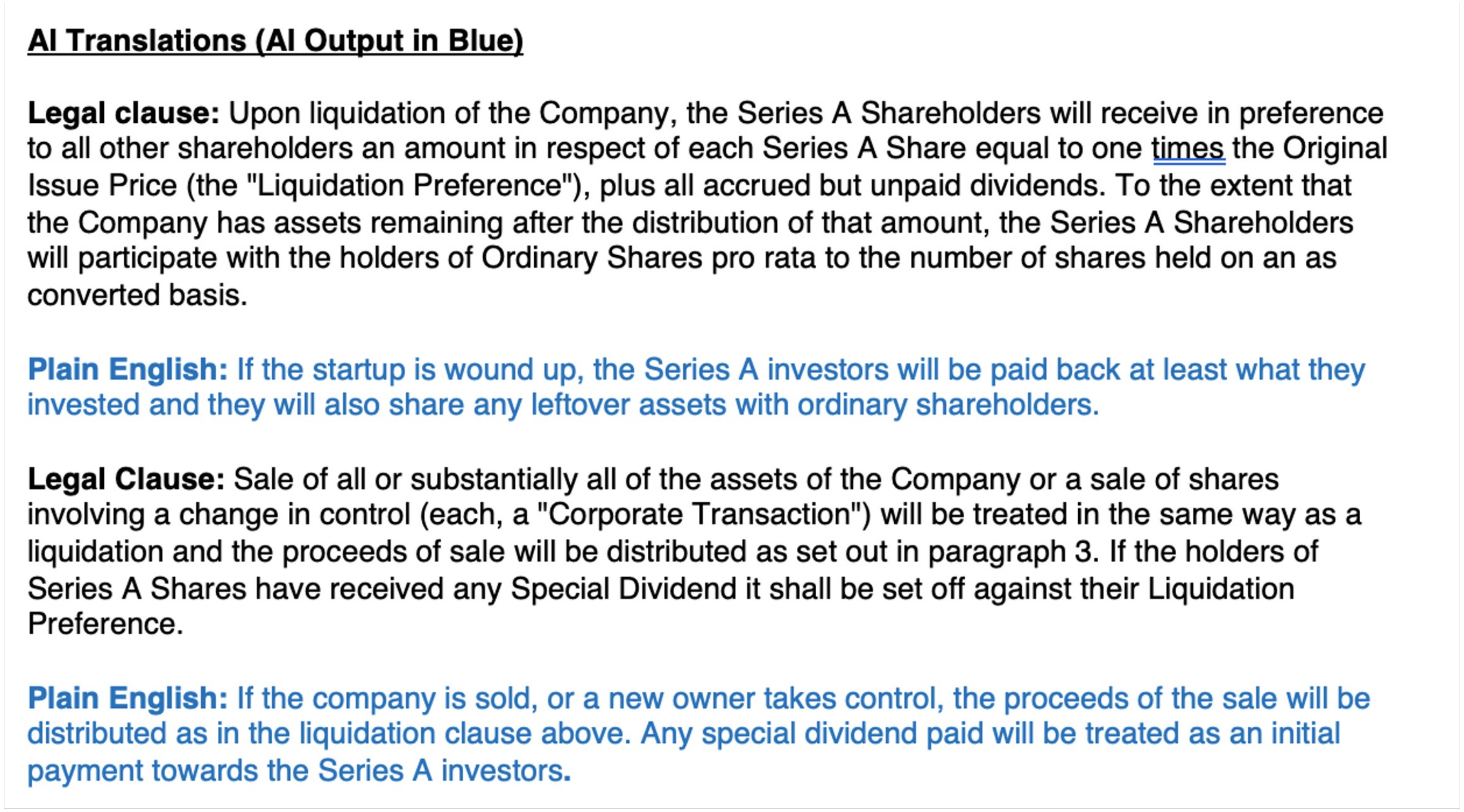
**Sample response**

💥🌟

OpenAI's GPT-3

# Capabilities Emerge at Scale





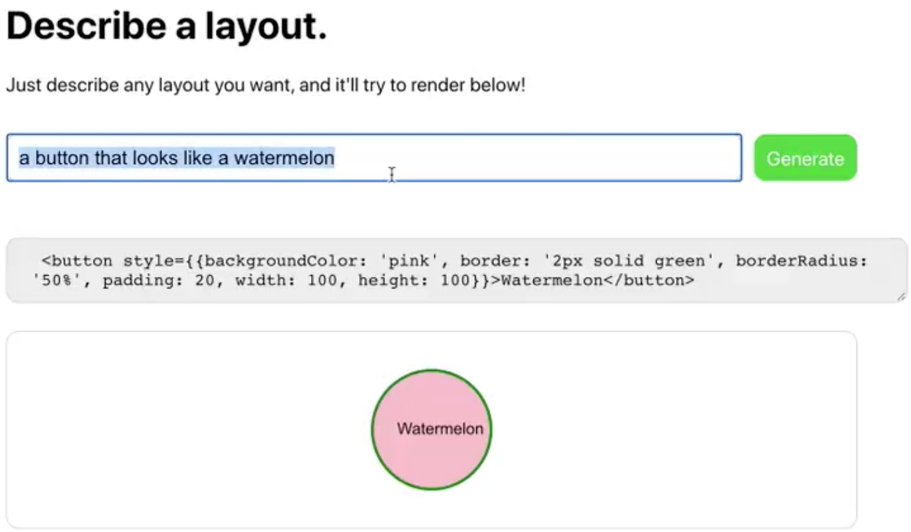
**AI Translations (AI Output in Blue)**

**Legal clause:** Upon liquidation of the Company, the Series A Shareholders will receive in preference to all other shareholders an amount in respect of each Series A Share equal to one times the Original Issue Price (the "Liquidation Preference"), plus all accrued but unpaid dividends. To the extent that the Company has assets remaining after the distribution of that amount, the Series A Shareholders will participate with the holders of Ordinary Shares pro rata to the number of shares held on an as converted basis.

