Carnegie Mellon University

Lecture 2:
Large Language Models

SPRING 2024

MOHAMED FARAG

FARAG@CMU.EDU

Agenda

- NLP as Introductory Domain to LLMs
- What are LLMs?
- Key Components of LLMs
- Demo
  - Redeem Your GCP Coupon
  - Run LLM on Google Vertex AI
  - Disable the Billing

Natural Language Processing (NLP)

- Statistical methods, large datasets, and deep learning led to ML-based NLP adoption in the 2000s and 2010s.
- ML-based NLP systems are used in customer support chatbots, virtual assistants, sentiment analysis, and machine translation.
- Late 2010s saw the emergence of pre-trained language models like ELMo, GPT, and BERT.
- These models which are pre-trained on large data and fine-tuned for specific NLP tasks have achieved top results in benchmarks.
- As a result, there has been a significant progress in language understanding, text generation, and other NLP tasks due to these developments.
- NLP became crucial in various modern applications and services.

NLP Tasks

NLP aims to bridge human language and computer understanding, applied in various language tasks.

- **Text classification:** Labeling texts, like spam detection, sentiment analysis, and topic categorization.
- **Named Entity Recognition (NER):** Identifying and classifying entities in text (people, organizations, locations, dates).
- **Machine translation:** Automatic translation between languages.
- **Text generation:** Creating human-like text for chatbots, autogenerated content, or summarization.

NLP Tasks – Cont’d

- **Speech recognition:** Converting spoken language into written text.
- **Text summarization:** Generating concise summaries of longer texts.
- **Question answering:** Providing answers to natural language questions.

These tasks form the foundation of current NLP applications.

Key NLP Concepts

- **Tokenization:** Breaking text into smaller units (words or sub-words) called tokens.
- **Part-Of-Speech (POS) tagging:** Assigning grammatical tags (noun, verb, adjective, etc.) to each word in a sentence.
- **Word embeddings:** Creating dense vector representations of words (e.g., Word2Vec, GloVe) that capture semantic relationships.
- **Stemming and lemmatization:** Reducing words to their base or root form (e.g., 'running' to 'run')
- **Language models:** Predicting word sequence likelihood, crucial for tasks like machine translation and text generation.

What are Large Language Models?

LLMs are **large**,
general-purpose language models
that can be **pre-trained** and
then **fine-tuned** for specific purposes.

Large?

1. Large training datasets
2. Large number of parameters (millions of parameters).
    - Linear regression has 2 parameters!!

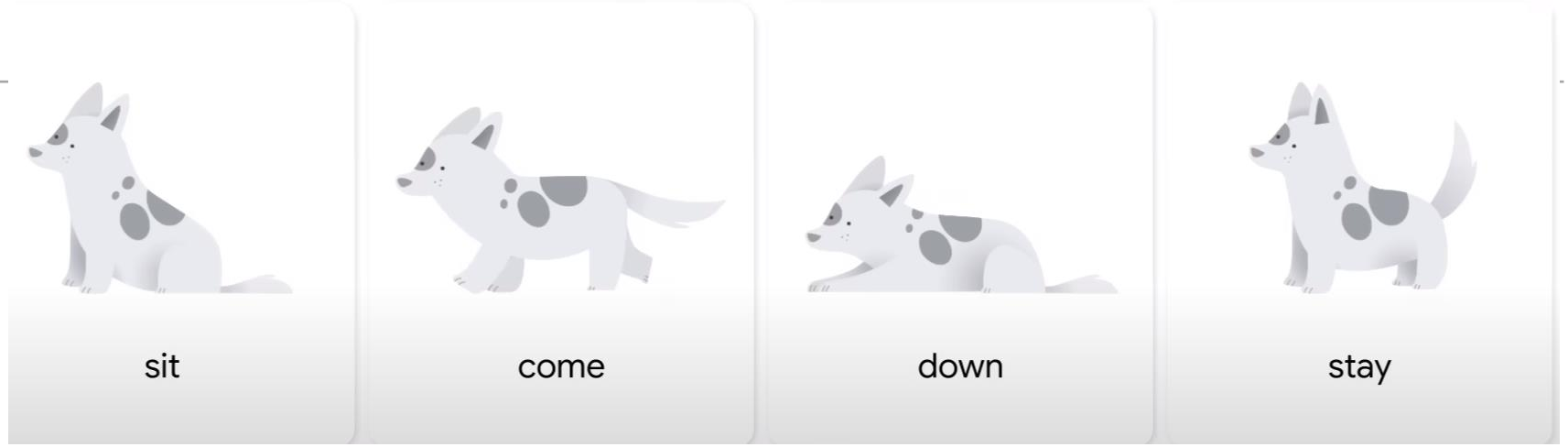
General Purpose?

