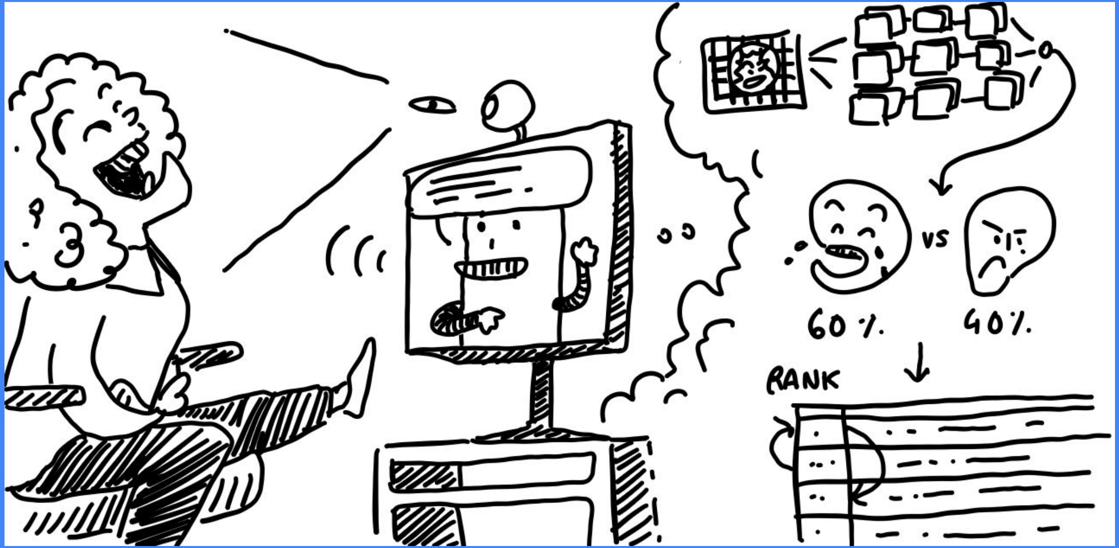


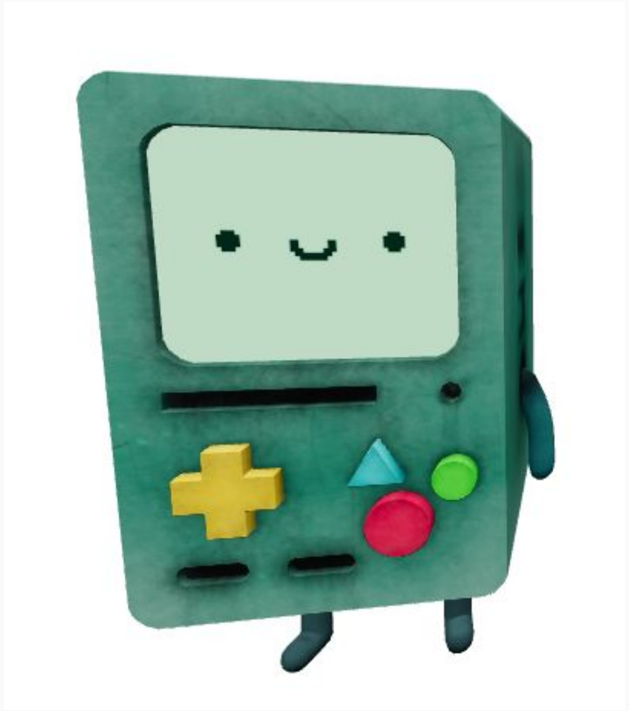
Can I joke on you?

