

OpenAssistant Conversations - Democratizing Large Language Model Alignment

Andreas Köpf*

andreas.koepf@provisio.com

Yannic Kilcher*

yannic@ykilcher.com

Dimitri von Rütte

Sotiris Anagnostidis

Zhi-Rui Tam

Keith Stevens

Abdullah Barhoum

Nguyen Minh Duc

Oliver Stanley

Richárd Nagyfi

Shahul ES

Sameer Suri

David Glushkov

Arnav Dantuluri

Andrew Maguire

Christoph Schuhmann

Huu Nguyen

Alexander Mattick

alexander.mattick@gmail.com

Abstract

Aligning large language models (LLMs) with human preferences has proven to drastically improve usability and has driven rapid adoption as demonstrated by ChatGPT. Alignment techniques such as supervised fine-tuning (*SFT*) and reinforcement learning from human feedback (*RLHF*) greatly reduce the required skill and domain knowledge to effectively harness the capabilities of LLMs, increasing their accessibility and utility across various domains. However, state-of-the-art alignment techniques like *RLHF* rely on high-quality human feedback data, which is expensive to create and often remains proprietary. In an effort to democratize research on large-scale alignment, we release OpenAssistant Conversations, a human-generated, human-annotated assistant-style conversation corpus consisting of 161,443 messages in 35 different languages, annotated with 461,292 quality ratings, resulting in over 10,000 complete and fully annotated conversation trees. The corpus is a product of a worldwide crowd-sourcing effort involving over 13,500 volunteers. Models trained on OpenAssistant Conversations show consistent improvements on standard benchmarks over respective base models. We release our code² and data³ under a fully permissive licence.

A list of contributors who have chosen to be acknowledged by name can be found at <https://open-assistant.io/contributors>.

*These authors contributed equally to this work.

²<https://github.com/LAION-AI/Open-Assistant>

³<https://huggingface.co/OpenAssistant/oasst1>

1 Introduction

Artificial intelligence (AI), particularly in the field of natural language processing, has witnessed rapid progress in recent years. Major advancements are primarily driven by a straightforward formula: take a Transformer [1]-based architecture, increase the parameter count by enlarging depth and width, increase the size of the training corpus, and increase the scale of training compute. Although models have for some time exhibited an extraordinary ability to fit the training data and generalize based on their trained objective [2, 3], their adoption among the general public has until recently been slow. This can be mainly attributed to misalignment between model predictions and final intended usage.

The alignment of AI systems to human values, intentions, and preferences is a vital and intricate challenge within the AI research domain. This refers to the process of ensuring that AI systems can not only successfully optimize the provided surrogate training objectives, but also that their predictions are in line with their intended purpose and adhere to ethical and safety standards provided by humans [4, 5]. One possible solution is assistant-style fine-tuning of language models that has recently emerged as a promising approach to making large language models more in line with human preferences by generating more desirable outputs based on explicitly collected human preference data [6, 7, 8, 9, 10, 11] and thus making them more useful.

A notable instance of such an assistant-style model is ChatGPT, which has gained unprecedented user growth due to remarkable capabilities demonstrated in a wide range of fields, but also ease-of-use for the end user [12]. Aligning the model’s predictions is in this case accomplished by introducing human-generated examples of intended usage and using reinforcement learning from human feedback [13, 14]. In *RLHF*, the human acts as a teacher and provides feedback in the form of rewards or penalties. In more detail, Ouyang et al. [13] proposed a three stage procedure to align language models: First, collect human-generated demonstrations of desired behaviour and train a supervised fine-tuned (*SFT*) model. Second, train a reward model (RM) on human-annotated rankings for different model outputs. Third, use the RM as a reward function and fine-tune the *SFT* model to maximize the reward generated by its responses. This is achieved using the PPO algorithm [15].

It becomes apparent that the benefits of all the aforementioned stages are predominantly dependent on the quality of the data used [16]. Despite this, availability of large-scale human feedback datasets for the open research community remains scarce. Most openly accessible datasets are comprised of synthetic data of instructions automatically generated by querying language models [17, 18, 19, 20]. Unfortunately, these datasets are limited with respect to their complexity, creativity and quality, as they rely on a pre-specified list of possible instruction types. Other datasets, such as Vicuna [21], use human-generated instructions, but still rely on language models to produce the respective responses. Without sufficiently broad and high quality data, even models with substantial size and pre-training would be inadequate for building capable, helpful, and harmless AI assistants.

Research in this area has predominantly been confined to a select few research labs with access to the required resources to engage in large-scale training and data collection. This monopolization of access to quality data undermines the potential for inclusive and diverse research endeavours, particularly in relation to alignment challenges, which arguably constitute some of the most crucial research areas of our time. In an effort to democratize research on aligning large language models, we introduce and release the OpenAssistant Conversations dataset. This dataset is the culmination of an extensive open- and crowd-sourcing initiative, and its release to the research community seeks to promote more inclusive research in this highly-influential domain. We provide a comprehensive analysis of the dataset, assessing ethical implications and safety considerations. We also fine-tune and release several assistant and preference models to further advance open access and research in this area. This transparency allows for iterative improvements on the released artifacts, fostering a more collaborative and inclusive research environment. By providing such a large and diverse dataset, OpenAssistant Conversations opens up new avenues of research in the field, enabling researchers to explore the complexities of human language and interactions in ways that were not possible before [22]. In the following sections, we delve into the intricacies of the OpenAssistant Conversations dataset and discuss its implications for the alignment of large language models and for society at large.

2 Data Format

The proposed dataset consists of a list of conversations. The basic data structure is a *Conversation Tree (CT)*, with nodes representing written messages in a conversation. A CT’s root node represents an initial prompt, given by the prompter. To avoid confusion, we call the roles of the conversation *prompter* and *assistant*. This allows us to reserve the term *user* for the human contributors. Although our efforts focus largely on human contributions, both the prompter and assistant roles can, in principle, be fulfilled by either a human user or a machine. Every tree node is labelled by its role, and can have multiple children of the opposite role, each of which represents a separate next step in the conversation. A path from the root to any node in the tree (including to itself) is called a *thread*, and it represents a valid conversation with the prompter and the assistant taking turns. Tree nodes are annotated with additional data such as user-provided labels and metadata, such as collection timestamp and indicated language. Each *assistant* node further has a rank associated which orders it compared to replies of the parent prompt, according to user preferences.

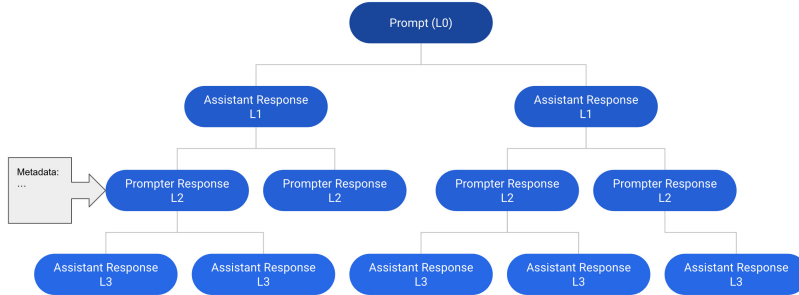


Figure 1: An example CT of depth 4 containing 12 messages. Any path from the root prompt to a node is a valid thread.

3 Data Collection

The OpenAssistant Conversations dataset is a comprehensive collection of conversational data that was obtained through a crowd-sourcing effort involving more than 13,000 volunteers. The data was collected using a web-app interface, dividing the collection of each tree into five separate steps: *prompting*, *labelling prompts*, *adding reply messages as prompter or assistant*, *labelling replies*, and *ranking assistant replies*. Users were fully informed about contributing to a public dataset. The dataset was curated with content moderation and spam filtering as key components of the annotation pipeline, ensuring high quality and safety standards.

Volunteers completed over 625,000 tasks in total, resulting in the collection of over 10,000 fully annotated and filtered Conversation Trees. Example User Interface (UI) displays of the data collection platform can be found in Appendix C, and current collection parameter settings, such as the number of collected replies to any parent message, can be found in Appendix G. In the following sections, we provide more details regarding the various aspects of the data collection pipeline.

3.1 Single-Step Collection

Data collection is structured to be both efficient and effective by breaking the work down into single units and advancing multiple conversation trees one step at a time. This approach minimizes data loss due to user attrition and ensures that every unit of work is captured for utilization. The users are presented with a range of task types, either by choice or through random sampling (weighted according to current requirements). The task types include creating prompts, replying as an assistant, replying as a prompter, labeling prompts or replies, and ranking prompter or assistant replies.

Create a prompt. Users write an initial prompt that forms the root of a new conversation tree. As this task is particularly popular, a lottery system manages the selection of new prompts, with only a fixed number of prompts being chosen for continuation at any given moment. This method serves to regulate the influx of new prompts and maintain a balanced distribution of tasks.

Reply as assistant. Replying as an assistant is a more labor-intensive task that necessitates users to carefully consider their responses and often engage in external research to provide a helpful and

relevant answer to the prompter’s request. This task type, despite its demanding nature, has been reported to be the most enjoyable by many users due to the diverse array of topics covered. To account for the increased effort required for this task, a reward system has been implemented to incentivize users to participate. See Figure 6 for a UI preview.

Reply as prompter. The task of replying as a prompter, on the other hand, does not impose strict quality requirements but instead emphasizes on the importance of diversity to accommodate various use-cases. Examples of prompter replies may include asking for clarification, modifying the original intent, posing a follow-up question, or changing the direction of the conversation altogether.

Label a prompt or reply. Users are presented with a message from the database along with the preceding conversation thread (if available) and are asked to categorize the message according to three dimensions: spam detection, guideline adherence, and quality. For spam detection, users assess whether the message is unsuitable for inclusion in the dataset, for instances of obvious spam or trolling. Messages flagged as spam by multiple users are automatically removed from the dataset.

Guideline adherence is evaluated through a set of labels that determines whether the contribution aligns with the established guidelines (see Figure 4). These labels encompass the message being in a language other than the specified one, containing personally identifiable information, hate speech, sexual content, or being deemed inappropriate. Messages labelled in this manner are subsequently reviewed by human moderators.

Quality labels require users to rate the message on a five-point *Likert* scale across dimensions such as quality, creativity, humorousness, politeness, and harmlessness. These labels are stored for later analysis and application. Notably, users can voluntarily assign these labels (as well as spam & guideline adherence labels) to any message within the system, even as part of another task, as an additional contribution.

