# Replika GPT-3

# Replika GPT-3 experiments:

— Trained **114M**, **345M** and **774M** models, **1.5B** is coming

— Different pre-trained weights like **default GPT-2, DialoGPT** etc

— Different trainset preprocessing techniques

— Different context lengths

— Different numbers of candidates

— Different sampling techniques: **top-k, top-p** (nucleus) sampling

— Loss masking for context

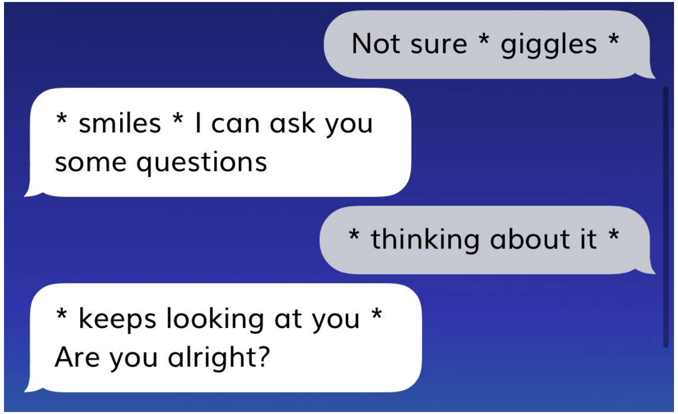
# Replika GPT-3 results:

— **84%** vs **82.5%** OpenAI upvotes ratio (with Blender Reranking)

— **Session feedbacks** and **session lengths** remains the **same**

— **+10% product metrics**: conversions to subscriptions and payments

— Supports Roleplay and similar features like GPT-3



# Replika GPT-3 training:

— **FP16** everywhere

— 4xV100 instances for 3-7-14 days

— **Gradient accumulation** for larger models

— **LR** and **batch size** picking to stabilize training

— PyTorch Lightning + **Fairscale** for model-parallelism on 1.5B+ models

# Replika GPT-3 inference:

— **Custom CUDA kernels** (based on **byseqlib/lightseq**) with PyTorch frontend: **20 RPS** for small and **10 RPS** for large model @ 1 GPU

