**My road map in this project**

references :

https://www.youtube.com/@AndrejKarpathy

https://www.youtube.com/watch?v=zduSFxRajkE

https://github.com/brevdev/notebooks/blob/main/mistral-finetune-own-data.ipynb

https://medium.com/@thakermadhav/build-your-own-rag-with-mistral-7b-and-langchain-97d0c92fa146

https://www.langchain.com/

<https://huggingface.co/blog/how-to-generate>

Tokenization: <https://github.com/openai/openai-cookbook>

Fine-tuning: https: //[github.com/brevdev/notebooks/blob/main/mistral-finetune-own-data.ipynb](http://github.com/brevdev/notebooks/blob/main/mistral-finetune-own-data.ipynb)[first link]

Future: <https://github.com/ashishpatel26/LLM-Finetuning> It is advanced for fine-tuning after the first link

The below link is for Tokenization, especially for LLm:

<https://github.com/openai/openai-cookbook/blob/main/examples/How_to_count_tokens_with_tiktoken.ipynb>

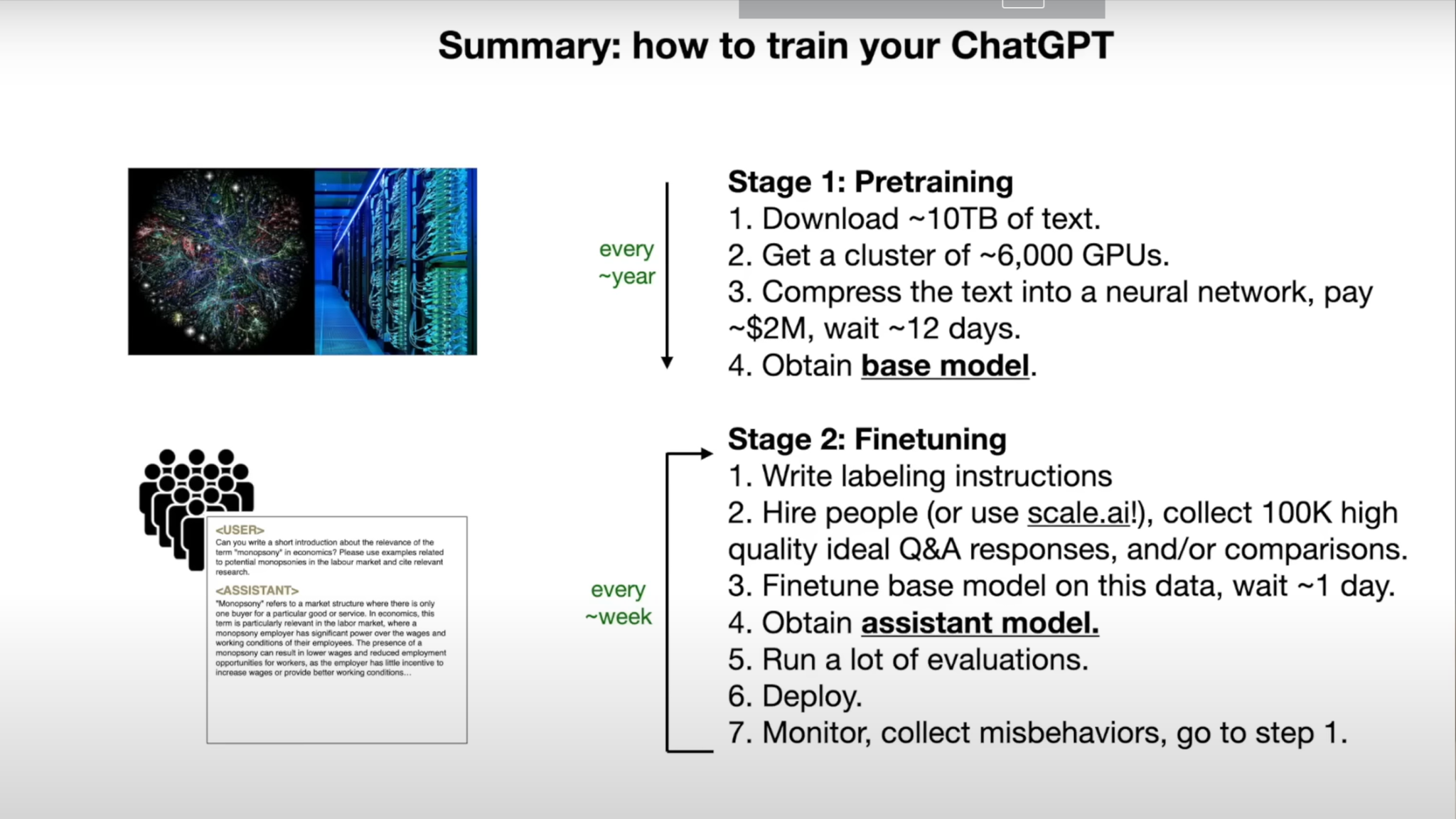
Self-supervised: <https://arxiv.org/pdf/2310.06825>