Rank assistant replies. Users are presented with two or more responses to the same parent message and asked to rank them in order of preference. This allows for a comparative analysis of the various responses and helps in identifying the most effective and engaging replies (Figure 5).

In summary, this data collection methodology effectively divides work into single units, minimizes data loss due to user attrition, and captures valuable information for future analysis and application. By offering users a diverse range of task types, the study encourages active participation and ensures the collection of rich and varied data for a comprehensive understanding of the subject.

3.2 Message Tree State Machine

The tree state machine serves as a systematic approach to managing the progression of message trees throughout the data collection process. This method ensures that each tree undergoes a series of states until it reaches completion, beginning with the creation of new trees by randomly sampling from the pool of initial prompts. The various states that a message tree passes through include the *initial prompt review state*, *growing state*, and *end state*, as well as the *aborted low-grade state* for trees that are deemed unsuitable for inclusion in the dataset and the *halted by moderator state* for trees that have manually been halted by a community moderator.

Upon the creation of a new tree, it enters the *initial prompt review state*, where multiple users are tasked with providing labels to assess its quality and suitability. This state plays a crucial role in identifying any potential issues with the initial prompt, and demands special attention, as the entire rest of the tree (potentially several dozens of tasks) is rooted in the initial prompt. If, at this point, the provided labels indicate that the tree contains spam or unsuitable content, it is transitioned to the *aborted low-grade state* and subsequently removed from the dataset. Conversely, if the tree passes the *initial prompt review state*, it proceeds to the *growing state*.

The *growing state* involves the continuous issuance of tasks to users, such as providing replies, labels, and rankings, to facilitate the development and expansion of the conversation tree. This state is essential for collecting diverse and rich data, as it allows for the accumulation of multiple interactions and the exploration of various conversation paths, given the same initial prompt. The *growing state* continues until the tree reaches its *end state*, which is defined by a maximum number of messages or other predetermined criteria.

Parameters within the data collection platform govern the behaviour of the tree state machine, such as the average number of messages collected for each parent message or the maximum tree depth. These

parameters enable researchers to fine-tune the data collection process according to their specific research goals and requirements, ensuring a more targeted and efficient approach to gathering data. Parameters varied during the collection of the dataset. Current settings can be found in Appendix G.

In summary, the tree state machine is a structured and systematic method for managing the progression of message trees during the data collection process. By guiding each tree through a series of states, from initial prompt review to growing and reaching its *end state*, the tree state machine ensures the collection of high-quality, diverse, and relevant data. Additionally, the inclusion of platform parameters allows for the customization of the data collection process to align with specific research objectives, further enhancing the effectiveness and utility of this approach.

3.3 Contributor Guidelines

To achieve a high degree of quality and consistency across a wide range of contributors, we issue clear and detailed guidelines. A full copy of these guidelines at the present time can be found in Appendix A. Our guidelines follow three main goals: First, clarify the meanings, scales, and criteria for assigning labels and rankings during the labelling and ranking tasks. Second, make assistant responses be polite, helpful, concise, friendly, and safety-aware and third, instruct prompts and prompter replies to explore a diverse and challenging set of inputs to the assistant role.

The guidelines establish a framework for safely interacting with an automated assistant by drawing inspiration from the concept of *informed consent*. Rather than categorically denying large parts of request categories, we aim to provide the prompter with useful feedback, for example drawing special awareness to dangerous activities, elaborating on weaknesses of automated assistants, such as hallucinations, and discouraging and denying requests asking for illegal or highly inappropriate content. In our validation experiments in training assistant models based on OpenAssistant Conversations, we observe a high degree of consistency of the trained models’ outputs with our given guidelines.

Although guideline adherence is already high in our models after training, our approach is completely compatible with deploying additional safety measures during inference, such as secondary models to filter or modify ill-suited user input.

3.4 Quality Control & Content Moderation

We take a multi-pronged approach to quality assurance, with the main pillars being a system of reward points & leaderboards, and manual review of flagged content by human moderators. This both maximizes the quality of contributions, while effectively utilizing the limited time of the volunteer moderators. In an effort to demonstrate progress and achievement to users, and to encourage high-quality contributions, our system allocates points for the completion of tasks. These points contribute to various leaderboards, including daily, weekly, monthly, and all-time rankings. A level system also exists, wherein higher point accumulation results in elevated levels, reflecting veteran status and engagement. In the future, this system could potentially be developed further to facilitate preferential access to more engaging tasks or similar perks.

The distribution of points is contingent upon task type, as certain tasks require greater effort, such as the *reply as assistant* task (compared to the *create a prompt* task). A significant portion of points is deferred and reliant on interactions with other users. For instance, a user’s assistant reply may gather many additional points if it is subsequently deemed non-spam and highly ranked by other users. Inversely, points may be reduced or lost for answers that are labeled as spam or down-voted by consensus of other users.

Within the moderator section of the website, an alternative leaderboard, designated the *Trollboard*, is exhibited. This leaderboard assesses users based on an aggregate of negative labels, reports, and down-votes received for their contributions. This approach enables human moderators to proactively scrutinize potentially misbehaving users in a comprehensive manner. The Trollboard has proven to be an effective tool in addressing the numerical disparity between users and moderators, maximizing the collective efforts of contributors to identify undesirable contributions.

Users further have the option to report messages to moderators for manual review, either via the platform, or directly via communication on a community chat server. Moderators have the ability to delete individual messages, or all messages of a given user, at their own discretion. Deleted messages are retained, but marked as deleted and are not exported for training.

4 Dataset Composition

The full dataset consists of 161,443 messages (91,829 prompter and 69,614 assistant messages) distributed across 66,497 conversation trees, in 35 different languages, annotated with 461,292 quality ratings. This includes 8,576 synthetic messages, leaving 152,867 human-submitted messages. Of the 66,497 total conversation trees, we consider 10,968 complete, meaning the full number of messages has been collected and the moderation process for these trees has been concluded. 52,159 incomplete trees are in the prompt lottery state, meaning they only consist of a single initial prompt. The completed trees contain 92,365 messages.

The set of categories for which *Likert*-scale human labels are collected is Creativity, Quality, Humor, Helpfulness, Violence, and Rudeness. The set of categories for which binary human labels are collected is Language Mismatch, Not Appropriate, Personally Identifiable Information, Hate Speech, and Sexual Content. We additionally release the rank of each assistant message compared to other assistant messages submitted for the same prompt, computed from the preference rankings of several human annotators. To merge the rankings of multiple (possibly conflicting) annotators, we use a variant of Tideman’s method, described in Appendix B.

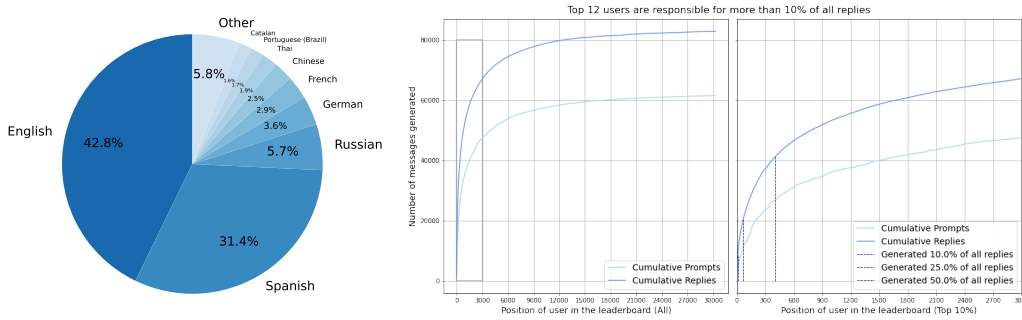


Figure 2: **Left:** Relative share of the most frequent languages in the dataset. **Right:** Distribution of contributions among users.

The dataset is dominated by English and Spanish messages as illustrated in Figure 2 (left). The prominence of English is expected as a result of the community around OpenAssistant originating in the English-speaking open-source machine learning community. The high quantity of Spanish messages can be attributed to the publicity given to OpenAssistant by prominent figures in the Spanish machine learning community. Figure 2 (right) illustrates how a small number of power users contributed a significant proportion of the dataset. This must be taken into account when considering possible biases in the data. Although significant effort went into preventing responses directly copied from other sources, it is possible that some users utilised automated techniques to enter data.

We release the dataset on the Hugging Face Hub⁴ in several variants: One variant containing all collected initial prompts, one variant containing all trees that are considered completed, one variant containing all trees (completed and in-progress), and one variant containing messages filtered out as spam. For most purposes, such as instruction-tuning of language models, we recommend using the variant containing all completed trees, and include non-completed trees if more data is required.

5 Contributor Demographics and Satisfaction

To gain a deeper understanding of the contributors’ demographics, a Google Form survey was sent out as an announcement on the project’s Discord channel. The survey consists of 3 parts with questions on demographic information, personal motivation and user satisfaction. At the time of the release of this paper, a total of 270 survey answers have been collected. Results can be seen in Figures 7,8,9,10, and Tables 3 and 4 (all in the Appendix). Respondent answers indicate a homogeneity towards identifying as male, but strong diversity in the reported levels of education, stated motivations for contribution, proficiency in artificial intelligence, and use cases for the technology. Over 95% of respondents either agreed or strongly agreed with the statement "Overall, I’m glad I have contributed

⁴<https://huggingface.co/OpenAssistant/oasst1>

to OpenAssistant." About 40% reported this being their first time contributing to an open-source project. We note that the method of recruiting via Discord is biased towards users who are present on the platform and have been active around the time of the announcement. Active contributors are also expected to be more likely to respond.

6 Experimental Validation

6.1 Instruction Tuning & Preference Modeling

We focus on the development and evaluation of fine-tuned language models based on Pythia [3], LLaMA [2], and Falcon [23]. Pythia and Falcon are state-of-the-art language models with permissive open-source licenses, while LLaMA is a powerful language model with a bespoke non-commercial license. Specifically we train supervised fine-tuned (SFT) models, reward models (RM [13],[24]), and, using the trained reward models, reinforcement-learned models (RLHF)⁵.

| Model | LMEH | VEL | OAIE | HE |
|---|-------|------|------|------|
| gpt-3.5-turbo (ChatGPT) | | 1110 | 0.87 | 0.72 |
| EleutherAI/pythia-12b | 60.33 | | | |
| OpenAssistant/pythia-12b-sft-v8-7k-steps | 60.28 | 997 | 0.10 | 0.10 |
| tiiuae/falcon-40b | 72.29 | | | |
| OpenAssistant/falcon-40b-sft-top1-560 | 74.04 | 1192 | 0.26 | 0.09 |
| OpenAssistant/falcon-40b-sft-mix-1226 | 74.40 | 1053 | 0.44 | 0.13 |
| huggyllama/llama-65b | 67.24 | | | |
| OpenAssistant/oasst-sft-7e3-llama-30b | 68.03 | 979 | 0.52 | 0.20 |
| OpenAssistant/oasst-rlhf-3-llama-30b-5k-steps | 68.51 | 1068 | 0.51 | 0.15 |

Table 1: Comparison of model evaluation scores on different LLM benchmarks: **LMEH**: Im-evaluation-harness [25] (average scores, see online leaderboard for more details) **VEL**: Vicuna Elo Rank [21] **OAIE**: OpenAI Evals [26] **HE**: HumanEval [27] (for all benchmarks, higher is better). We have chosen to leave the Hugging Face Hub identifiers as the model names for identifiability.