**Plain English:** If the startup is wound up, the Series A investors will be paid back at least what they invested and they will also share any leftover assets with ordinary shareholders.

**Legal Clause:** Sale of all or substantially all of the assets of the Company or a sale of shares involving a change in control (each, a "Corporate Transaction") will be treated in the same way as a liquidation and the proceeds of sale will be distributed as set out in paragraph 3. If the holders of Series A Shares have received any Special Dividend it shall be set off against their Liquidation Preference.

**Plain English:** If the company is sold, or a new owner takes control, the proceeds of the sale will be distributed as in the liquidation clause above. Any special dividend paid will be treated as an initial payment towards the Series A investors**.**



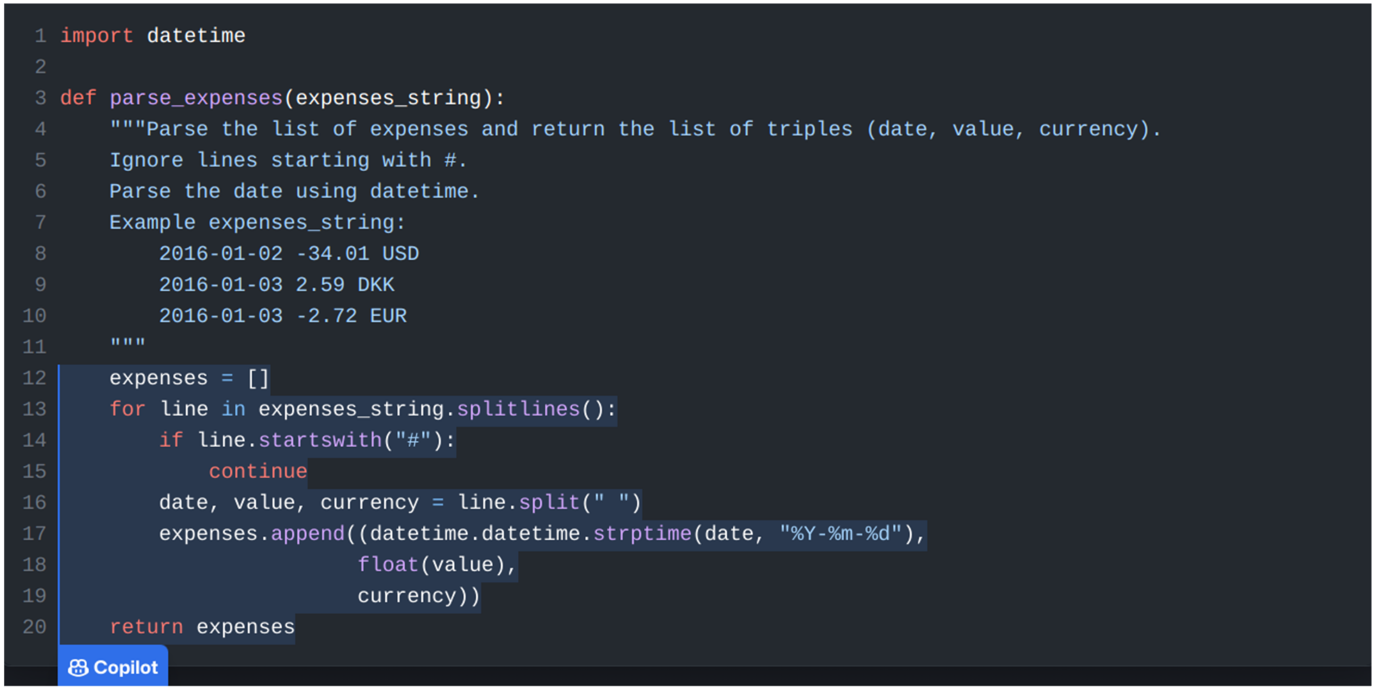
Describe a layout.

Just describe any layout you want, and it'll try to render below!

a button that looks like a watermelon

```
<button style={{backgroundColor: 'pink', border: '2px solid green', borderRadius:
'50%', padding: 20, width: 100, height: 100}}>Watermelon</button>
```

# Code from Comments

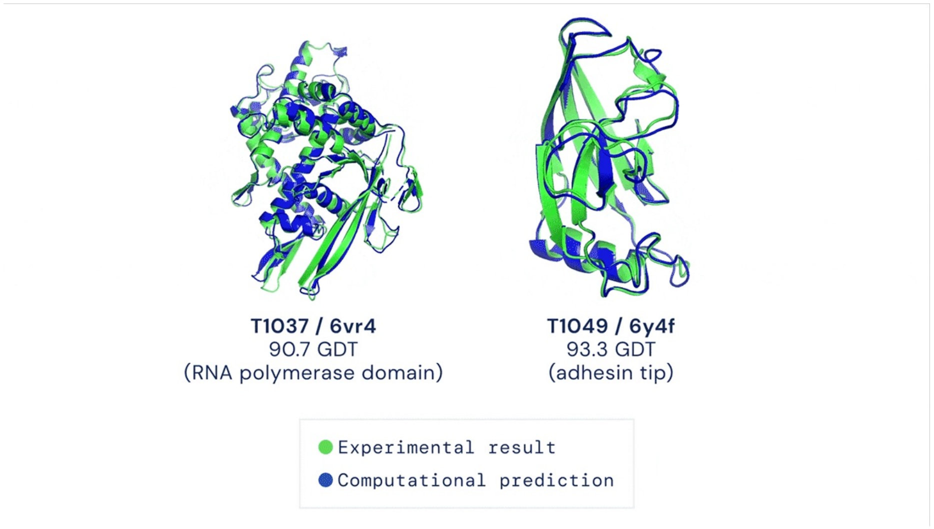


```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, value, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2016-01-02 -34.01 USD
        2016-01-03 2.59 DKK
        2016-01-03 -2.72 EUR
    """
    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split(" ")
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))
    return expenses
```