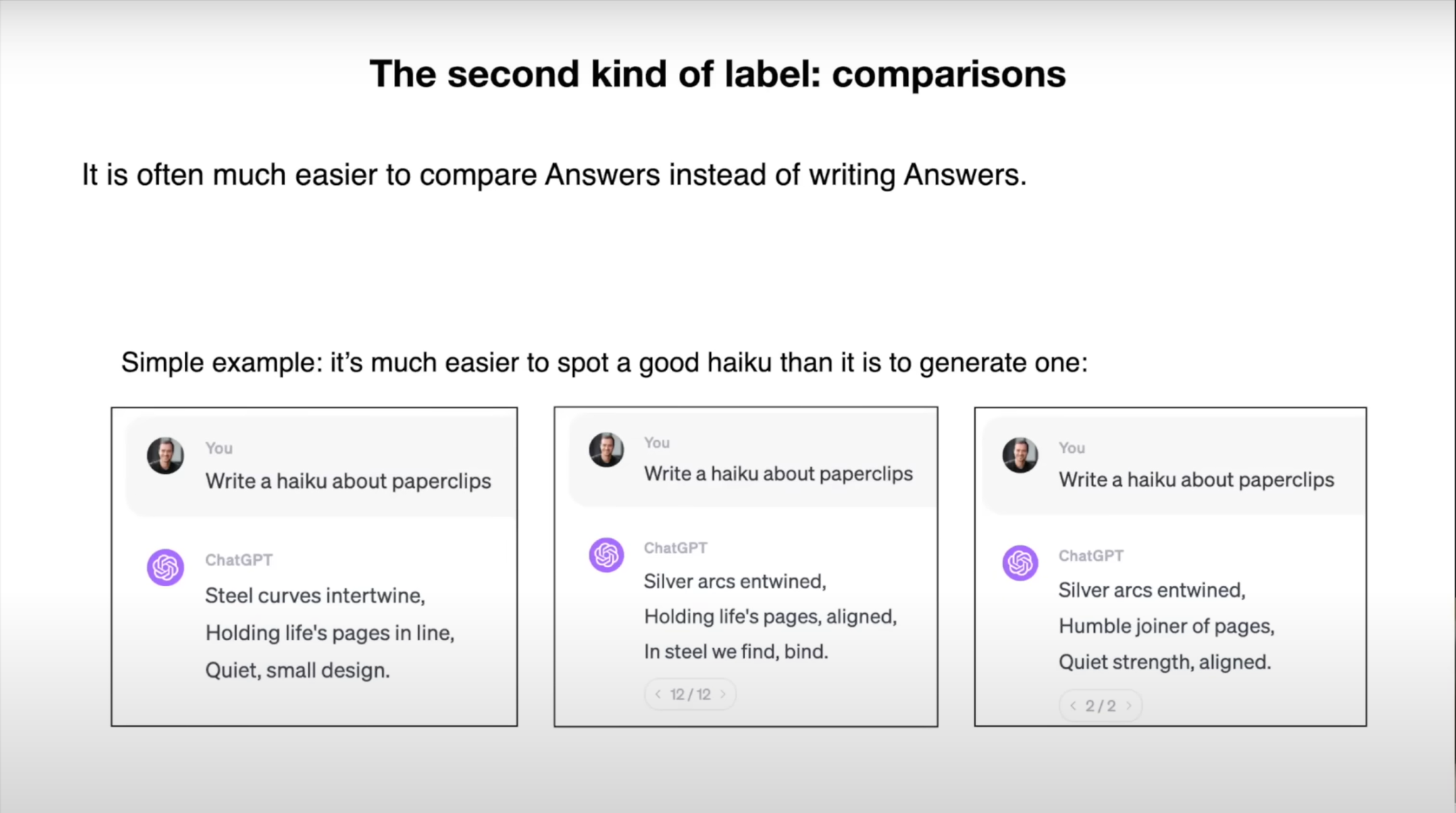
Maybe it is a survey for constraint decoding: <https://arxiv.org/pdf/2403.01632>

