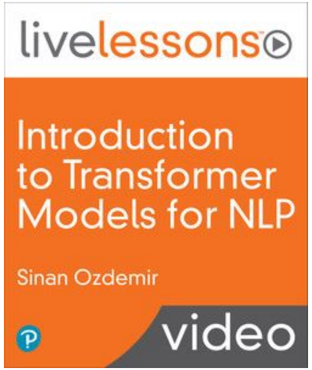
Large Language Models and ChatGPT in 3 Weeks

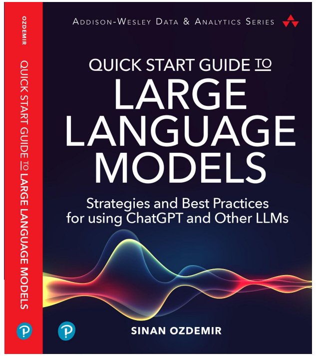
Week 2 - Getting Actionable Results and Cost Projecting with LLMs and GPT

Sinan Ozdemir

Data Scientist, Entrepreneur, Author, Lecturer

Welcome!





Pearson

My name is **Sinan Ozdemir** (in/sinan-ozdemir + @prof_oz)

- Current **founder** of Loop Genius (using AI to help entrepreneurs get their first 100 customers)
- Current **lecturer** for O'Reilly and Pearson
- Founder of Kylie.ai (Funded by OpenAI Founder + Acquired)
- **Masters** in Theoretical Math from **Johns Hopkins**
- Former lecturer of Data Science at Johns Hopkins

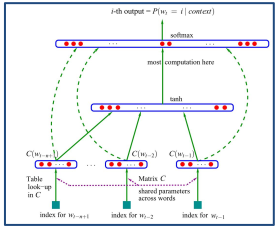
Author of ML textbooks and online series, including

- The Principles of Data Science
- Introduction to Transformer Models for NLP
- Quick Start Guide to LLMs (Top 10 in NLP on Amazon)

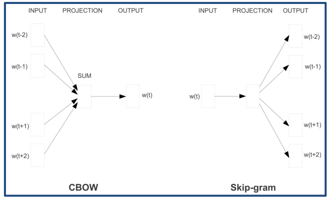
Quick Recap

Brief History of Modern NLP

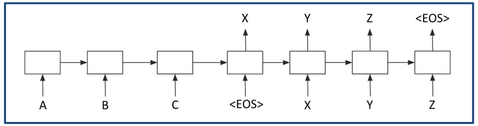
2001
Neural Language Models

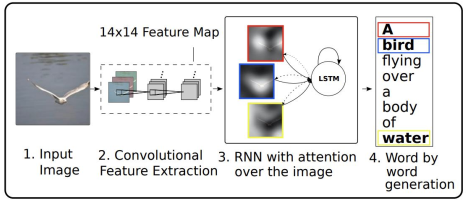


2013
encoding semantic meaning with Word2vec



2014 - 2017
Seq2seq + Attention





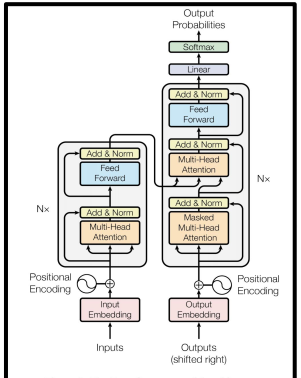


Figure 1: The Transformer - model architecture.

2017 - Present
Transformers + Large Language Models

Bengio et al. https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf
Mikolov et al. https://arxiv.org/abs/1301.3781
Xu et al. http://proceedings.mlr.press/v37/xuc15.pdf
https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Auto-__ Language Model Use Cases

Auto-regressive Models

1. Predicting next word in a sentence (auto-complete)
2. Natural Language Generation (NLG)
3. GPT Family

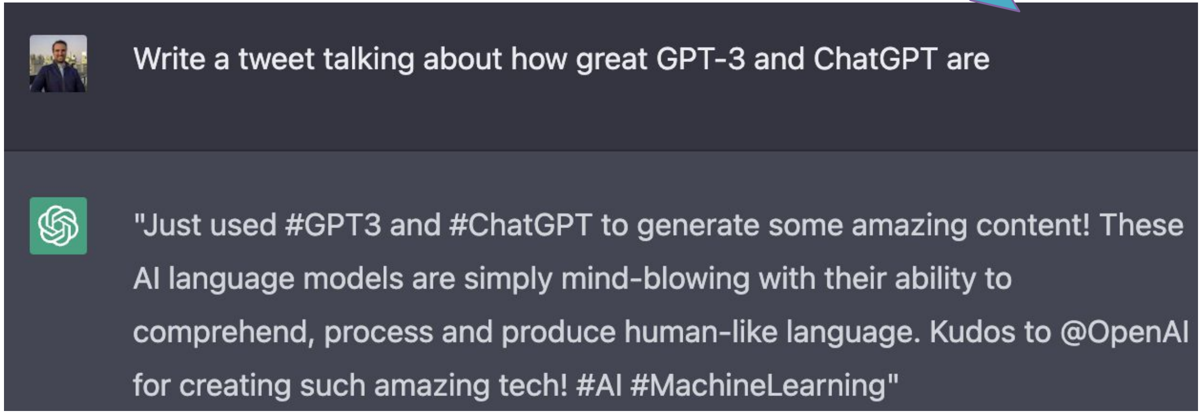
Auto-encoding Models

1. Comprehensive understanding and encoding of entire sequences of tokens
2. Natural Language Understanding (NLU)
3. BERT Family

Using the ChatGPT Playground

Write an instruction to the LLM, and see the response

Prompt



LLM Response

Source: ChatGPT Playground

Using LLMs

We can use LLMs in (generally) three ways:

1. **Encode** text into semantic vectors with little/no fine-tuning
   a. Eg. Creating an information retrieval system using BERT vectors

2. Fine-tune a pre-trained LLM to perform a very specific task using **Transfer Learning**
   a. Eg. Fine-tuning BERT to classify sequences with labels

3. Ask an LLM to solve a task it was pre-trained to solve or could intuit
   a. Eg. **Prompting** GPT3 to write a blog post
   b. Eg. **Prompting** T5 to perform language translation

Does the LLM know enough for my task?