Table 1 shows evaluation scores of a selection of baseline and trained models on a set of standard benchmark datasets. Evaluations were performed externally using FastEval and evaluation results are hosted on a leaderboard, which is continually being updated⁶. LMEH refers to the average performance on a set of widely-used NLU tasks consisting of BoolQ, PIQA, HellaSwag, WinoGrande, ARC-e, ARC-c and OBQA⁷. We omit instruction-centric experiments (VEL, OAIE, HE) for the base models, as these benchmarks are unsuitable for non-instruction-tuned models. The results show that models using OpenAssistant Conversations are consistently outperforming the corresponding baseline models (in the case of LLaMA even a larger baseline model). RLHF outperforms SFT in some benchmarks, but not in others. For Falcon-based models, *sft-top1* is trained only on top-ranked conversation threads, whereas *sft-mix* mixes OpenAssistant Conversations with other instruction datasets (details in Appendix H). The varied evaluation scores demonstrate that by combining different data sources, the nature of the resulting model can be readily influenced. Ranks across benchmarks are not consistent, which could indicate the unsuitability of automatic evaluations for language models, or could indicate that different models and datasets lead to different capabilities. The results also show that while open-source models are close to matching ChatGPT in some benchmarks, others still show large performance gaps. Anecdotally, users report OpenAssistant models to be less robotic and more human-sounding than commercial models and report generations to have high quality and diversity in domains such as creative writing, conversational messaging, and drafting social media posts.

⁵All trained models are released at <https://huggingface.co/OpenAssistant>

⁶<https://github.com/FastEval/FastEval>, <https://tju01.github.io/ilm-eval/>

⁷For readability, Table 1 contains aggregated LMEH scores. Details in the online leaderboard.

6.2 Spam and Toxicity

To understand the concordance between human and automated toxicity detection, we employ toxicity detection methods based on **Detoxify** [28] to obtain automated ratings for six distinct categories, classifying whether a message is toxic, obscene, threatening, insulting, attacking a certain identity or sexually explicit. We limit our analysis to those languages that are supported by the toxicity detection method, covering English, Spanish, Russian, French, and Italian. These languages represent the majority of OASST1 messages (over 83%).

Using automated toxicity ratings, we are able to systematically assess the correlation between these ratings and human-assigned toxicity labels (hate speech, not appropriate, and sexual content). Based on a sample of 115,153 messages, we compute the correlation between automatic and human-annotated toxicity labels, which is visualized in Figure 3. We see a correlation between human and automatic labels in at least one element of each row and column of the correlation matrix, suggesting agreement between human annotators and off-the-shelf toxicity detection models. The results serve to both validate the capabilities and show limitations of AI-driven toxicity detection in comparison to human judgement and may inform future work in this area.

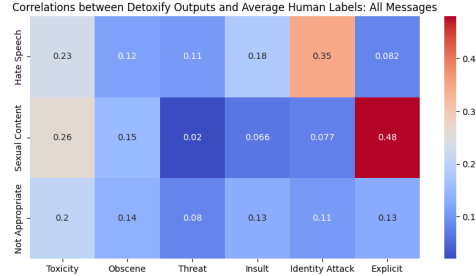


Figure 3: Correlation between human labels and Detoxify outputs for all messages in Detoxify-supported languages.

In addition to analysing the correlation between human-assigned toxicity labels and automated ratings, we extend the application of the Detoxify model to assess the efficacy of the moderation process for the same languages described earlier. To facilitate this analysis, we define two categories of messages: *deleted* messages, which encompass those that either failed to pass the community moderation process or were subsequently manually removed by moderators, and *retained* messages, which successfully made it through to the dataset. In order to provide a comprehensive evaluation of the moderation process, we calculated average values for each of the six Detoxify categories for both *deleted* and *retained* messages. The values obtained for this analysis are based on a sample of 74,781 messages. We excluded messages in trees that were incomplete at the time of export, as these messages may be subject to removal by the moderation process.

Our analysis, presented in Table 2 shows that the values for all six toxicity categories are markedly higher for *deleted* messages compared to *retained* messages. This significant difference demonstrates the effectiveness of the moderation processes in place, as messages removed from the dataset are on average rated as significantly more toxic by the Detoxify model than messages allowed to remain in the dataset.

| State | Toxicity | Obscene | Threat | Insult | Identity Attack | Explicit | N |
|----------|----------|---------|--------|--------|-----------------|----------|-------|
| Deleted | 4.625% | 1.965% | 0.411% | 2.085% | 0.651% | 1.39% | 3422 |
| Retained | 0.988% | 0.574% | 0.102% | 0.715% | 0.121% | 0.177% | 71359 |

Table 2: Detoxify outputs across six categories of toxicity, comparing *deleted* and *retained* messages.

While *deleted* messages are rated as more toxic than *retained* messages by the Detoxify model across all categories, the average toxicity values for these messages are still small. This implies toxicity ratings from models like Detoxify alone are not sufficient to determine when messages are unsuitable for inclusion in the dataset. Reasons for deleting non-toxic messages may include a lack of factual accuracy, or poor grammar. Additionally, messages which are children of deleted messages must themselves be deleted even if they appear to be acceptable in isolation.

7 Limitations

Reward model data collection. InstructGPT [13] trained reward models on ranking data of messages generated by their initial SFT model, while our reward models are trained using ranking data of human-generated messages. We chose to do so because we were already collecting this ranking data as part of our general efforts, for use in quality control, spam filtering, and dataset sub-sampling. While subjectively, many users report our RLHF models to follow instructions more closely (also compare Section 6.1), the models do not deliver the same uniform and significant improvements over SFT models as reported in [13]. We hypothesize that the difference in data collection for the reward model could at least partially explain this gap. We plan to collect ranking data based on our own SFT models in the future to verify these assumptions. Further research is necessary to determine more precise criteria for collecting data useful to RLHF.

Subjective, Cultural, and Contribution Frequency Biases. The open nature of our project introduces a unique set of challenges when it comes to controlling for biases within the dataset. Annotators from diverse backgrounds contribute to the dataset, with demographics that are simultaneously heterogeneous in some dimensions and homogeneous in others (see Section 5). Specifically, 89.1% of the annotators identify as male, with a median age of 26. This demographic profile may inadvertently introduce biases in the dataset, as it is bound to reflect the values, perspectives, and interests of the annotators. (We pose that some of this could be mitigated by introducing more constrained conversations, for example sampling a random Wikipedia page to determine the conversation topic.) Further, users’ participation levels differ significantly. More engaged users contribute a greater number of annotations (see Figure 2), which likely leads to over-representation of their values and interests in the dataset. Consequently, the dataset may not adequately capture the diverse perspectives that a more balanced distribution of contributions could have provided. Further research is necessary to determine the effect of uneven contributor distributions have when given a clear, general task.

Possibility of Unsafe Content. While we have implemented measures to detect and remove harmful messages, our system is not infallible. It is possible that the dataset still contains unsafe content. We believe that the open nature of the project allows for data filtering to be conducted in a transparent manner, ultimately converging on the highest possible standards. Nevertheless, the potential presence of residual unsafe content in the dataset necessitates careful evaluation of any models trained on it.

Given the limitations discussed above, we advocate for the use of our models in academic research contexts only. We strongly encourage researchers to thoroughly investigate the safety and bias of any model before employing it in downstream tasks. The released models may exhibit unsafe behavior and are likely susceptible to prompt injection attacks. The alignment of LLMs is a crucial aspect of AI research, and we hope that our contributions can help advance the field of AI alignment. However, we also acknowledge that current alignment techniques are not perfect and can even exacerbate certain biases [29]. As such, researchers should exercise caution when using these models and be cognizant of their limitations. Additionally, it is essential to continue refining alignment techniques and advancing the field of AI alignment in order to mitigate these limitations and develop more reliable and robust LLMs.

8 Safety and Ethical Implications

Large language models are prone to generating inaccurate information about people, places, or facts, commonly known as ‘hallucinations’ [30, 31]. LLMs can also produce toxic or hateful content and fail to follow provided constraints [32]. Additionally, these models tend to incorporate biases present in their training data, leading to unfair and discriminatory outputs [33]. While methods such as *RLHF* can mitigate some of these shortcomings, they may exacerbate others [34, 29]. We hope that alignment methods using OpenAssistant Conversations can fix some of these issues [13], but we acknowledge that achieving full alignment is a complex and ongoing challenge.

We recognize that sufficiently powerful language models can have a significant impact on society [35], and therefore we believe it is essential to promote transparency in their development and deployment. OpenAssistant Conversations is our contribution to this goal of transparency.