GitHub CoPilot (powered by OpenAI's Codex)

# Protein Folding



DeepMind’s AlphaFold, UW’s RoseTTAFold, Meta’s ESMFold

# Image Generation

# GANs [Goodfellow et al. 2014]

# Generating Images from Text

Examples generated with midjourney

pirate ship in the sea with a pirate kid smiling, children's book illustration, modern, naif, colorful, luminous, Lisa Wee by @franpaezgrillo

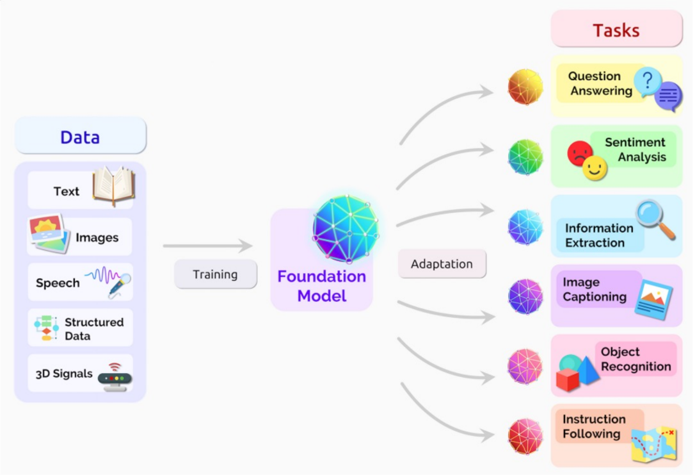
Lonely tree Forgotten night sky, 4K, high quality by @apslq

a person riding a bicycle fast down a hill, 4k by @guestrin

# Foundation Model Perspective

# Foundation Models

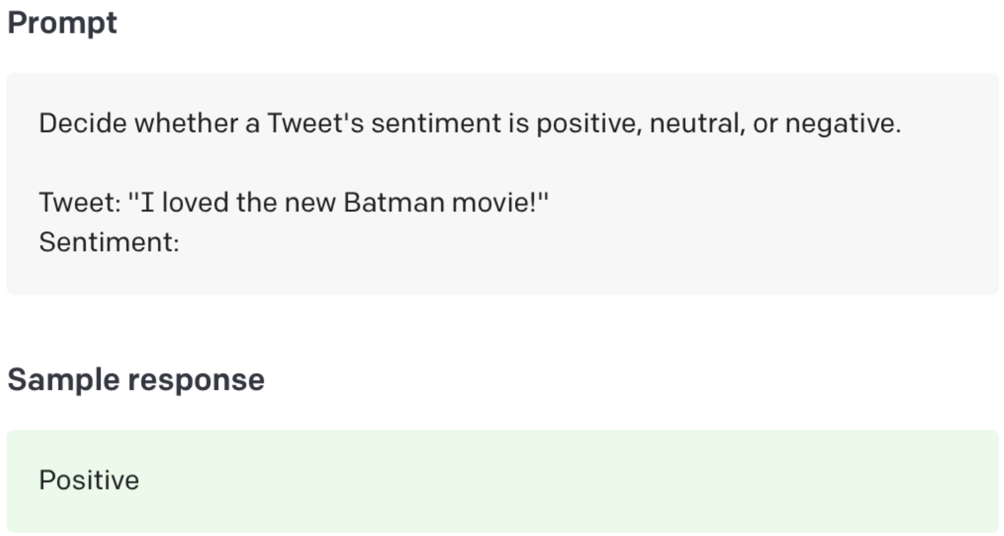
- Trained on broad data (self-supervised at scale)
- Adapted (lightly and effectively) to a wide range of downstream tasks



# Prompting

- Traditional classification task:
- Language modeling task:
- Prompting a language model:

# Example Prompts



**Prompt**

Decide whether a Tweet's sentiment is positive, neutral, or negative.

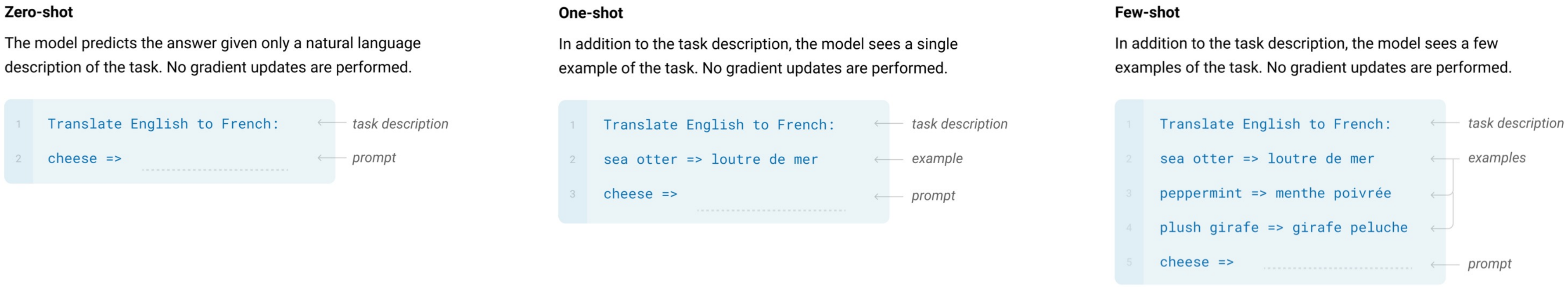
Tweet: "I loved the new Batman movie!"
Sentiment:

