Quotes Recommender System

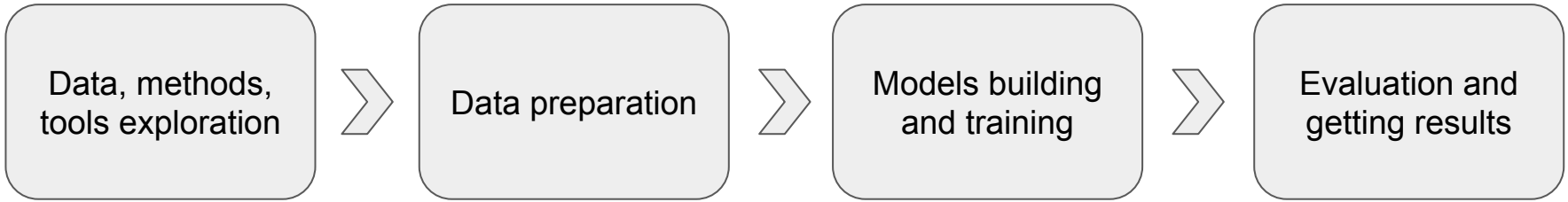
Vladimir Makharev, Artem Batalov, Georgii Budnik

What is our project about?

is to develop a system that *recommends quotes based on the answer* to the question:

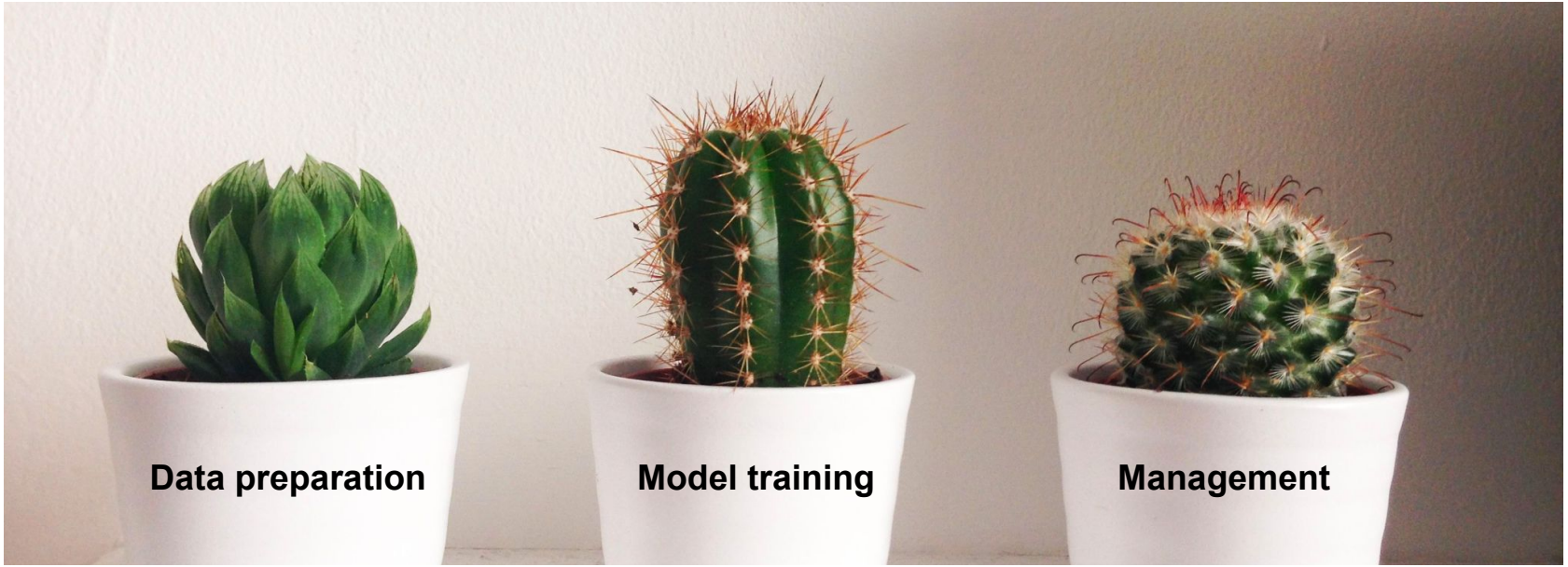
How was your day?

