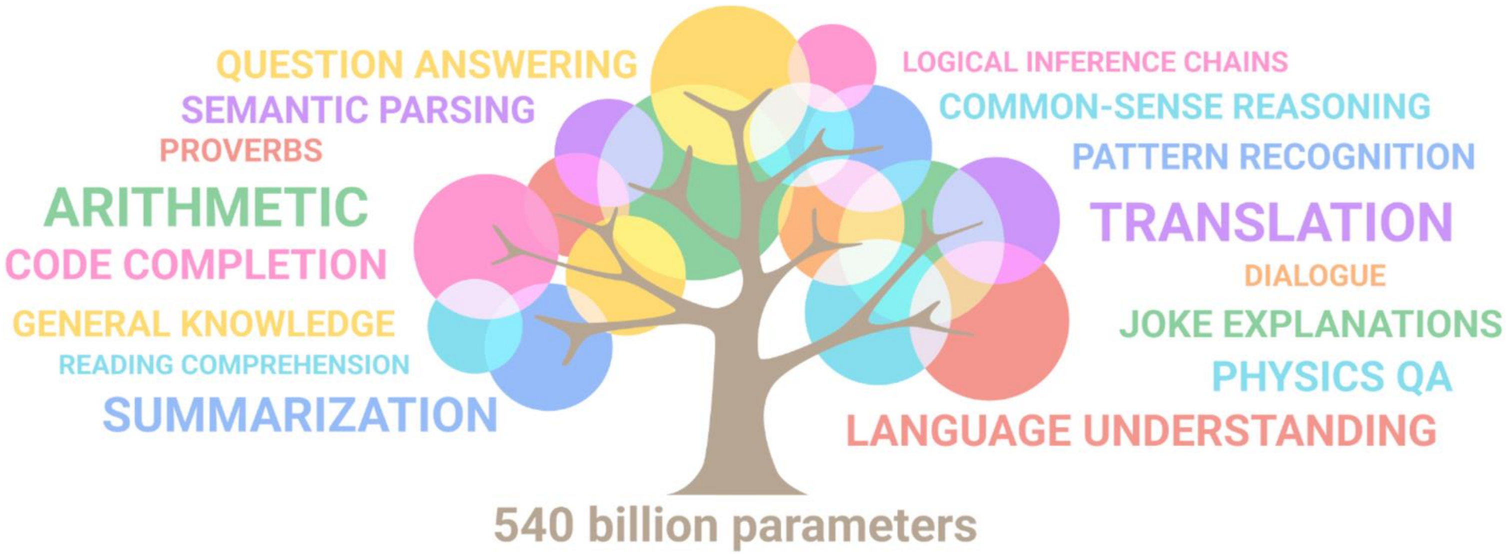
BIA660
Web Mining (with LLMs)

Outline

- **Introduction** — Name, major, interest, expectation from this course etc.
- **What is Web Mining?**
- **Syllabus**
- **Course Logistics**
  - Python Installation,
  - Poll Everywhere, Datacamp assignments
  - Class project

What is Web Mining?

In short, use large language models (LLMs) and machine learning techniques to analyze online text for insights



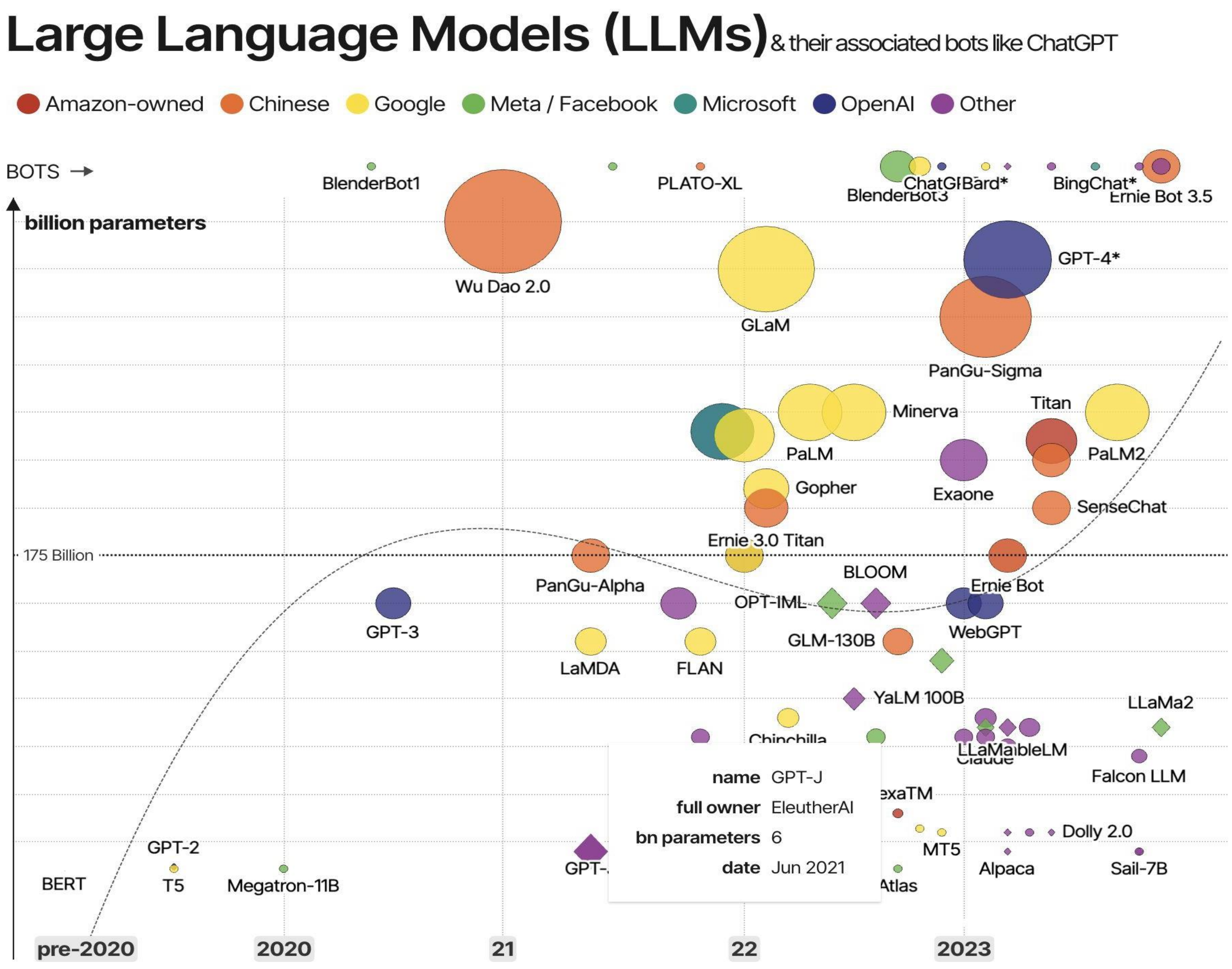
Let's first understand basics behind LLMs

References:

https://github.com/Hannibal046/Awesome-LLM

https://docs.google.com/presentation/d/1TTyePrw-p_xxUbi3rbmBI3QQpSsTI1btaQuAUvvNc8w/edit#slide=id.g206fa25c94c_0_24

Exponential Growth in LLMs



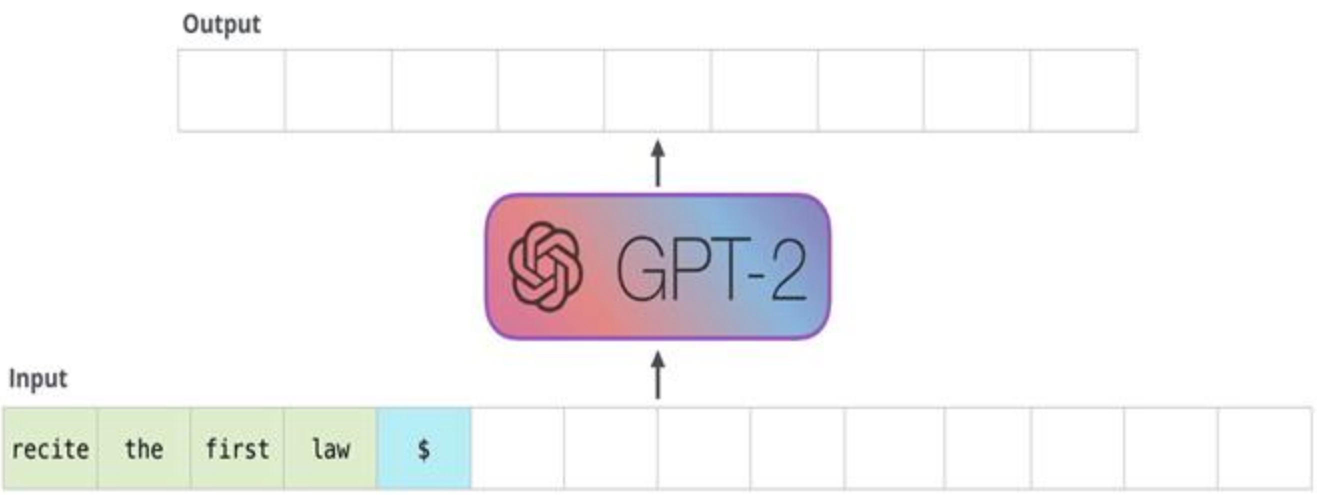
Scaling Laws

"These results show that ***language modeling performance improves smoothly and predictably as we appropriately scale up model size, data, and compute****. We expect that larger language models will perform better and be more sample efficient than current models."* (Kaplan, et al. 2020)

What is Language Model?