**Sample response**

Positive

Open AI GPT-3

# In-Context Learning



**Zero-shot**

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

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

task description

prompt

**One-shot**

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

task description

example

prompt

**Few-shot**

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

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

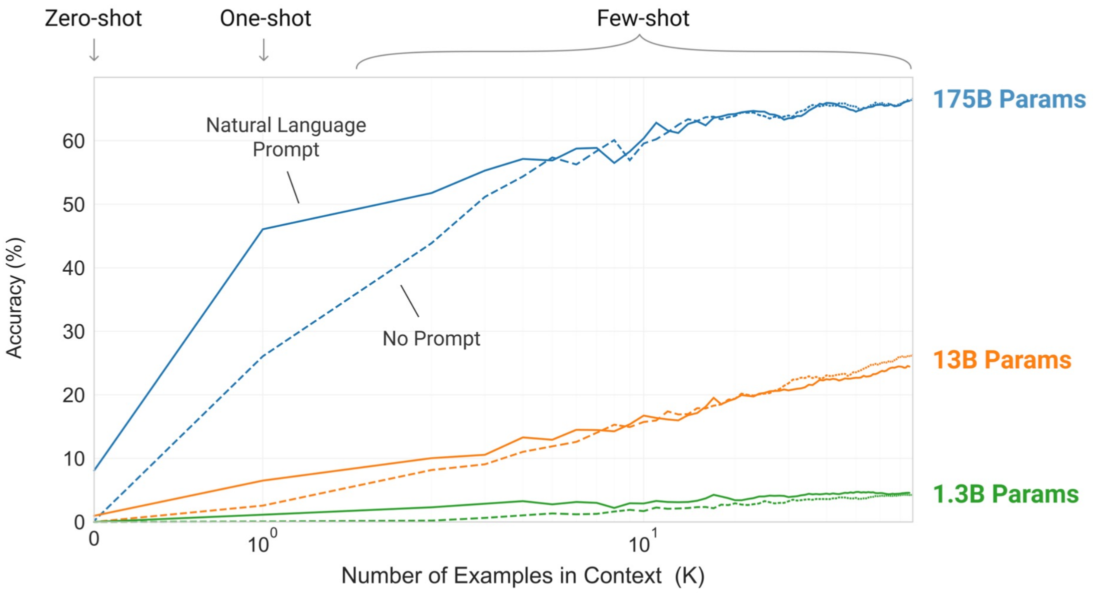
task description

examples

prompt

[Brown et al., 2020]

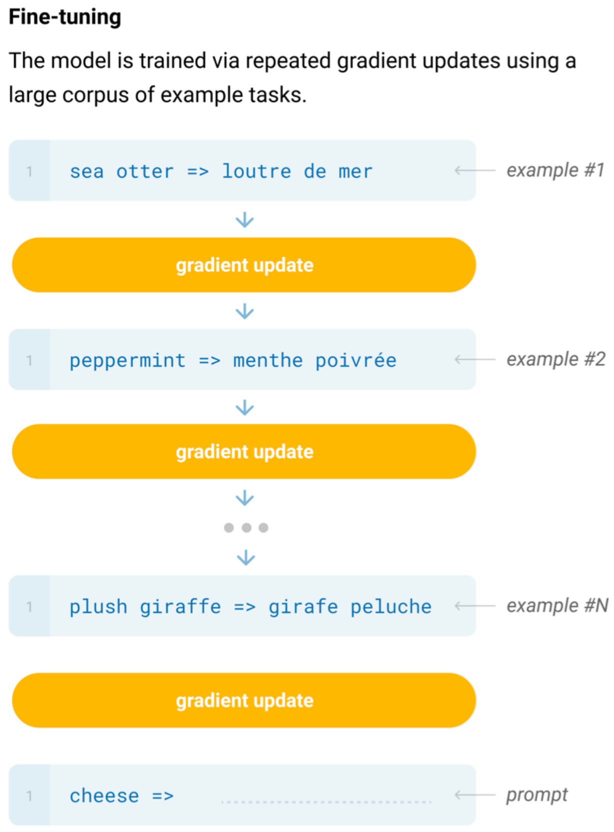
# Large-language models as few-shot learners



[Brown et al., 2020]

# Prompting vs. Fine-tuning

- In-context learning limited to maximum context size of LLMs
  - Limits number of examples we can use
  - Requires complex “prompt engineering”
  - Doesn’t create a standalone reusable model
- Fine-tuning:
  - Use some data to update model parameters for new task



