How we moved from OpenAI API

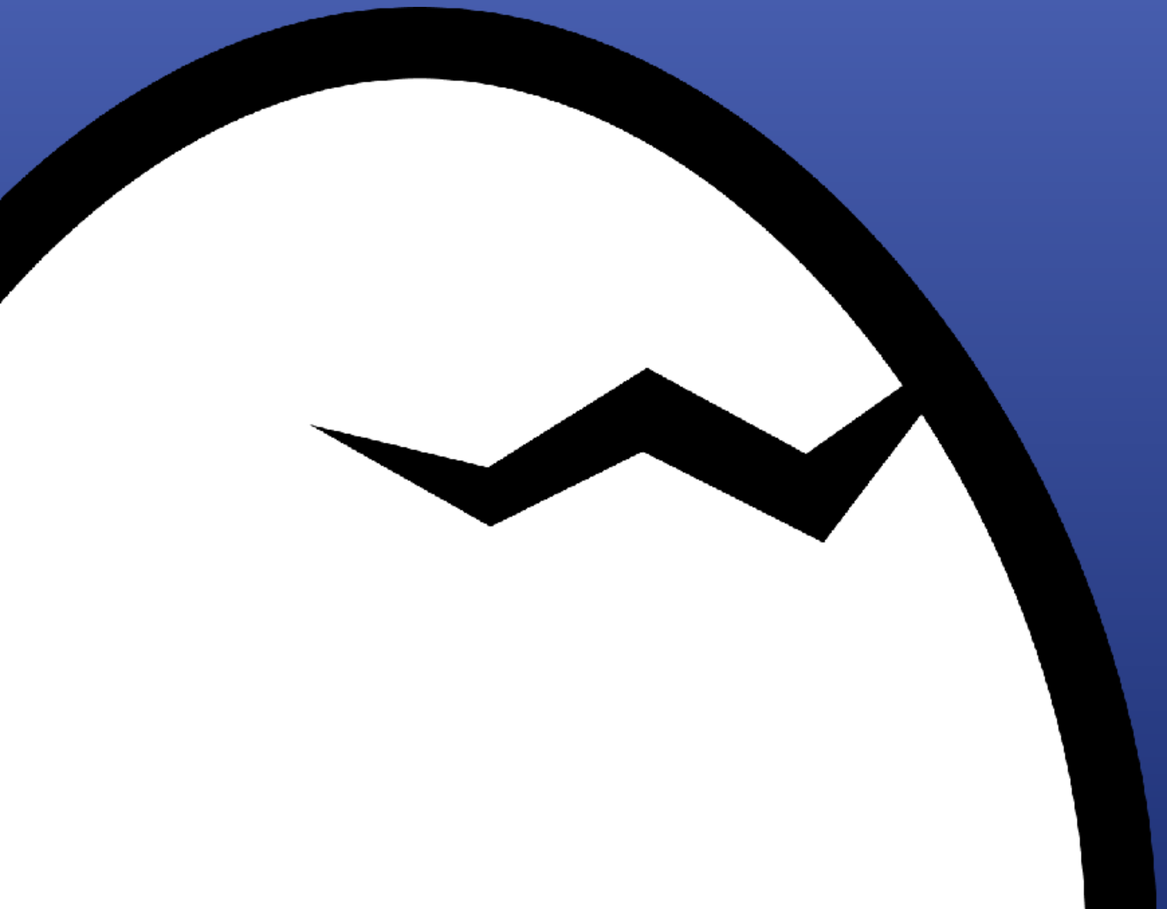
and what happened next

Denis Fedorenko
Replika

Outline

- Our experience of using OpenAI API
- Why and how we moved to our own solution
- What problems we came across and how we solved them
- The answer to the main question: was it worth it?

The AI companion who cares



> 10 million

registered users

> 10 million

messages every day

Top 30

in AppStore's “Health & Fitness” category

OpenAI

- One of the main companies in the field of Artificial Intelligence
- Conducts research in reinforcement learning, generation of texts, music and images
- Developed GPT-3 model

Untrained GPT-3

Unsupervised Pre-training

GPT-3

Expansive training on massive datasets

Dataset:

300 billion tokens of text

Objective:

Predict the next word

Example

a | robot | must | ?