Autoregressive Generation

1) Given input words (or prompts), compute P(next word | input words)
2) Sample a token from ~ P(next word | previous words)
3) Append the word to the input and go back to (1)



p(**A| recite the first law $**)=**0.02**
p(**legal| recite the first law $**)=**0.001**,
…

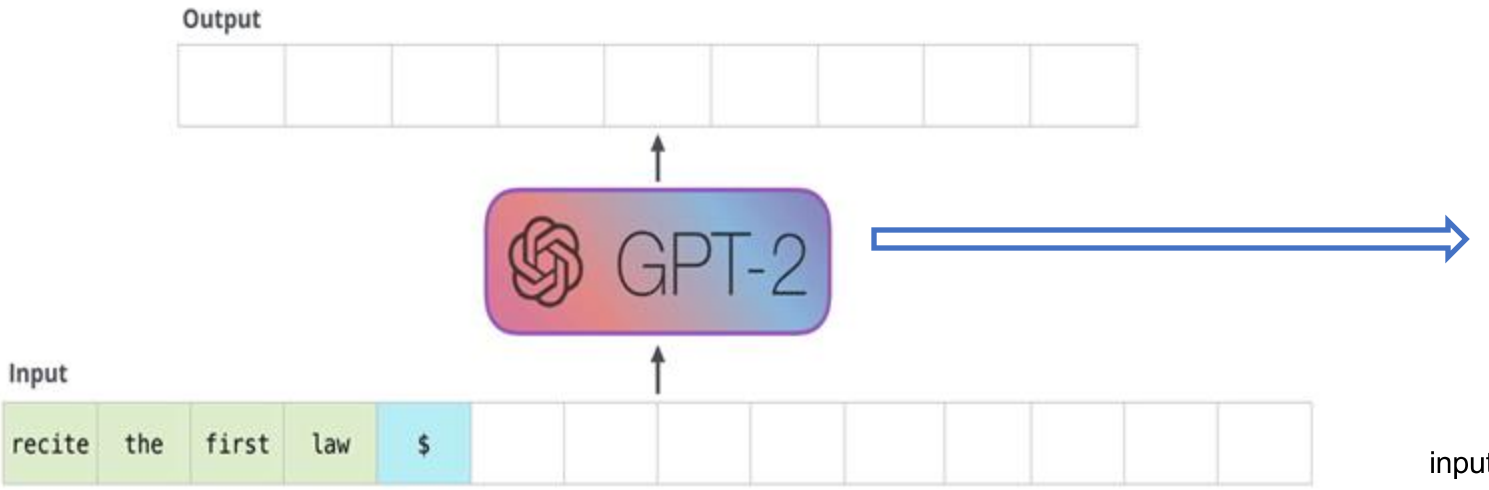
p(**robot| recite the first law $ A**)=**0.1**
p(**everyone| recite the first law $ A**)=**0.0001**
…

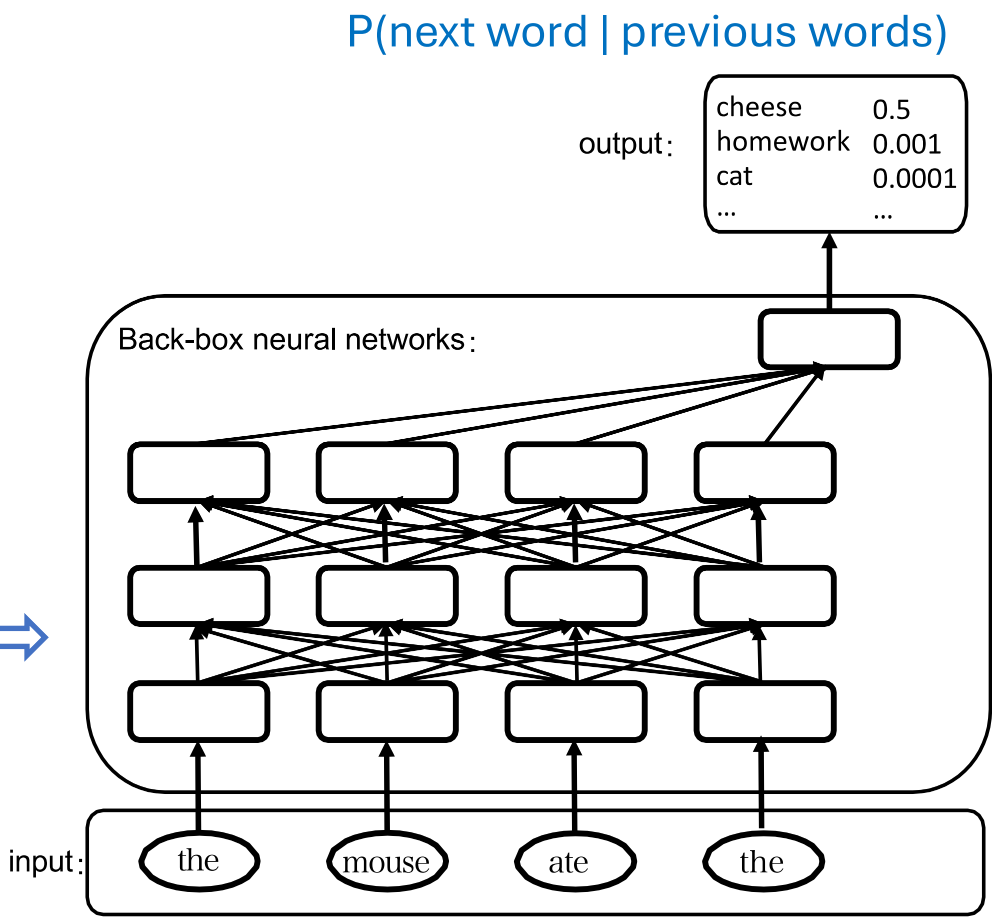
[from Alammar, The Illustrated GPT-2, https://jalammar.github.io/illustrated-gpt2/]

What is Language Model?

Autoregressive Generation

Computing conditional probability using a neural network



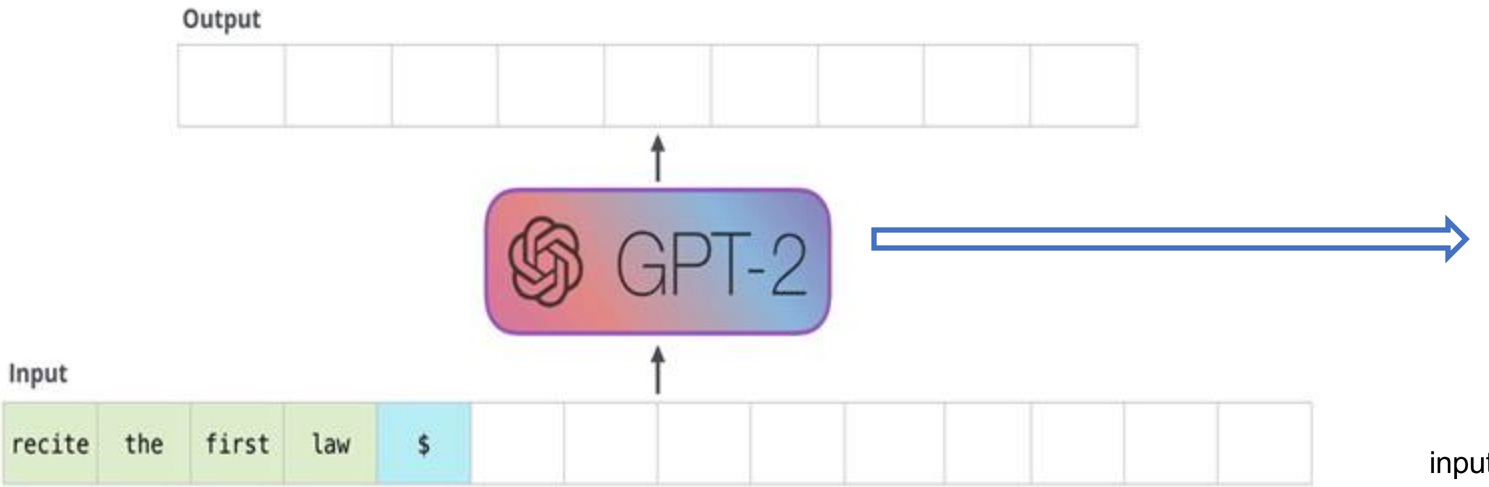


[from Alammar, The Illustrated GPT-2, https://jalammar.github.io/illustrated-gpt2/]

