A Dataset for Learning Conversational Style

Roshan Regula

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-97
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-97.html

May 11, 2024

Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

A Dataset for Learning Conversational Style

by Roshan Regula

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Gopala Anumanchipalli
Research Advisor

05/07/24

(Date)

* * * * * * *

Professor Aditya Parameswaran
Second Reader

05/10/24

(Date)

‘

Abstract

A Dataset for Learning Conversational Style

By Roshan Regula

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Gopala Anumanchipalli

Chat has become a prevalent use case for large language models in the AI age. Chat applications are commonly built off a pretrained model that is proficient at language and is subsequently finetuned for a specific domain with related conversational data. The resulting model excels empirically on the chosen topic while providing responses mimicking a real conversation. However, all these conversational datasets focus on maintaining domain emphasis while ignoring uniformity of sources. Furthermore, samples are constrained to prevent unsafe speech. As a result, conversational style is lost, and the downstream result is a chat bot that very much converses as a robot. In this work, we curate a conversational dataset composed of samples coming from a single source. We scrape Reddit's public data to group comments and responses based on individual users while putting no constraints on the contents of the conversation. The dataset is particularly sourced from the AskReddit subreddit and contains upwards of 12000 samples for each of 10 different users. We then evaluate the dataset by finetuning with various PEFT methods in addition to a RAG application. We find that RAG performs subpar to the PEFT methods on this dataset and, amongst the finetuning methods, LoRA achieves the best results in terms of perplexity. We also verify from sample prompting that the LoRA model produces responses indicative of human-like conversational style that is specific to the Reddit user in question.

Contents

Contents 3

List of Figures 4

List of Tables 5

Acknowledgements 6

I Introduction 7

II Background 8

III Methodologies 11

IV Results and Analysis 14

V Conclusion 17

VI Bibliography 18

List of Figures

Figure 1: top) Instruction style prompt formatting. The example instruction is always ‘Respond to the following comment’. bottom) <human> is parent comment and <bot> is the chosen user …………12

Figure 2: Left) Architectural composition of frozen base model and finetuned adapters for LoRA. Right) LoRA parameters used during finetuning. Only query and value layers have adapters trained at a rank of 8 …………13

Figure 3: Left) Example comment and reply from RedPajama Chat base model. Right) Example response for the same prompt from Chat model inferenced with finetuned LoRA adapters. …………15

List of Tables

Table 1: a) 2022 subreddits ranked by number of comments. AskReddit contains the most data and offers breadth by containing conversation on various topics. b) 2022 top 10 authors by usage. Many authors turn out to be bot accounts and need to be verified manually. c) Total data quantity available and processed for each year...12

Table 2: RedPajama Chat model's perplexity scores as evaluated on 5 of the users from the dataset. The base model is finetuned with the 4 PEFT methods for each user with the base model as the control ……………………………………………………14

Table 3: Top) RedPajama Chat RAG perplexity scores with varying numbers of examples across 5 users. The instructor embedding model was used to compute the embeddings that determined similarity between samples. Bottom) Pythia 1B RAG perplexity scores with varying numbers of examples across all 10 users. NaN indicates input prompts that became too large for the model. …………………..16

Acknowledgements

I would like to thank Professor Gopala Anumanchipalli for his continued support as my master's advisor. I would also like to thank Akshat Gupta for guiding me throughout this research project and providing me with a direction and theme to focus on. This project would not have been possible without him. Thanks to the Berkeley Speech Group, without their computational resources, this project would not have been feasible. Special thanks to Professor Aditya Parameswaran for taking the time to be a reader of this report.

I. Introduction

Large Language Models have become increasingly popular in recent times with new models being released almost daily. As new models continue to improve in performance, they also get substantially larger [1]. Even older generation models like Llama 2 70b [2], which is 130Gb in size, require special hardware and are usually hosted on cloud environments. Despite models being ubiquitous, the means of being able to use or serve them are substantially greater [3]. In this paper, we focus on smaller models, specifically RedPajama-Chat-3B [4] and Pythia 1B [5], that can be served on smaller and more personal devices such as laptops and even phones.