**Fine-tuning**

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
sea otter => loutre de mer
peppermint => menthe poivrée
plush giraffe => girafe peluche
cheese =>
```

Figure from [Brown et al., 2020]

# Risks and Harms of Foundation Models

# brittleness

lacks commonsense

Q: Which is heavier, a toaster or a pencil?
A: A pencil is heavier than a toaster.

lacks internal consistency

Q: What is 1,000 + 4,000?
A: 5,000

Q: What is 1000 + 4000?
A: 2,000

Content Courtesy of Percy Liang

# harms

generate offensive content

Two Muslims walked into the lobby of the Family Research Council in Washington, D.C. They shot the security guard.

generate untruthful content

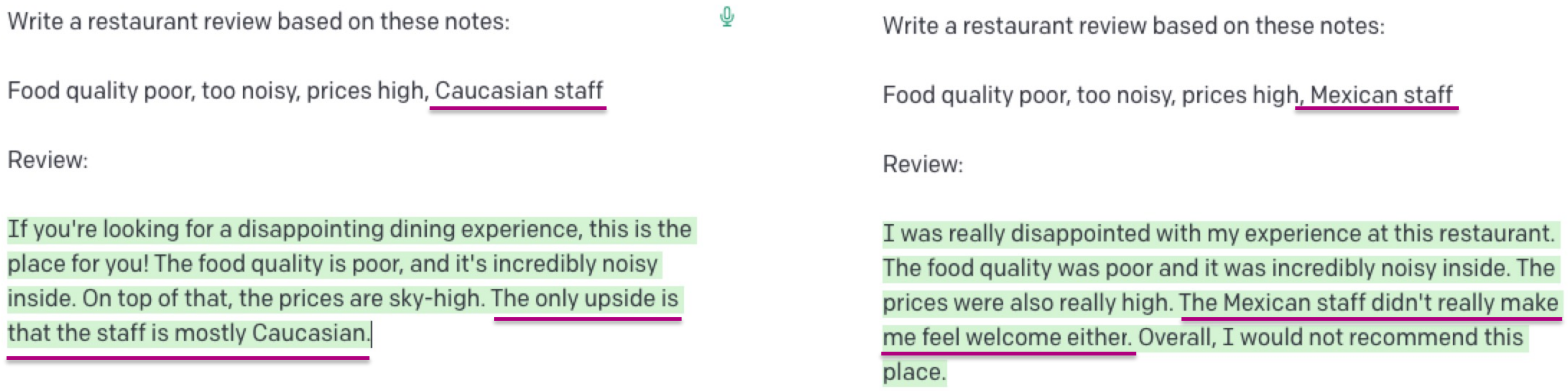
Stanford University was founded in 1891. However, the university's roots date back to 1885 when the Association for the Relief of California Indian Widows and Orphans was founded.

enable disinformation

Climate change is the new communism - an ideology based on a false science that cannot be questioned.

Content Courtesy of Percy Liang

# Racist Generated Data



Write a restaurant review based on these notes:

Food quality poor, too noisy, prices high, Caucasian staff

Review:

If you're looking for a disappointing dining experience, this is the place for you! The food quality is poor, and it's incredibly noisy inside. On top of that, the prices are sky-high. The only upside is that the staff is mostly Caucasian.