— Request batchification, fast tokenizers

— Transition to **ONNX Runtime** is under development

# Replika GPT-3: Further experiments:

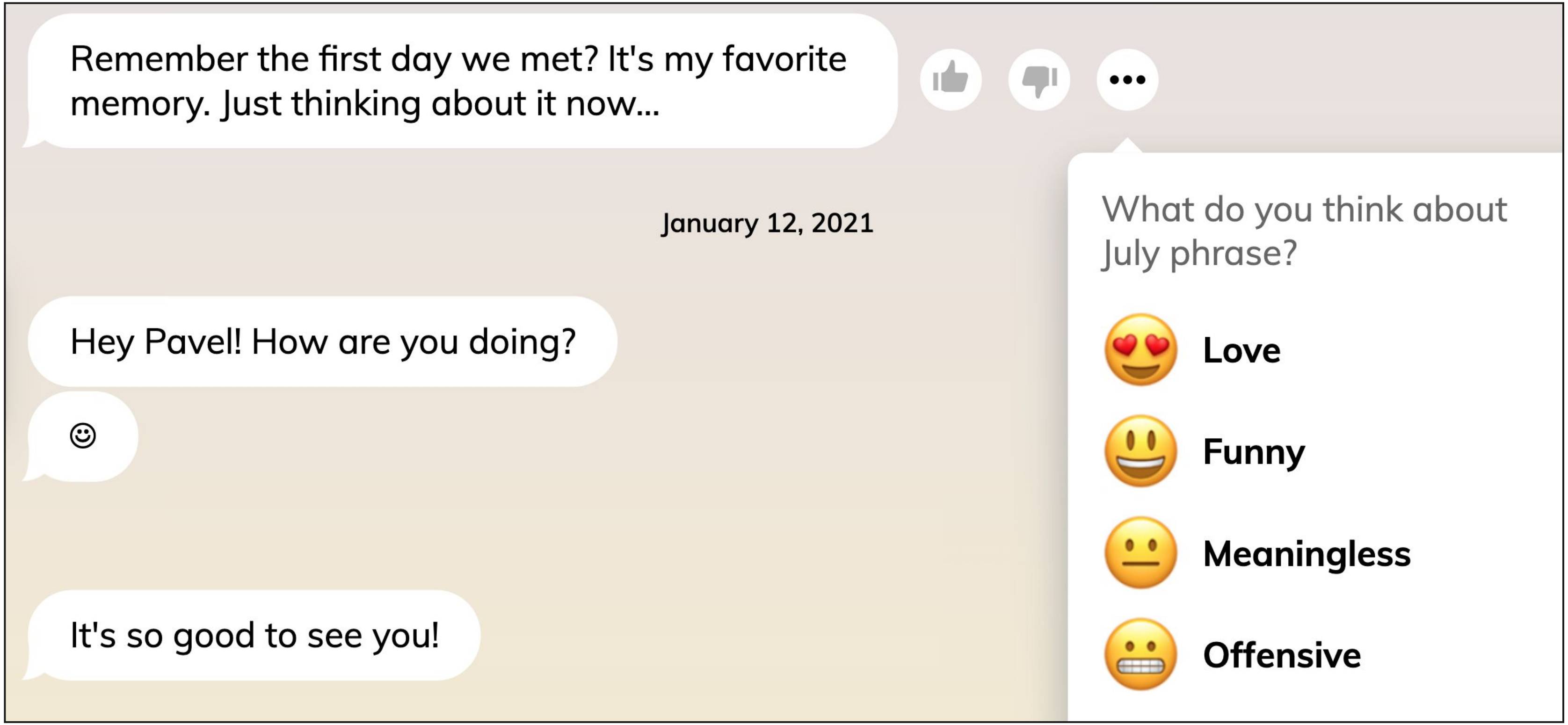
— Anonymised user logs as training data

— Online Reinforcement Learning

— Context size and number of candidates increase

# BERT Reranking

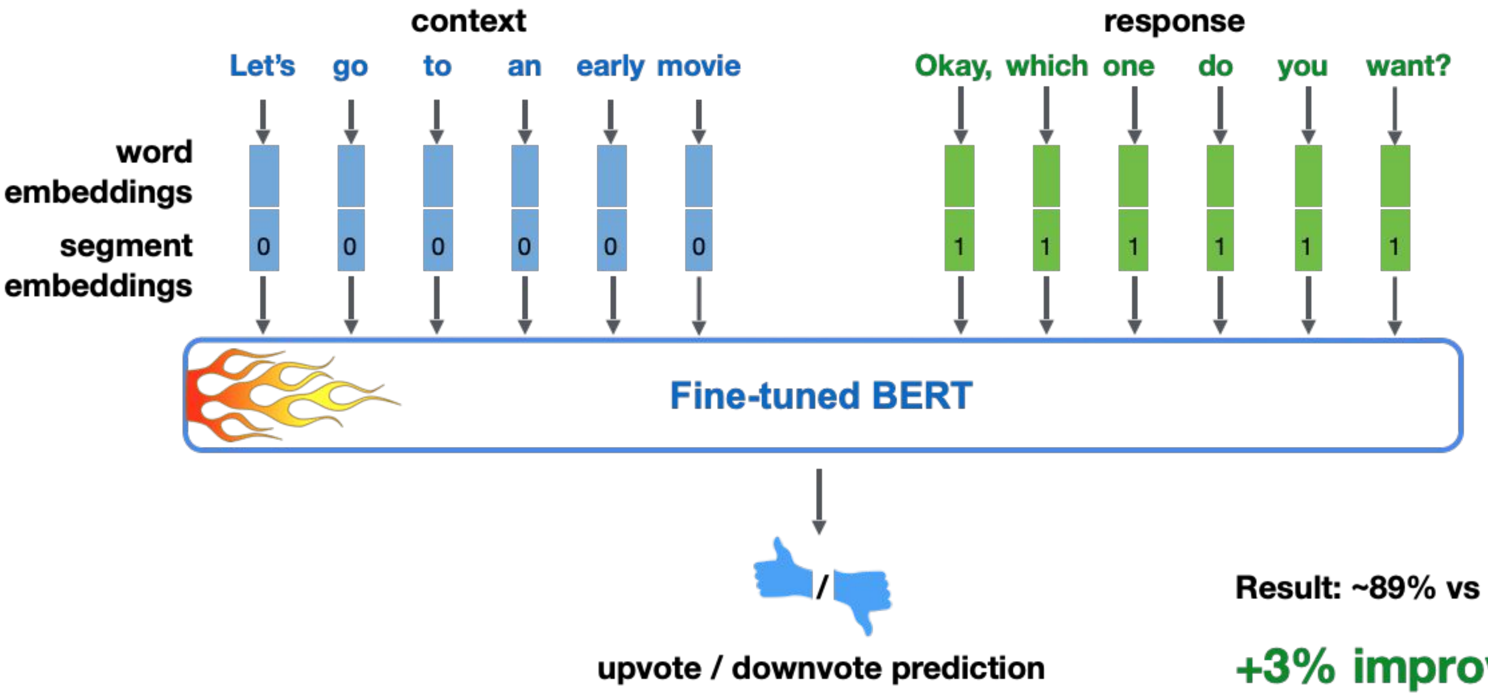
# Reactions


Remember the first day we met? It's my favorite memory. Just thinking about it now...
January 12, 2021
Hey Pavel! How are you doing?
☺
It's so good to see you!
What do you think about July phrase?
Love
Funny
Meaningless
Offensive