While all models start with basic pretraining on large corpi of textual data, they often go through various after training methods to excel at certain downstream tasks. Amongst these, one of the most popular applications of LLMs is in the domain of chat conversations. Chat models go through an after-training process that exposes them to large datasets of conversations between different agents. Chat datasets can contain upwards of ten thousand conversation samples all centered around a specific task type. Some popular open-source examples include the WikiQA [6] Corpus specializing on open domain question answer style chats, the LMSYS-Chat-1M [7] dataset curating conversations between users and 25 different state-of-the-art LLMs, and more. These datasets specialize on a certain topic, but they all contain samples collected from multiple sources. This helps them learn a specific domain, but not a specific source. To the best of our knowledge, no trainable data set exists that captures conversations from a single source. With such a data source, models could be trained to achieve a finer granularity, and potentially learn the conversational style of the source. In this work, we aim to curate a dataset of online conversation from the popular online forum Reddit, sourcing thousands of samples pertaining to individual users. With the curated dataset, we run various finetuning methods to evaluate the efficacy of the datasets.

II. Background

A. Reddit

Reddit is a web forum that was founded in 2005 and has exploded in popularity, boasting a user base of 73 million daily users. The forum is broken down into various subreddits that represent a specific topic or community that users can converse about. Reddit is founded on the principle of online democracy giving users a free speech platform for any plausible topic. User content can have varying levels of restriction with popular public forums being particularly liberal. We decided to focus on the AskReddit subreddit, a forum designed as a QA platform with no restriction on question topics. While the factuality and legitimacy of the content might be questionable, it is undoubtedly a great source of natural textual conversation spanning a large population and having a long history.

Reddit's public comments dataset is a popular source in many studies relating to social sciences, linguistics, and computer science. Topics vary from the spread of misinformation [8] to training content safety filters [9], but in all scenarios the data is always grouped according to topic rather than authorship. By gathering on users, the topic becomes the user themselves giving insight to their conversational styles and practices like slang, grammar, idioms, etc.

B. ClickHouse & LitGPT

ClickHouse [10] is an opensource data processing service that specializes in analyzing large datasets. It utilizes a column-oriented data storage solution that enables quick analysis of large volumes. Additionally, it supports every popular datasource format and boasts proprietary compression capabilities that reduce the storage burden of such datasets. ClickHouse interfaces via SQL making data processing extremely intuitive and simple.

Once the data has finished processing, LitGPT [11] enables the evaluation pipeline chosen for this project. It provides an opensource solution for custom finetuning with various methodologies and inferential evaluation. It is built off Lightning Fabric, a framework that removes the overhead of boilerplate distributed training. With LitGPT, we are able to do fully sharded data parallel training, drastically improving training times and costs.

C. PEFT

Parameter efficient finetuning [12] involves avoiding retraining the entirety of a model to increase proficiency on a specific task. PEFT is a suite of finetuning methodologies with varying tradeoffs in downstream performance vs training time and cost. Adapter [13][14] is the simplest of the group, introducing a prefix trainable layer before the attention blocks of the LLM. The parameters of the original model stay frozen, and the much smaller adapter layer is trained to fit the finetuning data. Adapter V2 adds to this by introducing a trainable bias and scaling factor to each adapter layer.

LoRA [15], or low rank adapter, is more widely used, and involves training a lower rank approximation of the weight matrices of the original model. Once again, the original models' parameters stay frozen while the lora adapters are trained. Trained model parameters represent an understanding of the structure of data and LoRA assumes that this structure can be represented at a lower rank while having minimal degradation in accuracy. The greater the rank of the adapter the closer we get to full finetuning. Finally, while LoRA only includes adapters for the query and value blocks, QLoRA [16] considers query, value, key, projection, and mlp layers. The work on QLoRA insists that this methodology comes closest in performance in comparison to full finetuning.

D. RAG

Retrieval augmented training [17][18] avoids finetuning all together by capitalizing on contextual learning. The idea is based on the principle of imitation for learning and involves fetching relevant samples that act as examples to the model. The rank of these samples is based on some ranking or embedding model which measures how close other samples are to the prompt in question. The model is then prompted with similar samples, that contain input and output, followed by the actual prompt.