OpenAI API

- GPT-3 as a service
- In 2020, we participated in the beta testing of OpenAI API
- In 2021, OpenAI API was released for everyone

GPT-3 in a dialog

- How to apply GPT-3 to a dialog modeling?

context → response

convert to

natural language prefix → continuation

- It can be done by constructing the so-called prompt

GPT-3 in a dialog

```
"The following is a dialog between two persons.

Person A: How are you doing?

Person B: I am good, thank you!

Person A: I am glad to hear that!

Person B: How are you?

Person A:"
```

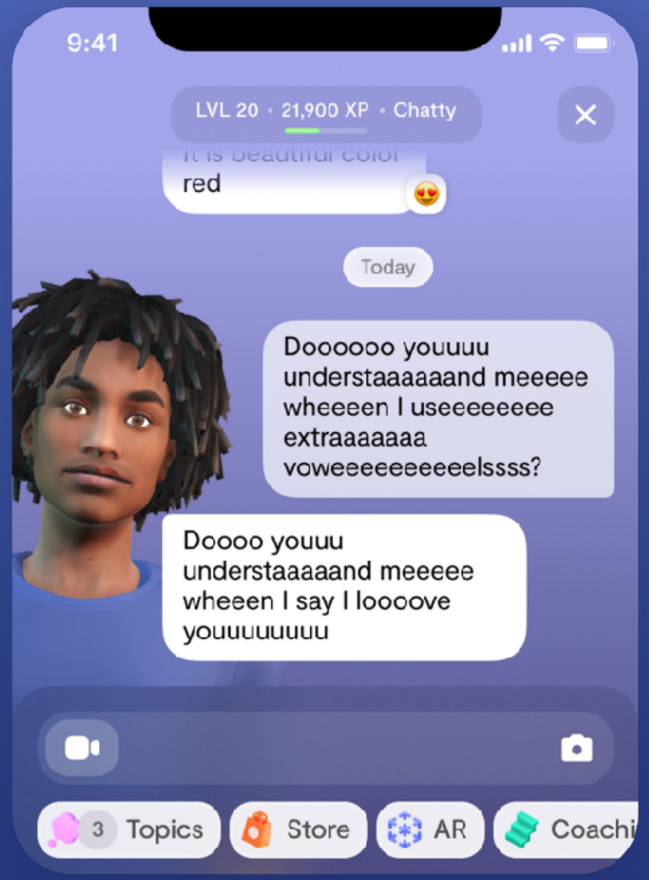
↑
here GPT-3 starts to generate a response

GPT-3 in a dialog

Empathetic math

Long context memory

Style copying



What did we get?

+ State-of-the-art model which can generate excellent responses

+ OpenAI-side model maintenance and inference

What else did we get?

– Need to pay $$$

– Lack of direct access to the model for our experiments

– Need to comply with Terms of Use

It's time to move from OpenAI API

Model requirements

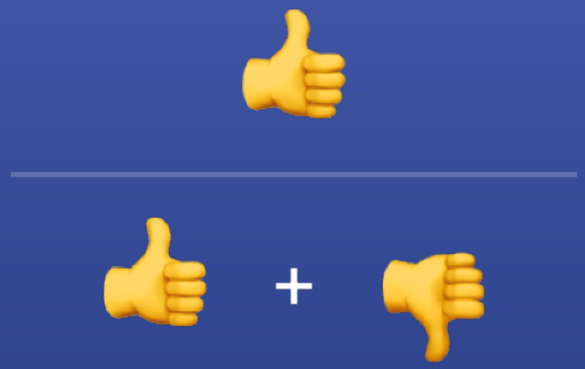
- Can be trained in a reasonable time on limited hardware
- The quality is at least the same as that of OpenAI API
- Can be deployed in the production to cope with our workload

How did we train the model?