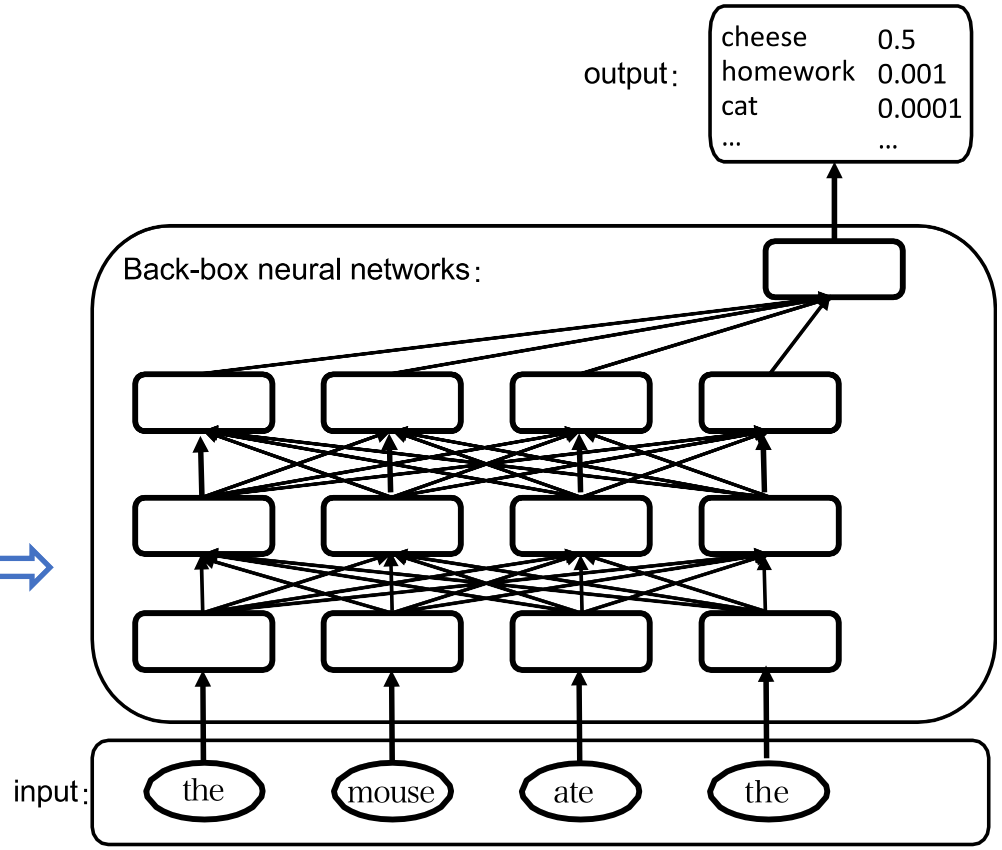
What is Language Model?

Autoregressive Generation

Computing conditional probability using a neural network



P(next word | previous words)



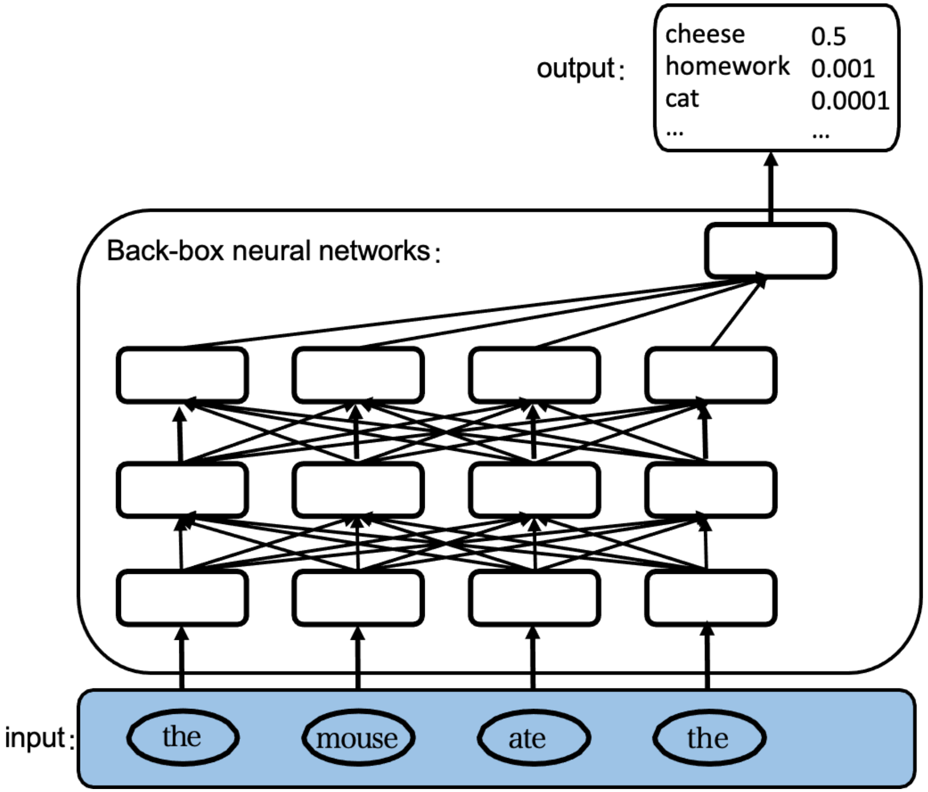
[from Alammar, The Illustrated GPT-2, https://jalammar.github.io/illustrated-gpt2/]

Unpack Language Model: Tokenization

- Tokenization: Converts any string into a sequence of tokens. e.g.,

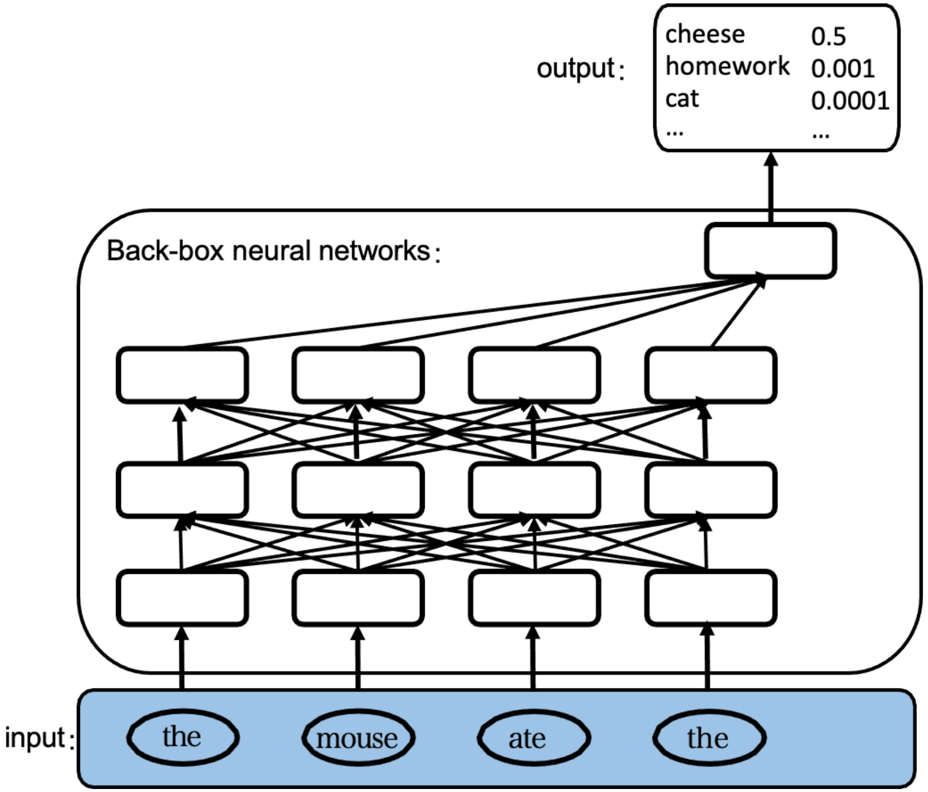
  the mouse ate the cheese⇒[the,mouse,ate,the,cheese]

- How to tokenize?
  - Split by space, by punctuation, ...
- What makes good tokens?
  - Not too many, e.g., not every character
  - Not too few : e.g., mother-in-law or father-in-law, 1 token or 3 tokens?
  - Each token should be a linguistically or statistically meaningful unit.