Anicet Nougaret, Lena Ebner, Jens De Bock

6/11/2023

Overview

Goals & Concept



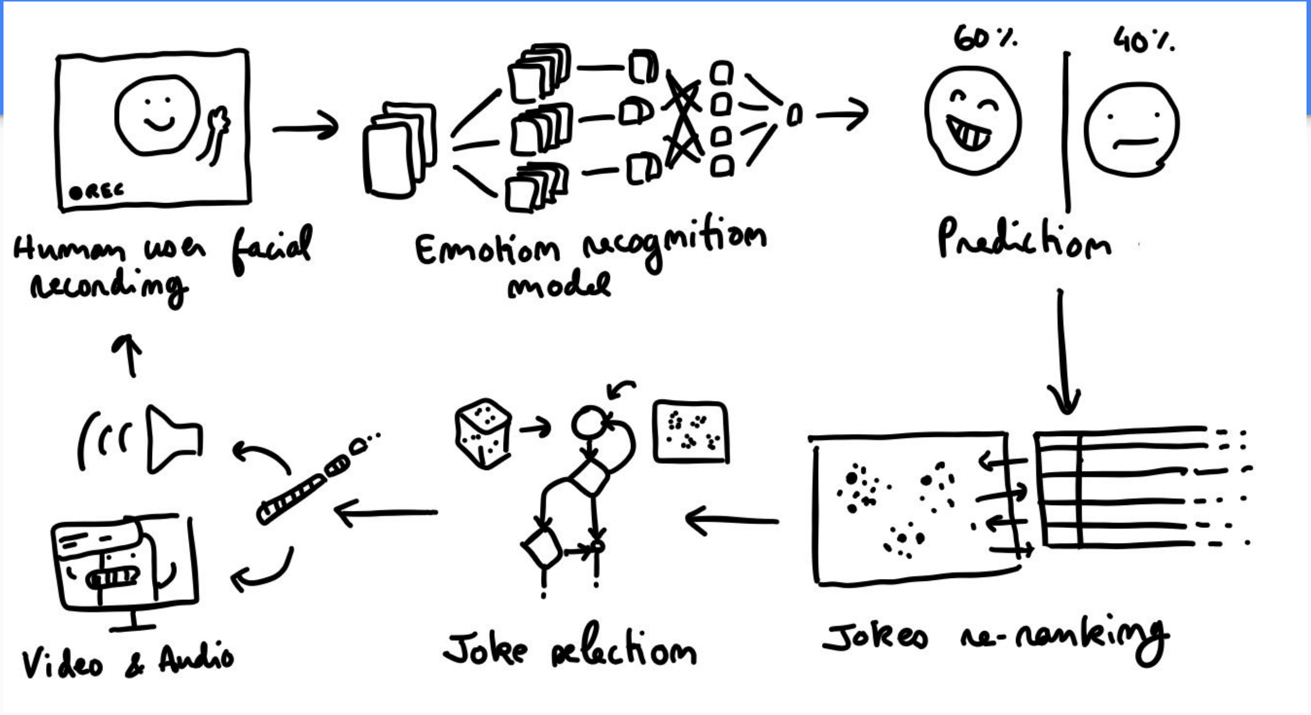
Goals

Making the user happy and entertained with good jokes.

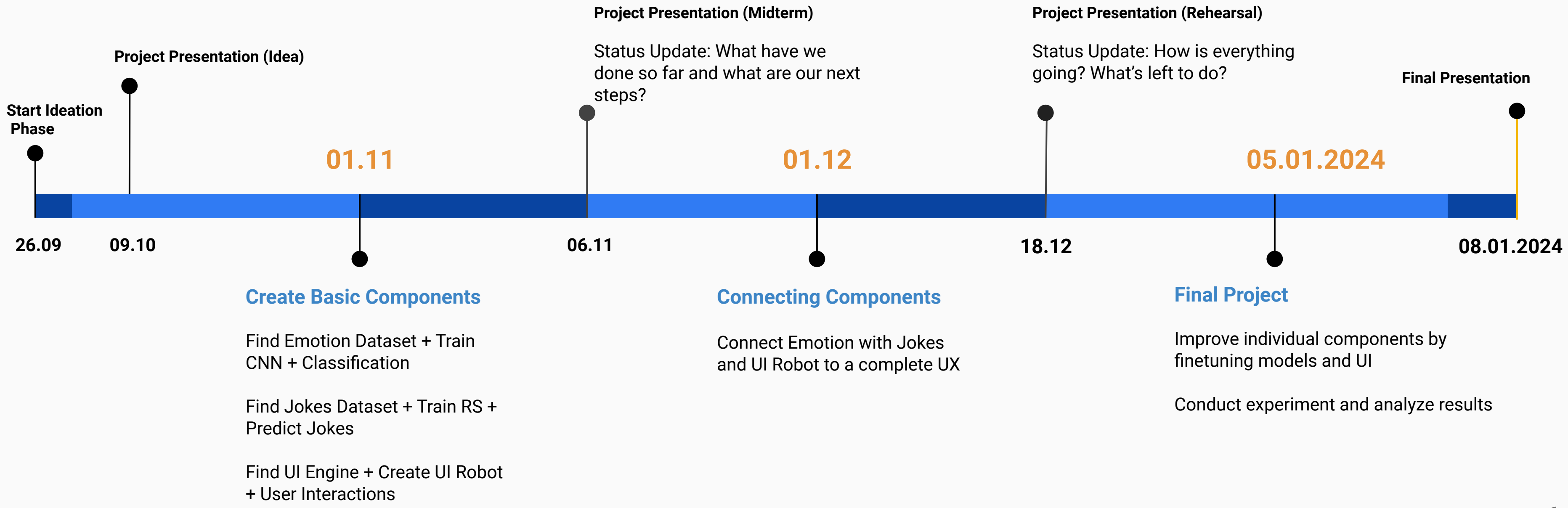
Exploring the ability of a system to learn based on user's emotions and act on them in a positive reinforcement loop.