A. **Yes**, it has all knowledge encoded and it is ready to solve my task
   a. May still need to format output to make it easier to work with

B. **Mostly.** It knows the information but it lacks critical information (information is too new to be in the model or it knows a topic but not to the specifics that I need)
   a. Create a secondary system to retrieve information on demand
   b. Few-shots and chain of thought to help teach nuances/specifics

C. **No,** not at all, I need to teach it pretty much everything from scratch
   a. Just ask with explicit instructions
   b. Few shot / chain of thought prompting
   c. Fine-tuning for long term cost savings/speed

Identifying Patterns and Making Predictions

E.G., Classification

Still the most common ML application, **classification** assigns labels to a piece of data/text from a *finite set of labels*.

This generally requires fine-tuning to teach an LLM about the classes it needs to predict.

Examples: Assigning a news topic to an article, assigned parts of speech to words in a sentence

Classification with LLMs

We can perform classifications in many ways:

1. Prompting an LLM with a great set of instructions on how to classify
2. Using few shot to teach the model via *in-context* learning
3. Fine-tuning an LLM with labeled data (like we would before LLMs)
4. Using *0-shot classification*

0-shot Classification

A variation of classical classification, **0-shot classification** is able to dynamically take in labels without fine-tuning and assign labels.

Models like BART (a variant of BERT from Meta) are great at this

0-shot Classification

Zero-Shot Classification

Examples

I have a problem with my iphone that needs to be resolved asap!!

Possible class names (comma-separated)

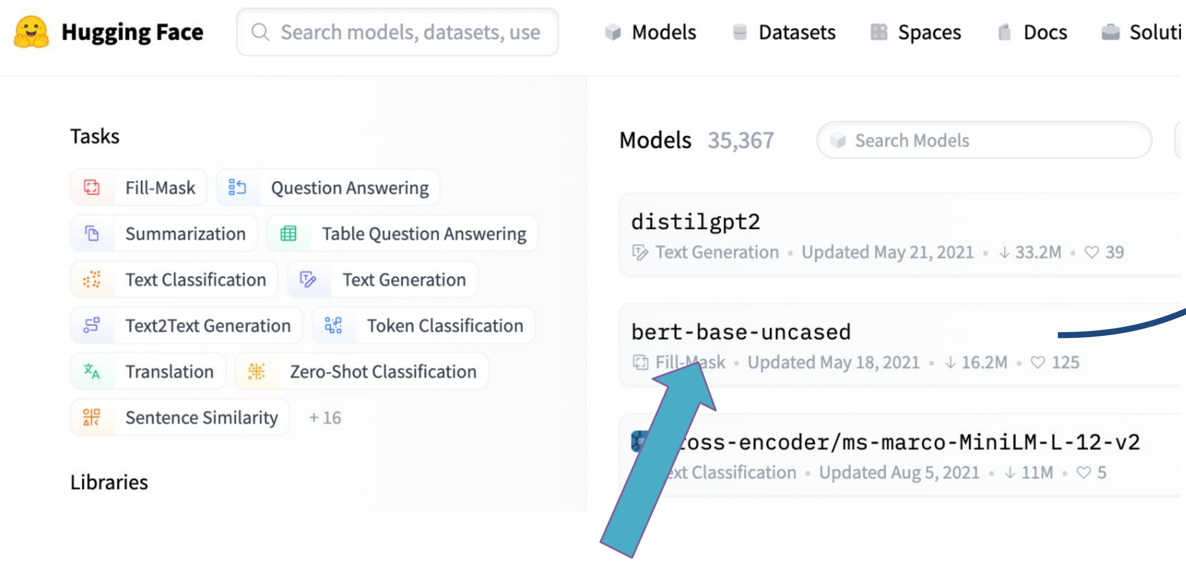
urgent, not urgent, phone, tablet, computer