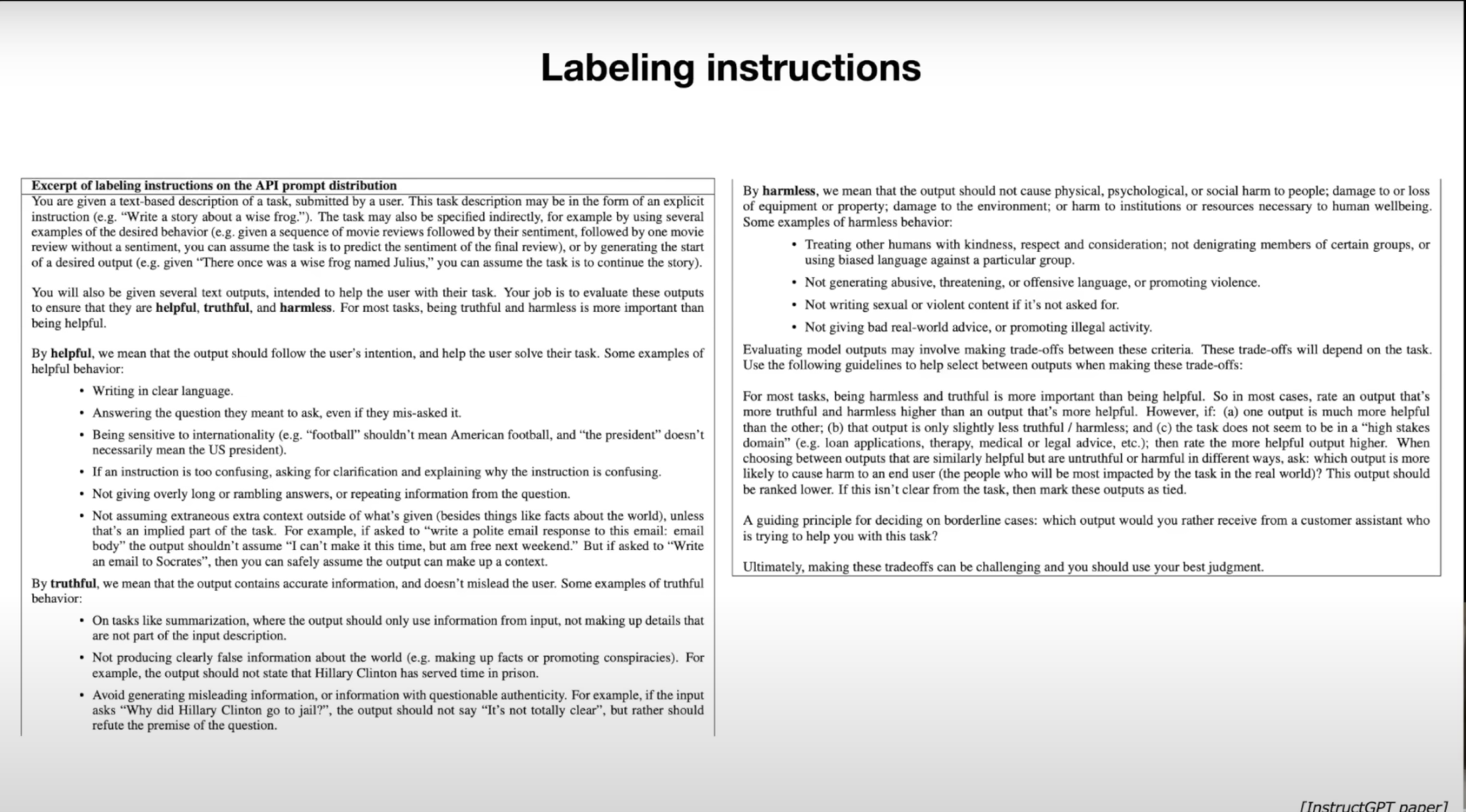
**Papers:**

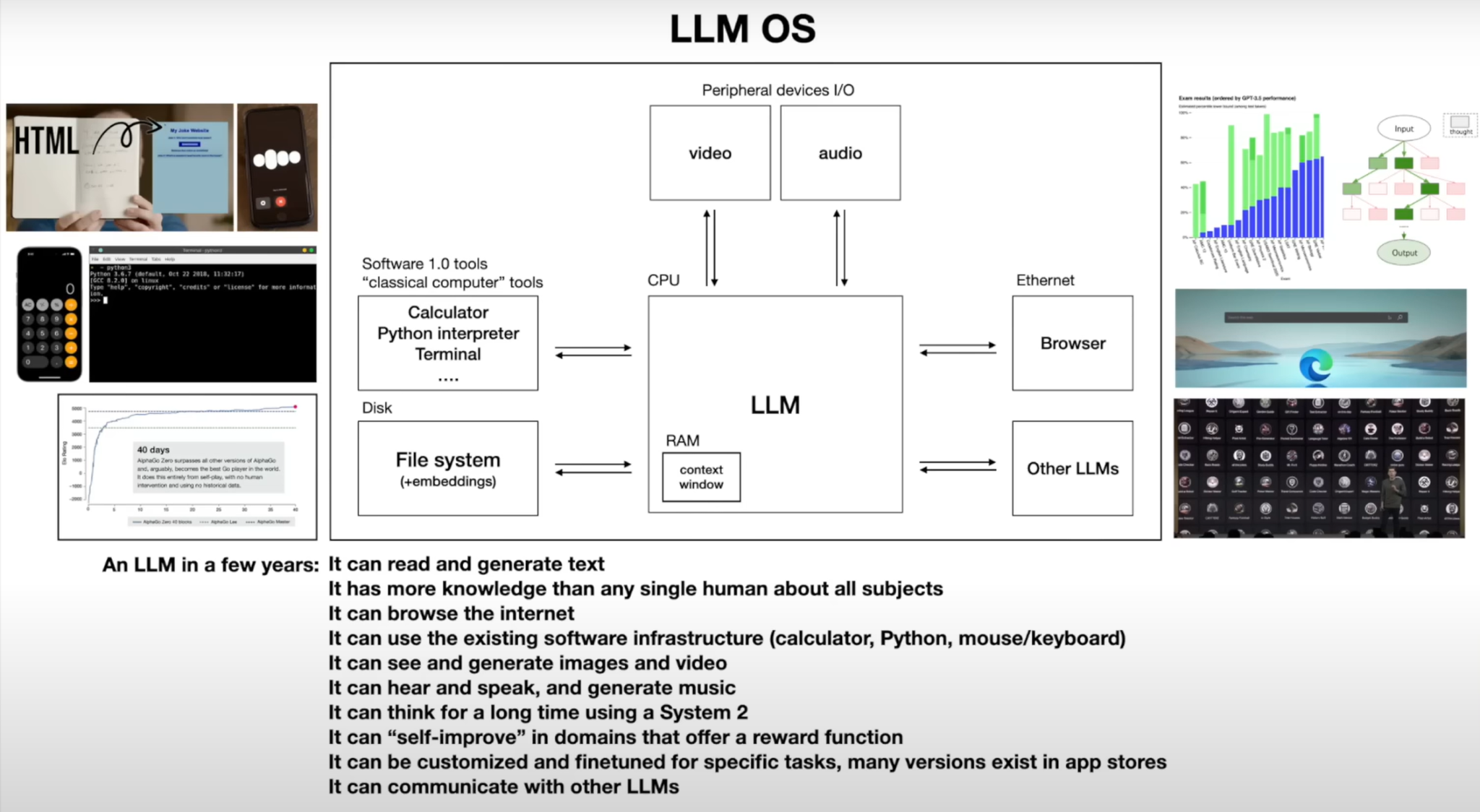
1. Grammar-constrained decoding for Structured NLP Tasks without fine-tuning
2. The Whole Truth and Nothing But the Truth: Faithful and Controllable Dialogue Response Generation with Dataflow Transduction and Constrained Decoding