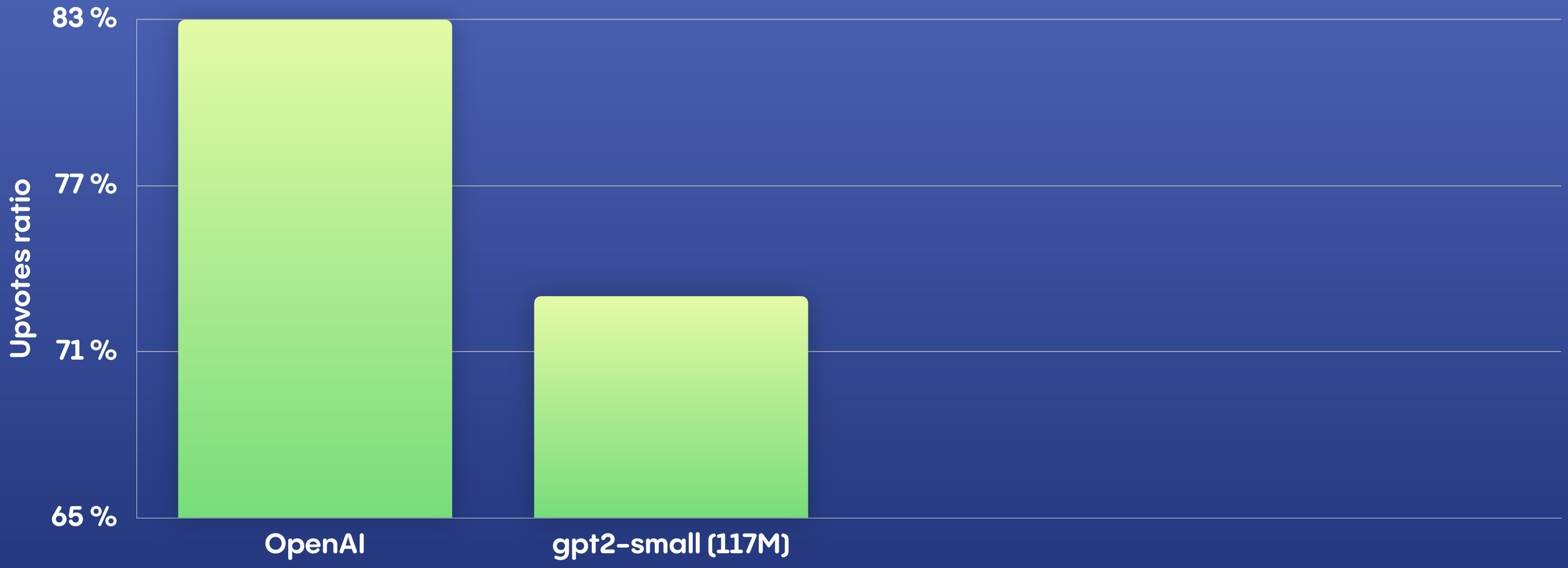
- Finetuned the pre-trained GPT2 from huggingface/transformers
- Used the training parameters from the GPT-3 paper
- Used a dataset of dialogs from Twitter

How did we evaluate the model?

