



OpenAssistant

Vision & Roadmap

OpenAssistant is a **chat-based assistant** that understands **tasks**, can interact with **third-party systems**, and **retrieve information** dynamically to do so.

It can be extended and personalized easily and is developed as free, open-source software.

Your Conversational Assistant

State-of-the-Art chat assistant that can be personalized to your needs

Retrieval via Search Engines

External, upgradeable knowledge: No need for billions of parameters.



Interfacing w/ external systems

Usage of APIs and third-party applications, described via language & demonstrations.

A building block for developers

Integrate OpenAssistant into your application.

OpenAssistant unifies all knowledge work in one place

- Uses modern deep learning
- Runs on consumer hardware
- Trains on human feedback
- Free and open

Our Vision

We want OpenAssistant to be the single, unifying platform that all other systems use to interface with humans.

Our Roadmap



Minimum Viable Prototype

- Data Collection Pipeline
- RL on Human Feedback
- Assistant v1 usable
- Out January 2023!

Growing Up

- Retrieval Augmentation
- Rapid Personalization
- Using External Tools

Growing Out

- Third-Party Extensions
- Device Control
- Multi-Modality

How did we get here?

- What do you need?

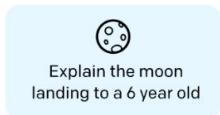
Getting to MVP

We follow [InstructGPT](#)

Step 1

Collect demonstration data, and train a supervised policy.

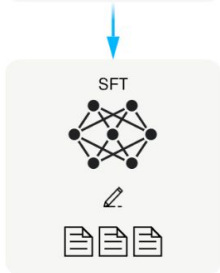
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



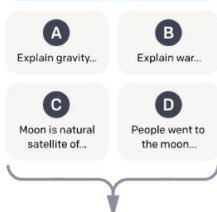
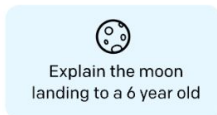
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

Collect comparison data, and train a reward model.

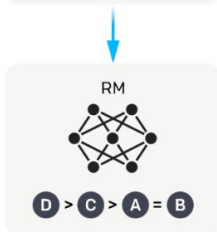
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

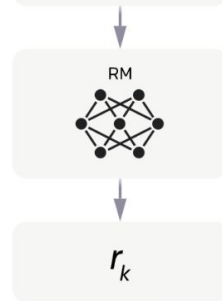
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

1) Supervised Fine-Tuning on Human Demonstrations

- We need to collect (human) demonstrations of assistant interactions
 - Read our [Data Structures Overview](#) to see how
 - We estimate about 50k* demonstrations
- Fine-tuning a base model on the collected data
 - Candidates: GPT-J, CodeGen(surprisingly promising), FlanT5, GPT-JT
 - Can use pseudo-data (e.g. from QA dataset) before we have the real data
- Additionally, collect instruction datasets
 - Quora, StackOverflow, appropriate subreddits, ...
 - Training an "instruction detector" would allow us to e.g. filter Twitter for good data

* InstructGPT has 13k, 33k, and 31k samples for the three steps, respectively

2) Training a Reward Model & RLHF

- We need to collect rankings of interactions
 - Again, read our [Data Structures Overview](#) to see how
- Reward Model Training could also use Active Learning
 - Keeps humans in the loop
 - Drastically decreases needed data
- Reinforcement Learning against the Reward Model
 - Follow InstructGPT and use PPO

Main Efforts

- Data Collection Code → Backend, website, and discord bot to collect data
- Instruction Dataset Gathering → Scraping & cleaning web data
- Gamification → Leaderboards & more, to make data collection more fun
- Model Training → Experiments on pseudo- and real-data
- Infrastructure → Collection, training, and inference
- Data Collection → This is the bulk of the work
- Data Augmentation → Making more data from little data
- Privacy and Safety → Protecting sensitive data

Principles

- We put the human in the center
- We need to get the MVP out fast, while we still have momentum
- We pull in one direction
- We are pragmatic
- We aim for models that can (or could, with some effort) be run on consumer hardware
- We rapidly validate our ML experiments on a small scale, before going to a supercluster

Where to go from here?

- Read the [Data Structures Documentation](#)
- Come to [our repository](#) and grab an issue
- Join the [LAION Discord](#) and the [YK Discord](#)