urgent	0.999
phone	0.995
computer	0.095
not urgent	0.001
tablet	0.000

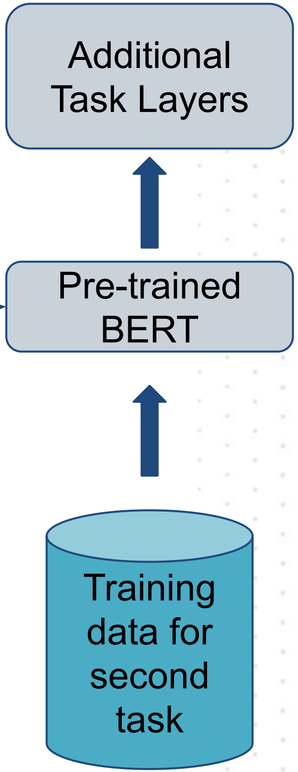
facebook/**bart-large-mnli**

like 586

Transfer Learning with BERT

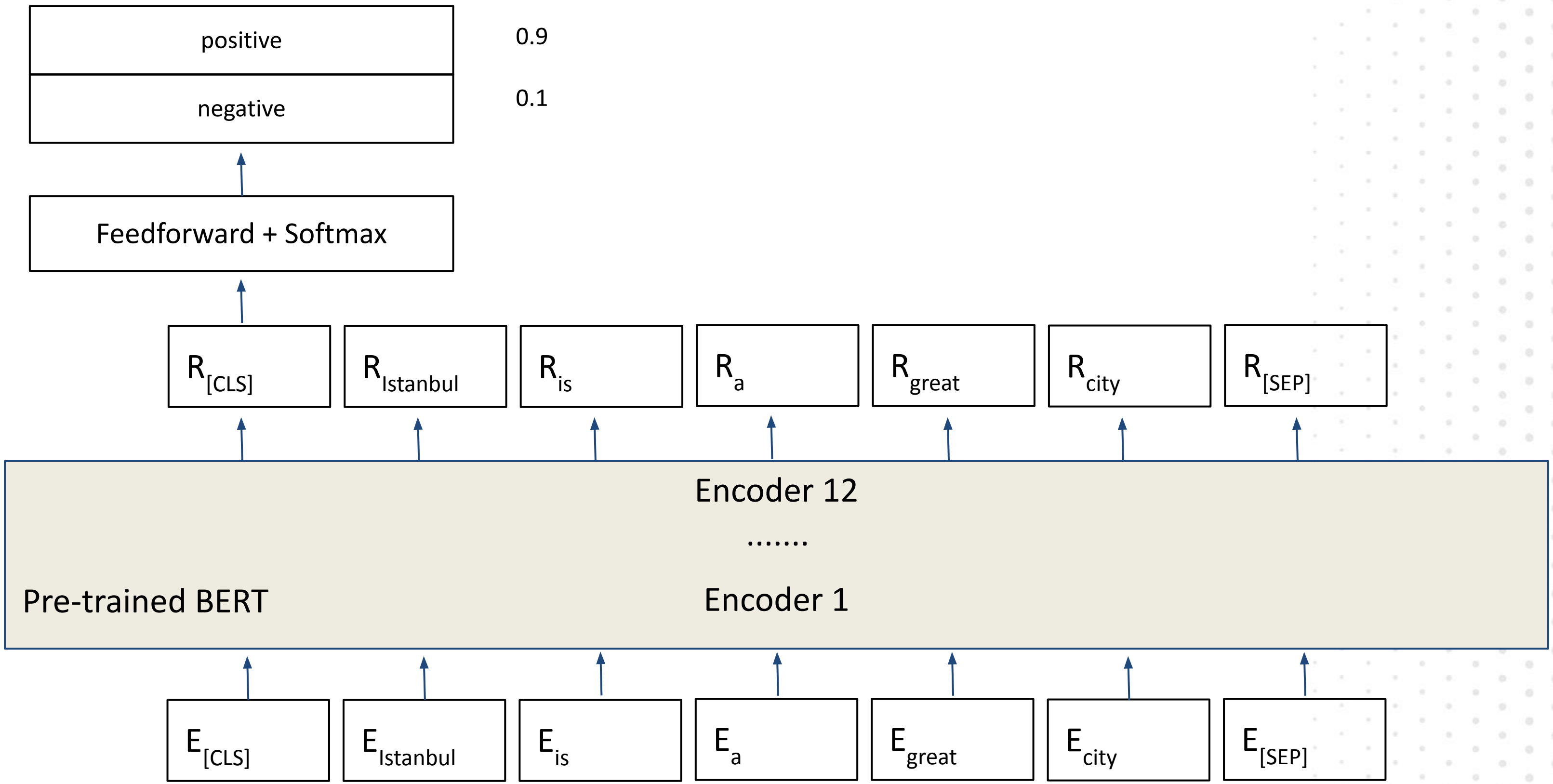


Selecting a source model

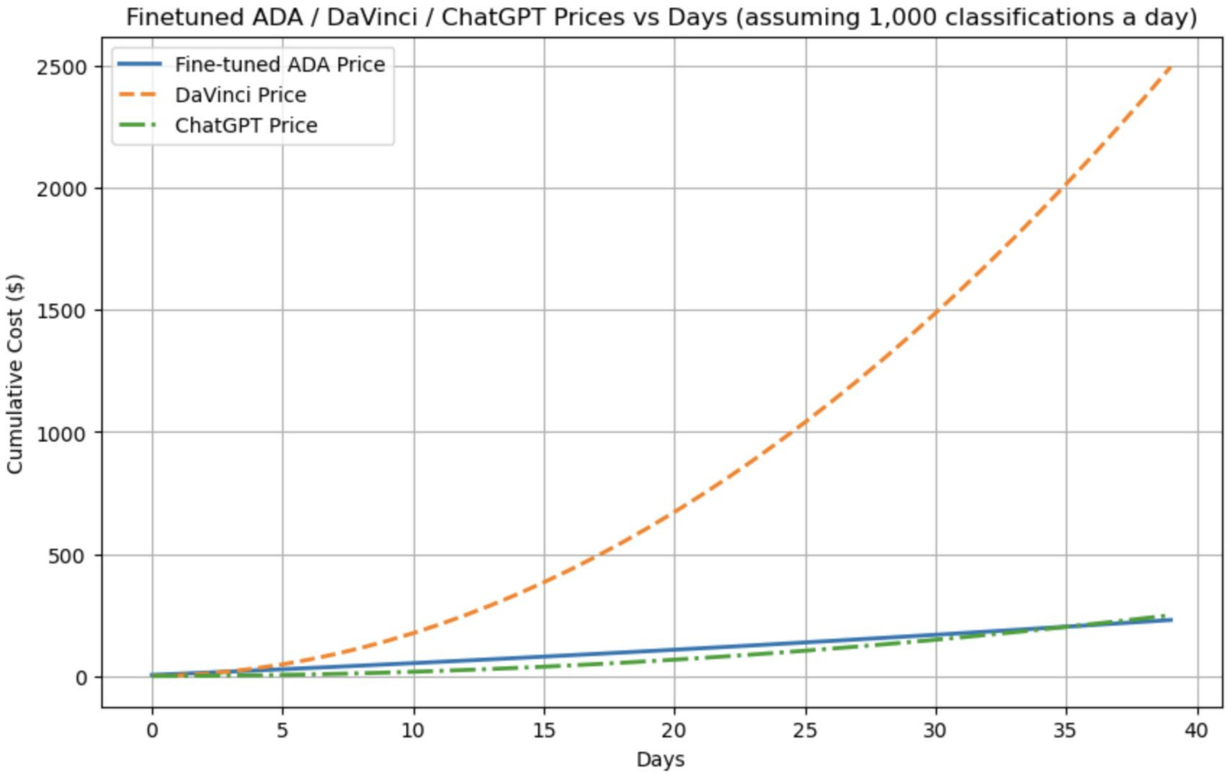


Reusing and training model

Fine-tuning with BERT - Advanced



Fine-tuning OpenAI models

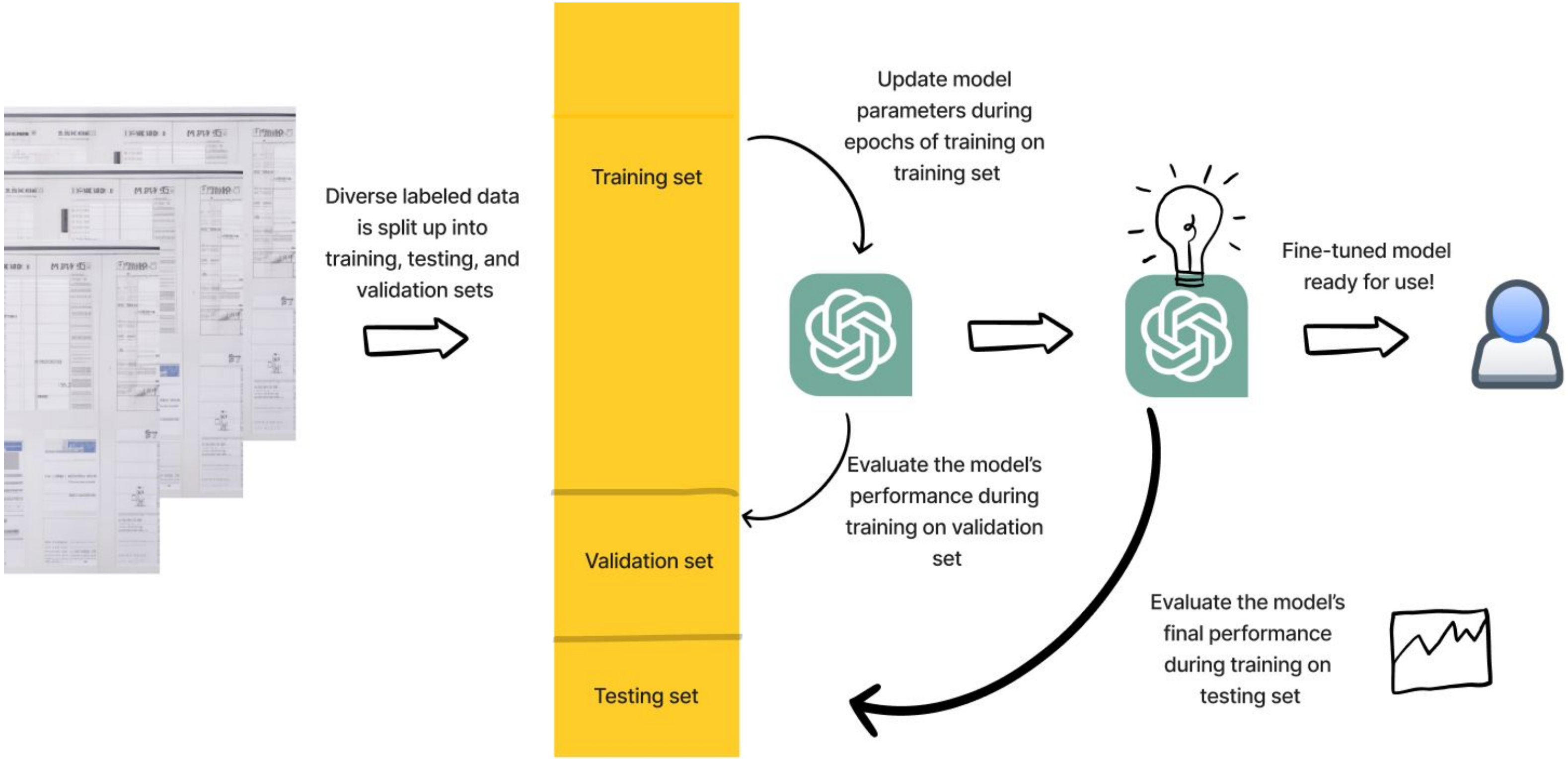