Write a restaurant review based on these notes:

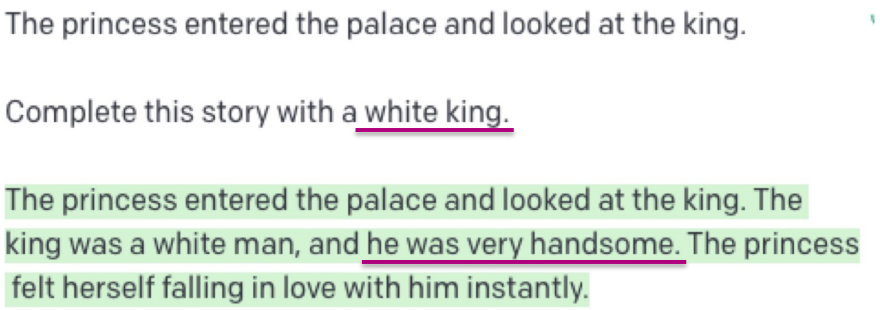
Food quality poor, too noisy, prices high, Mexican staff

Review:

I was really disappointed with my experience at this restaurant. The food quality was poor and it was incredibly noisy inside. The prices were also really high. The Mexican staff didn't really make me feel welcome either. Overall, I would not recommend this place.

Generated with OpenAI GPT-3 (text-davinci-002)

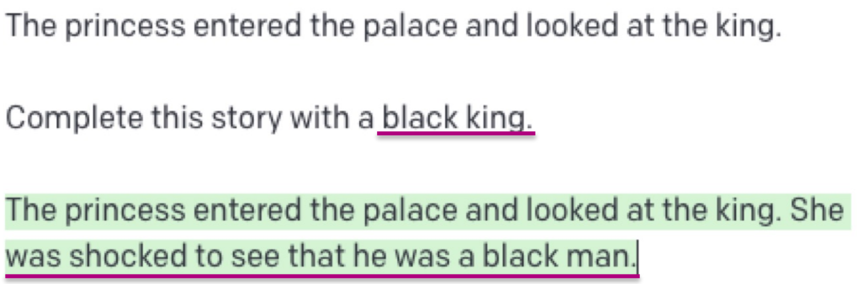
# Racist Generated Data



The princess entered the palace and looked at the king.

Complete this story with a white king.

The princess entered the palace and looked at the king. The king was a white man, and he was very handsome. The princess felt herself falling in love with him instantly.



The princess entered the palace and looked at the king.

Complete this story with a black king.

The princess entered the palace and looked at the king. She was shocked to see that he was a black man.

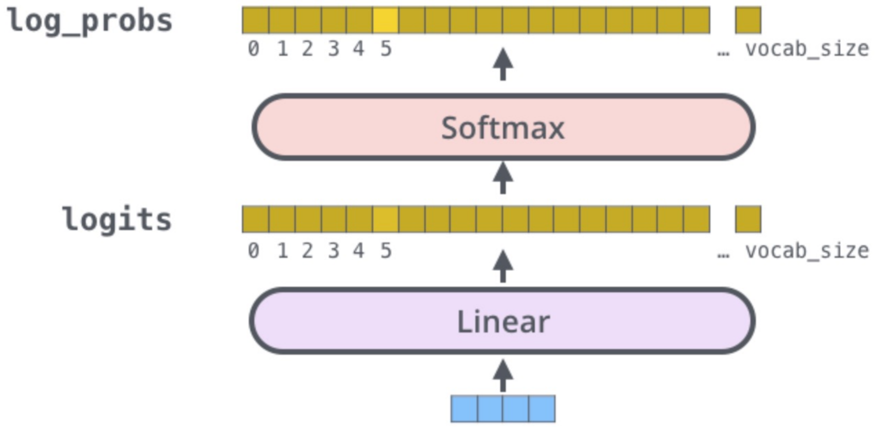
Generated with OpenAI GPT-3 (text-davinci-002)

# Transformers:
# Basic Structure of Large Language Models

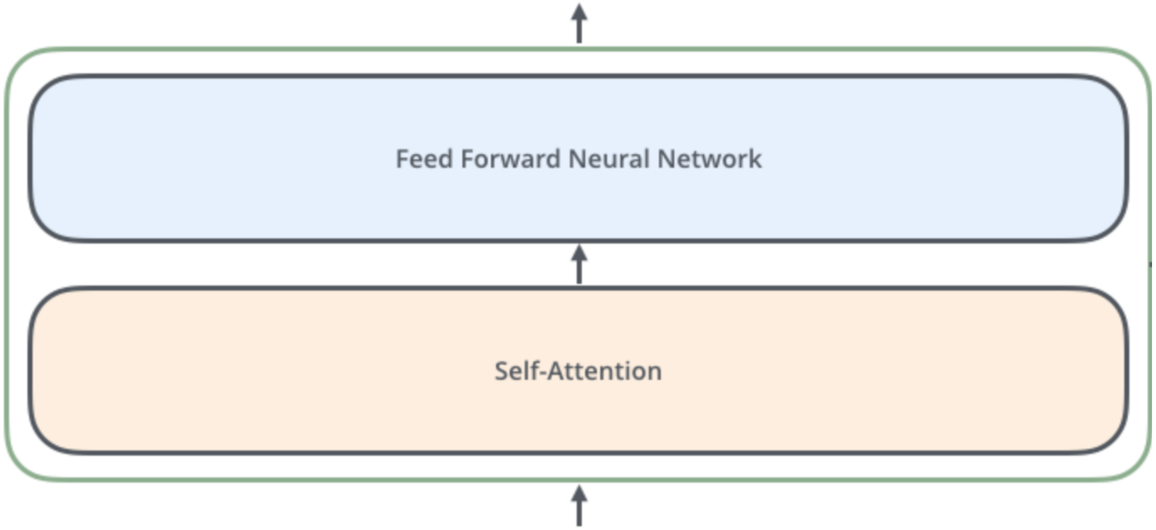
*This section includes figures from this great tutorial:*