Timeline


Data, methods, tools exploration
Data preparation
Models building and training
Evaluation and getting results

Our team

* responsibilities can be combined



Vladimir Makharev

Artem Batalov

Georgii Budnik

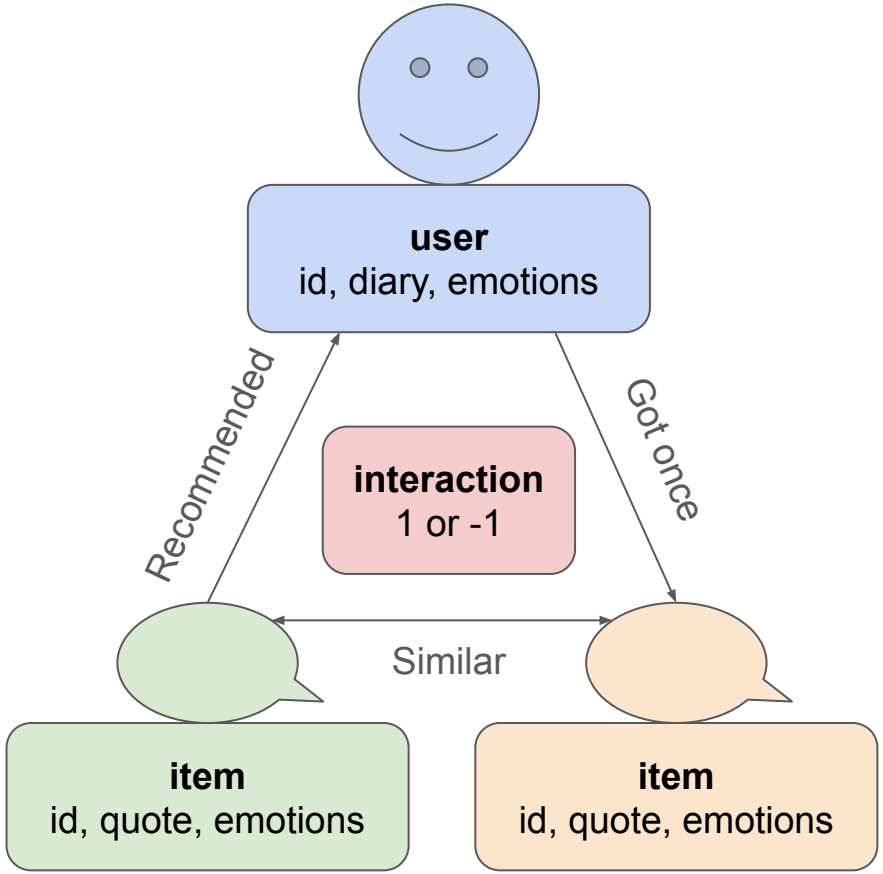
Problem setting

Content-based recommender system

- **user**: (id + diary text + text emotions)
- **item**: (id + quote text + text emotions)
- **interaction**: (-1 or 1) a value that denotes whether user liked the item or not

Preprocessing:

- Embeddings of texts
- Emotions scores of texts



Data preparation

1. Datasets used:
   a. Combined Diaries dataset of 1648 samples [**users**]
   b. Quotes dataset of 852 samples [**items**]
2. Attempt to unify emotions labels between datasets
3. Labeling datasets using classifiers
   a. roberta-base-go_emotions (27 classes + Neutral)
   b. twitter-roberta-base-emotion-multilabel-latest (11 classes)

text	optimism	anger	…
‘Today I have …	0.623463432111	0.124124214132	…

4. Using baseline model create a dataset of `<diary, quote>` pairs
5. Using GPT models create Interactions dataset (to solve cold start problem)

Data preparation: samples from baseline dataset

< diary, quote >

- My family was the most salient part of my day, since … I am so excited to see them hit developmental and social milestones.

“I'm possessed by love — but isn't everybody?”

- Freddie Mercury

- Yesterday, I had to go to work. It was my first day back to work after my weekend, so I was pretty frustrated and sad. It was a pretty boring day overall.

“I used to get upset by people not understanding me, but I’ve made a career out of it now.”

- Ozzy Osbourne

Data preparation: prompts used to generate Interactions dataset

< diary, quote, interaction >

519 samples, `gpt-4`

System prompt:

```
“You are an expert recommendation evaluator. Now you must evaluate recommended quote of famous person for the person diary note. Just write '1' if quote is suitable, and '-1' if unsuitable"
```

Instruction prompt:

```
"Text: {Text}\n\nQuote: {e.Quote}”
```

Data preparation: prompts used to generate Interactions dataset

< diary, quote, interaction >

544 samples, `gpt-3.5-turbo`

System prompt:

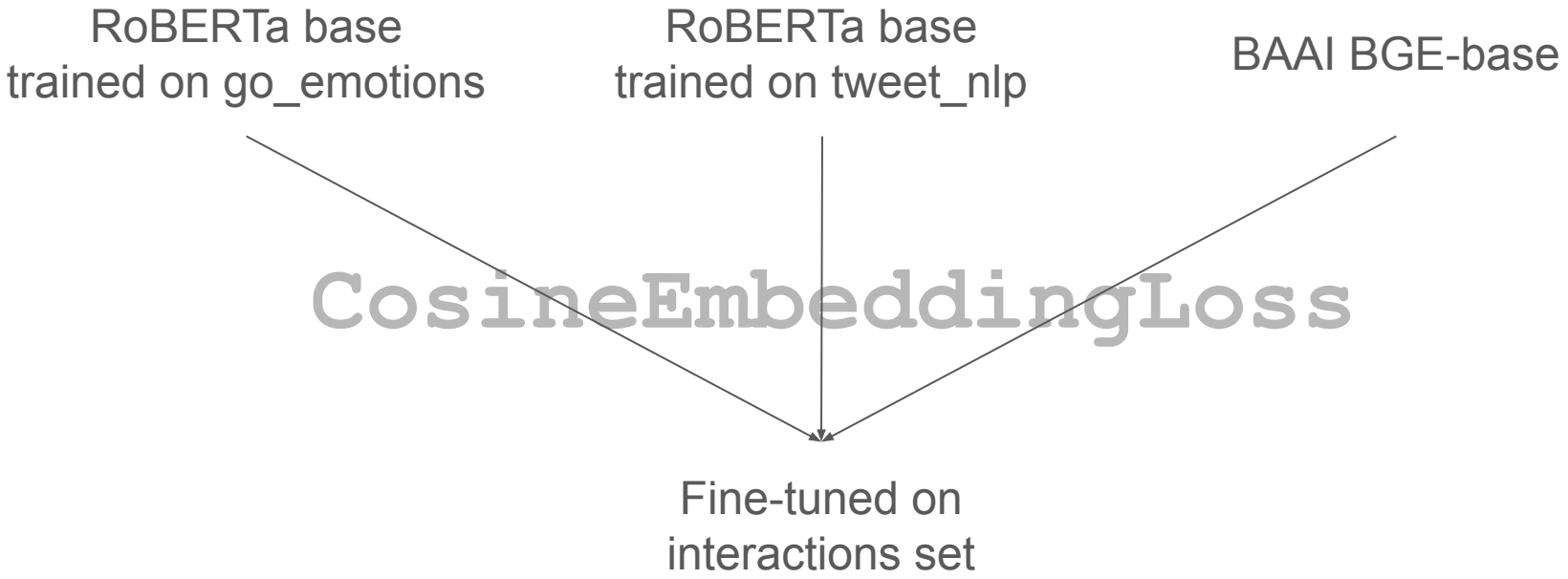
"Imagine you are the primary evaluator of the recommendation system. System recommend quote to the diary text by user (user answers to the question \"How was your day?\"). The system have cold start problem, that is why we ask you to solve a specific task (will be announced in the next prompt). You have a dataset of pairs (diary text, quote text). The diary texts were collected journal entries from people reflecting on their day. The quote was selected by baseline model that uses cosine similarity distance between embeddings from multi-label (28 emotions) classifier on reddit texts."

Instruction prompt:

"Your task is to evaluate each pair of diary text and quote text how much (in your opinion) the quote fits the corresponding diary, giving a rating of "1" - if it fits enough or "-1" - if it is not fits enough. Make decisions based on how the quote could lift the mood of the user and mood of most people who would be shown this quote for a similar text. Write ONLY "1" or "-1"\n\nDiary text: {e.Text}\nQuote text: {e.Quote}" .

Models building and training

Models use Cosine Similarity and threshold value to select most close quotes



Evaluation results

Test dataset of 70 samples from Interactions dataset by `gpt-3.5-turbo`

We introduce our own `Custom-GPT-Score` metric. This metric represents accuracy, i.e., number of positive (equal to 1) interactions out of all samples

```
evaluate_openai(diaries, reddit_recommended, model="gpt-3.5-turbo")
```

```
100%|██████████| 70/70 [00:34<00:00,  2.03it/s, retry=1]
0.38571428571428573
```

```
evaluate_openai(diaries, reddit_finetuned_recommended, model="gpt-3.5-turbo")
```

```
100%|██████████| 70/70 [00:33<00:00,  2.08it/s, retry=1]
0.44285714285714284
```

```
evaluate_openai(diaries, twitter_recommended, model="gpt-3.5-turbo")
```

```
100%|██████████| 70/70 [00:33<00:00,  2.08it/s, retry=1]
0.4
```

```
evaluate_openai(diaries, twitter_finetuned_recommended, model="gpt-3.5-turbo")
```

```
100%|██████████| 70/70 [00:30<00:00,  2.27it/s, retry=1]
0.4857142857142857
```

```
evaluate_openai(diaries, bge_base_recommended, model="gpt-3.5-turbo")
```