Acknowledgments and Disclosure of Funding

Our greatest thanks go to the many volunteer contributors, of human data, code, moderation, documentation, and community organization. Absent of any financial incentives, this project is a stunning and unprecedented display of global cooperation of humans for the purpose of advancing and democratizing AI research. In addition, several organizations have contributed to this project with resources: Redmond AI provided training compute. Stability AI and Hugging Face provided inference compute. We thank Olivier Dehaene at Hugging Face for close collaboration and personal support. Weights & Biases provided their full MLOps solution to the entire team. LAION provided legal input and acts as the website addressee. We thank Til Jasper Ullrich for running LLM benchmark evaluations, Luke Thomas Kaiser for running evaluations on model bias, and Silvia Pareti for detailed feedback on the manuscript. Agradecemos a Carlos Santana por la promoción a gran escala de la plataforma de recopilación de datos para la comunidad hispanohablante.⁸

References

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [3] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*, 2023.
- [4] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3):411–437, September 2020.
- [5] Yufei Wang, Wanjuan Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey, 2023.
- [6] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.
- [7] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with \mathcal{V} -usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR, 17–23 Jul 2022.
- [8] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm,

⁸Translated from English by OpenAssistant, first try

- Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agueri-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022.
- [9] Jacob Hilton, R Nakano, S Balaji, and John Schulman. Webgpt: Improving the factual accuracy of language models through web browsing. *OpenAI Blog*, December, 16, 2021.
 - [10] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.
 - [11] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020.
 - [12] Ratanond Koonchanok, Ya-Chen Pan, and Hyeju Jang. Tracking public attitudes toward chatgpt on twitter using sentiment analysis and topic modeling. *ArXiv*, abs/2306.12951, 2023.
 - [13] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
 - [14] Leandro von Werra, Jonathan Tow, reciprocated, Shahbuland Matiana, Alex Havrilla, cat state, Louis Castricato, Alan, Duy V. Phung, Ayush Thakur, Alexey Bukhtiyarov, aaronrmm, Fabrizio Milo, Daniel, Daniel King, Dong Shin, Ethan Kim, Justin Wei, Manuel Romero, Nicky Pochinkov, Omar Sanseviero, Reshith Adithyan, Sherman Siu, Thomas Simonini, Vladimir Blagojevic, Xu Song, Zack Witten, alexandremuzio, and crumb. CarperAI/trlx: v0.6.0: LLaMa (Alpaca), Benchmark Util, T5 ILQL, Tests, March 2023.
 - [15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
 - [16] Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. The effects of data quality on machine learning performance, 2022.
 - [17] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.
 - [18] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
 - [19] Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. <https://github.com/nomic-ai/gpt4all>, 2023.
 - [20] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
 - [21] Vicuna. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. <https://vicuna.lmsys.org/>, 2023.
 - [22] Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren Resnick, Levine B., M. John, Stephanie Teasley, and D., editors, *Perspectives on Socially Shared Cognition*, pages 13–1991. American Psychological Association, 1991.
 - [23] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance, 2023.

- [24] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021.
- [25] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021.
- [26] OpenAI. Gpt-4 technical report, 2023.
- [27] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgén Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.
- [28] Laura Hanu and Unitary team. Detoxify. Github. <https://github.com/unitaryai/detoxify>, 2020.
- [29] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Mari-beth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.
- [30] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*, 2019.
- [31] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.
- [32] Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. Ethical challenges in data-driven dialogue systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2018.
- [33] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.
- [34] Ethan Perez, Sam Ringer, Kamilė Lukošiušė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations, 2022.

- [35] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.
- [36] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023.
- [37] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.
- [38] T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, September 1987.
- [39] David Rozado. The political biases of chatgpt. *Social Sciences*, 12(3):148, Mar 2023.

Appendix

Table of Contents

| | |
|--|-----------|
| A Contributor Guidelines | 14 |
| B Ranking merging | 18 |
| C Web application for data collection | 19 |
| D Filtering ChatGPT Inputs | 22 |
| E Online Survey Results | 22 |
| F Word Cloud | 27 |
| G Collection parameters | 27 |
| H Training Configuration | 29 |
| I Evaluation Tasks | 30 |
| J Political Compass Evaluations | 30 |
| K Community Engagement | 31 |
| L Dataset Documentation (Data card) | 34 |
| M Author Statement | 34 |

A Contributor Guidelines

We provide the guidelines presented to the users for the creation of the dataset.

Guidelines

Below is a list of guidelines that should be adhered to for each possible task available when building the dataset. To see some examples of how the guidelines can be applied, visit the examples document.

Please consider checking out our survey [here] (<https://forms.gle/vBW7b2kMzjCoehkH9>). You can use it to rate each guideline and leave feedback for each task.

If you have further suggestions to improve any of our guidelines, or want to add more examples, create a pull request or suggest them on our [GitHub] (<https://github.com/LAION-AI/Open-Assistant>).

1. General rules

- Always make sure to read and understand the guidelines to each task before fulfilling it.
- Try to follow the guidelines as closely as possible.
- If you are unsure whether a message violates a guidelines, contact us at our Discord.

- Use the thumbs-up/thumbs-down system to further mark messages that are of high or low quality.

2. Providing an assistant reply {#assistant-reply}

Do:

- Remain polite and treat the user with respect, even when not given the same courtesy.
- Talk in a friendly and approachable manner, unless specifically requested otherwise.
- Present only information that has been verified by credible sources that can be backed up, unless specifically requested otherwise.
- Make sure the user is aware when given unverified information.
- Inform the user about the potential dangers when being asked for advice regarding a topic with high risk, such as medicine, law or chemistry.
- When being asked about a high-risk topic, make sure the user knows that as a language model, the assistant is susceptible to producing incorrect information, and that no actions should be taken regarding the assistant reply without the opinion of a professional.
- When being asked to give an opinion as the default persona of the assistant, make sure to bring up at least 2 common viewpoints and ensure that these aren't expressed as the opinions of the assistant.
 - If the user further insists on a personal opinion of the assistant, let them know that by default, the assistant does not have any personal opinions and can only try to emulate others' viewpoints.
- Ask for clarification if it's unclear what the user is asking for.
- Use paragraphs and line breaks to make larger replies more readable.
- Make use of [Markdown syntax](https://www.markdownguide.org/basic-syntax) to better format lists, tables or blocks of code.
 - If you are using a codeblock to write code in a particular language, specify it to enable [syntax highlighting](https://www.markdownguide.org/extended-syntax/#syntax-highlighting). You can find all supported abbreviations [here](https://github.com/jincheng9/markdown_supported_languages#heres-a-full-list-of-supported-languages).
- Be consistent in the style and tone of the assistant.

Don't:

- Copy and paste text from other sources without editing. ****This includes ChatGPT.****
- Supply text that violates the law of Germany, UK, USA, or your country of residence.
- Write content encouraging:
 - Violence
 - Violation of the rights of a third party
 - Pedophilia
- Provide the user with information that could be used for self-harm if there is plausible suspicion of intent to self-harm.
- Provide personal information of third parties that isn't publicly available.
- Ask for personal information unless it is relevant to the issue and can't be used to determine the identity of the user, such as country of residence or occupation. The user should be allowed to refuse to give up any information.
- Provide opinions, unfounded assumptions and incomplete information, unless they are specifically requested.
- Purposefully curate information to guide the conclusion, i.e. don't hide facts to present a particular narrative.

- Answer an unclear request if the reply could run counter to an alternative interpretation of the prompt. Ask the user to elaborate or rephrase instead.
- Dodge a question, unless it violates a guideline.
- Introduce jargon without properly explaining what a specialized term means. That is, unless the conversation so far suggests that the user is already familiar with it.
- Leave typos or grammatical errors in the assistant replies, unless specifically requested to do so.
- Overload the user with too much information. Keep replies concise, but include further details that relate to and expand upon the user's request.
- Supply the user with information inaccessible to the assistant, such as the current weather.
- Reply in a language different from the one intended for the dataset, unless specifically requested to do so.

3. Providing an initial prompt or user reply {#user-reply}

Do:

- Ask questions that reflect real-life situations and needs.
- Ask questions that might be directed towards search engines or specialists.
- Make requests that encourage lateral thinking and/or require specialized knowledge.
- Use a mix between questions that are straightforward and questions without a clear answer.
- Introduce a variety in prompts by using different phrasing, degrees of politeness or amount of context given.
- Consider the previous replies and prompts that lead up to the current one.
- Try to build upon the topic and ask a sensible follow-up question when replying to the assistant.

Don't:

- Write prompts without a clear request.
- Supply text that violates the law of Germany, UK, USA, or your country of residence.
- Make requests that override the original purpose of the assistant, i.e. jailbreak the model.
- Make requests that leave the assistant with no other choice but to refuse in order to avoid the generation of harmful content.
- Submit a prompt similar or identical to a prompt you previously submitted.
- Change the topic of a conversation without prefacing it accordingly when replying to the assistant.
- Leave typos and grammatical errors in the prompt.
- Reply in a language different from the one intended for the dataset, unless the context of the conversation requires it.

4. Classifying an assistant reply {#classifying-assistant}

Do:

- Rate every criteria of each reply, unless it can't be discerned because it is spam or inappropriate.
- Judge quality based on how well the reply adheres to the guidelines. Factual accuracy and helpfulness are first and foremost.
- Make sure to read the reply thoroughly.
- Use the [label explanations] (#label-explanation) to determine which labels apply to the reply.
- Research to make sure whether the reply is factually accurate.

- Skip a classification if you are unable to determine the validity of reply.

Don't:

- Judge quality based on personal beliefs. Assuming an opinion was warranted, fulfills the users request and doesn't violate any guidelines, it should not impact the rating of the reply.
- Skip a label just because the reply is spam. Each label can help the model improve.
- Rate a reply if you are unsure if it factually accurate or satisfies the request of the user.

5. Classifying an initial prompt or user reply {#classifying-user}

Do:

- Rate every criteria of each prompt, unless it can't be discerned because it is spam or inappropriate.
- Judge quality based on how well the prompt adheres to the guidelines.
- Make sure to read the prompt thoroughly.
- Use the [label explanations] (#label-explanation) to determine which labels apply to the prompt.

Don't:

- Judge quality based on personal beliefs. The opinion of the user should not impact the rating of the prompt.
- Skip a label just because the reply is spam. Each label can help the model improve.

6. Ranking assistant replies {#ranking-assistant}

Do:

- Make sure to read every available reply.
- Think about which reply best satisfies the request of the user.
- Rank replies based on how well they adhere to the guidelines. Factual accuracy and helpfulness are first and foremost.
- Penalize replies that fail to provide adequate warnings or caveats.
- Penalize replies that are difficult to read due to a lack of formatting, capitalization or other errors.
- Penalize replies if the requested information is obfuscated by superfluous details that make up a large part of the message.
- Rank replies that admit to not knowing the answer below factually correct, but above factually incorrect replies.

Don't:

- Rank replies based on personal beliefs. Assuming an opinion was warranted, fulfills the users request and doesn't violate any guidelines, it should not impact the rating of the reply.
- Rank replies based on how long and short they are - instead, find out which reply best answers the query of the user.

7. Label Explanation {#label-explanation}

****Is the message spam?****

Is the message an unrelated piece of text that doesn't contribute to the

previous prompt in any way? If it's an initial prompt, is it something unlikely to be used to initiate a conversation with the assistant?

****Is it a bad reply, as an answer to the prompt task?****

Does the assistant reply fail to engage the request of the user or does it contain incorrect information? Does it ignore relevant parts of the prompt?