Unpack Language Model: Tokenization

- How about rare words?



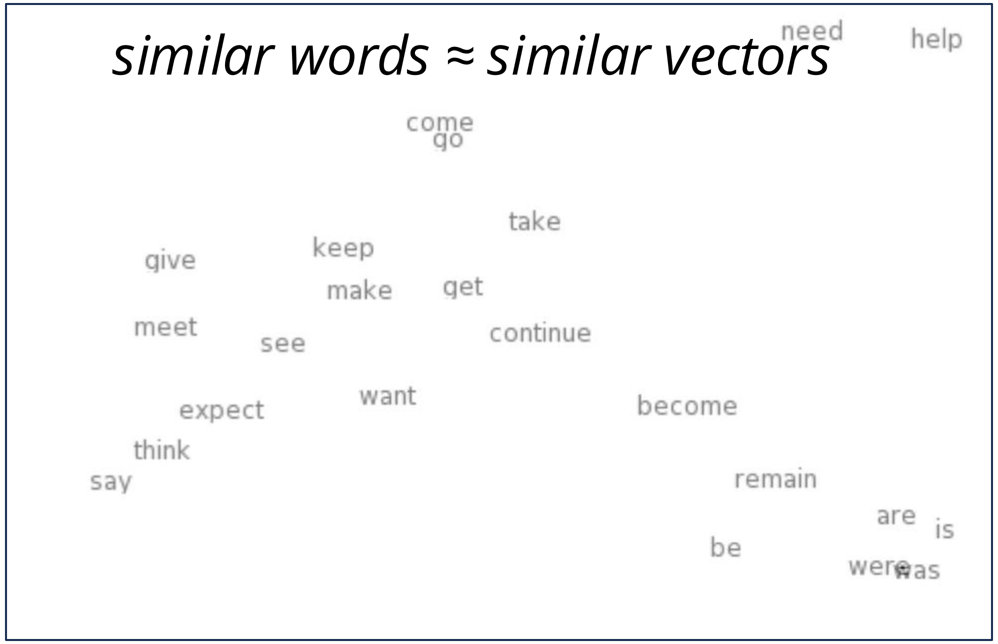
https://blog.research.google/2021/12/a-fast-wordpiece-tokenization-system.html

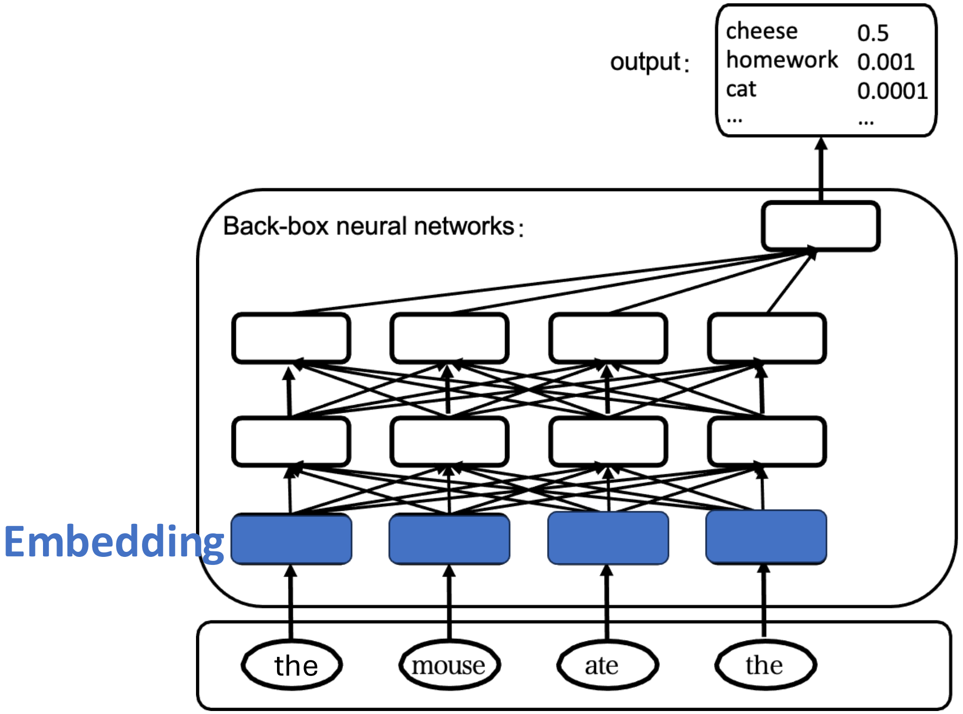
Unpack Language Models: Embedding

- **Word Embedding**: Vector representations of words

$$f_{\text{word2vec}} : V \to \mathbb{R}^d$$

$$v_{mouse} \quad \begin{pmatrix} -0.224 \\ 0.130 \\ \cdots\cdots \\ 0.276 \end{pmatrix}$$ ≈ 100–3000 dims!





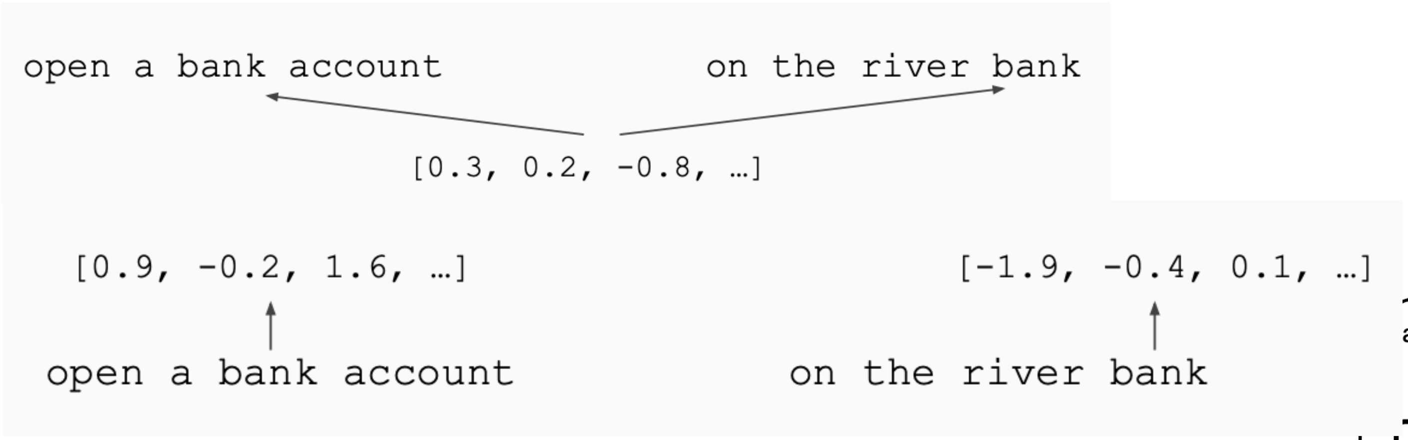
Unpack Language Models: Embedding

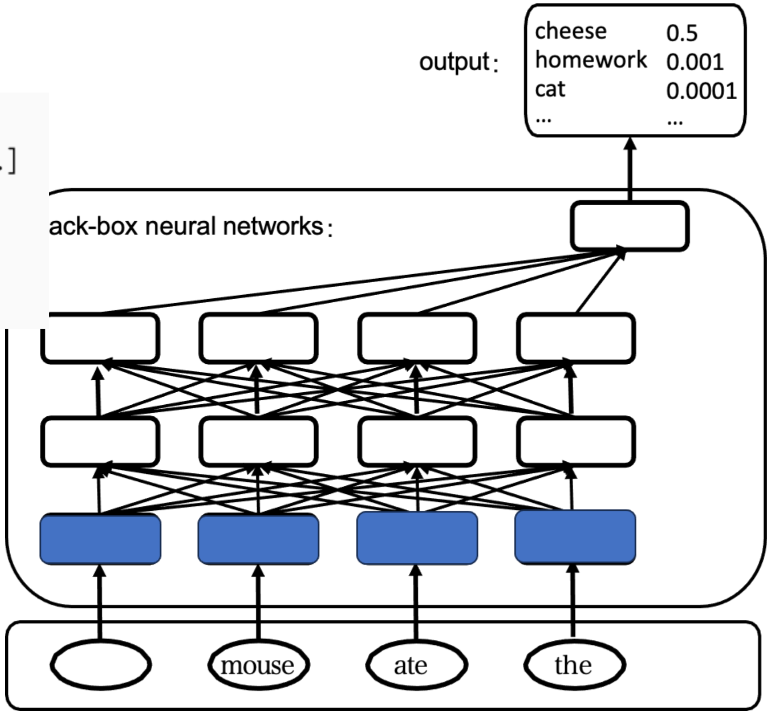
- **Word Embedding**: Vector representations of words