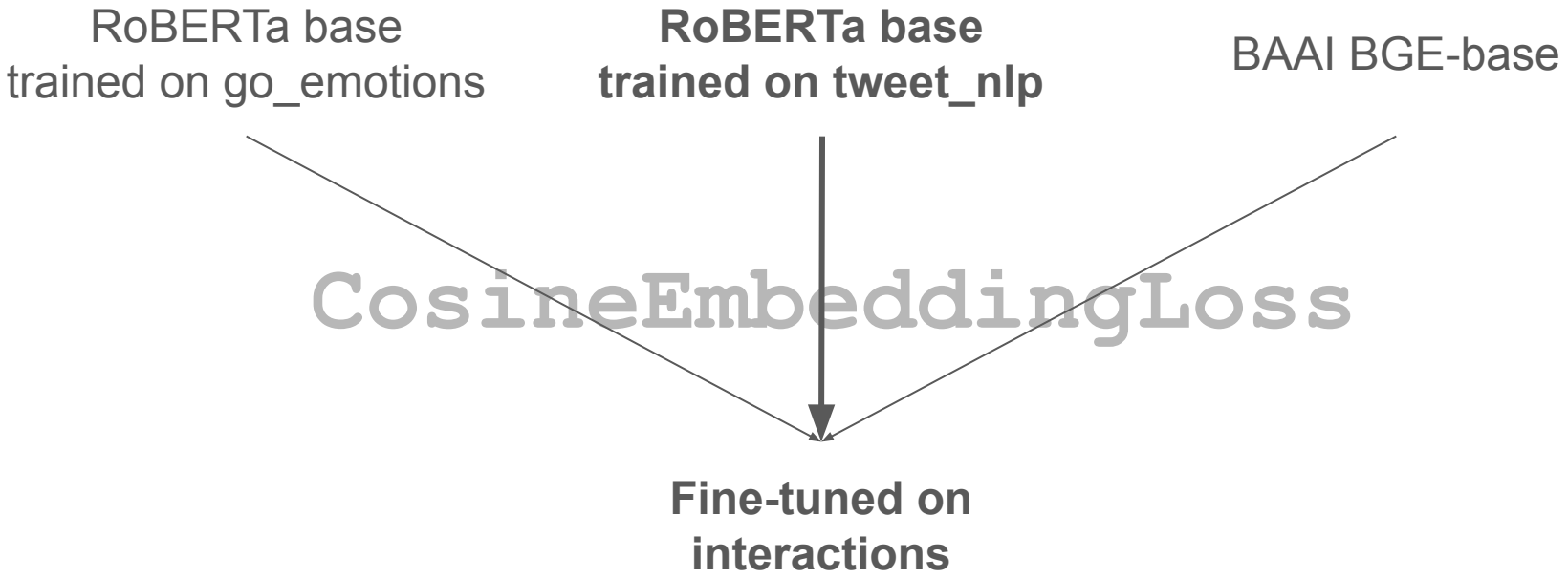
```
100%|██████████| 70/70 [00:33<00:00,  2.11it/s, retry=1]
0.4
```

```
evaluate_openai(diaries, bge_recommended, model="gpt-3.5-turbo")
```

```
100%|██████████| 70/70 [00:42<00:00,  1.65it/s, retry=1]
0.44285714285714284
```

Evaluation results

Models use Cosine Similarity and threshold value to select most close quotes



Attempt to use RecTools

New dataset:

- user_id
- item_id
- weight
- datetime
- user_features
- item_features

Split to train and test

Models

- PopularModel
- RandomModel
- PureSVDModel

k=10 recommendations/user

Get recommendations data (user_id, item_id, score, rank)

Compute metrics:

- Accuracy
- IntraListDiversity
- MCC
- MeanInvUserFreq

Other metrics were not representative… due to **issues with data…**

Conclusion

We assumed:

- diary and quote domain is very difficult to analyze by ML
- challenging to simulate the real user’s data
- need human evaluation to be conducted

The system successfully demonstrates the potential of using NLP and DL techniques in developing a recommendation system tailored to emotional contexts.

Demonstration

“Do the best you can until you know better”
- Maya Angelou, American poet

Questions?