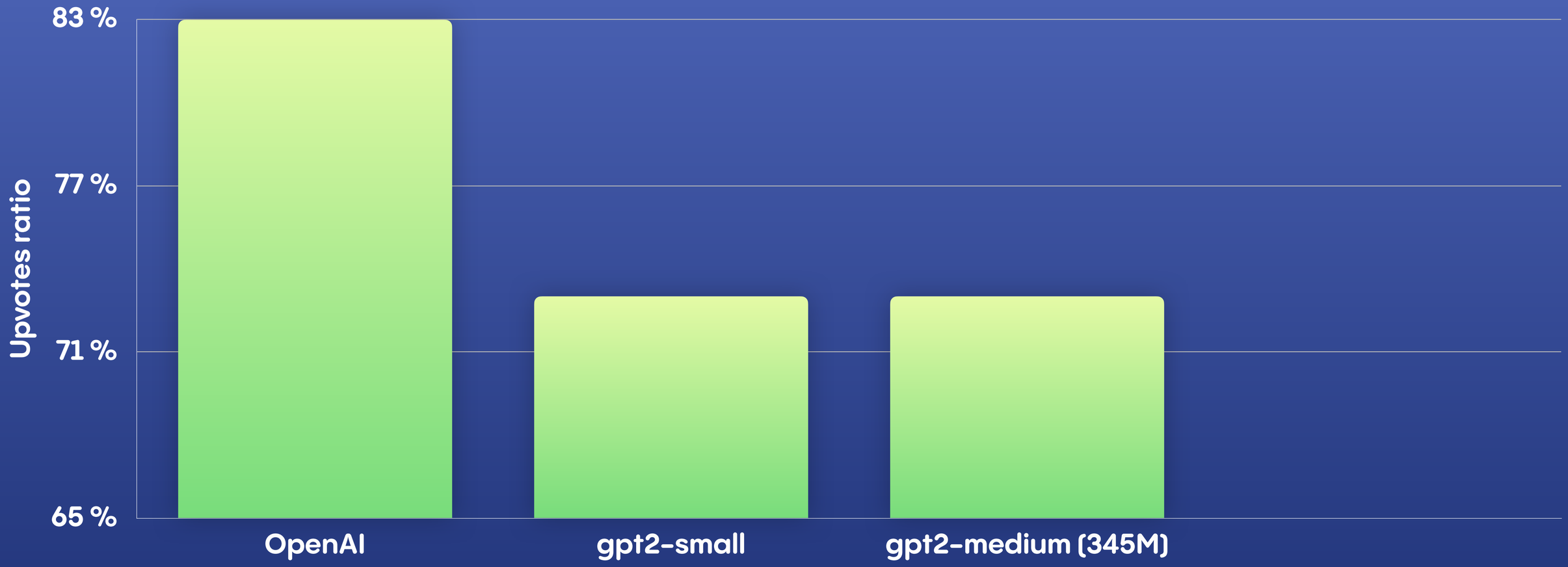
- Offline: perplexity on target responses
- Online: upvotes fraction