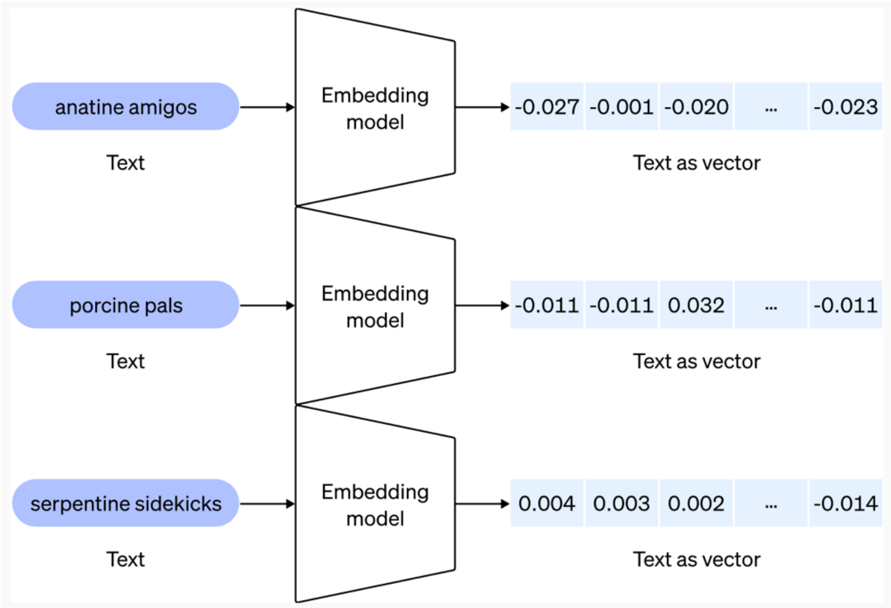
Context-free

vs.

Contextualized







Text Embedding for Retrieval Augmented Generation (RAG)

https://openai.com/index/new-embedding-models-and-api-updates/

Unpack Language Models: Model Architecture

Language Model : Assign a probability to any sequence of words $x_1, x_2, \ldots, x_n$ (a.k.a. tokens),

$$P(x_1, x_2, \ldots, x_n)$$

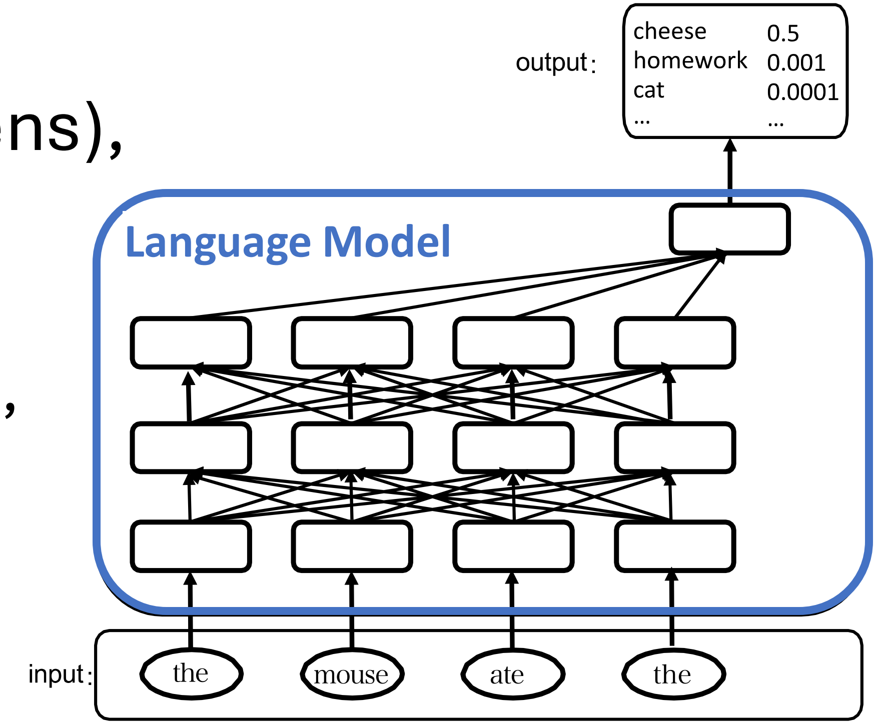
Example:

Given vocabulary $V = \{$ate,ball,cheese,mouse,the, homework, ...$\}$, the model may estimate:

p(the,mouse,ate,the,cheese)=**0.02**,

p(the,cheese,ate,the,mouse)=0.001,

p(mouse,the,the,cheese,ate)=0.00001



The probability intuitively tells us how “good” a sequence of tokens is

What “good” means?

Modeling Natural Language

Language Model: Assign a probability to any sequence of words $x_1, x_2, \ldots, x_n$ (a.k.a. tokens).

$$\boldsymbol{P(x_1, x_2, \ldots, x_n)}$$

*uh5r-0%9806 98e*59y G8Svv/,]]\vhiut8Gr*

Low probability

ChatGPT is all you need

high probability

Modeling Natural Language

Generation: Given a language model, we can sample a sequence of n words $x_1, x_2, \ldots, x_n$ based on the probabilities

Example:

Given a language model:

p(the,mouse,ate,the,cheese)=**0.02**,
p(the,cheese,ate,the,mouse)=0.001,
p(mouse,the,the,cheese,ate)=0.0001,
…

Sample a sequence of five words

Issue: How likely you can generate a sequence you want?

Autoregressive Language Model

Chain Rule of Probability:

$$P(x_1, x_2, \dots, x_n) = P(x_1)\, P(x_2|x_1)\, P(x_3|x_1, x_2)\dots P(x_n|x_{1:n-1})$$
$$= \prod_{i=1}^{n} P(x_i|x_{1:i-1})$$

$\boldsymbol{P(x_i|x_{1:i-1})}$ is a **conditional probability distribution** of the next token x_i given the previous tokens $x_{1:i-1}$.

Example:

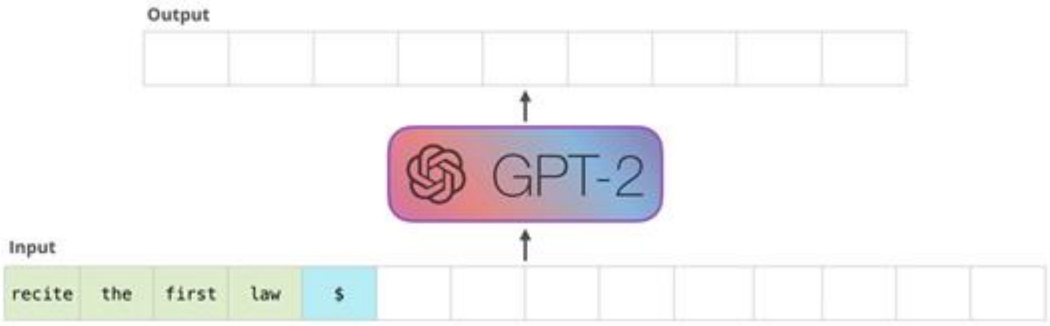
p(the,mouse,ate,the,cheese)=p(the) x
p(mouse | the) x
p(ate | the, mouse) x
p(the | the, mouse, ate) x
p(cheese | the, mouse, ate, the)

How to compute conditional probability?