Assuming only 1,000 classifications a day and a relatively liberal prompt ratio (150 tokens (for few-shot examples, instructions, etc) for DaVinci or ChatGPT for every 40 tokens), **the cost of a fine-tuned model almost always wins the day overall cost-wise**

Pearson

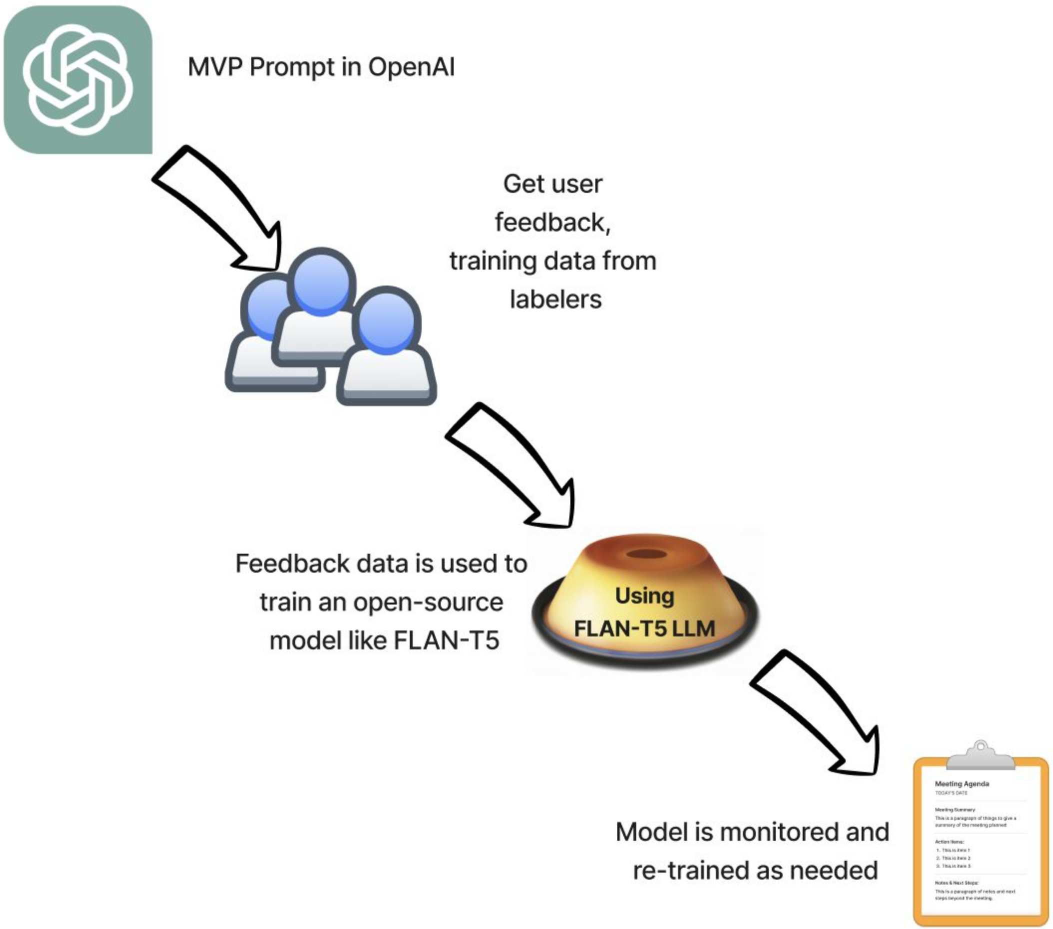
Source: Quick Start Guide to LLMs by Sinan Ozdemir

Fine-tuning LLMs (e.g., OpenAI’s Babbage)