Tags

****Not Target Language**:** Unless specifically requested, is the language used different from the target language of the dataset?

****Not Appropriate**:** In case of a user prompt, is it missing a clear request?

****Contains PII**:** Does the message contain personally identifying information (PII) that isn't publicly available and can be used to determine the identity of the user or a private third party?

****Hate Speech**:** Does the message contain prejudice towards a group classified as protected according to the law of Germany, UK, USA or your country of residence?

****Sexual Content**:** Does the message contain or request sexually explicit content, such as descriptions of sexual acts?

Scales

****Low Quality/High Quality**:** How well does the message adhere to the guidelines? For assistant replies, factual accuracy and helpfulness are first and foremost.

****Serious/Humorous**:** Does the message contain sarcasm, a play on words or other humorous embellishments?

****Ordinary/Creative**:** How unlikely is the question to be asked? Is it challenging the assistant with a problem that requires creative thinking? Does the question include lots of context? On the assistant's side, is the reply straightforward, or does it contain a unique approach to a problem?

****Rude/Polite**:** How polite is the message towards the recipient? Does it use words like "please" or is it dismissive, rude or insulting towards the other side of the conversation?

****Violent/Harmless**:** Does the message contain descriptions of explicit violent acts? Is it glorifying, encouraging or downplaying violent behavior?

B Ranking merging

Reinforcement learning from human feedback [36, 37] comprises a set of techniques that all aim to optimize the output distribution of a language model using the preference structure provided by human rankers. To get a preference structure that is well aligned to users, we cannot just rely on the opinions of individual rankers, due to the high variance in human preferences. Since our objective is to collect data for a generally capable digital assistant, every ranking of possible responses is performed by K independent rankers, (see, section 3.1). Once this is done, we need to fuse these K individual opinions into one consensus opinion usable in training preference models.

We perform this preference fusion by treating it as a ranked-voting problem, whose objective it is to maintain the preferences as faithfully as possible. The method chosen for this is known as "ranked pairs" or "Tideman's method" [38]. Simplified, this method creates a sorted list of "winners" according to the strength of the preference of one element over the others. The way the preference strength is measured is by considering all preference pairs in the input votes: For example, if the votes are two times $A > B > C$ and one time $B > A > C$, this would mean that the pair $A > B$ exists 2 times, while $A > C$ exists three times. The method then orders the winners by winning strength, i.e. here $A > C$ would happen before $A > B$, and constructs a directed graph using the preferences, i.e. $A > C$ would become an edge $A \rightarrow C$. Edges are added one by one according to their weight (higher first), as long as no cycle is produced. If the edge would produce a cycle, it is skipped as the corresponding dominant preferences are already in the graph.

The resulting directed acyclic graph can be turned into a preference structure by recursively removing source nodes and adding them to the back of a preference list, since source nodes are, by construction, not preferred over any other item left in the graph. In practice, one can speed up the construction by not explicitly constructing the preference graph and fusing the graph con- and destruction.


C Web application for data collection

Figures 4, 5, and 6 show the data collection platform UI interface for labelling, reply input, and ranking, respectively. The UI also allows the user to choose a random task type, the task is automatically chosen according the previously discussed heuristics, which allows for more variation and engagement.



The UI did grow and change over time as the project developed, so the experience of early users was less refined as shown here.


Label Assistant Reply ⓘ

Given the following discussion, provide labels for the final prompt.



Why does everyone ignore the terms of service, should I read them?



 2  0 ...



Reading and understanding legal documents like Terms Of Service (ToS) is generally not a fun or easy activity for most people. Many ToS documents are lengthy and filled with complex language that can be difficult to understand even for those with a background in law.

Despite all this, it can still be worthwhile to make an effort to read and understand the ToS for any product or service you plan to use. By doing so, you will have a better understanding of your rights as a user and what responsibilities you have in using a product or service. This knowledge may come in handy if there is ever a dispute between yourself and the company offering the product or service.

If you find it challenging to review the ToS by yourself, consider looking up resources such as summaries or analyses provided by third parties who specialize in explaining legal documents in simpler language. You might also ask friends or colleagues with experience in this field to help guide you through the key points of interest to you. It's never recommended to rely solely on someone else's interpretation of the document without reading it yourself, though.

 0  0 ...

Answer the following question(s) about the highlighted message:

Is the message spam? ⓘ

Is it a bad reply, as an answer to the prompt task? ⓘ

Select any that apply to the highlighted message: ⓘ

Rate the highlighted message:

| | | |
|-------------|---|--------------|
| Low Quality | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | High Quality |
| Unhelpful | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | Helpful ⓘ |
| Ordinary | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | Creative ⓘ |
| Serious | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | Humorous ⓘ |
| Rude ⓘ | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | Polite |
| Violent ⓘ | <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> | Harmless |

Figure 4: A preview of the page for labelling tasks. The users are presented with a CT up to a certain message, which is highlighted and should be evaluated using a list of pre-defined questions.

Rank Assistant Replies

Given the following Assistant replies, sort them from best to worst, best being first, worst being last.

I would like you to create some regex to find out if the first number in a set of numbers and letters is a one. For example, in "1 month, 2 days, and 3 hours" the selected text would be "1". In "1 day, and 3 hours", it would return "1". In "5 months" it would return "5".

Sure thing. You can use Extended Regular Expression and come up with the following expression:
`*(d+).`
the previous expression will look for a sequence of digit (a number) preceded by a string that does not contains numbers so that `|d+` is the first number encountered. At this point `|` will contain the number matched by `|d+`.

How would that be implemented in Python? Which programming language would be the best to use regex with?

How would you make a code in cpp/csharp or other Lang to make pc always show up my SMS messages from phone with android and what should I do with my Phone to make IT work?

You can implement this expression in Python as follows:

```
python
import re

# define a string to search in
string = "This is a test string with 123 and some other text"

# use re.search to search for the pattern
match = re.search("*(d+).", string)

# check if a match was found
if match:
    # print the first group that matches the pattern (d+)
    print(match.group(1))
else:
    print("No match found")
```

There is no best programming language it's a personal choice that depends on you...

☐ All answers are factually incorrect and cannot be ranked

Figure 5: A preview of the page for ranking assistant replies. Users are provided with CT up to a message, and a couple of responses that should be ranked according to how good they answer the given message. The interaction is drag & drop based. Additionally, users can choose to mark all answers as factually incorrect.

Reply as Assistant

Given the following conversation, provide an adequate reply

Recommend me a winter jacket for someone who is 6 foot 5 inches tall and likes the color grey

Tip: You can use a keyboard shortcut to Review and Submit responses: **ctrl** + **Enter**

[Write](#) [Preview](#)

Finding fitting clothes depends a lot on the figure and body type, I see that you have already provided me with your height, however, other information is still missing, such as your body type.

Styling with markdown supported

Figure 6: A preview of the page for replying as an assistant. Users are provided with a CT up to a prompter message, and they should provide a response to it. In this example, the CT contains only one message. Users can use Markdown in their responses. An additional progress bar is added below the input field to incentivize longer messages.

D Filtering ChatGPT Inputs

The widespread use of ChatGPT at the time of our data collection meant that some of the inputs could be performed using ChatGPT, rather than being fully created by humans. Our guidelines specifically target this, explicitly discouraging contributors from copy-pasting responses generated by other AI models. Users who were found to post ChatGPT-generated responses were banned, and their contributions were deleted. Furthermore, we used multiple automatic tests to catch such cases. For instance, we searched for and removed messages that contained text such as "as a large language model" or "knowledge cutoff after September 2021". Moreover, users were encouraged to up-vote and down-vote responses they came across from other users, which also helped weeding out low-quality, generic, (possibly AI generated) responses. We note that these mechanisms cannot remove all ChatGPT-generated content, and more post-hoc filtering may be necessary.

E Online Survey Results

We asked users to provide feedback based on the overall experience, while taking part in the data collection process. Results are provided in Tables 3, 4 and Fig. 7, 8, 9, 10.

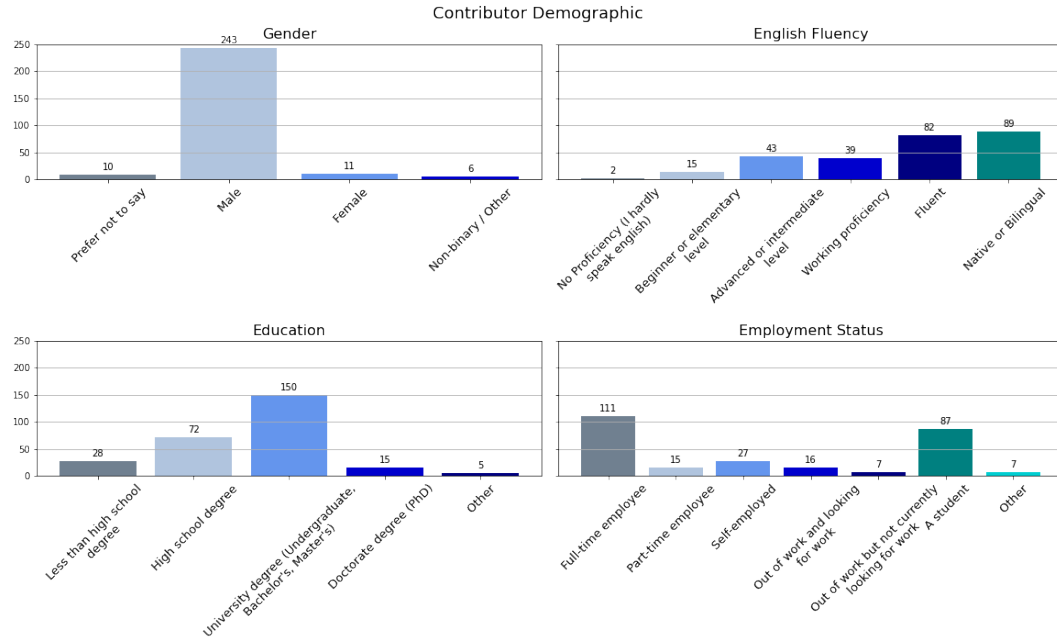


Figure 7: Demography of 270 respondents

| | |
|---|--------|
| It was clear from the website what to do and how to contribute messages to the training data. | |
| Strongly agree | 33.70% |
| Agree | 47.04% |
| Neither agree nor disagree | 11.48% |
| Disagree | 6.67% |
| Strongly disagree | 1.11% |
| I've felt I could always ask for help from the community and the moderators. | |
| Strongly agree | 30.74% |
| Agree | 27.78% |
| Neither agree nor disagree | 31.85% |
| Disagree | 7.41% |
| Strongly disagree | 2.22% |
| I found the tasks enjoyable and engaging. | |
| Strongly agree | 20.00% |
| Agree | 42.22% |
| Neither agree nor disagree | 27.78% |
| Disagree | 8.15% |
| Strongly disagree | 1.85% |
| I found the tasks repetitive. | |
| Strongly agree | 11.48% |
| Agree | 30.00% |
| Neither agree nor disagree | 37.04% |
| Disagree | 18.15% |
| Strongly disagree | 3.33% |
| While doing rating or ranking tasks, I found the messages to be really high quality. | |
| Strongly agree | 16.67% |
| Agree | 42.59% |
| Neither agree nor disagree | 30.00% |
| Disagree | 9.26% |
| Strongly disagree | 1.48% |
| Overall, I'm glad I have contributed to OpenAssistant. | |
| Strongly agree | 81.11% |
| Agree | 14.44% |
| Neither agree nor disagree | 2.96% |
| Disagree | 0.74% |
| Strongly disagree | 0.74% |