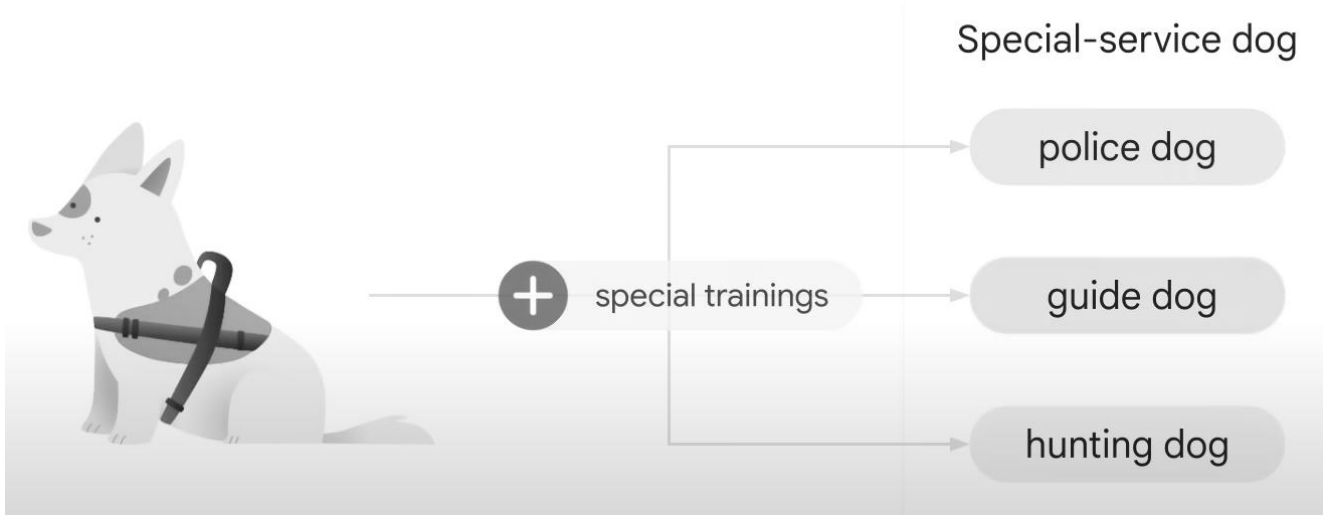
1. Commonality of human languages
2. Resource restrictions