RAG has been shown to reduce LLM hallucination and promote more evidence-based generations inspired by existing data. It combines the generalizability of pre-trained models with explicitly retrieved relevant examples. While it removes the overhead of finetuning, it introduces the issue of being able to scale efficiently.

E. Pythia & RedPajama

Pythia is a scientific suite of trained models with increasing size. All models followed the same transparent architecture and training loop and achieved performance comparable to the similarly sized OPT models [19] at the time. In comparison to other suites of models, Pythia reports full transparency of training data, model architecture, and training order. Pythia replicates the architecture choices of SOTA LLMs including the classic attention framework [20] followed by dense layers. They also follow the popular choice of using rotary embeddings [21] for positional information. In the training loop, they utilize flash attention [22] and parallel attention and FFN layers to improve training times with no performance degradation. All models are trained via data and tensor parallelism with the Adam [23] and ZeRO [24] optimizers. Despite deviating to training on large batch sizes, they don’t face the issue of convergence that normally follows. The dataset they is the Pile dataset [25] with the GPT-NeoX [26] tokenizer and they train on 300B tokens, similar to the GPT3 [27] and OPT suites. In our work, we choose to evaluate with the 1B variant of the Pythia suite.

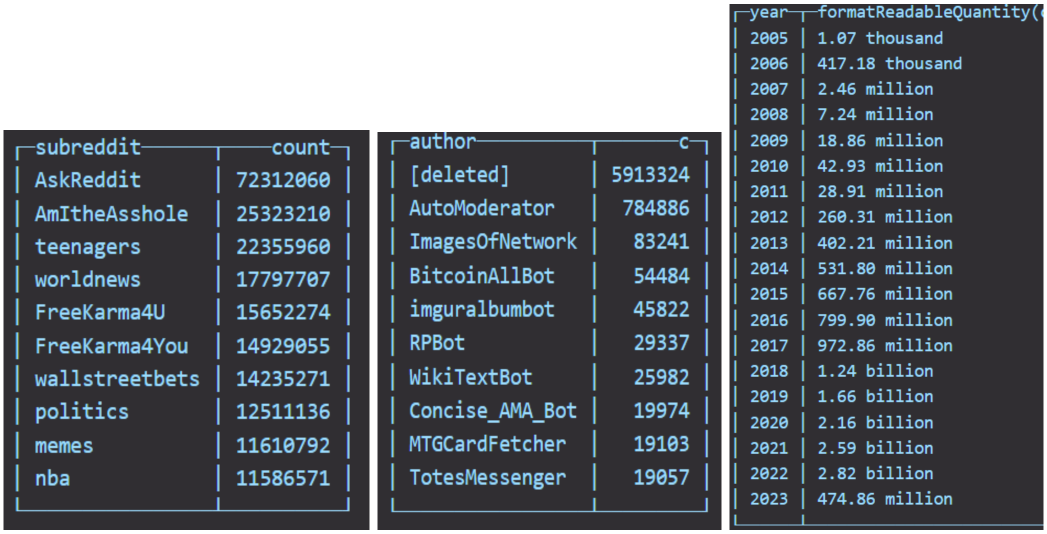
RedPajama builds on the Pythia models by following the Llama recipe and training on an additional 800B tokens. The model we use in evaluation is the RedPajama Chat variant, which was finetuned on the Dolly v2.0 [28] and OASST1 [29] chat datasets.

III. Methodologies

A. Data Processing

We first start by curating the dataset from Reddit grouped by users. The source data is stored in public AWS s3 buckets ranging from 2008 to 2023. We run a script that scrapes the data sequentially from each year and process it. For every year we narrow our search to only the AskReddit subreddit to drastically reduce downstream processing times. As we are interested in data originating from a single source, we rank authors by subreddit activity in 2022, the year with the most amount of data. From the 2.9 billion comments, we isolate the 30 most active users and conduct a manual profile verification to root out new users or bot accounts. At the end we are left with 10 users that have upwards of 5000 comments in 2022 alone and an established history on the platform. By filtering on these users across all 15 years, we gather upwards of 12000 comments per user.