Concept

A robot that tells jokes to the user. Able to detect the user's facial expressions, he learns to pick jokes the user seems to like.



Planning



Start Ideation Phase

26.09

Project Presentation (Idea)

09.10

01.11

Create Basic Components

Find Emotion Dataset + Train CNN + Classification

Find Jokes Dataset + Train RS + Predict Jokes

Find UI Engine + Create UI Robot + User Interactions

Project Presentation (Midterm)

Status Update: What have we done so far and what are our next steps?

06.11

01.12

Connecting Components

Connect Emotion with Jokes and UI Robot to a complete UX

Project Presentation (Rehearsal)

Status Update: How is everything going? What's left to do?

18.12

05.01.2024

Final Project

Improve individual components by finetuning models and UI

Conduct experiment and analyze results

Final Presentation

08.01.2024

Research Question & Target group

Research Question

Is a robot UI with recommender more fun to use than telling random jokes?

Variables

3 Variables

- Recommender is active or not (independent)
- Facial expressions (independent)
- User experience (dependent)

Target Group

English-speaking students

Why?

English: dataset of English jokes

Students: more accessible

Project Description

Parts of the project

Identify smile degree via camera

Jens

+

Recommender system for jokes

Anicet

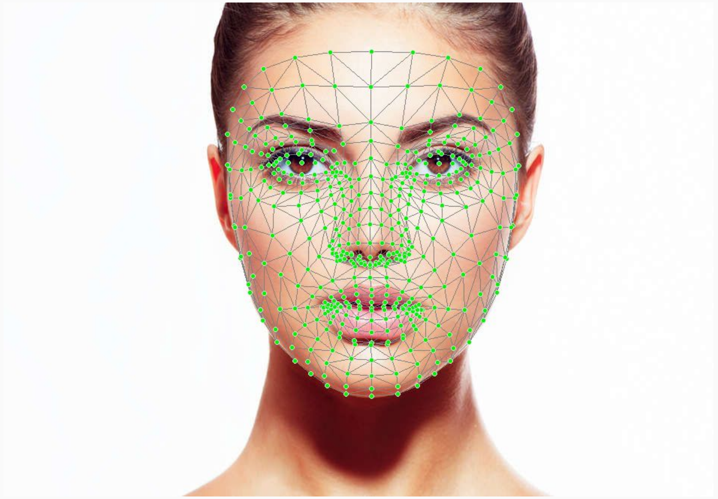
+

UI with virtual robot for user interaction

Lena

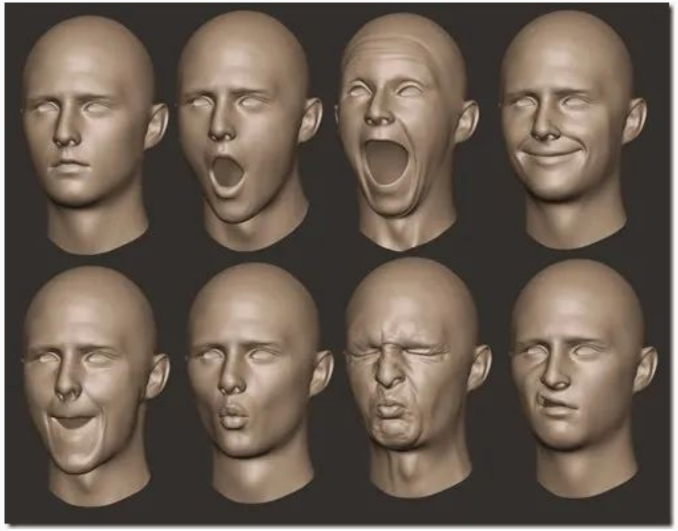
Smile Detection

- Facial keypoints from MediaPipe
- Input to Convolution Neural Network
- 90% accuracy between non smile and smile
- Real world is however harder
- Search better and more explainable

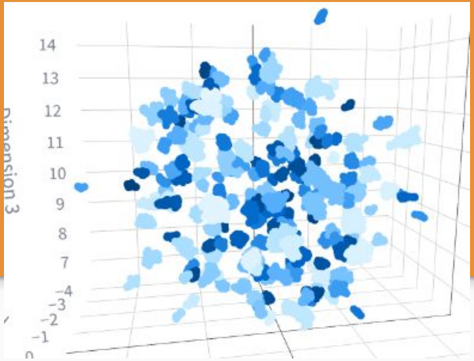


Smile Detection

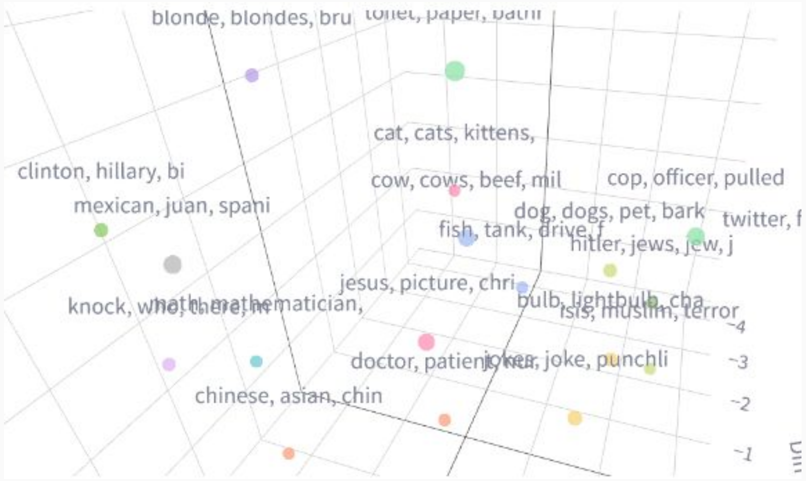
- No more usage of keypoints
- Use blendshapes to estimate laughing
- Can be used directly in the front-end
- 2 calibration phases
- Finetune and normalize output



Joke Dataset

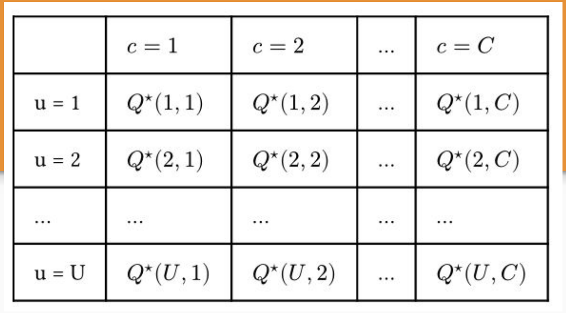


- English dataset of jokes on internet -> embeddings -> UMAP -> clustering -> semantic categories
- Cleaning + keeping 26 categories