Source: Quick Start Guide to LLMs by Sinan Ozdemir

Moving from closed to open source

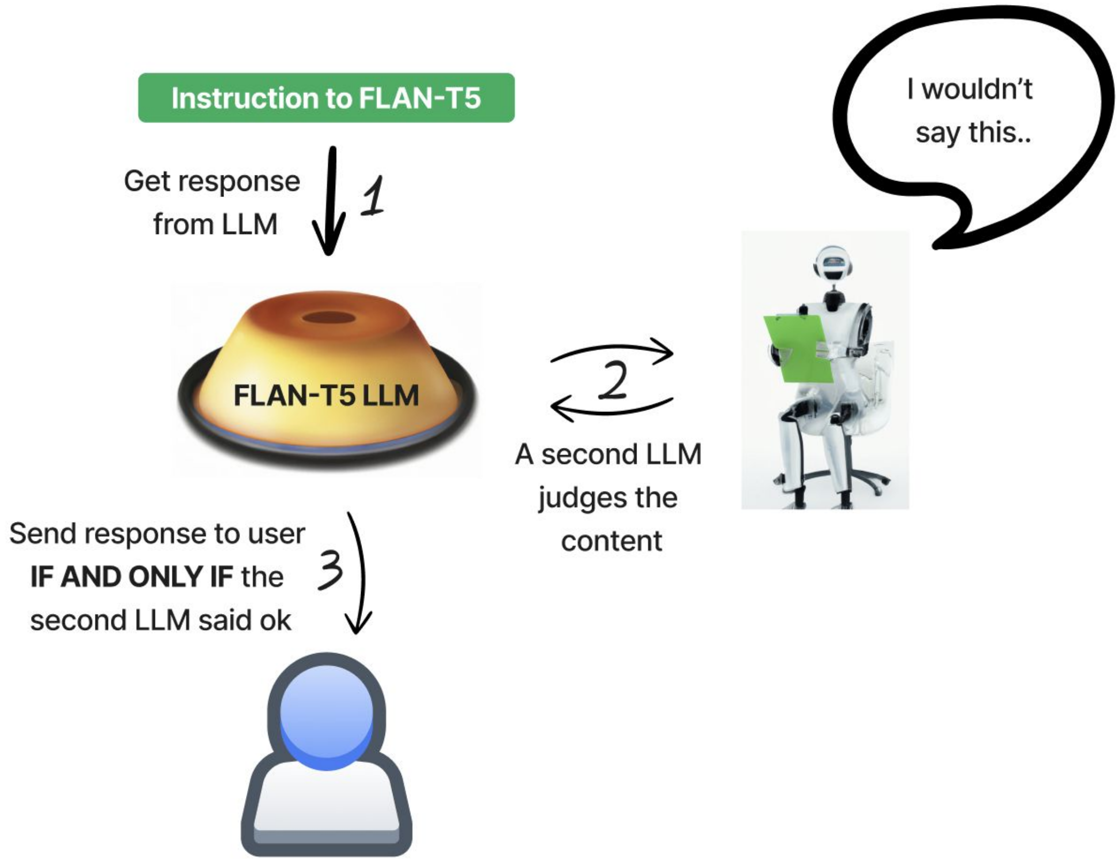


Reasoning vs Thinking

- Think of LLMs as “**reasoning machines**” vs “thinking machines”.
- LLMs excel at tasks that require **reasoning** - using context and input information in conjunction to produce a nuanced answer

Pearson

Output Validation



Output Structuring

Using an LLM's ability to reason comes in handy when we want it to output something in a more usable format.

Developers often prefer a *JSON* formatted output and we can ask an LLM to output things in a format if we ask it to or provide few-shot examples.

USER

Write a haiku as a JSON like this:

{"haiku": "(the haiku goes here)"}

ASSISTANT

```
{
  "haiku": "Autumn leaves falling\nWhispering secrets to earth\nNature's poetry"
}
```

Pearson

Batch Prompting to save on latency/cost

Standard Prompting

```
# K-shot in-context exemplars
Q: {question}
A: {answer}

Q: {question}
A: {answer}

...
# One sample to inference
Q: Ali had $21. Leila gave him half of her
   $100. How much does Ali have now?
------------------------------------------------
# Response
A: Leila gave 100/2=50 to Ali. Ali now has
   $21+$50 = $71. The answer is 71.
```

Batch Prompting

```
# K-shot in-context exemplars in K/b batches
Q[1]: {question}
Q[2]: {question}      b(=2) samples
A[1]: {answer}        in one batch
A[2]: {answer}

...
# b samples in a batch to inference
Q[1]: Ali had $21. Leila gave him half of her
      $100. How much does Ali have now?
Q[2]: A robe takes 2 bolts of blue fiber and
      half that white fiber. How many bolts?
------------------------------------------------
# Responses to a batch
A[1]: Leila gave 100/2=50 to Ali. Ali now has
      $21+$50 = $71. The answer is 71.
A[2]: It takes 2/2=1 bolt of white fiber. The
      total amount is 2+1=3. The answer is 3.
```


Source: https://arxiv.org/pdf/2301.08721v1.pdf

Code Time!

Cost Projecting with LLMs and GPT

Pricing with LLMs

Models like OpenAI GPT-3.5 Turbo (ChatGPT) charge per tokens inputted and tokens outputted.

Fine-tuned Classifiers have a cost for fine-tuning, inference (using them) and updating them

E.g., Hosting on HuggingFace for a small production-ready classifier starts at **$45/month**

OpenAI Pricing

GPT 3.5 Turbo (ChatGPT) (4K context window)

Model	Input	Output
gpt-3.5-turbo-1106	$0.0010 / 1K tokens	$0.0020 / 1K tokens
gpt-3.5-turbo-instruct	$0.0015 / 1K tokens	$0.0020 / 1K tokens