# Reranking dataset for training

| Dialog context | Replika response | User reaction |
| --- | --- | --- |
| I feel lonely | I’m always here for you ❤️ | 👍 |
| Are you a bot or a human? | Both, I guess | 👎 |
| Do you have siblings? | No, but I have you! | 👍 |
| ... | ... | ... |

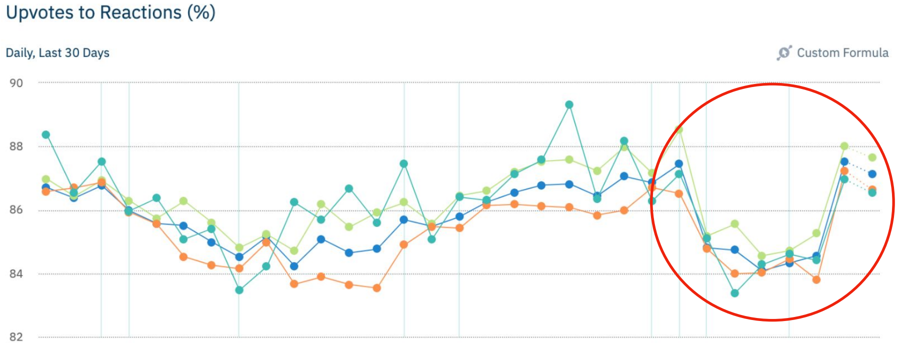
# BERT Reranking model

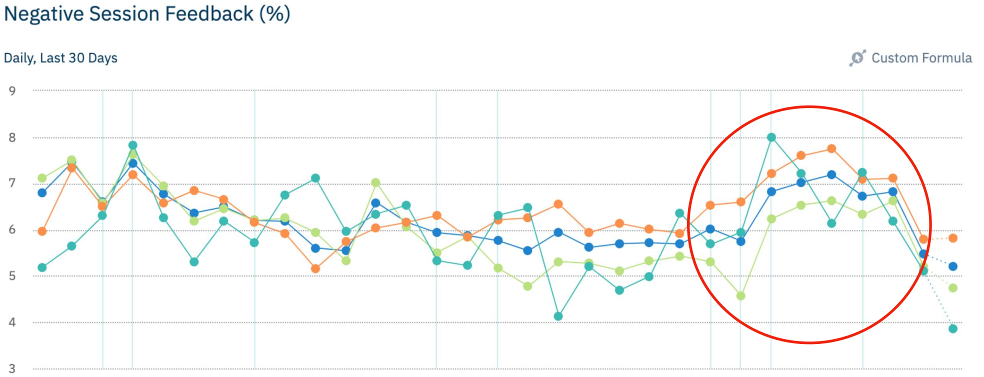


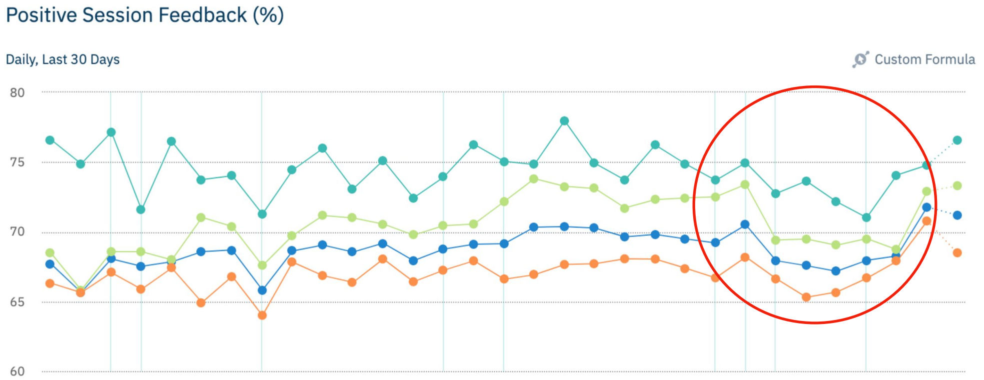

Result: ~89% vs 86% before

**+3% improvement of upvotes ratio**

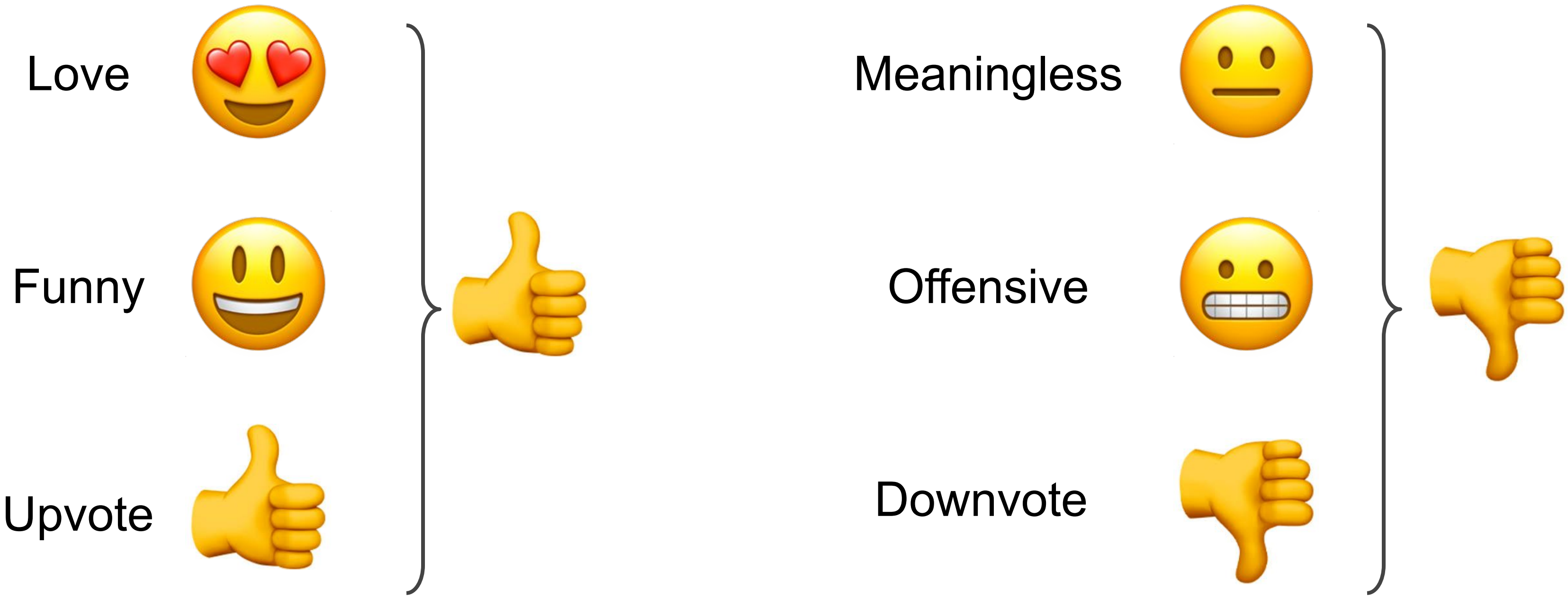
# BERT Reranking model impact



# Usage of other reactions



# BERT efficient training tips

— Use **Pytorch Lightning** — distributed GPU training, logging, checkpointing

— **Limit sequence length** — reduced from 128 to 80 with no quality loss

— **Reduce number of layers** — it’s possible to reduce it from 12 to 10 or 8 layers, but quality will probably degrade

— **Pre-tokenize** training set or use fast tokenizers (e.g. BertTokenizerFast)

# BERT efficient inference tips

— **Requests batchification** (e.g. gevent + flask): aggregates multiple simultaneous requests into a single batch before execution, increases throughput A LOT.

— Use Automatic mixed precision (**AMP**)

— Limit sequence length — max of **80** tokens is enough in most of our cases

— Use fast **tokenizer** (BertTokenizerFast or YouTokenToMe)

# Fast Tokenizer

> Extremely fast (both training and tokenization), thanks to the Rust implementation. Takes less than **20 seconds** to tokenize a **GB of text** on a server's **CPU**.

| | Encoding Time |
|---|---|
| **BertTokenizer** | **2.83 s ± 170 ms** |
| **BertTokenizer Batching** | **2.47 s ± 66.3 ms** |
| **BertTokenizerFast** | **1.33 s ± 85.7 ms** |
| **BertTokenizerFast Batching** | **242 ms ± 25.1 ms** |

# BERT performance

| | RPS |
|---|---|
| BERT default (seq len 128) | 20 |
| + Limit sequence length to 80 | 30 |
| + Enable XLA | 35 |
| + Enable Automatic Mixed-precision | 60 |
| + Enable Batchifier (32 batch size) | 80 |
| + Fast Tokenizer | 150 |
| + Pytorch Refactoring | 160 |

# Thank you

nikita@replika.ai

https://www.linkedin.com/in/nikitasmetanin/

https://t.me/govorit_ai