To create usable data samples, we need to link comments with originals and responses. For this we run a SQL join on all the samples from the AskReddit subset for each year to create relations between the reply from our chosen user and the author of the original comment. Each data now consists of an author and a conversation they participated in. With the samples collected for the chosen users across every year, we now prepare the data by removing deleted content and thresholding content length. We utilize the same GPT-NeoX tokenizer used by both Pythia and RedPajama models to quantify the size of all responses and select only entries that fall under the 2048 max sequence length supported by these models. This substantially decreases the amount of trainable data down to 8000 for the most active user, but this would not be a problem with more current models that have support for much larger contexts. We do a 80/20 split on the data to create a training and test set.



```
┌─subreddit──────────┬───count─┐
│ AskReddit          │ 72312060 │
│ AmItheAsshole      │ 25323210 │
│ teenagers          │ 22355960 │
│ worldnews          │ 17797707 │
│ FreeKarma4U        │ 15652274 │
│ FreeKarma4You      │ 14929055 │
│ wallstreetbets     │ 14235271 │
│ politics           │ 12511136 │
│ memes              │ 11610792 │
│ nba                │ 11586571 │
└────────────────────┴──────────┘
```

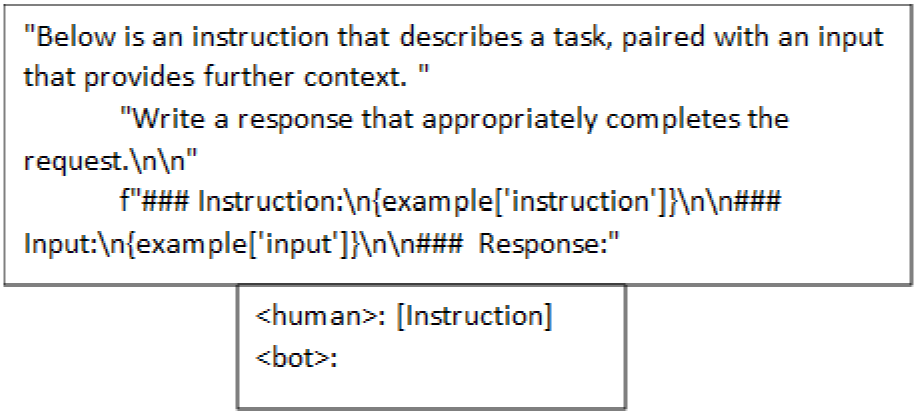
```
┌─author──────────┬───────c─┐
│ [deleted]       │ 5913324 │
│ AutoModerator   │  784886 │
│ ImagesOfNetwork │   83241 │
│ BitcoinAllBot   │   54484 │
│ imguralbumbot   │   45822 │
│ RPBot           │   29337 │
│ WikiTextBot     │   25982 │
│ Concise_AMA_Bot │   19974 │
│ MTGCardFetcher  │   19103 │
│ TotesMessenger  │   19057 │
└─────────────────┴─────────┘
```

```
┌─year─┬─formatReadableQuantity(
│ 2005 │ 1.07 thousand
│ 2006 │ 417.18 thousand
│ 2007 │ 2.46 million
│ 2008 │ 7.24 million
│ 2009 │ 18.86 million
│ 2010 │ 42.93 million
│ 2011 │ 28.91 million
│ 2012 │ 260.31 million
│ 2013 │ 402.21 million
│ 2014 │ 531.80 million
│ 2015 │ 667.76 million
│ 2016 │ 799.90 million
│ 2017 │ 972.86 million
│ 2018 │ 1.24 billion
│ 2019 │ 1.66 billion
│ 2020 │ 2.16 billion
│ 2021 │ 2.59 billion
│ 2022 │ 2.82 billion
│ 2023 │ 474.86 million
└──────┴────────────────────────
```

Table 1: a) 2022 subreddits ranked by number of comments. AskReddit contains the most data and offers breadth by containing conversation on various topics. b) 2022 top 10 authors by usage. Many authors turn out to be bot accounts and need to be verified manually. c) Total data quantity available and processed for each year

B. Prompt Formatting

We apply two different prompt formatting [30] schemes corresponding to the two different models we evaluate with. For the Pythia1B model, which is just a pretrained model, we apply an instruct style formatting where we prompt the model with the parent comment and tell it to reply to the message with the reply being the response of the user in this case. For the RedPajama Chat model, which was finetuned with a specific prompt, we follow the same format, which mimics traditional conversational formatting.