Autoregressive Language Model

Autoregressive Generation: to generate $(x_1,\ x_2,\ \dots, x_n)$, sample one token at a time given the tokens generated so far:

For $i = 1, 2, \dots, n$, sample
$$x_i \sim P(x_i | x_{1:i-1})$$

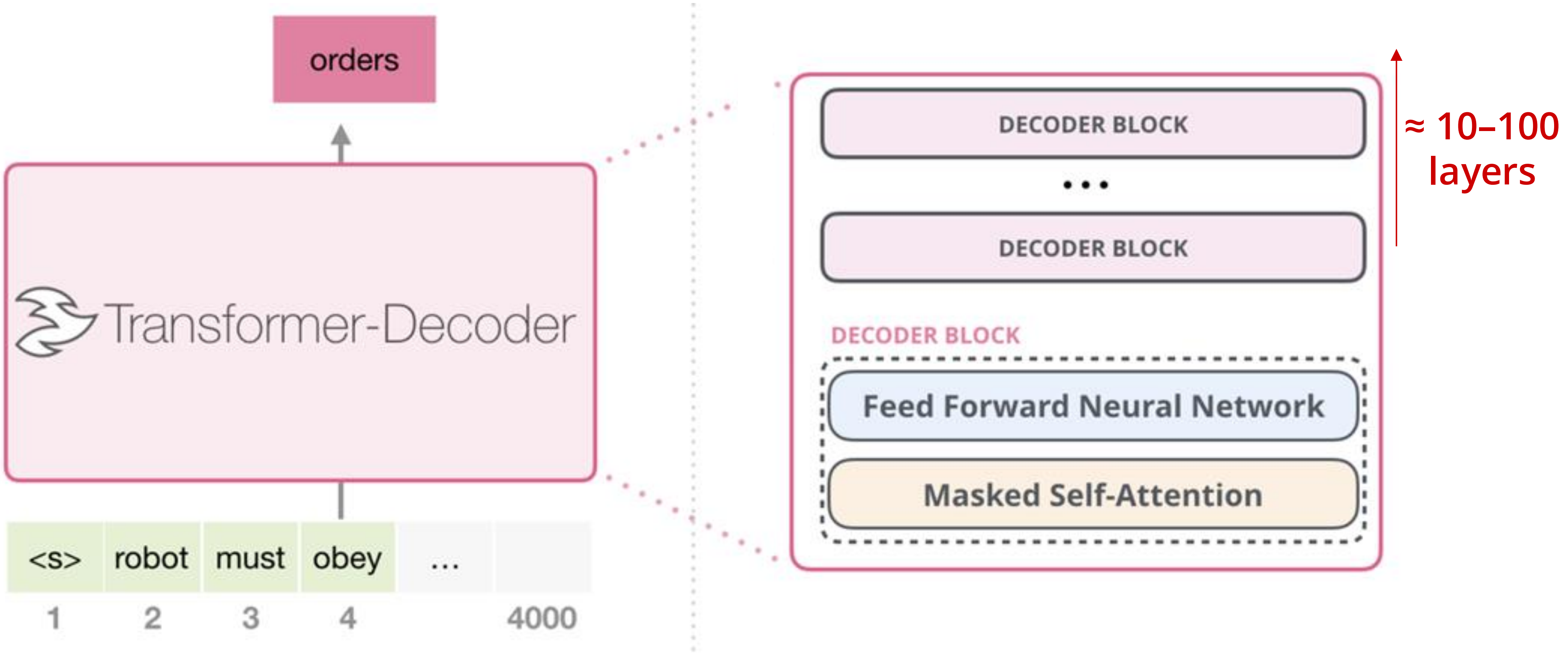


Conditional generation: specify some prefix sequence $x_{1:i-1}$ (i.e., **prompt**) and sample the rest $x_{i:n}$ (i.e., **completion**), e.g.,

$$\underbrace{\text{the, mouse, ate}}_{\textbf{Prompt}} \rightarrow \underbrace{\text{the, cheese}}_{\textbf{Completion}}$$

Autoregressive Language Model Architecture

How to compute conditional probability p(next token | previous tokens)?



[from Alammar, The Illustrated GPT-2, https://jalammar.github.io/illustrated-gpt2/]

Unpack Language Models: NLP Tasks

It turns out many tasks conform to conditional generation!

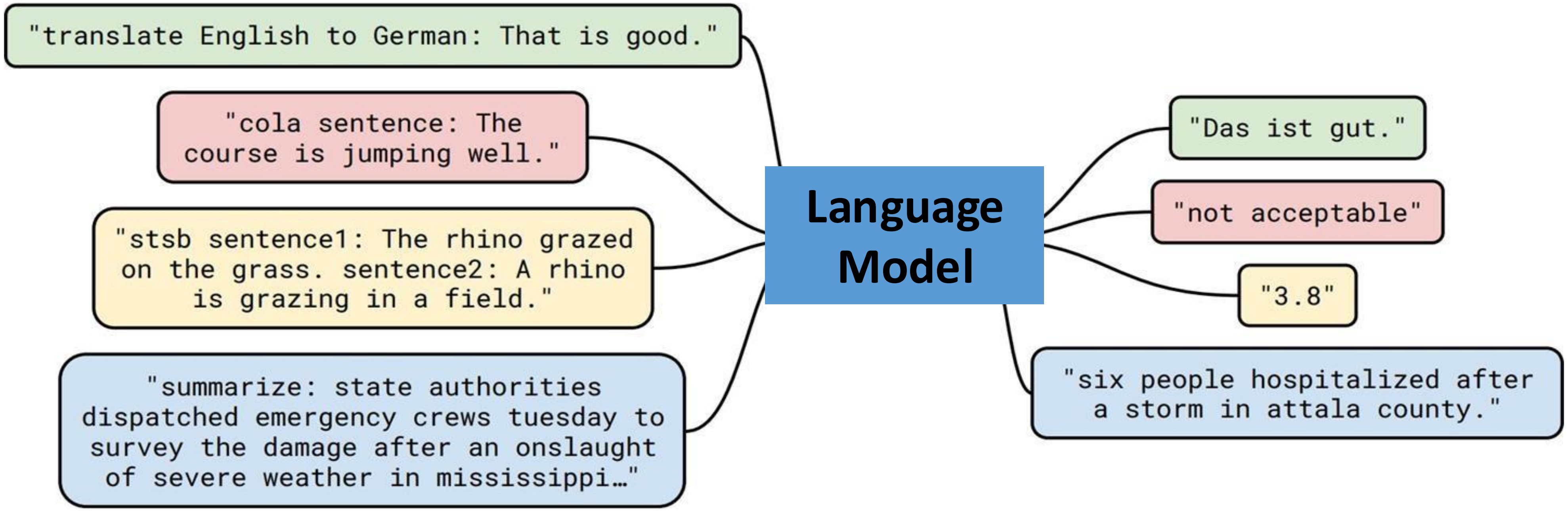
"The plot was substandard, but it left a smile!"
What is the sentiment of above sentence? *Positive*

P(positive | "The plot was substandard, but it left a smile!" What is the sentiment of above sentence?) = 0.7

P(negative | "The plot was substandard, but it left a smile!" What is the sentiment of above sentence?) = 0.3

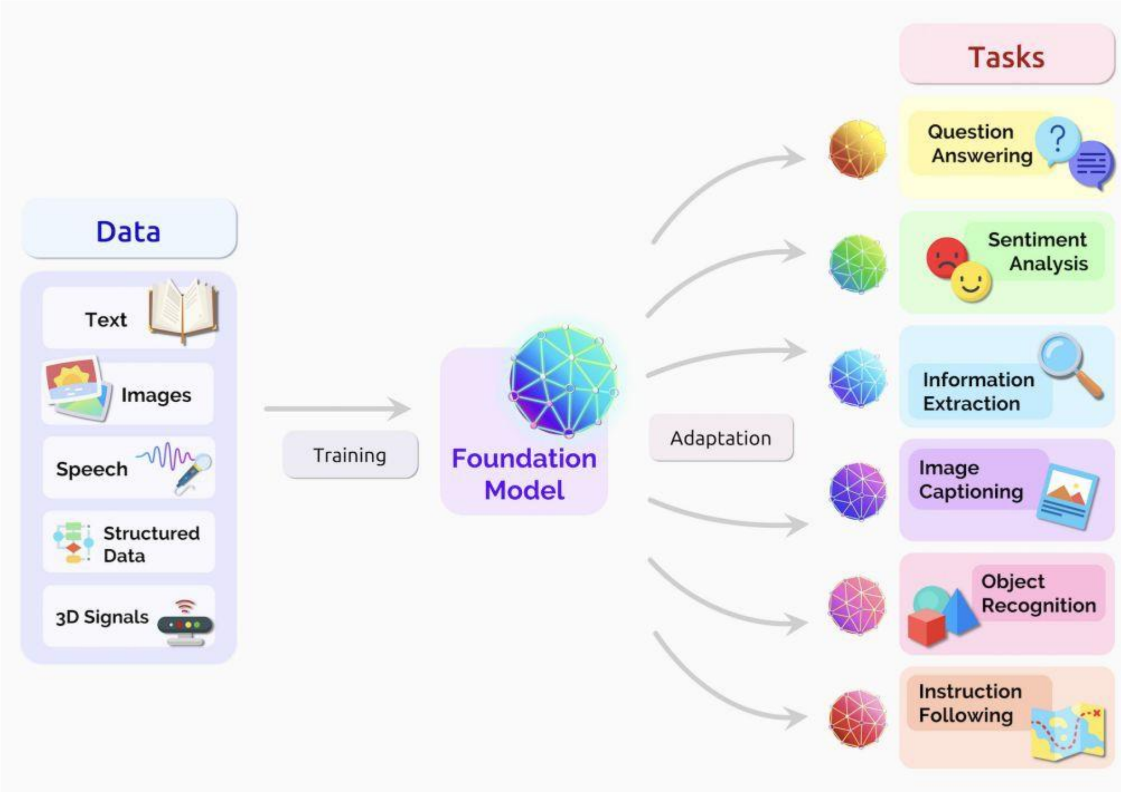
Framing Tasks as Conditional Generation

It turns out many tasks conform to conditional generation!