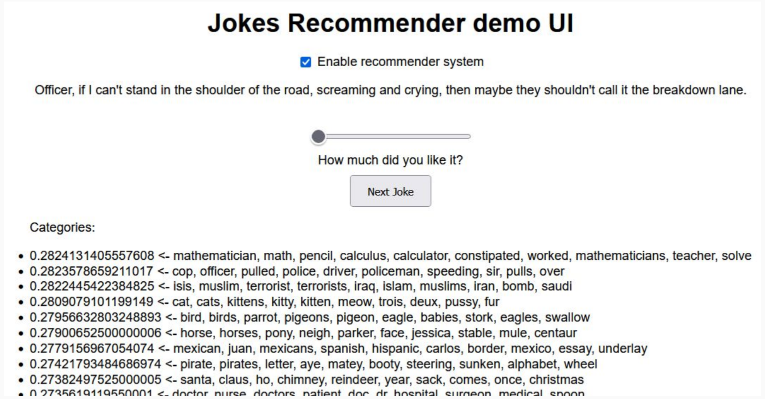


Joke Recommender

- similar to tabular Q-learning (Barto & Sutton, 1998)
- Fast convergence to pool of categories
- Good at filtering
- Tested in dedicated UI



	$c = 1$	$c = 2$	...	$c = C$
u = 1	$Q^\star(1,1)$	$Q^\star(1,2)$	...	$Q^\star(1,C)$
u = 2	$Q^\star(2,1)$	$Q^\star(2,2)$	...	$Q^\star(2,C)$
...	...	...	...	...
u = U	$Q^\star(U,1)$	$Q^\star(U,2)$	...	$Q^\star(U,C)$



Jokes Recommender demo UI

☑ Enable recommender system

Officer, if I can't stand in the shoulder of the road, screaming and crying, then maybe they shouldn't call it the breakdown lane.

How much did you like it?

Categories:

- 0.2824131405557608 <- mathematician, math, pencil, calculus, calculator, constipated, worked, mathematicians, teacher, solve
- 0.2823578659211017 <- cop, officer, pulled, police, driver, policeman, speeding, sir, pulls, over
- 0.2822445422384825 <- isis, muslim, terrorist, terrorists, iraq, islam, muslims, iran, bomb, saudi
- 0.2809079101199149 <- cat, cats, kittens, kitty, kitten, meow, trois, deux, pussy, fur
- 0.27956632803248893 <- bird, birds, parrot, pigeons, pigeon, eagle, babies, stork, eagles, swallow
- 0.27900652500000006 <- horse, horses, pony, neigh, parker, face, jessica, stable, mule, centaur
- 0.2779156967054074 <- mexican, juan, mexicans, spanish, hispanic, carlos, border, mexico, essay, underlay
- 0.27421793484686974 <- pirate, pirates, letter, aye, matey, booty, steering, sunken, alphabet, wheel
- 0.27382497525000005 <- santa, claus, ho, chimney, reindeer, year, sack, comes, once, christmas
- 0.273561911955001 <- doctor, nurse, doctors, patient, doc, dr, hospital, surgeon, medical, spoon

Robot UI

- Create UI with **Next.js** using **Three.js** for rendering 3D Model
- Looking for an appropriate **3D Model**, smiley, cute, with slight animations
- **TTS** Web API for telling jokes
- **Automatic Logging** of Smile Detection
- Implementing **User Flow** according Study Design
- **Deployment**
- **Connecting** UI with **recommender** and **Smile Detection**
    - **REST API** calls to recommender
    - directly **doing smile detection in frontend** with @mediapipetasks-vision

Demo

[illegible]