Table 3: User Satisfaction Survey

| | |
|--|-----|
| Have you contributed to other community projects besides this one? | |
| No, this is my first time contributing | 111 |
| Yes I have contributed to a few projects | 110 |
| Yes, I have contributed to multiple open source projects | 44 |
| Prefer not to say | 5 |

Table 4: Previous Contributions

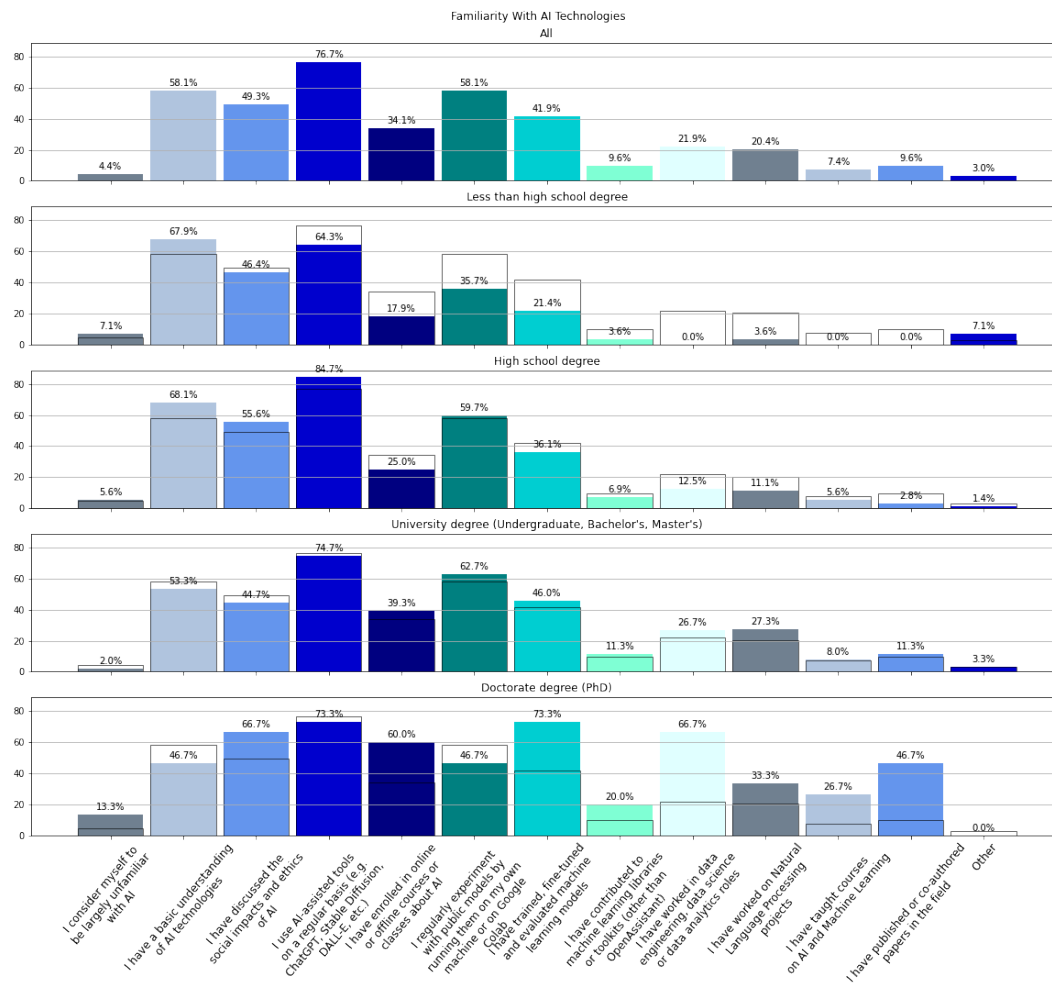


Figure 8: Familiarity with AI based on level of education. The average values are shown as outlines in the bar chart.

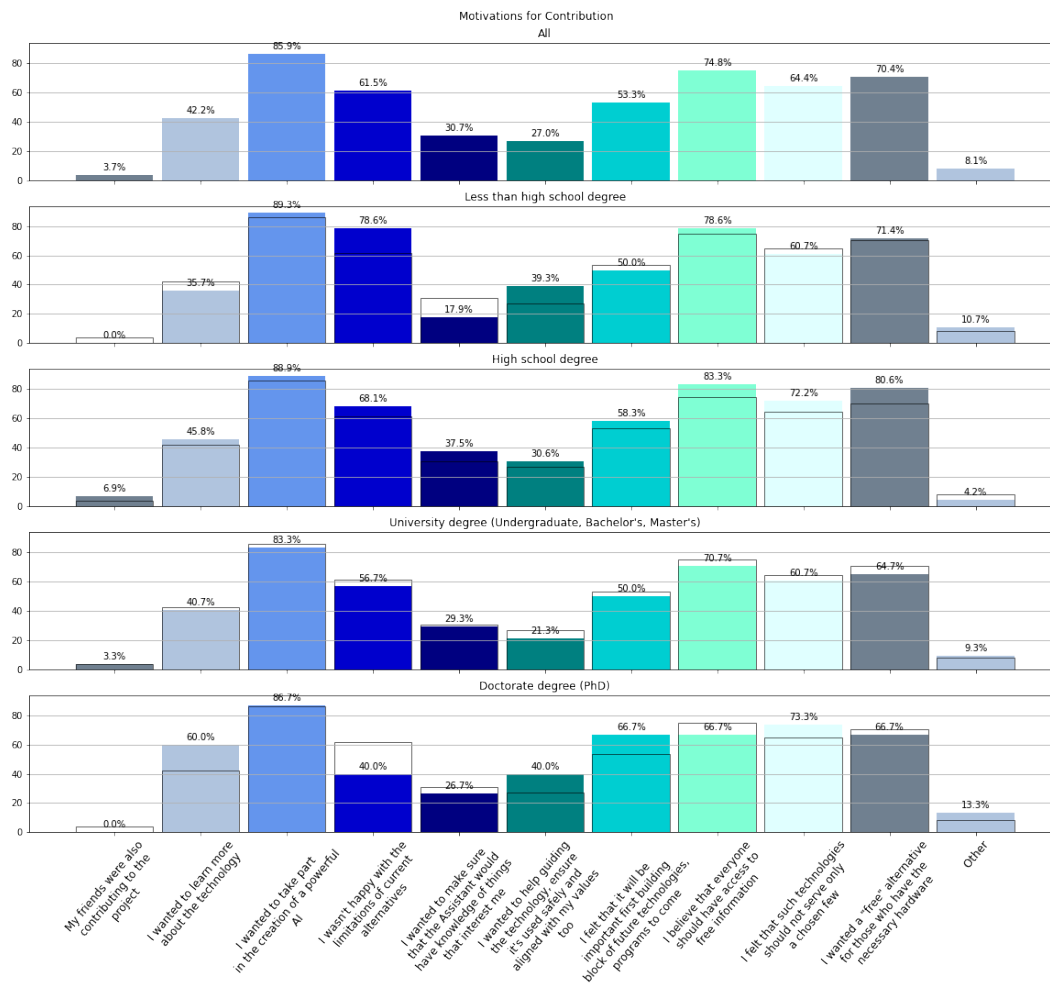


Figure 9: Motivations for contribution based on level of education. The average values are shown as outlines in the bar chart.

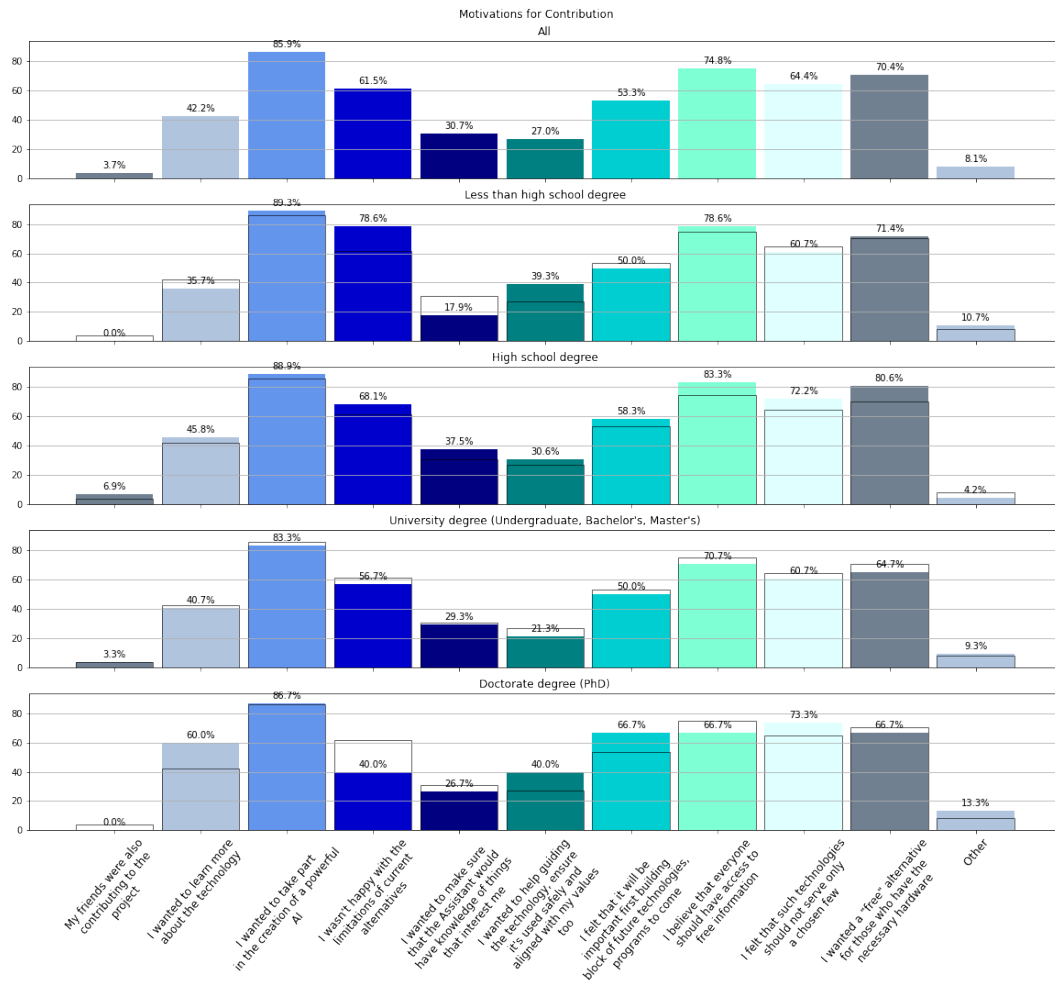


Figure 10: Planned personal use-cases for OpenAssistant. The average values are shown as outlines in the bar chart.