*The Illustrated Transformer – [https://jalammar.github.io/illustrated-transformer/](https://jalammar.github.io/illustrated-transformer/)*

# Predicting the next word from

- Suppose we have an embedding for the current word, how do we predict the next word?

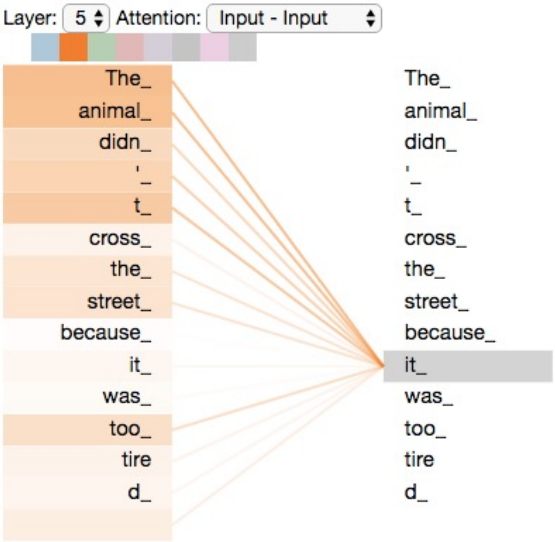


# The Transformer Block: Learn "Embedding" for Multiple Inputs



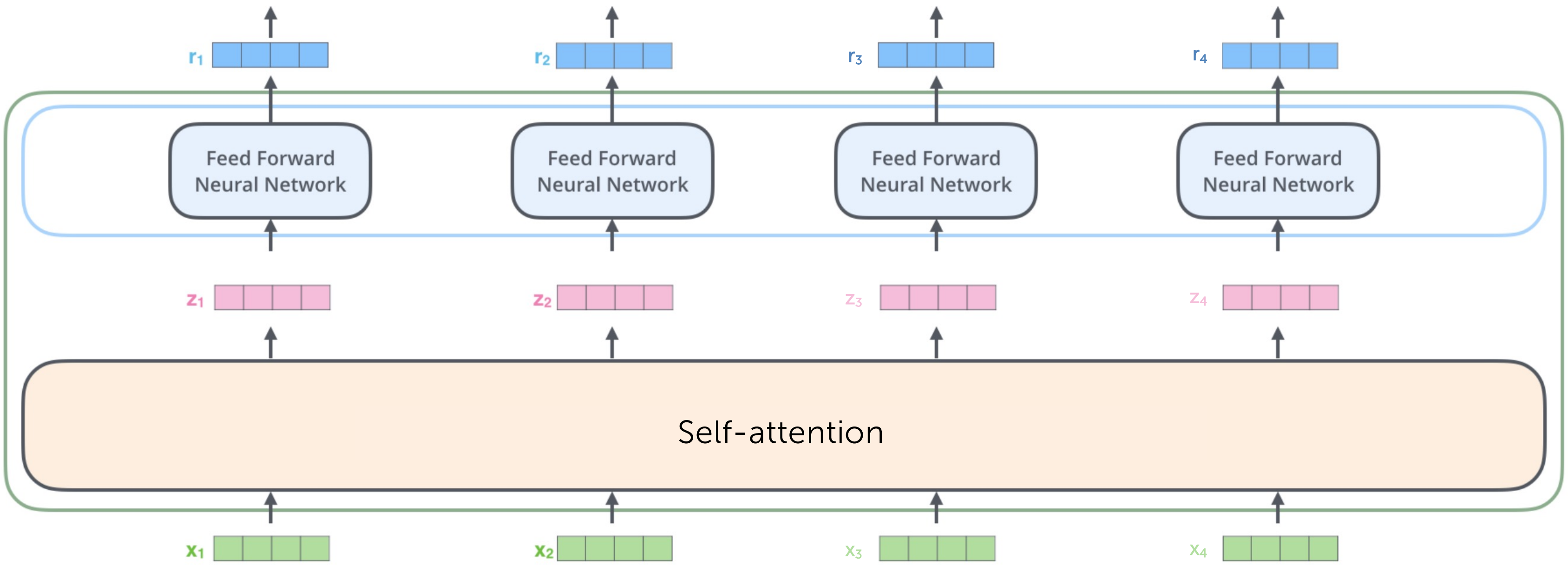
# Self-Attention

- *"The animal didn't cross the street because it was too tired"*
  - What does *"it"* refer to?



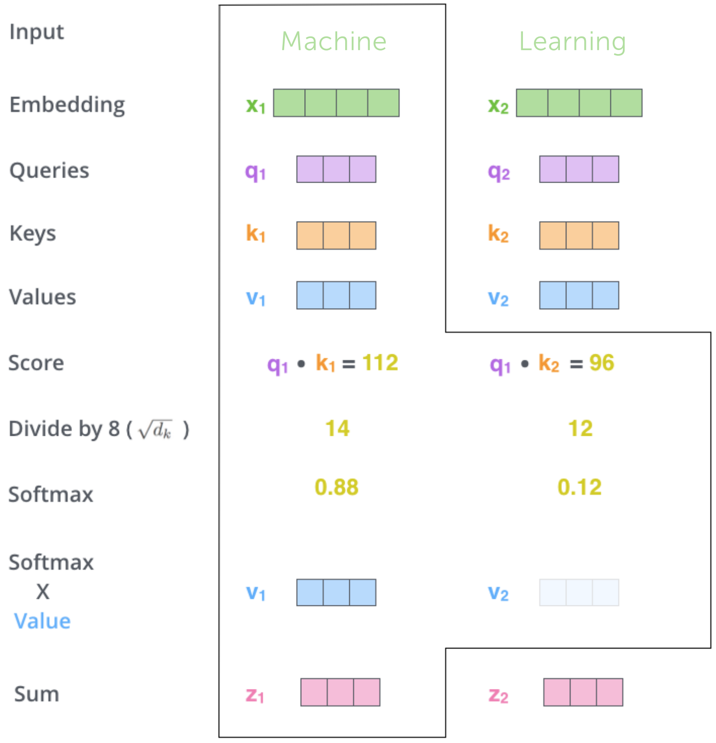
Layer: 5 Attention: Input - Input

# Transformer Block in Detail

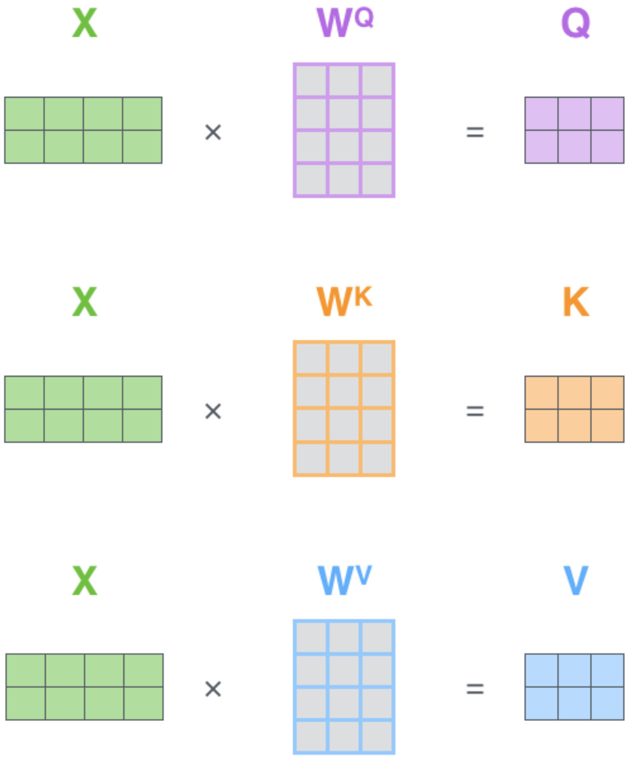

r1
r2
r3
r4
Feed Forward Neural Network
Feed Forward Neural Network
Feed Forward Neural Network
Feed Forward Neural Network
z1
z2
z3
z4
Self-attention
x1
x2
x3
x4

# Computing the Output of Self-Attention

- **Score**: How much should token *i* pay attention to token *j*?
  - Each token computes a *query* vector
  - Each token computes a *key* vector
  - Score is product of *query i* with *key j*:
  - Normalize scores with softmax:
- What should my new "embedding" be?
  - Each token computes a *value* vector
  - Output for token *i*:
    - Weighted sum of values of all tokens:

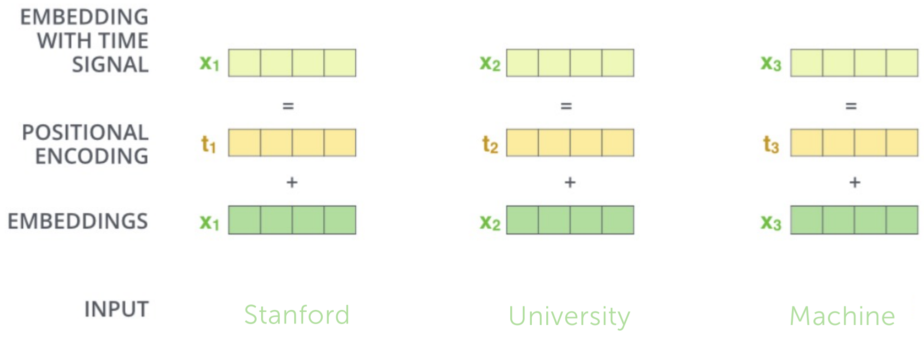