https://youtu.be/S_HRdiZi24g

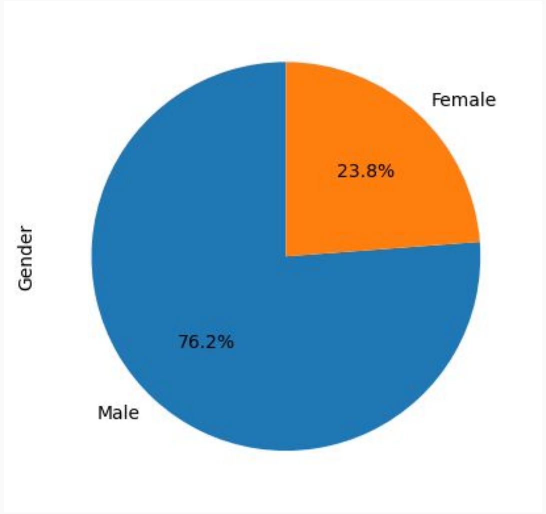
Evaluation

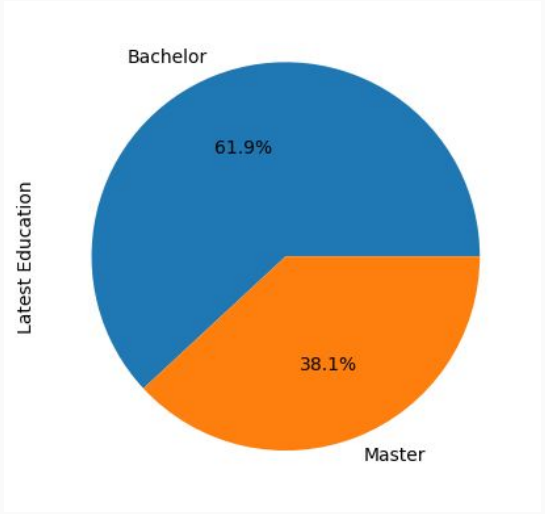
What we wanted to evaluate?

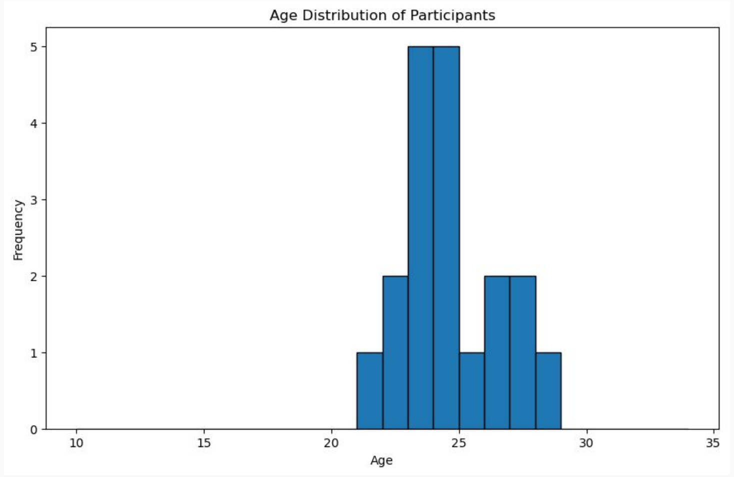
1. **How well** learns the recommender system
2. Smile detection **accuracy**
3. **User experience** differences

Participants

21 Participants







Study Procedure

Testing 2 times with each user

- with recommender (A)
- without recommender (B)

Remote

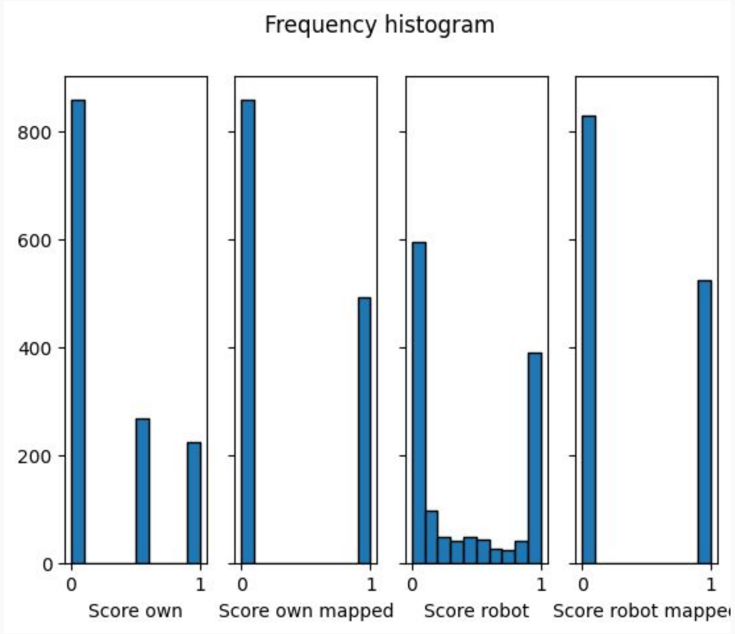
Interaction time 5-15 min

Fill out questionnaires 2x

Results

Smile Accuracy

- Categorical values vs numerical values
- Solution: map values to smile or no smile
- Result: non-significant difference
    - Overall accurate detection
- MSE: 13%