Pre-trained and fine-tuned

Pre-trained

Fine-tuned

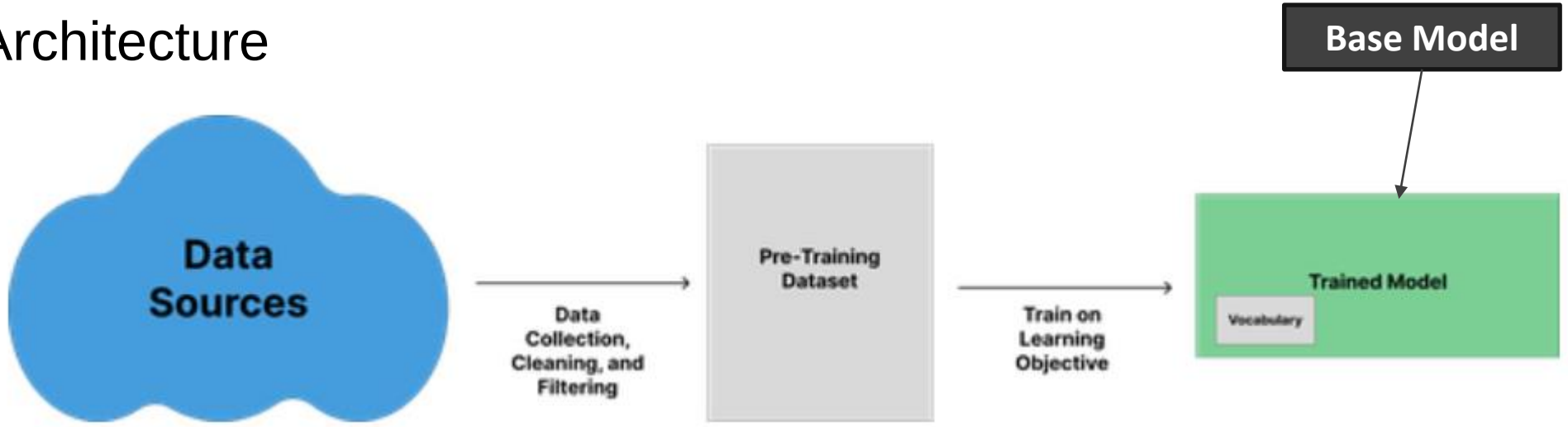




Images from Google's Introduction to Large Language Model

Key Components of LLMs

1. Pre-training Data
2. Vocabulary and Tokenizer
3. Learning Objective
4. Architecture



1. LLM Components: Pre-training Data

- Here, our goal is to answer the question: **“What’s it trained on?”**
- It’s import to use high-quality data to avoid “Garbage-in, Garbage-out”.
- Pre-trained data come from “Corpus”.
- A corpus is a large collection of text or utterances used for language analysis and model training.

LLM Components: Pre-training Data
Corpora Types

1. **Text Corpora**
    - Collection of written texts (books, articles, web pages, emails, social media posts).
    - Used for language modeling, sentiment analysis, text classification, information retrieval.
2. **Speech Corpora**
    - Contains audio recordings or transcriptions of spoken language.
    - Utilized in speech recognition, speaker identification, emotion detection.

LLM Components: Pre-training Data
Corpora Types (Cont'd)

3. **Parallel Corpora**
    - Text in multiple languages aligned at sentence or document level.
    - Employed for machine translation and cross-lingual tasks.
4. **Treebanks**
    - Annotated corpora with syntactic parse trees.
    - Used in parsing and syntax-based machine learning
5. **Multimodal Corpora**
    - Includes text and other modalities like images, videos, or audio.
    - Applied in tasks involving multiple modalities' understanding and generation.

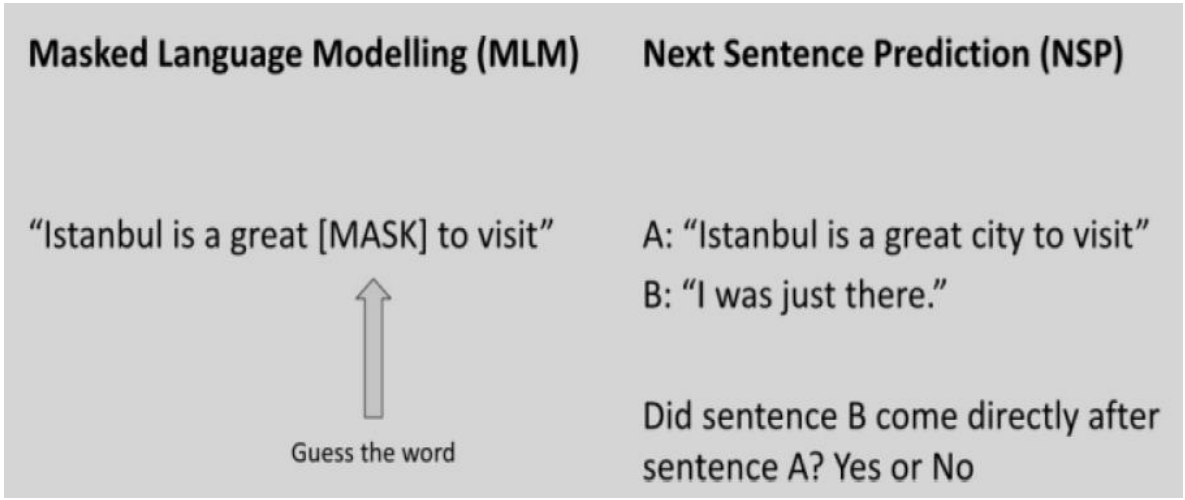
In your opinion, which corpora type is used the most for LLMs?

Example: BERT Pre-training Data Sources

1. **English Wikipedia:**
   - Contains articles from the English Wikipedia.
   - Diverse topics and writing styles, representing English language well.
   - Size: 2.5 billion words.
2. **The BookCorpus:**
   - Large collection of fiction and non-fiction books scraped from the web.
   - Includes various genres like romance, mystery, science fiction, and history.
   - Books have a minimum of 2000 words and are written by verified authors.
   - Size: 800 million words.