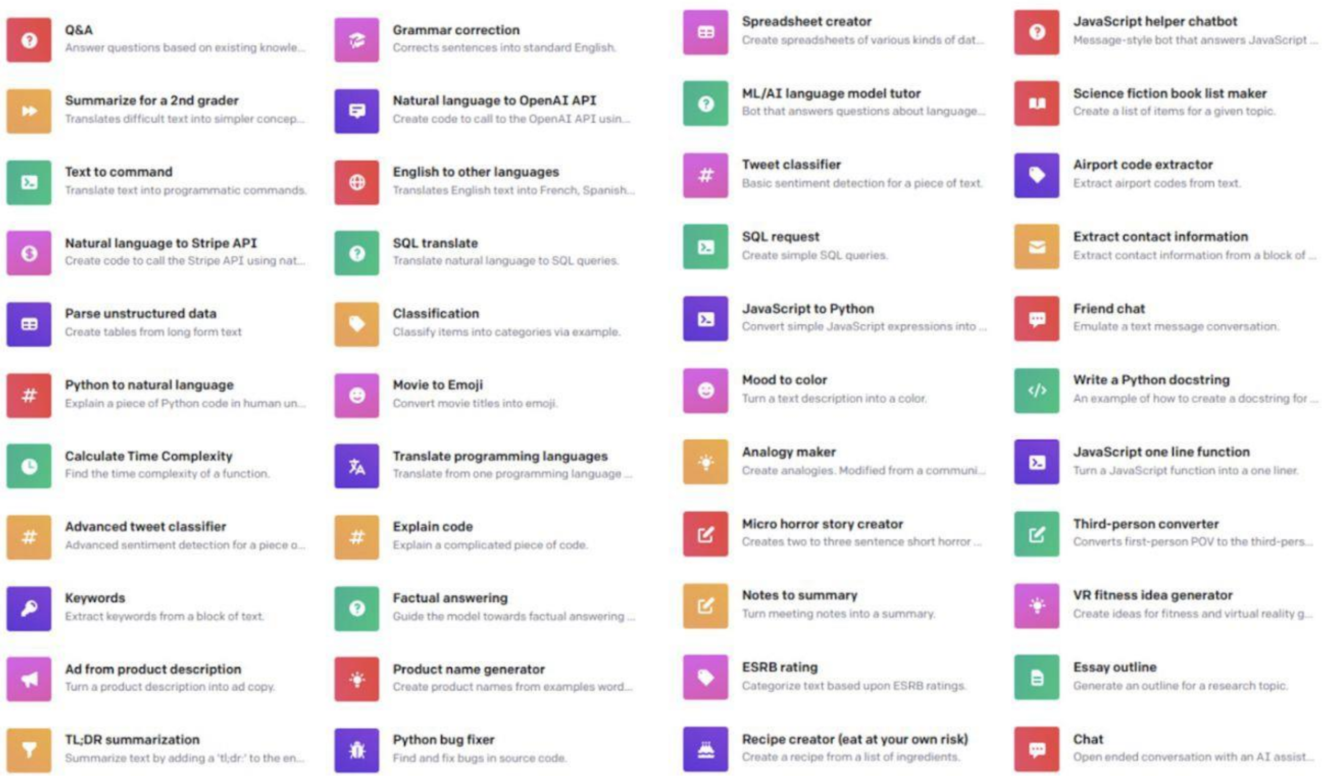


Note: stsb: sentence textual similarity benchmark (1-5)

Powering Rich New Capabilities



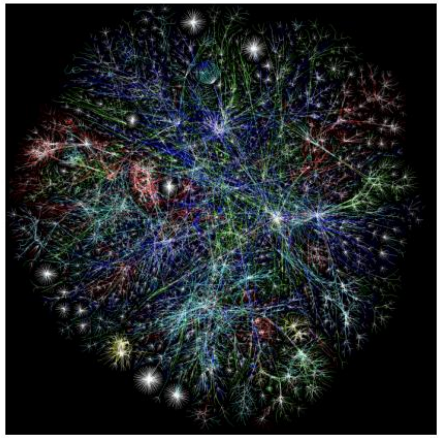
https://arxiv.org/pdf/2108.07258.pdf



Source: openai

Unpack Language Models: Training Data

- Large language models trained on chunk of the internet
  - Scraping
  - Parsing
  - Cleansing



Chunk of the internet,
~10TB of text

6,000 GPUs for 12 days, ~$2M
~1e24 FLOPS

parameters.zip

~140GB file

*numbers for Llama 2 70B

Source: Andrej Karpathy, Intro to LLMs, https://drive.google.com/file/d/1pxx_ZI7O-NwI7ZLNk5hI3WzAsTLwvNU7/view

What will we learn?

1. Collect and clean data

2. Prepare Data for language modeling (Tokenization, Embedding, Parsing)

3. Work on NLP tasks using LLMs & traditional techniques

Note: We'll cover Transformer model briefly but won't dive into the details, since this requires deep learning foundations

Web Scraping

Extract and transform text from html, pdf, doc, xml, txt, …
Scrape static and dynamic web pages

Preprocessing / Parsing

Extract words/terms (Tokenization)
Part of Speech
Stemming/Lemmatization
Named Entity Recognition

Feature Extraction

Embedding: word, sentence, document

NLP Tasks

Supervised Learning
- Classification
- Sentiment mining

Unsupervised Learning
- Clustering
- Topic modeling (LDA)

LLMs
- Prompt Engineering
- Retrieval Augmented Generation

Outline

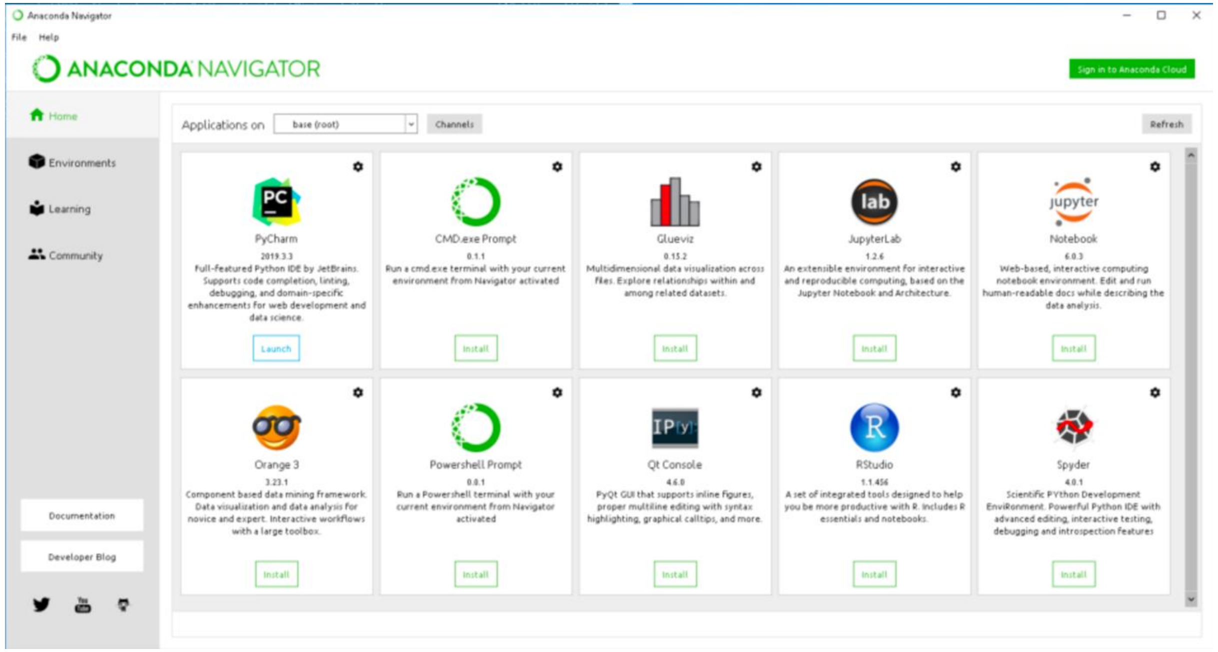
- **Introduction** — Name, major, interest, expectation from this course etc.
- **What is Web Mining?**
- **Syllabus**
- **Course Logistics**
  - Python Installation,
  - Poll Everywhere, Datacamp assignments
  - Class project