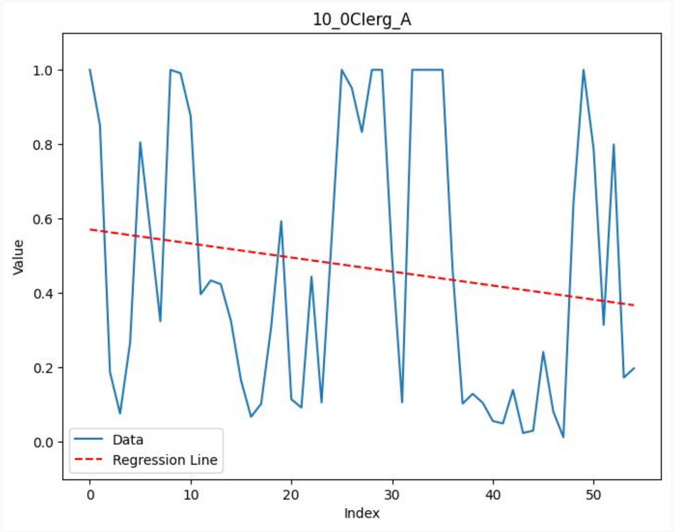


Using a recommender system or not

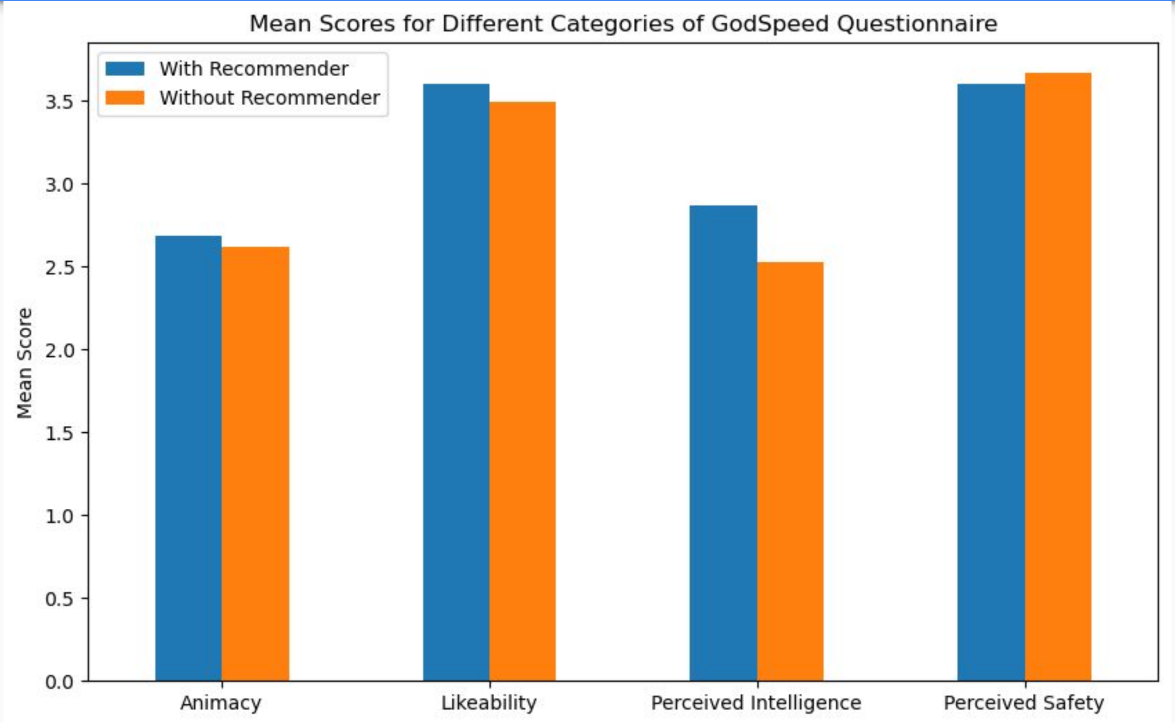
- Compare detected values from both experiments
- Result: non-significant difference
  - Recommender system does not have a big influence
- Reasons:
  - Recommender system based on themes
  - Dataset has a lot of flawed jokes