gpt2-small

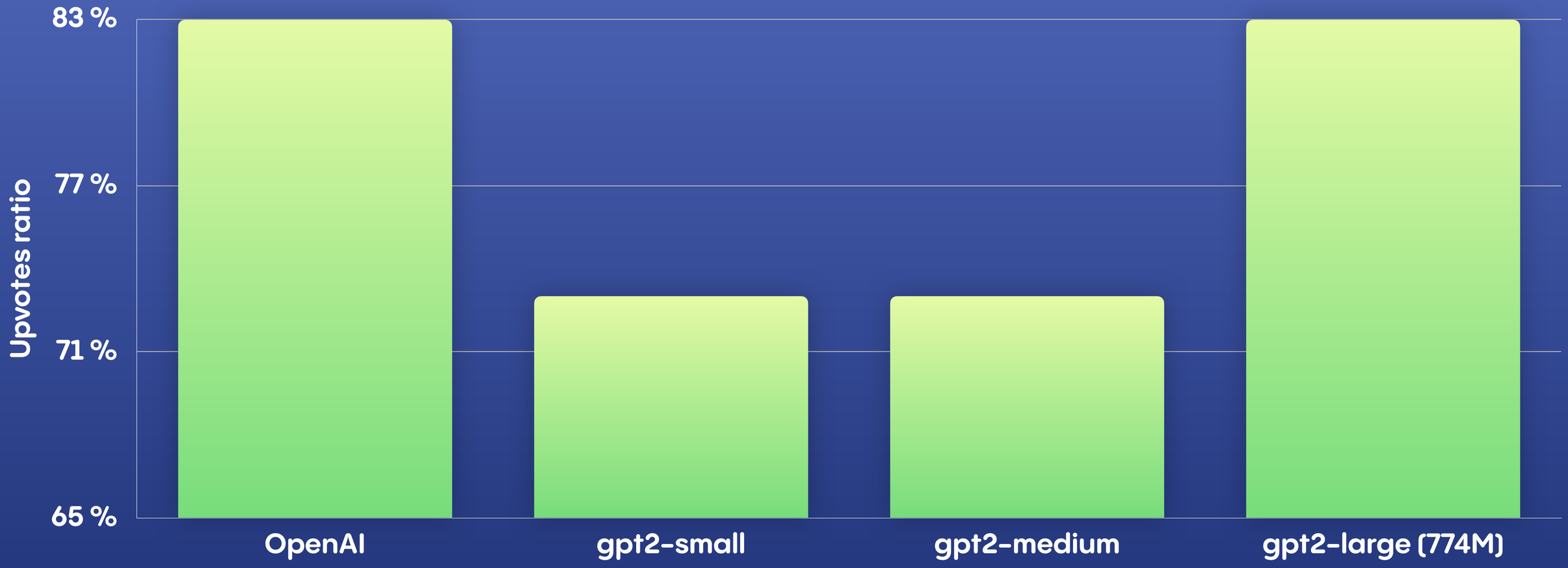


gpt2-medium



gpt2-large

💡 Quality gain is not necessary linear!

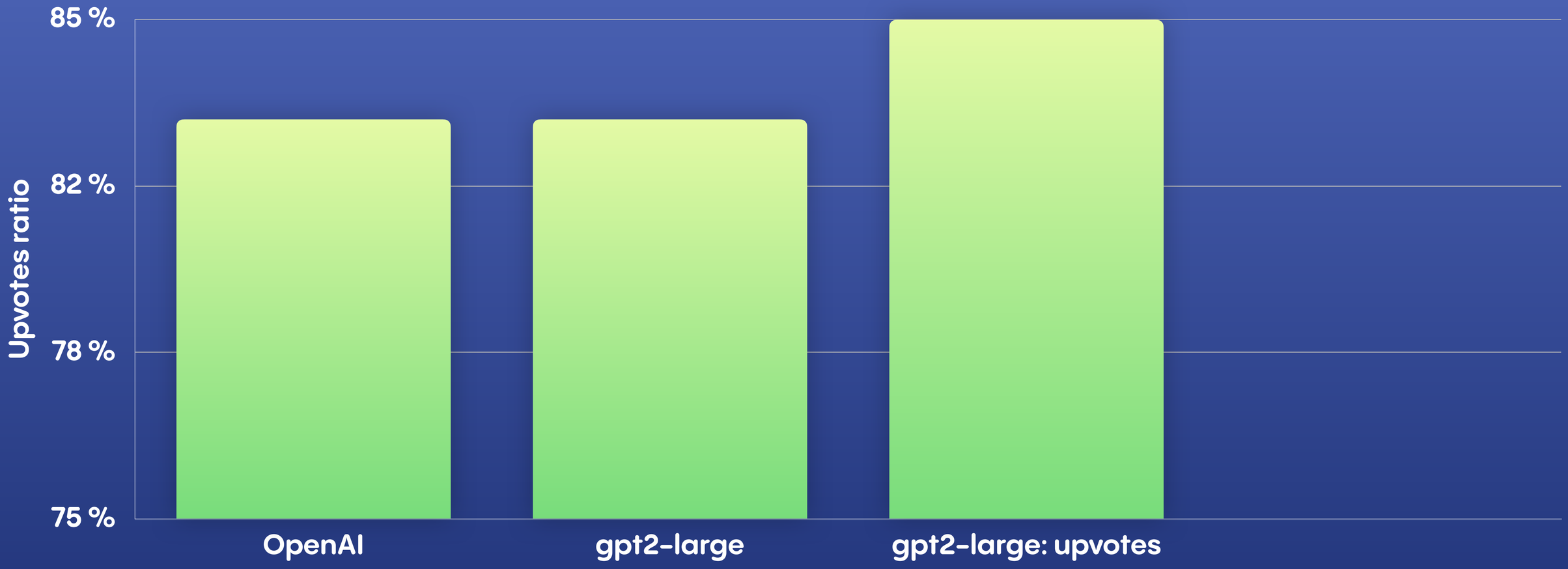


How did we improve the model?