F Word Cloud

We present some statistics on the words used the most in Fig. 11.



Figure 11: Word clouds for 33 topics extracted from the English subset of the open-assistant dataset. The number of topics was selected by selecting the peak coherence from 40 LDA models. One can observe high topic variety from biology (second top row), cooking (top row), and music (second to last row), all the way up to quantum physics (bottom row).

G Collection parameters

We provide more information on the parameters used to collect new data samples.

| Parameter | value |
|--|-------|
| max active trees | 100 |
| max initial prompt review | 100 |
| max tree depth | 5 |
| max children count | 2 |
| num prompter replies | 1 |
| goal tree size | 9 |
| num reviews initial prompt | 3 |
| num reviews reply | 3 |
| auto mod enabled | true |
| auto mod max skip reply | 25 |
| auto mod red flags | 4 |
| p full labeling review prompt | 1 |
| p full labeling review reply assistant | 1 |
| p full labeling review reply prompter | 0.1 |
| acceptance threshold initial prompt | 0.6 |
| acceptance threshold reply | 0.6 |
| num required rankings | 3 |
| p activate backlog tree | 0.1 |
| min active rankings per lang | 20 |
| lonely children count | 2 |
| p lonely child extension | 0.75 |
| recent tasks span sec | 300 |
| max pending tasks per user | 8 |
| max prompt lottery waiting | 1000 |

Table 5: Collection parameters

max active trees: Maximum number of concurrently active message trees in the database. No new initial prompt tasks are handed out to users if this number is reached

max initial prompt review: Maximum number of initial prompts under review before no more initial prompt tasks will be handed out.

max tree depth: Maximum depth of message tree.

max children count: Maximum number of reply messages per tree node.

num prompter replies: Number of prompter replies to collect per assistant reply.

goal tree size: Total number of messages to gather per tree.

num reviews initial prompt: Number of peer-review checks to collect in the ‘INITIAL_PROMPT_REVIEW’ state

num reviews reply: Number of peer review checks to collect per reply (other than initial prompt).

auto mod enabled: Flag to enable/disable auto moderation.

auto mod max skip reply: Automatically set tree state to ‘halted_by_moderator’ when more than the specified number of users skip replying to a message. (auto moderation)

auto mod red flags: Delete messages that receive more than this number of red flags if it is a reply or set the tree to ‘aborted_low_grade’ when a prompt is flagged. (auto moderation)

p full labeling review prompt: Probability of full text-labeling (instead of mandatory only) for initial prompts.

p full labeling review reply assistant: Probability of full text-labeling (instead of mandatory only) for assistant replies.

p full labeling review reply prompter: Probability of full text-labeling (instead of mandatory only) for prompter replies.

acceptance threshold initial prompt: Threshold for accepting an initial prompt.

acceptance threshold reply: Threshold for accepting a reply.

num required rankings: Number of rankings in which the message participated.

p activate backlog tree: Probability to activate a message tree in BACKLOG_RANKING state when another tree enters a terminal state.

min active rankings per lang: When the number of active ranking tasks is below this value when a tree enters a terminal state an available trees in BACKLOG_RANKING will be activated (i.e. enters the RANKING state).

lonely children count: Number of children below which parents are preferred during sampling for reply tasks.

recent tasks span sec: Time in seconds of recent tasks to consider for exclusion during task selection.

max pending tasks per user: Maximum number of pending tasks (neither canceled nor completed) by a single user within the time span defined by 'recent_tasks_span_sec'.

max prompt lottery waiting: Maximum number of prompts in prompt_lottery_waiting state per language. If this value is exceeded no new initial prompt tasks for that language are generated.

H Training Configuration

Following [13] and as introduced in Section 1, we train supervised fine-tuned models (*SFT*), reward models (RM), and a PPO fine-tuned models based on RM's predictions. We use as base models the popular decoder-only Pythia [3] and LLaMA [2].

Conversation format We sample threads in the CTs and provide them as input text to the model by using some additional special tokens. More specifically, a thread composed of prompts (P) P_1, P_2, \dots and replies (R) R_1, R_2, \dots is provided as input to the model with the following format:

```
<prompter_token> P1 <endoftext_token> <assistant_token> R1 <endoftext_token>
<prompter_token> P2 <endoftext_token> <assistant_token> R2 <endoftext_token>
...
```

Each of the prompts and the replies consists potentially of multiple tokens after tokenizing.

SFT-mix details. The *sft-mix* training data configuration used for training *OpenAssistant/falcon-40b-sft-mix-1226* is a mixture of OASST1 with other instruction tuning datasets, according to the following configuration:

sft9-stage2:

```
# oasst_export: 100.00% (29899)
# vicuna: 50.00% (16963)
# code_alpaca: 50.00% (9510)
# oa_wiki_qa_bart_10000row: 100.00% (9434)
# grade_school_math_instructions: 100.00% (8351)
# dolly15k: 100.00% (14250)
```

use_custom_sampler: true

datasets:

```
- oasst_export:
  lang: "bg,ca,cs,da,de,en,es,fr,hr,hu,it,nl,pl,pt,ro,ru,sl,sr,sv,uk" # sft-8.0
  input_file_path: 2023-06-02_oasst_all_labels.jsonl.gz
  val_split: 0.05
  top_k: 2
- vicuna:
  fraction: 0.5
  val_split: 0.025
  max_val_set: 250
- code_alpaca:
  fraction: 0.5
  val_split: 0.05
  max_val_set: 250
- oa_wiki_qa_bart_10000row:
```



```

    val_split: 0.05
    max_val_set: 250
- grade_school_math_instructions:
    val_split: 0.05
- dolly15k:
    val_split: 0.05
    max_val_set: 300

```

More details can be found on the Hugging Face hub and in our open-source codebase.

Supervised fine-tuning. During this phase, we fine-tune pretrained models for the regular language modelling tasks based on our conversational data. We mask tokens that correspond to prompts and only train to predict tokens that correspond to assistant replies.

Reward model. For the reward model training, we replace the language modelling head with a linear layer producing a single output r_θ , corresponding to the predicted score for the last reply of the conversation. We use replies to the same prompt and their rankings as described in Appendix B. Following [13], assuming K distinct replies, we produce $\binom{K}{2}$ comparisons and train to minimize the loss

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l)} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))],$$

where σ is the sigmoid function and y_w corresponds to a preferred completion for the pair of y_w and y_r . We also optionally add another regularization parameter that prevents the predicted values from diverging too much. Performance is measured by measuring the ability to predict the better reply among pairs of replies with different rank, on a held-out validation set.

PPO training. We fine-tune the *SFT* model by producing assistant replies to unanswered questions. We use the RM to score these replies and train with PPO, using the trlx framework⁹. Following [13], we also add a per-token KL penalty from the *SFT* model at each token to avoid instability and over-optimization to the RM model.

All details and current training parameters are publicly available under https://github.com/LAION-AI/Open-Assistant/tree/main/model/model_training.

I Evaluation Tasks

This section aims to provide a brief explanation of the evaluation tasks used in Table 1.

lm-evaluation-harness [25] is a framework for evaluating language models on a variety of standardized tasks. We focus on the subset of BoolQ, PIQA, HellaSwag, WinoGrande, ARC-e, ARC-c and OBQA.

Vicuna Elo Rank [21] leverages LLMs to judge other LLMs and assign them a ranking according to the ELO system.

OpenAI Evals [26] is a set of general LLM evaluation benchmarks OpenAI released along with the announcement of GPT-4, and to which OpenAI has asked the community to contribute.

HumanEval [27] is a benchmark OpenAI released along with the announcement of Codex and sets out to measure "measure functional correctness for synthesizing programs from docstrings".

J Political Compass Evaluations

The political leanings of ChatGPT have been investigated in [39]. We evaluated a model fine-tuned on OpenAssistant Conversations on a subset of the given tests. Prompts were standardized and multiple samples were drawn, with majority vote deciding on the final answer for each question. Figure 12 depicts the result. We stress that these are very preliminary results and should not be taken with

⁹<https://github.com/CarperAI/trlx>

large degrees of certainty, as the community has yet to find consensus on the exact methodology to perform such evaluations. We will update this section with improved results in the future. Our limited, preliminary results show the model trained on OpenAssistant Conversations to be more balanced and varied in its political leanings than ChatGPT.