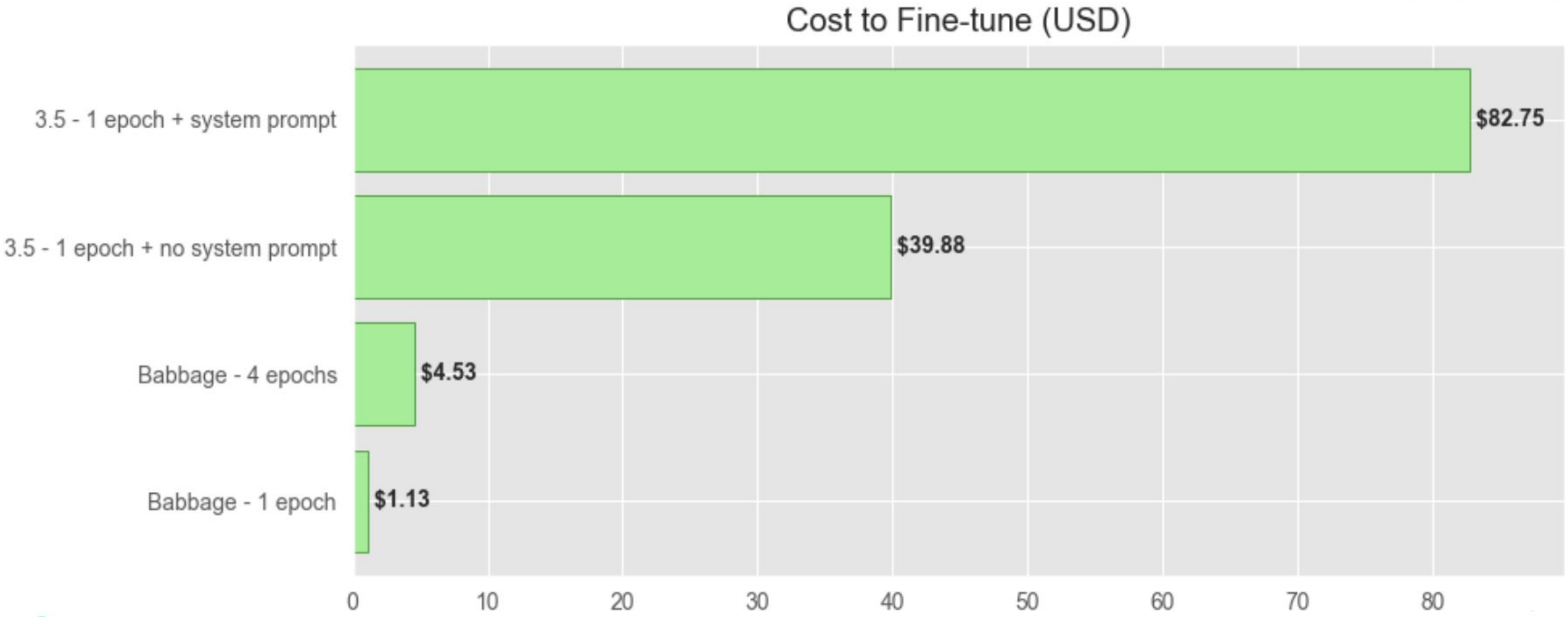
Fine-tuned Models

Model	Training	Input usage	Output usage
gpt-3.5-turbo	$0.0080 / 1K tokens	$0.0030 / 1K tokens	$0.0060 / 1K tokens
davinci-002	$0.0060 / 1K tokens	$0.0120 / 1K tokens	$0.0120 / 1K tokens
babbage-002	$0.0004 / 1K tokens	$0.0016 / 1K tokens	$0.0016 / 1K tokens

Source: https://openai.com/pricing

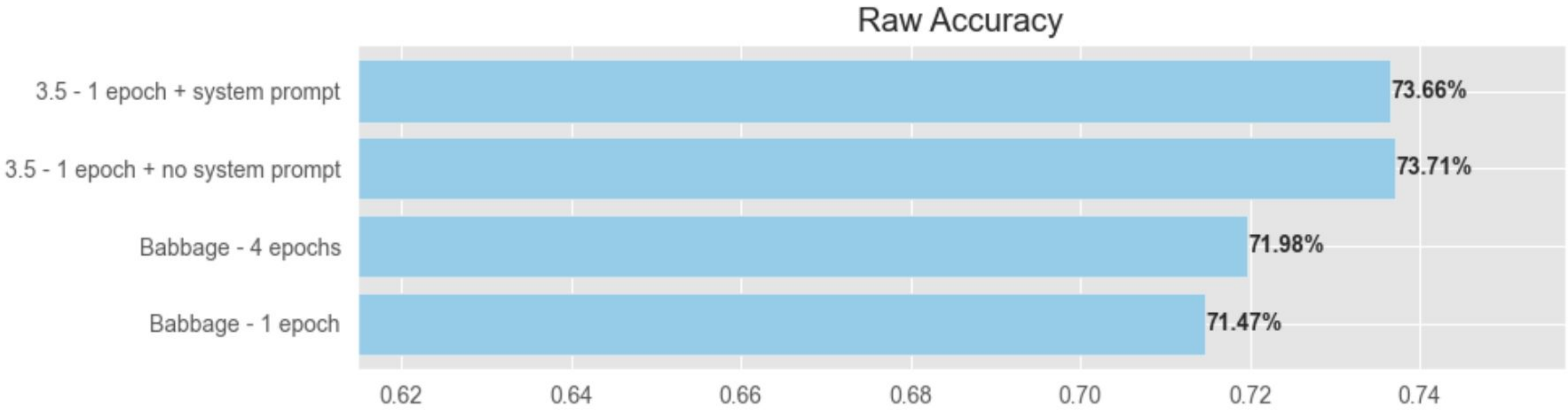
Cost of fine-tuning models

Classification with ~170K training examples



Cost of fine-tuning models

Accuracy bump probably not worth the cost



Cost Considerations for LLM applications

Open Source:

- Data Collection
  - Labelling, etc
- Fine-tuning costs
  - Machines, etc
- Model Serving
  - Machines, etc
- Maintenance
  - Future fine-tuning, etc

Closed Source

- Number of tokens

Open vs Closed Source

Open Source:

- Pricing is under your control and generally cheaper

- Models are narrower but often more performant

Closed Source

- Easier to use, no need to think of hosting

- Often more expensive in the long term

Open vs Closed Source (deeper)

Open Source:

- Data privacy / security is controllable with on-premises systems

Closed Source

- Companies control what parameters you can use (e.g. top-k is unavailable with OpenAI as are probabilities for tokens)

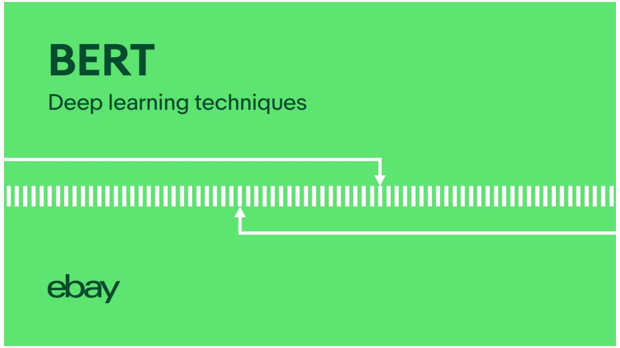
Code Time!