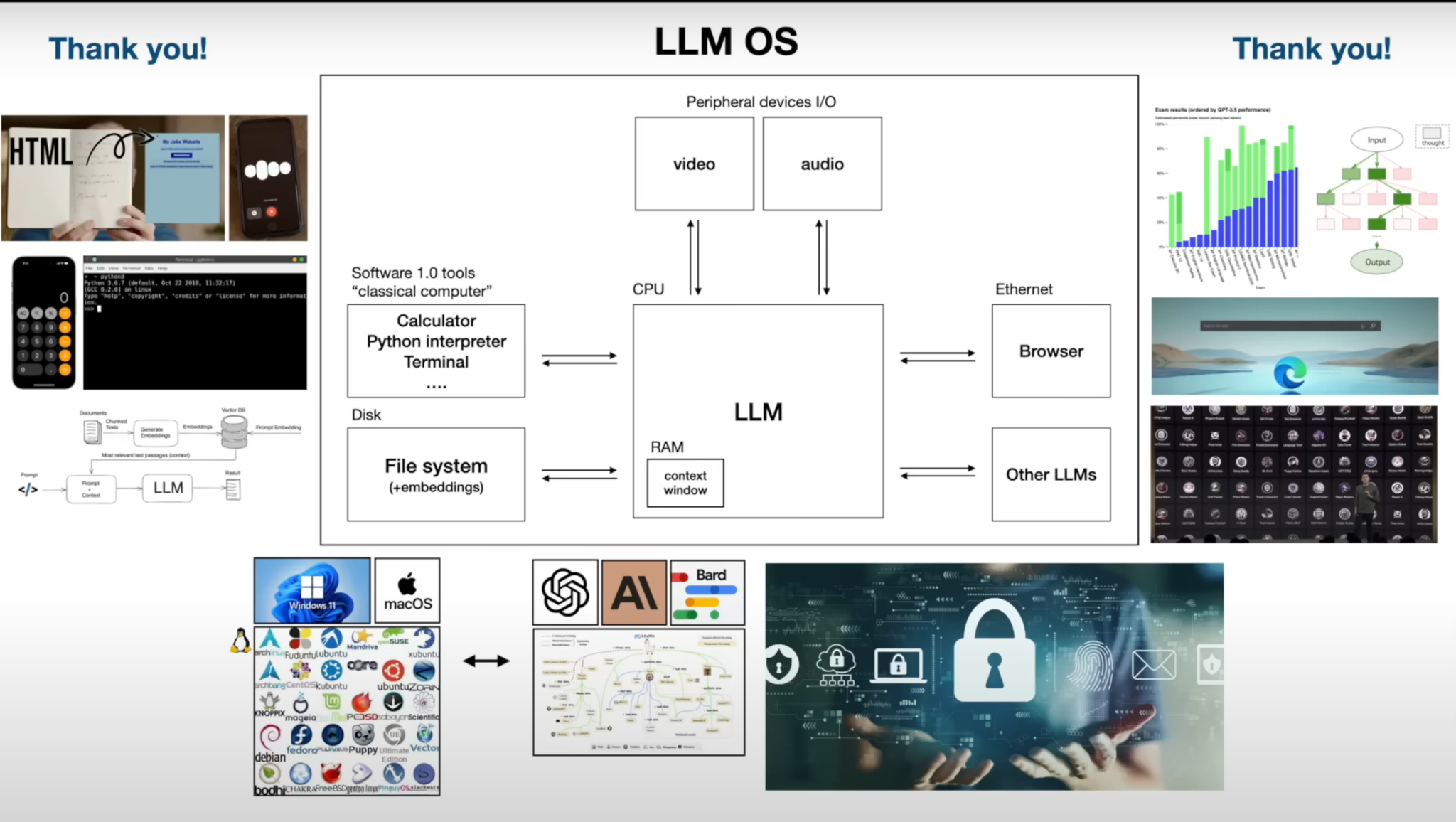
**First day of work on this project:**











My insight: This video was a short theory about LLM models. The presenter points to the key points of LLM.

The last part was exciting for me. It was about attacks in LLm models. I would like to know more about this.

**Task 2.** Read this paper [Grammar-Constrained Decoding for Structured NLP Tasks without Fine Tuning]

**Status:**

### **Introduction to the Paper**

The paper titled **"Grammar-Constrained Decoding for Structured NLP Tasks without Fine Tuning"** explores a method to improve how language models (like GPT-3) generate structured text.