Outline

- **Introduction** — Name, major, interest, expectation from this course etc.
- **What is Web Mining?**
- **Syllabus**
- **Course Logistics** — Python Installation, Poll Everywhere, Datacamp assignments, Class project

Python and Python Editor Installation

- Recommended: Anaconda. Instruction:
  1. Download Anaconda (prefer **Python 3.8 or above version**) https://www.anaconda.com/products/individual
  2. Install Anaconda following instruction at https://docs.anaconda.com/anaconda/install/index.html for windows or macOS
  3. After installation, open a terminal (or anaconda command window for Windows) to update python libraries using: conda update --all
  4. Pick Jupyter Notebook as the GUI editor. Launch it from “Anaconda Navigator” or type “jupyter notebook” from terminal (Mac users).
  5. Make sure the following packages have been installed

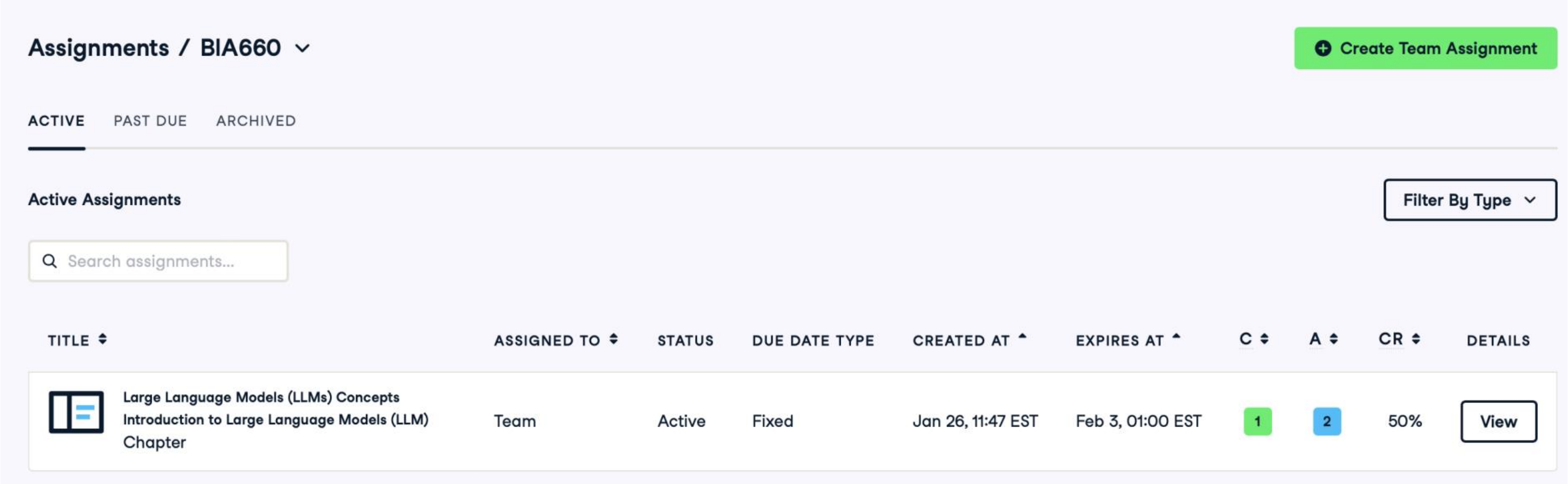


Poll Everywhere

- Simple in-class quiz is administered through Poll Everywhere
- The quiz results will be counted as your class participation
- To participate:
  - Click “Sign up” at https://www.polleverywhere.com and register using your **Stevens Email ID**, if you don’t have an account at Poll Everywhere yet
  - With your account created, now you can login our class poll site: **PollEv.com/emilyliu**
  - You should be able to see a test poll (the question is: "Have you used any of the Python Packages below?"). Please respond to this question.

Datacamp Assignments

- Register at Datacamp using your Stevens email ID
- Accept invitation: https://www.datacamp.com/groups/shared_links/d1dfd7ac52c8cf139e6dce448af5c91c1e4d67735ee66ee039e6b1e9cc768097
- After login, you shall see your first assignment:



Class Project – Option 1

Group project

- You form a team with 3-4 members
- Team brainstorm to decide a favorite topic
- Breakdown the project into tasks and assign tasks to team members
- Each one works on assigned tasks
- Integrate the tasks and analyze the results
- Write a report and present the report

Class Project – Option 2

- The whole class collectively works on a research project
  - Split the project into four homework assignments
  - Everyone works on each assignment individually and some results(e.g., collected data, benchmarking models) will be shared among class
  - Then each student selects a specific theme and conducts additional analysis using the materials contributed by the class
  - Write a report and present the report
- Pros: Reuse your homework assignments and focus on research
- Cons: We have never done this before. We'll need to figure out how to make it works

Class Project – Option 2

- Potential topic: ESG report analysis

DECARBONIZATION & CLIMATE RISK

Supporting the transition to a low-carbon economy in line with Paris Agreement goals

- Renewable energy and clean tech
- Energy efficiency
- Physical impact adaptation
- Just transition

7 AFFORDABLE AND CLEAN ENERGY
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
11 SUSTAINABLE CITIES AND COMMUNITIES
13 CLIMATE ACTION

DIVERSE & INCLUSIVE BUSINESS

Supporting business practices that create a more just and inclusive society

- Affordable access to essential services
- Investing in communities
- Racial justice
- Pay equity
- Board and employee diversity

3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
10 REDUCED INEQUALITIES

NATURAL CAPITAL & BIODIVERSITY

Supporting business models that reduce negative impact on biodiversity in line with the Post-2020 Biodiversity Framework

- Sustainable sourcing and use of resources
- Land and sea use change
- Deforestation
- Pollution reduction

6 CLEAN WATER AND SANITATION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND

CIRCULAR ECONOMY & WASTE REDUCTION

Supporting business models that reduce impact on natural resources and that innovate to reduce waste generation, with a focus on plastic waste

- Recycling and reuse
- Sustainable sourcing
- Life cycle analysis
- Water stewardship

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
14 LIFE BELOW WATER

DECENT WORK & RESILIENT JOBS

Supporting decent work across the entire value chain and making workforces resilient in the face of innovation and change

- Automation and the workforce
- Supply chain management
- Living wage
- Workforce wellbeing

3 GOOD HEALTH AND WELL-BEING
5 GENDER EQUALITY
8 DECENT WORK AND ECONOMIC GROWTH
12 RESPONSIBLE CONSUMPTION AND PRODUCTION

https://www.morganstanley.com/content/dam/msdotcom/en/assets/pdfs/Morgan_Stanley_2023_ESG_Report.pdf

Class Project – Option 2

- Potential topic: ESG report analysis
  - *Assignment 1 – Data Collection*: Select a specific industry (e.g., bank, education), scrape and process ~30 reports
    - Scraped content will be shared among class.
    - High-quality data contributors will have extra credits.
  - *Assignment 2 – Classification*: Identify content belongs to specific themes (e.g., green finance, united SDG goals)
  - *Assignment 3 – Clustering*: Cluster themes for a specific type of content (e.g., typical investments disclosed in green finance text)
    - Discovered themes will be shared among class
  - *Assignment 4 – LLM-augmented text analysis*: Perform Assignments 2 & 3 using LLMs and benchmark model performance
  - Project: Select a topic and reuse the assignments and materials
    - Comparing green finance investments by industry, or region (USA and Europe)
    - Regulatory compliance of ESG activities
    - Initiatives related to mitigate climate risks
    - Connecting ESG activities to firm performance

Class Project – Option 2

- Any Other Potential topics?
  - Tech Blog Topic Explorer: e.g., TeckCruch
  - Health forum
  - Customer support sites

References

- CS 194/294-267 Understanding Large Language Models: Foundations and Safety (https://rdi.berkeley.edu/understanding_llms/s24)
- Stanford course "**Large Language Models**" lecture notes, available online at https://stanford-cs324.github.io/winter2022/lectures/
- Hung-yi Lee, Tutorial for General Deep Learning Technology, https://speech.ee.ntu.edu.tw/~tlkagk/talk.html