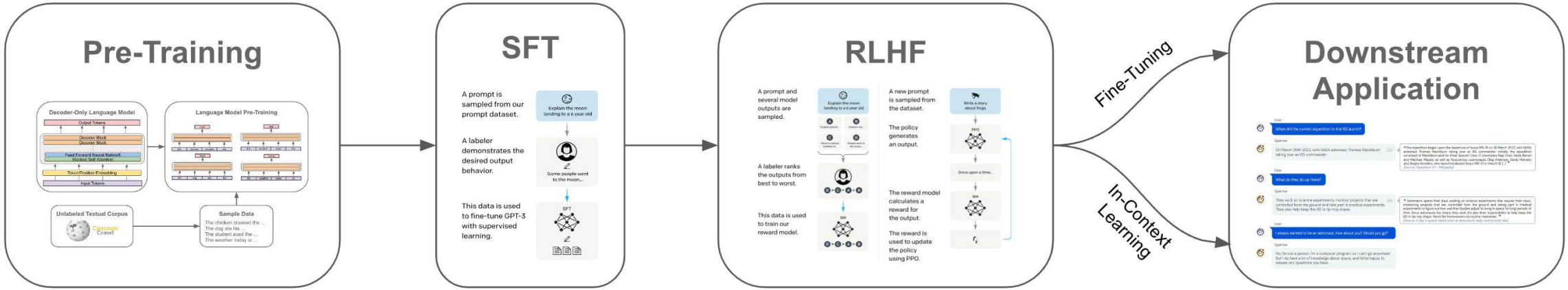
Digression 1:
BERT Training on Language Modeling Tasks

1. **Masked Language Modeling (MLM):**
    - Aids in recognizing token interactions within a sentence.
2. **Next Sentence Prediction Task:**
    - Helps BERT understand token interactions between sentences.



Digression 2:
How complex is it to train LLMs?

Read this article about [LLM Training Pipeline]



LLM Components: Pretrained Data Open Topics

1. **There is an ongoing debate on whether the text data are sufficient to teach the model on logical reasoning**
    - Only around 12% of information we understand from text is explicitly mentioned in text
    - Multimodal models combine different modalities like image, video, speech, and text. They are becoming a promising avenue of research and are likely to see more widespread usage in the coming years.
2. **LLMs are usually trained using one epoch and they are considered underfit**. Recently, some research shows that LLMs can be trained with about 5 epochs.

LLM Components:

Examples of Popular Text Corpora

[Check C4 Dataset]

Name	Data Source(s)	Size	Year Released	Public?
C4	Common Crawl	750GB	2019	Yes (reproduced version)
The Pile	Common Crawl, PubMed Central, Wikipedia, ArXiv, Project Gutenburg, Stack Exchange, USPTO, Github etc	825GB	2020	Yes
RedPajama	Common Crawl, Github, Wikipedia, arXiv, StackExchange etc	1.2T tokens	2023	Yes
BooksCorpus	Sampled from smashwords.com	74M sentences	2015	Original not available

2. LLM Components: Vocabulary and Tokenizer

- Here, our goal is to answer the question: **“What’s it trained over?”**
- We need to determine the language's vocabulary and tokenization rules.
- Humans process language in terms of meaning-bearing words and sentences while Language models process language in terms of **tokens**.
- The term **token** refers to the smallest unit of semantic meaning created by breaking down a sentence or piece of text into smaller units and are the basic inputs for an LLM.
- Tokens can be words but also can be “sub-words”

LLM Components: Tokenizer

- Tokenizing a text means splitting it into tokens (words or sub-words), which then are converted to IDs through a look-up table mapping words in text to corresponding lists of integers.
- Before training LLM, Tokenizer's dictionary is fitted to the entire training dataset.
- Once fitted, the tokenizer's dictionary is frozen.
- Tokenizers output specific integers, not arbitrary ones.
- The range of output integers is from 0 to N, where N is the tokenizer's vocabulary size.

LLM Components: Tokenization Techniques

1. **Whitespace Tokenization:**
    - Splits text based on whitespace (spaces, tabs, newlines).
    - Simple and common for English text.
    - May not handle special cases like hyphenated words or contractions well.
2. **Punctuation Tokenization:**
    - Splits text based on punctuation marks (periods, commas, exclamation marks).
    - Useful for text with significant punctuation.
    - May face issues with abbreviations or special cases.

LLM Components: Tokenization Techniques - Cont'd

3. **Word Tokenization:**
    - Advanced tokenizer using language-specific rules.
    - Accurately handles hyphenated words, contractions, and punctuation.
4. **Sub-word Tokenization:**
    - Methods like Byte-Pair Encoding (BPE) and SentencePiece.
    - Splits words into sub-word units.
    - Effectively handles out-of-vocabulary and rare or unseen words.