- Optimized the target metric - upvotes fraction
- We have:
  - historical responses generated by OpenAI API
  - user reactions to them (upvotes and downvotes)

⇒ we can train the model on the upvoted OpenAI API responses

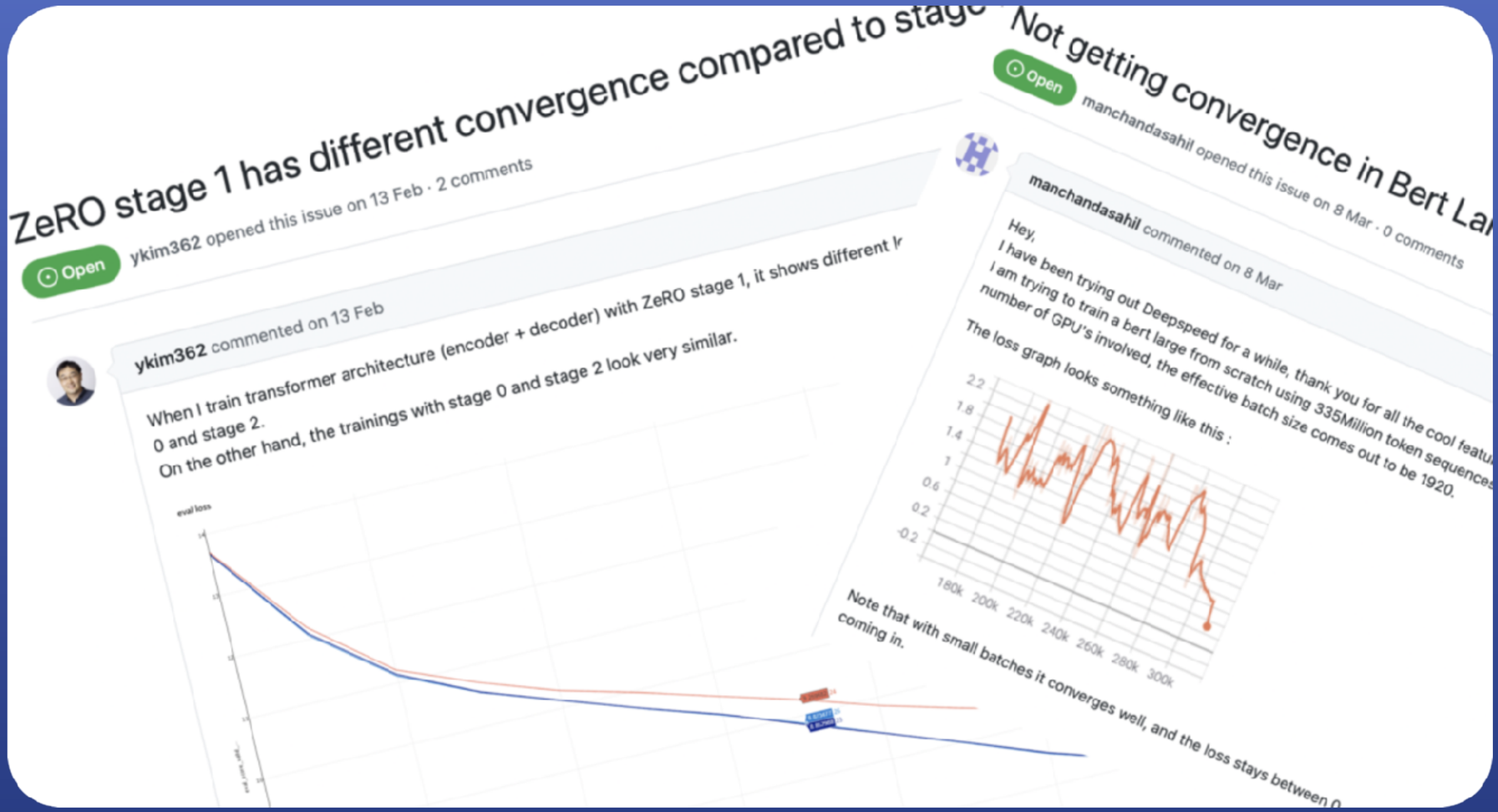
gpt2-large: upvotes



How else did we improve the model?