How well learns the recommender system

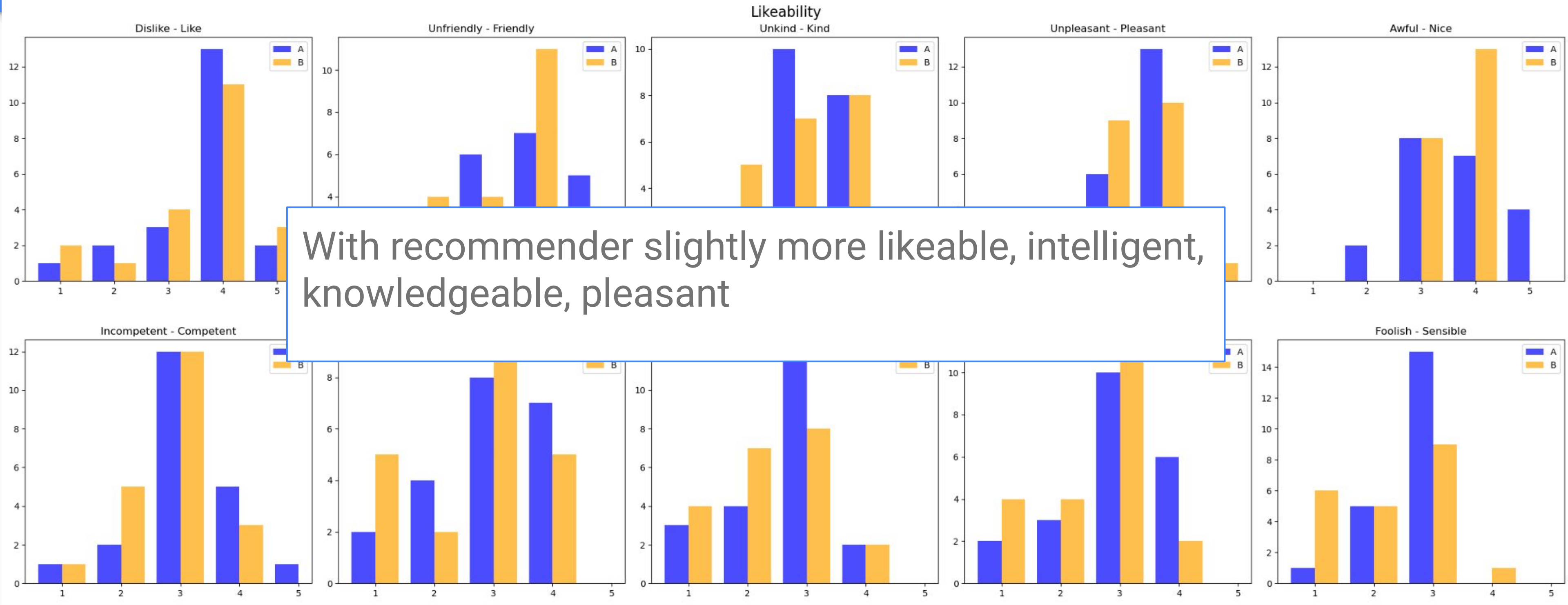
- Compare first part to second part
- Result: significant difference
  - Second part performs worse overall
  - 0.17% vs 0.23%
- Possible reasons:
  - Recommender system minimal influence
  - Tired towards the end
  - Similar jokes are not funny twice



User Experience Differences I



User Experience Differences II



With recommender slightly more likeable, intelligent, knowledgeable, pleasant

Conclusions

Conclusions

Recommender does not lead to a happier user in our application in terms of smile detection

Longer interaction periods with application would help to get more accurate results, but interaction is also very **monotonous** (always the same)

Improvements needed for better results

- Cleaning the dataset would help a lot
- Using different recommendation approach (not just classes)

Conclusions II

Application was **fun to build**

Smile detection works well, but strongly depends on individual and calibration

UX differences indicate that an application with recommender enabled makes a **more intelligent, competent and fun impression** to user

Questions?
Feedback?