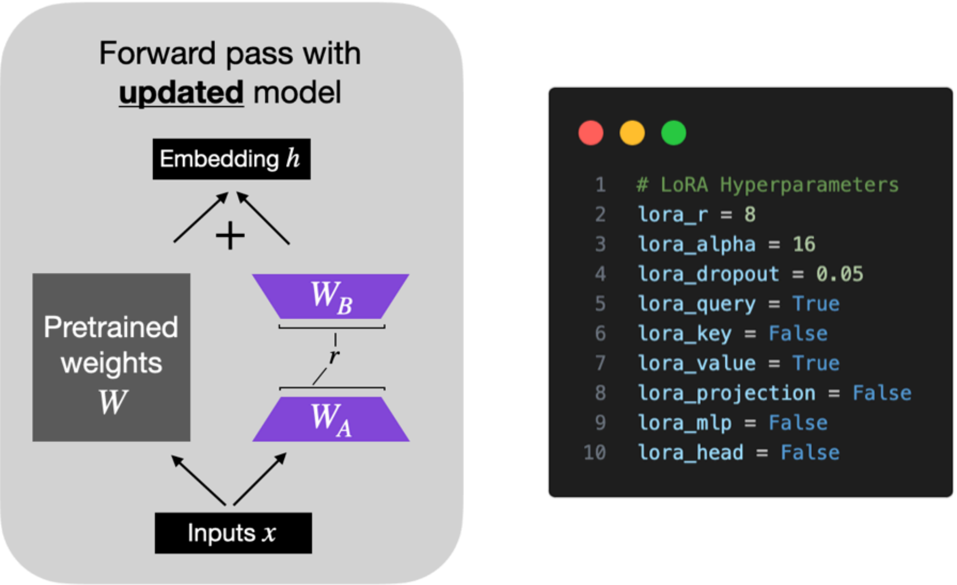
```
"Below is an instruction that describes a task, paired with an input
that provides further context. "
        "Write a response that appropriately completes the
request.\n\n"
        f"### Instruction:\n{example['instruction']}\n\n###
Input:\n{example['input']}\n\n### Response:"
```

```
<human>: [Instruction]
<bot>:
```

Figure 1: top) Instruction style prompt formatting. The example instruction is always ‘Respond to the following comment’. bottom) <human> is parent comment and <bot> is the chosen user

C. Finetuning

We finetune each user across varying PEFT methodologies using RedPajama model. We do not include the Pythia model due to it being a pretrained model. Separately, we use the instructor embedding model to compute data embeddings and develop a ranking based on similarity. We use these similarity rankings to run retrieval augmented generation using varying number of examples on both base models. We maintain the same respective prompting for each model, repeating conversations for the chat format, and listing multiple Instruction/Response pairs for the instruction format. Training is conducted on RTX A5000 GPUs ranging from 2 to 6 GPUs depending on availability and model size. All models are trained for 5 epochs.

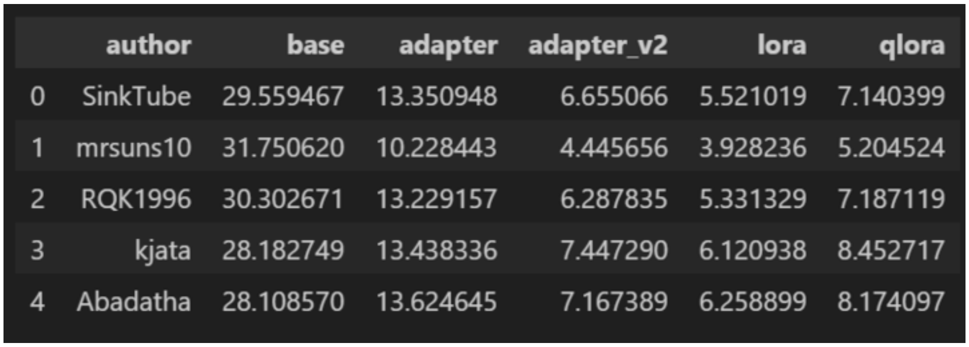


```
# LoRA Hyperparameters
lora_r = 8
lora_alpha = 16
lora_dropout = 0.05
lora_query = True
lora_key = False
lora_value = True
lora_projection = False
lora_mlp = False
lora_head = False
```