At first, I was required to learn the main concept of this topic:

**What is Structured text?** is the text that follows a specific format or set of rules, such as filling out a form or generating code.

**Structured NLP Tasks:** These tasks require the output to follow a specific format.

**Ex of Structured NLP Tasks:**

* **Information Extraction:** Extracting specific information from text.
* **Entity Disambiguation:** Identifying the correct entity (like a person or place) mentioned in the text.
* **Constituency Parsing:** Analyzing the grammatical structure of a sentence.

**What is the Grammar-Constrained Decoding (GCD)** This method ensures that the text the AI generates follows a predefined structure or grammar. Think of it as giving the AI strict rules when generating text.

I need clear and understandable examples of grammar constraint decoding for better understanding.

**Ex of GCD:**

1. Closed Information Extraction (cIE) is a struchtured NLP task

Closed Information Extraction (cIE) involves extracting structured information, specifically subject-relation-object triplets, from unstructured text.

**Task:** Extract subject-relation-object triplets from the text.

**Input Text:** "Marie Curie discovered radium in 1898."

**Desired Output:** [subject: Marie Curie, relation: discovered, object: radium, year: 1898]

**Grammar Constraints:**

* Subjects must be valid entity names.
* Relations must be valid relation names.
* Objects must be valid entity names.
* Years must be valid years.

**Grammar Rules:**

* S -> [subject] [relation] [object] [year]
* [subject] -> "Marie Curie" | "Albert Einstein" | "Isaac Newton"
* [relation] -> "discovered" | "invented" | "proposed"
* [object] -> "radium" | "the theory of relativity" | "calculus"
* [year] -> "in 1898" | "in 1905" | "in 1687"

**Decoding Steps:**

1. **Generate Subject:**
   * Allowed Tokens: "Marie Curie", "Albert Einstein", "Isaac Newton"
   * Model Suggests: "Marie Curie"
   * Output: "Marie Curie"
2. **Generate Relation:**
   * Allowed Tokens: "discovered", "invented", "proposed"
   * Model Suggests: "discovered"
   * Output: "discovered"
3. **Generate Object:**
   * Allowed Tokens: "radium", "the theory of relativity", "calculus"
   * Model Suggests: "radium"
   * Output: "radium"
4. **Generate Year:**
   * Allowed Tokens: "in 1898", "in 1905", "in 1687"
   * Model Suggests: "in 1898"
   * Output: "in 1898"

**Final Output:**

* [subject: Marie Curie, relation: discovered, object: radium, year: 1898]

### **Why is GCD Important?**

* **Current Challenges:** LLMs are good at generating text but struggle with tasks requiring a specific structure without fine-tuning (additional training for specific tasks).
* **Solution with GCD:** Using formal grammar (rules), GCD can guide the AI to produce text that follows the required structure without needing extra training.

### **How does GCD work?**

1. **Formal Grammars:** These are sets of rules that define the output structure. For example, in information extraction, the output might need to be in the format "subject-verb-object" (e.g., "John-buys-apple").
2. **Input-Dependent Grammars:** These adjust the rules based on the input text, allowing the AI to generate different structures as needed.

### **What are the Benefits of GCD?**

* **Flexibility:** GCD can be used for various tasks without fine-tuning the AI for each task.
* **Performance:** In experiments, GCD-enhanced models performed better than those without constraints, even matching or surpassing some task-specific models.

#### **Example of GCD:**

#### **Problem:**

LLMs are very good at generating text but need additional training to generate text that follows specific structures or formats.

#### **Solution: Grammar-constrained decoding (GCD)**

In this paper, GCD is a method proposed to guide the text generation of LLMs by using formal grammar to ensure the output follows a required structure.

### **Main Contributions of the Paper:**

1. **Unified Framework:** The paper demonstrates that many structured NLP tasks can be framed as grammar-constrained decoding problems.
2. **Input-Dependent Grammars:** Introduces grammars that can adjust based on the input text, allowing more flexibility in generating different structures for different inputs.
3. **Empirical Demonstration:** Shows through experiments that GCD-enhanced LLMs perform significantly better in tasks like information extraction, entity disambiguation, and constituency parsing