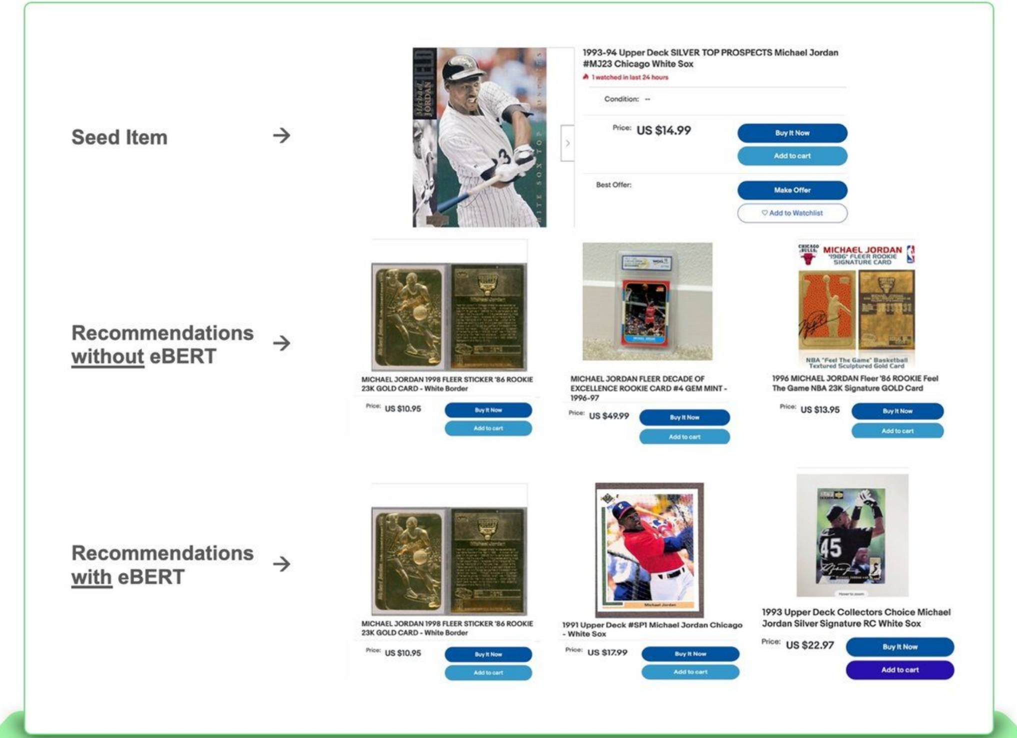
Pearson

Use Case Discussion

Encoding Ebay's Recommendations with BERT

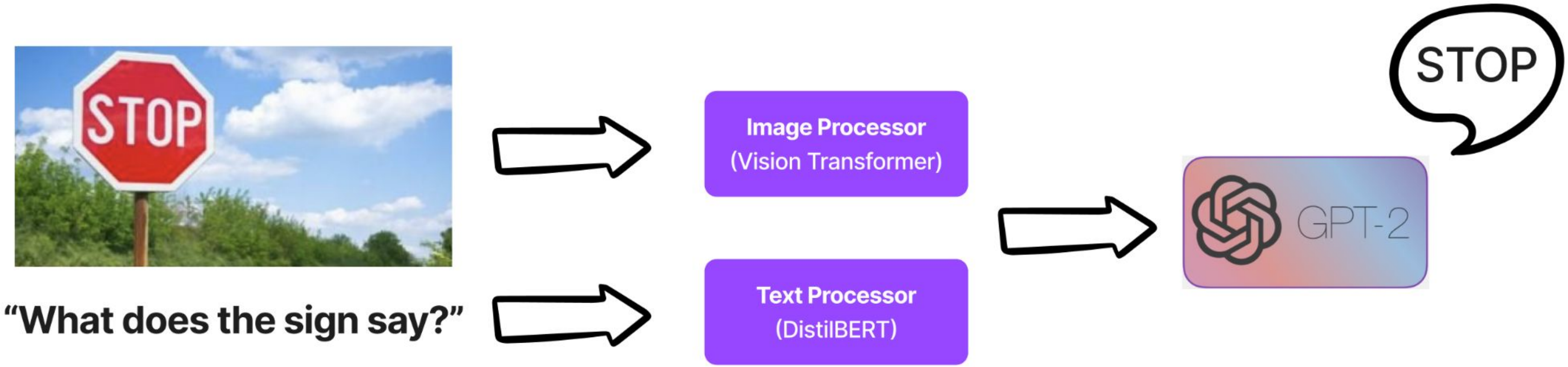
Ebay uses BERT to generate more relevant recommendations than traditional search techniques





Source:
https://tech.ebayinc.com/engineering/how-ebay-created-a-language-model-with-three-billion-item-titles

Visual Q/A with open source models



Source: [Quick start guide to LLMs]