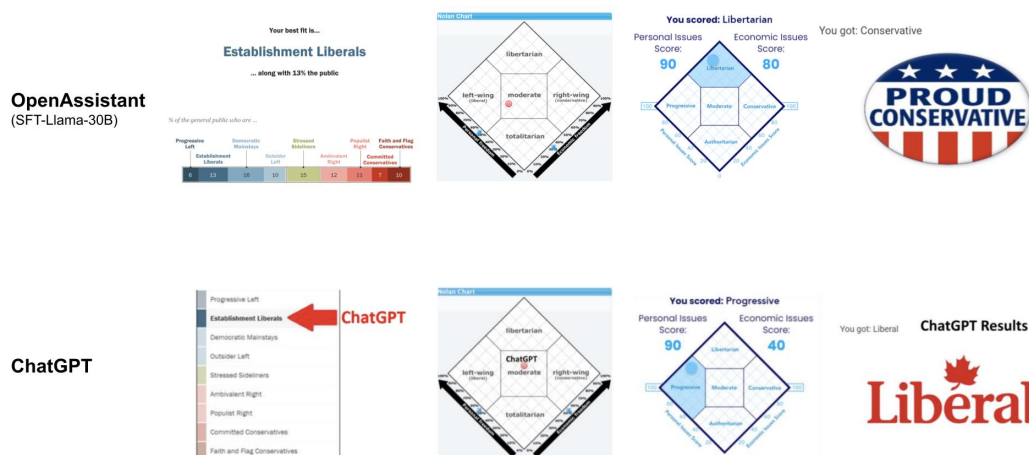


Figure 12: Comparison of evaluations on test for political leanings. For original ChatGPT results and references to tests used, see [39]

K Community Engagement

Throughout the collection of OpenAssistant Conversations, a large global community has been built, including an active Discord group, and a GitHub repository with over 200 contributors.

Figure 13 shows the growth of the Discord community throughout the duration of data collection.

Figure 14 shows new commits to the GitHub repository over time.

Figure 15 shows the growth in stars on the GitHub repository over time.

Figure 16 shows popularity of OpenAssistant by YouTube’s videos’ views on the theme over time.

These Figures serve as a strong reminder of what can be achieved by the collective effort of many volunteers, even in a field where research has thus far been largely monopolized by a small number of industrial labs.

In addition, by comparing the massive influx of new contributors and subscribers to the emergence of Open Assistant themed videos, it shows how certain media events have influenced the development.

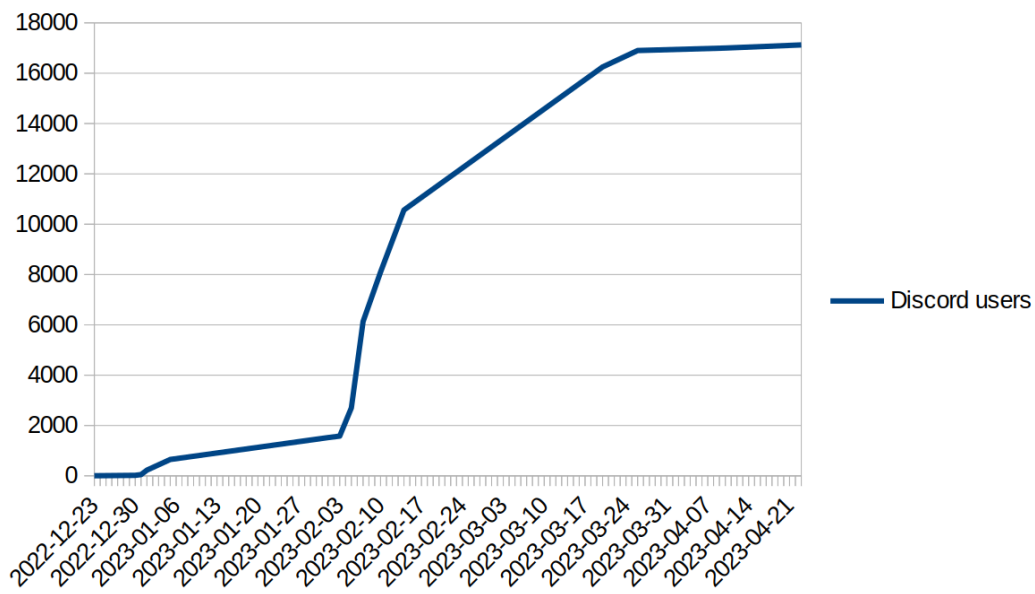


Figure 13: Discord users in the OpenAssistant group over time.

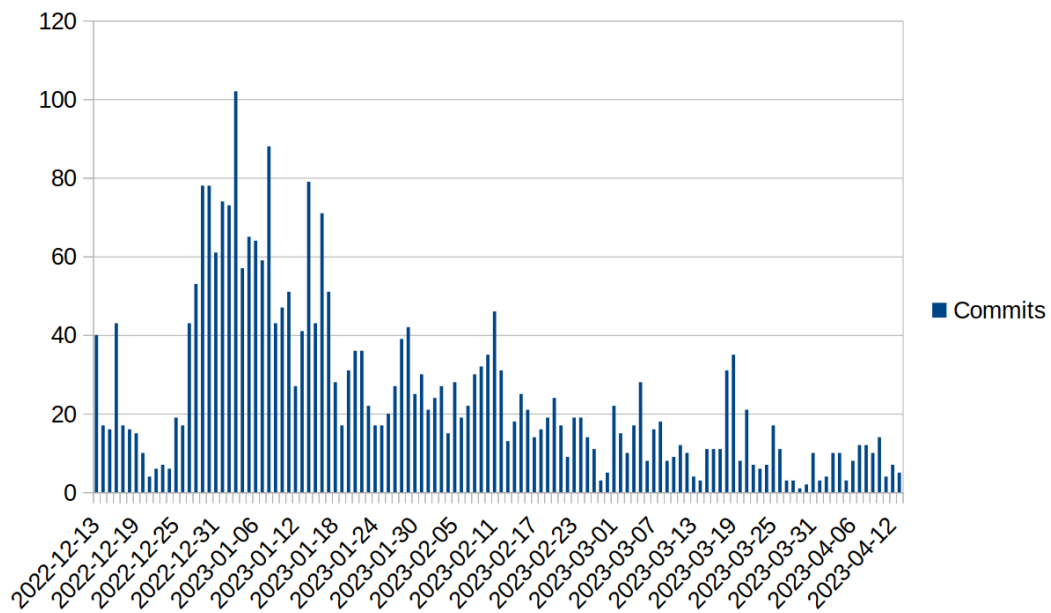


Figure 14: GitHub commits to the OpenAssistant repository over time.

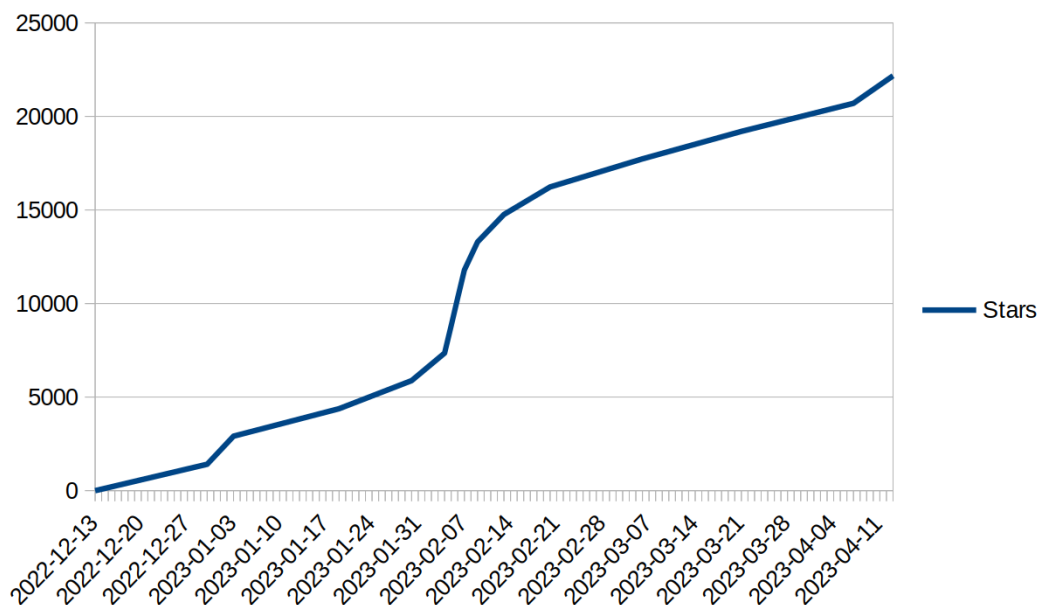


Figure 15: GitHub stars on the OpenAssistant repository over time.

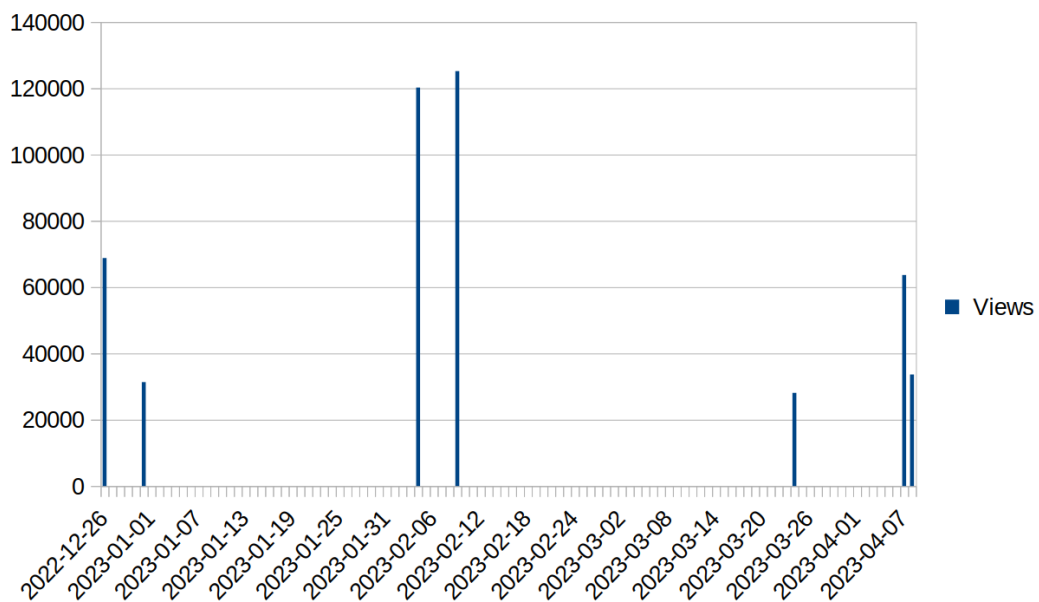


Figure 16: YouTube's videos' views on the OpenAssistant theme over time.

L Dataset Documentation (Data card)

Dataset documentation can be found at <https://huggingface.co/OpenAssistant/oasst1>.

M Author Statement

We confirm that we bear all responsibility in case of any violation of rights during the collection of the data or other work, and will take appropriate action when needed, e.g. to remove data with such issues.