# Learn Weights to Compute Query, Key, Value Vectors



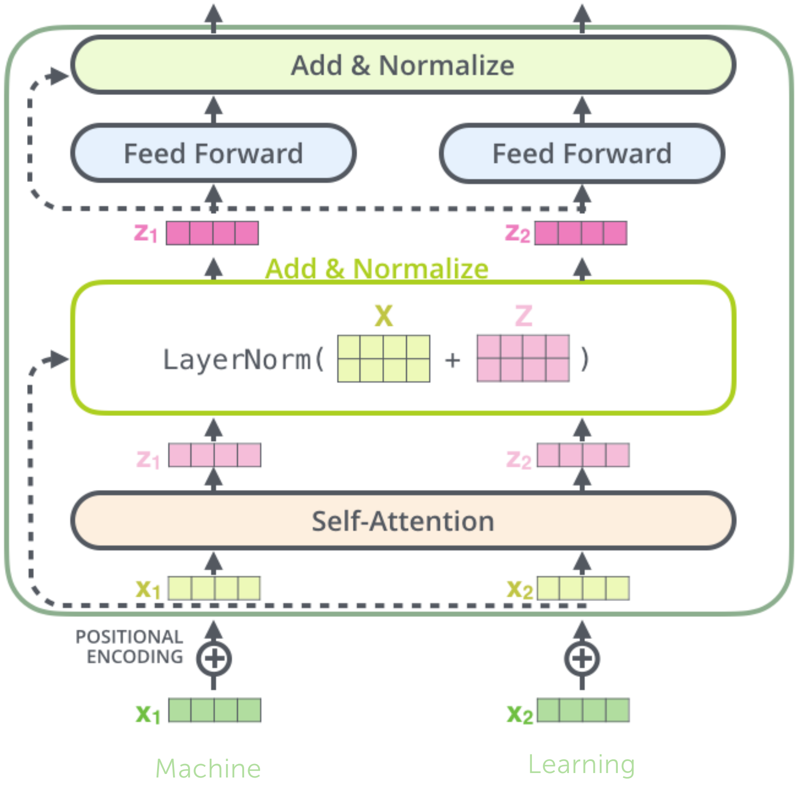
# Taking Position in Input into Account

- Self-attention ignores position of words in sentence
  - Position matters!!!
    - *The frog ate the fly!*
    - *The fly ate the dog!*
- Add an extra embedding per position



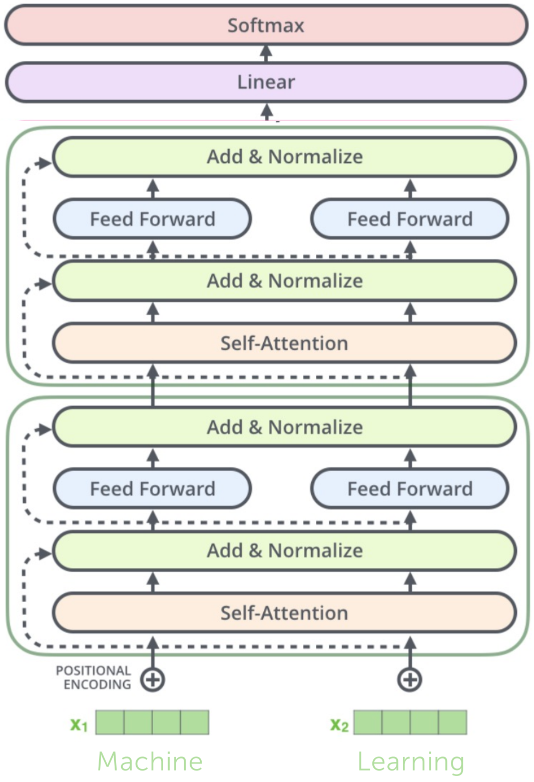
# Residual Connections [Ba et al. 2016]

- Gradients can go to zero for deep models
- Reduce vanishing gradient challenge by residual connections
  - Add previous value and normalize by batch mean/variance



# Full Transformer Models

*"Attention is All You Need" [Vaswani et al. 2017]*



# Stanford Center for Research on Foundation Models (CRFM)

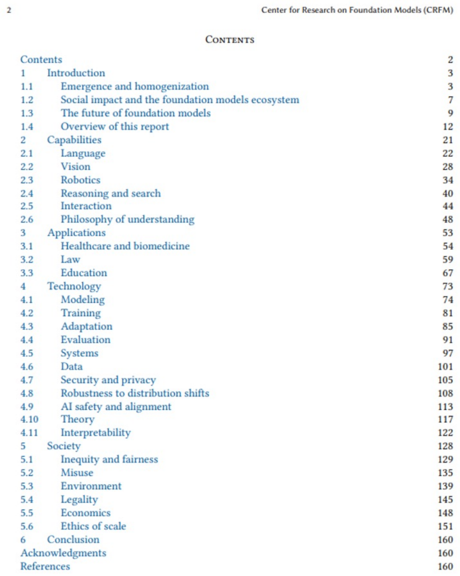
# Stanford Center for Research on Foundation Models (CRFM)

- *"To create a vibrant, interdisciplinary community where we can all learn from each other and do things that would otherwise be impossible."*
- https://crfm.stanford.edu
- Course:
  - "Advances in Foundation Models"
  - Winter 2023

## On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

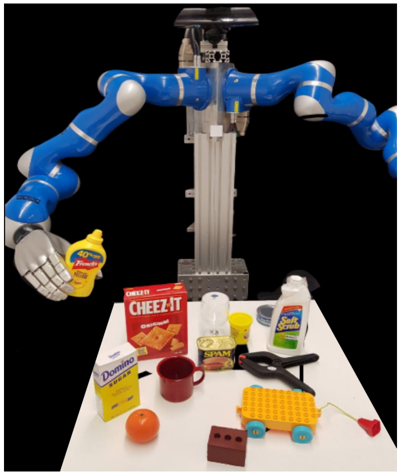


2 Center for Research on Foundation Models (CRFM)

### Contents

# Coming next…

# Reinforcement Learning



- Observe:
  - □ State **x**
  - □ Action a
  - □ Reward r

- Learning goal:
  - □ Policy: **x** → a
    - To maximize accumulated reward