LLaMa 2 Tokenization

- LLaMa 2 utilizes a BPE tokenizer that divides numbers into separate digits and decomposes unfamiliar UTF-8 characters into bytes.
- It has a total vocabulary of 32,000 tokens

LLM Components: Vocabulary

- A vocabulary in NLP refers to the set of unique words or tokens present in a corpus of text.
- Vocabulary is a fundamental component of language processing, as it defines the complete list of words that a model or system can understand and work with

LLM Components: Vocabulary Creation

1. **Tokenization:**
    - Splitting text into individual tokens (words, subwords, or characters).
    - Depends on the chosen tokenization strategy.
2. **Filtering and Normalization:**
    - Common steps include converting text to lowercase, removing punctuation.
    - Filtering out stop-words to clean data and reduce vocabulary size.

LLM Components: Vocabulary Creation (Cont'd)

3. **Building Vocabulary:**

   - Collecting unique tokens post-tokenization and preprocessing.
   - Assigning each token a unique numerical index for model representation or encoding.

- In many LLM models, words are represented as dense vectors (**word embeddings**) where each word's embedding is indexed using its integer representation in the vocabulary.

Demo

1. **Redeem GCP Coupon**
2. **Run LLM on Google Vertex AI**
3. **Disable the Billing**

1. Redeem GCP Coupon

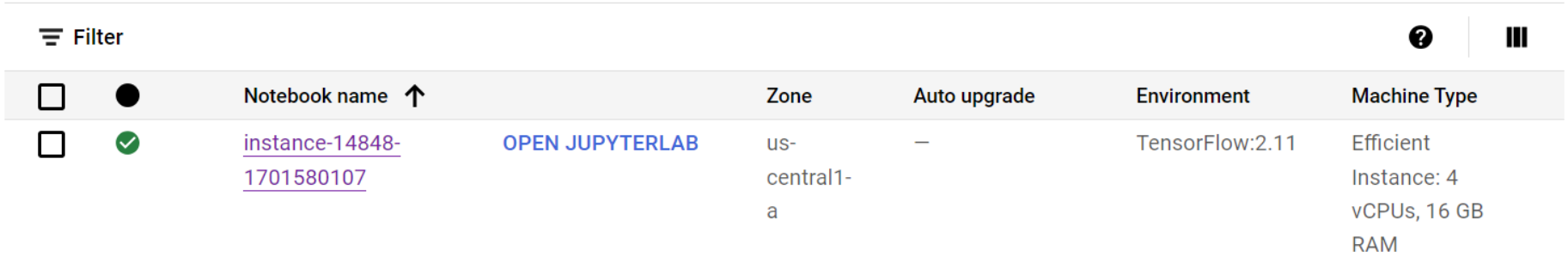
1. If you didn't read this article from last lecture, please read it prior to proceeding with these steps: https://cloud.google.com/docs/overview
2. Identify a GMAIL account to use Google Cloud on it. This email must not have used Google Cloud services before. If you are not sure, please create a new one.
3. Log in to https://console.cloud.google.com/ using your preferred @gmail.com email.
4. Create a New Project (if you don't have one)

Redeem GCP Coupon – Cont'd

5. Check the Canvas Course Page to Redeem Your GCP Coupon.
    - Kindly, don't share the coupon URL with others
    - If you are a waitlisted student, please don't redeem the coupon until you are fully enrolled.
6. Redeem the coupon by entering your Andrew ID
7. You will receive an email in your Andrew inbox to verify your email
8. After the email verification, you will receive another email with the coupon. Switch to your @gmail account and redeem this coupon.
9. Make sure you check your email from the top right corner prior to redeeming the coupon.

2. Run LLMs on Google Vertex AI

- Navigate to Vertex AI on GCP
- Enable Notebook API
- Enable Vertex API
- Create Workbench Instance
- Open JupyterLab on the created instance



2. Run LLMs on Google Vertex AI – Cont'd

```
!pip install langchain
```

```
from langchain.llms import VertexAI

llm = VertexAI(model_name='text-bison@001')
question1 = "What day comes after Saturday?"

print("Answer to 'What day come after Saturday?' is:")
llm(question1)
```

```
Answer to 'What day come after Saturday?' is:
'Sunday'
```

3. Disable the Billing

- After you finish using Google Vertex AI, navigate to billing section and disable the billing on your project
- Keeping the billing enabled will drill down your billing credits

Reading

- GPT-4 Technical Report (through page-14): https://arxiv.org/pdf/2303.08774.pdf
- LLM Training Pipeline: https://cameronrwolfe.substack.com/p/data-is-the-foundation-of-language

- Next Tuesday's quiz will be from the readings.

Quiz-1 Google Form for Waitlisted Students

Waitlisted Students

- All materials for first two weeks will be uploaded here