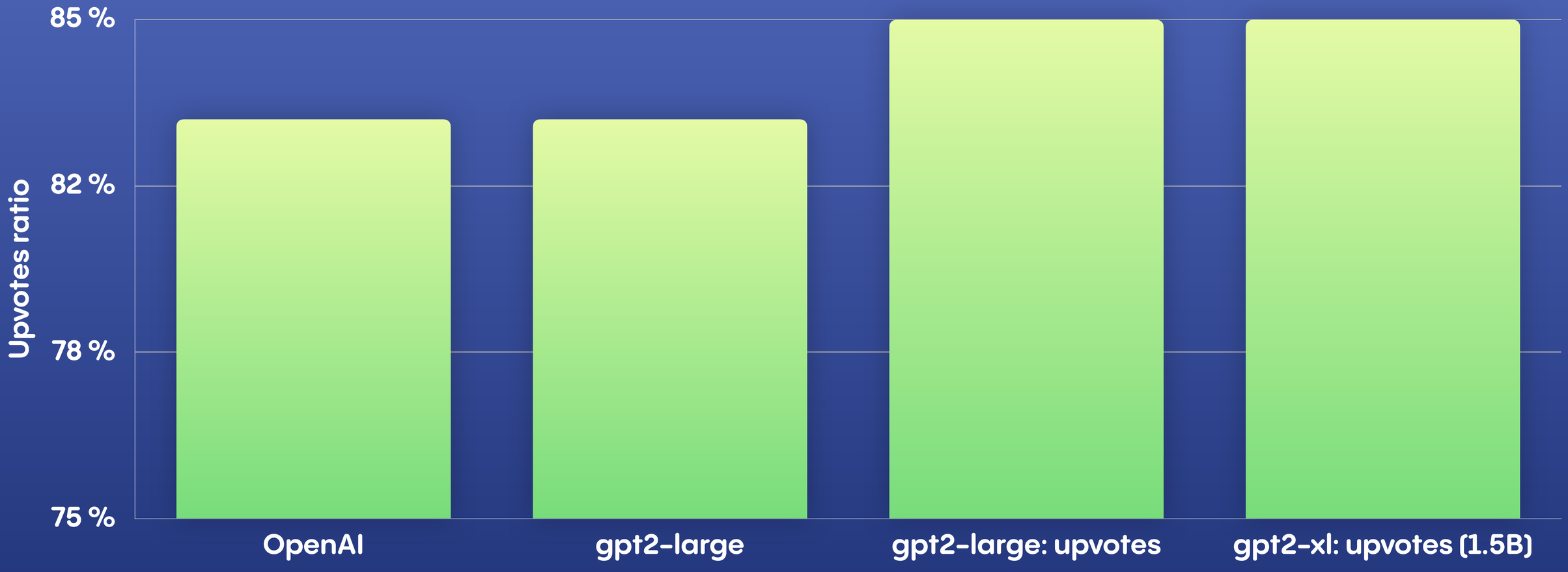
- Tried to increase the model to gpt2-xl (with 1.5B parameters)
- It is not trivial, since it requires a GPU with at least 32Gb of RAM
  - 💡 we used sharded data parallelism (DeepSpeed, Fairscale)
- It can be quite expensive
  - 💡 we adapted training for spot / preemptible instances

Not all solutions are quite stable



💡 Don't use combo of PytorchLightning + DeepSpeed/Fairscale!

gpt2-xl: upvotes



Inference

- We need to process 200 requests per second
- For each single request we generate 10 response candidates
- We applied basic optimizations:
  - ONNX: fp16, layer fusion, etc
  - Dynamic batching
  - Concurrent execution

Inference

We also took into account specificity of the model, so:

- Cached the result of the attention from the previous generation steps
- Limited the length of the input and output (100 tokens is enough)
- Tuned the number of response candidates depending on the current workload

What's the result?

- Our own dialog model that performs better than OpenAI API
- Our own infrastructure for effective model training and inference
- Invaluable experience

Was it worth it?

- For us – definitely, yes
- Generative model is a key component of a diverse and engaging dialog
- Having such a model, we can continuously improve it for our users and thereby make them happier

Thank you

denis@replika.ai