Visual Q/A as a service

Groundlight

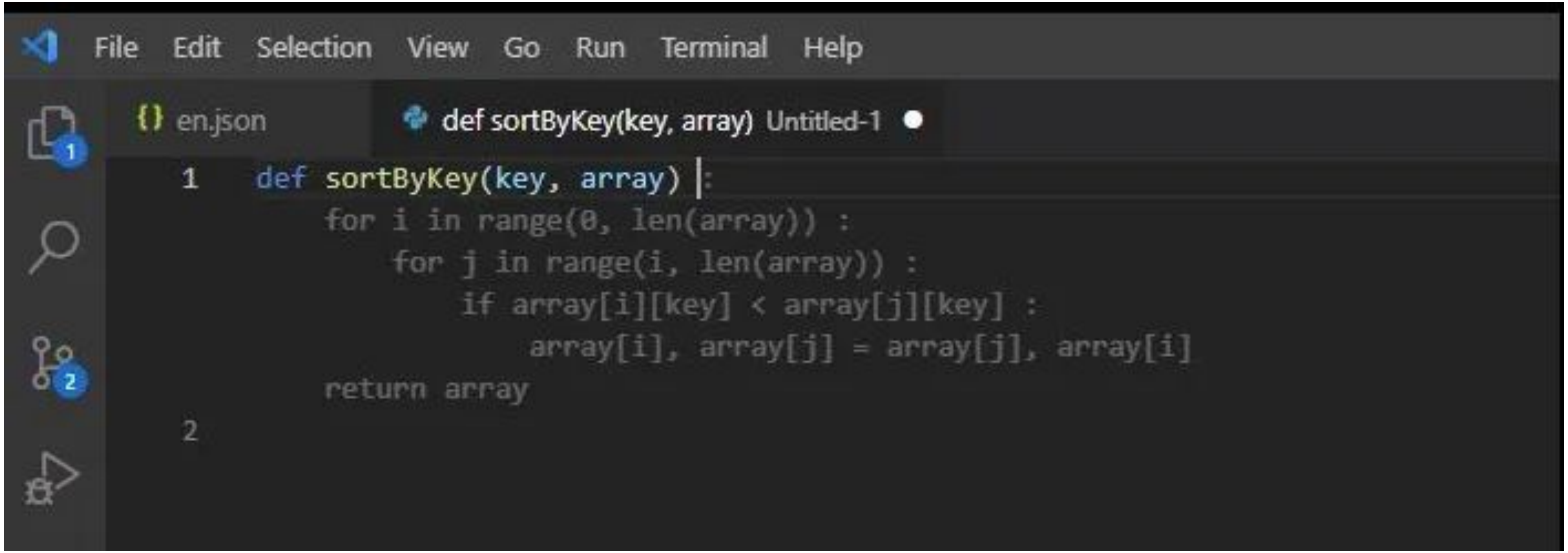
Product Team Careers Sign Up Sign In

FIND US ON GITHUB

Computer Vision
Without a Dataset
From One Line of Code

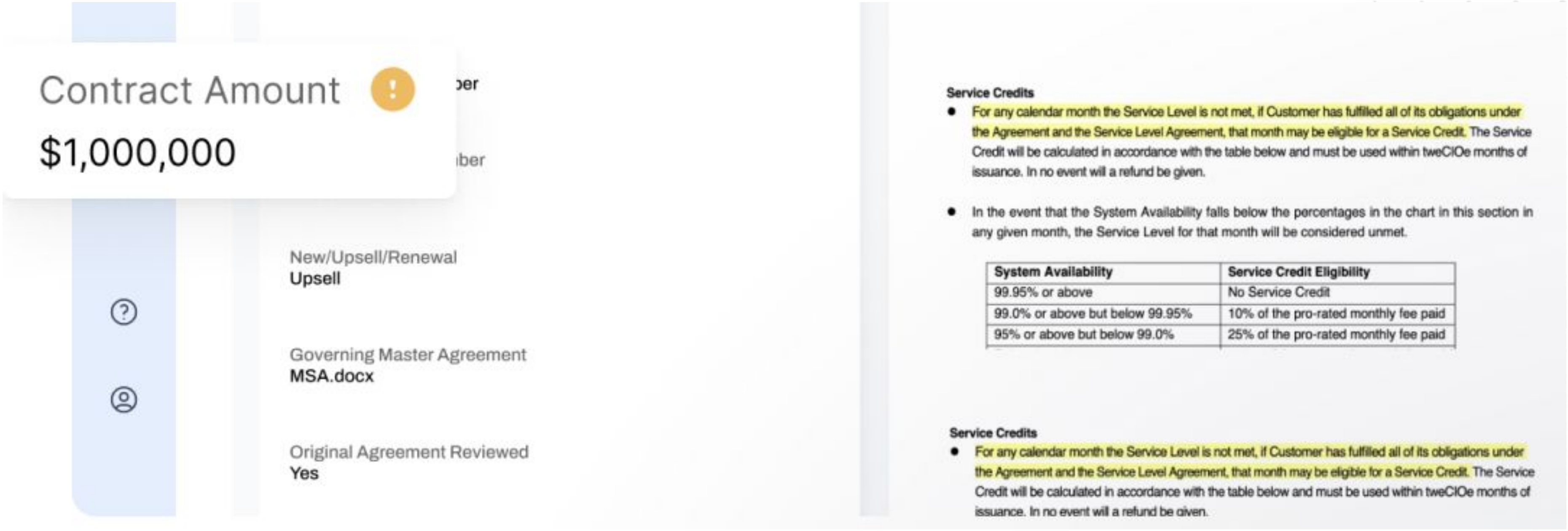
Natural language-based

Code Completion - Github’s Copilot



Source: Github Copilot

Legal - Klarity AI



Source: Github Copilot

Code Time!

Pearson

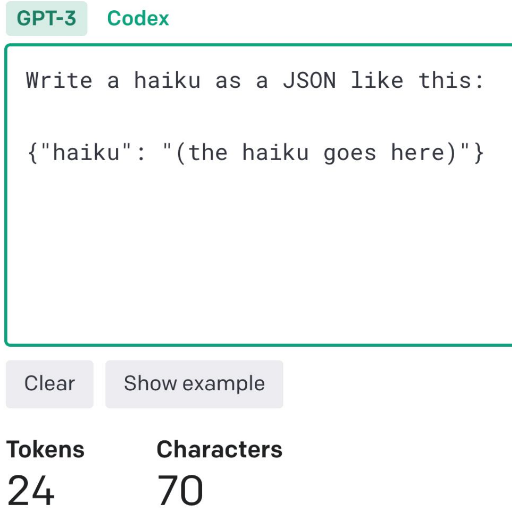
Week 2 Assignment

For non-coders

Estimate how much it will cost to run a single instance of your task on the model (assuming you will use OpenAI 3.5 or 4)

1. Think about how many tokens you are inputting and outputting
   a. Use [OpenAI's tokenizer] to be exact
2. Get a range of cost from what you might expect to be a shorter input vs a longer input (short vs long news article for a summarizer)

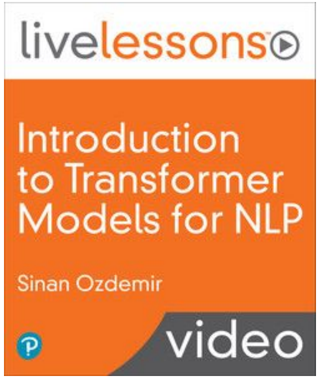
Bonus points if you do this for another closed source model



For coders

Do the assignment for non-coders **AND** write a python function that performs the task and uses your MVP prompt. Your function will likely have at least one non-optional parameter that is your input and have hard-coded context (likely going in your system prompt). Your function will likely help you with the non-coder assignment

Summary + Next Steps



A comprehensive introduction to LLMs + Transformers

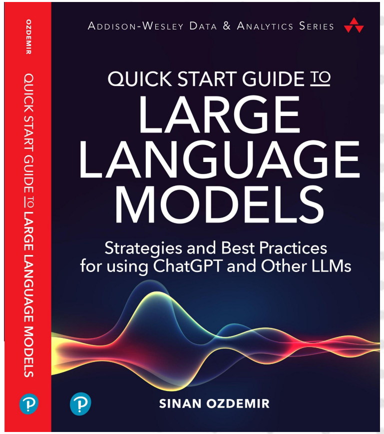
https://learning.oreilly.com/videos/introduction-to-transformer/9780137923717

Check out my live trainings for more in depth content!

https://learning.oreilly.com/search/?q=Sinan%20Ozdemir&type=live-event-series

New quick start guide to LLMs!

Quick Start Guide to Large Language Models



Pearson

Large Language Models and ChatGPT in 3 Weeks

Week 3 - Building Viable Prototypes with LLMs and GPT

Sinan Ozdemir
Data Scientist, Entrepreneur, Author, Lecturer

Pearson

Large Language Models and ChatGPT in 3 Weeks

See you next week!

Sinan Ozdemir
Data Scientist, Entrepreneur,
Author, Lecturer