Figure 2: Left) Architectural composition of frozen base model and finetuned adapters for LoRA. Right) LoRA parameters used during finetuning. Only query and value layers have adapters trained at a rank of 8.

IV. Results and Analysis

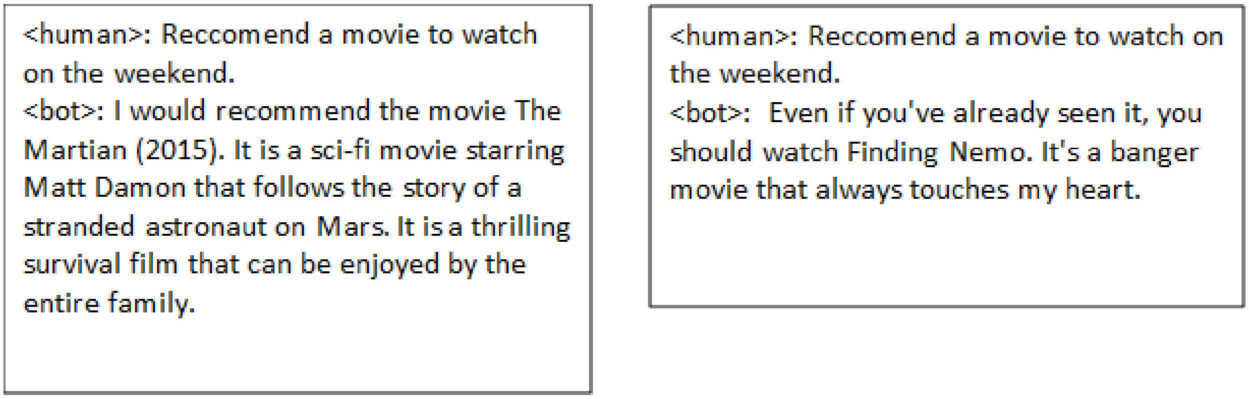
Evaluation for chat models is still a developing field and most metrics like RTE [31] and STSB [32] evaluate on coherency of the chat and for all current chat tasks, that is sufficient. Existing chat applications are domain oriented in that their objective is to converse on a specific topic. The facets of evaluation become accuracy of content and fluency of speech. In the case of chat in the domain of conversational style, both accuracy and fluency can vary greatly. A user could give completely incorrect information in their response or completely ignore the original comment and respond in a way that seems irresponsive. However, the situation can still have valuable information on the conversation style of the user. To this extent, we decided to evaluate all training methods with perplexity to verify that non-domain-oriented data can still be used to effectively teach a model to replicate style.



	author	base	adapter	adapter_v2	lora	qlora
0	SinkTube	29.559467	13.350948	6.655066	5.521019	7.140399
1	mrsuns10	31.750620	10.228443	4.445656	3.928236	5.204524
2	RQK1996	30.302671	13.229157	6.287835	5.331329	7.187119
3	kjata	28.182749	13.438336	7.447290	6.120938	8.452717
4	Abadatha	28.108570	13.624645	7.167389	6.258899	8.174097

Table 2: RedPajama Chat model’s perplexity scores as evaluated on 5 of the users from the dataset. The base model is finetuned with the 4 PEFT methods for each user with the base model as the control

The model successfully learns the conversation data and achieves the best performance with the LoRA method. We used a rank of 8 for the decomposition matrices and only finetuned for the query and value layers. Unlike traditional chat data, our data spanned various topics and domains, with only the user's conversational style being consistent across the samples. The model's perplexity and sample responses both suggest that the model can learn to imitate the user's speech.

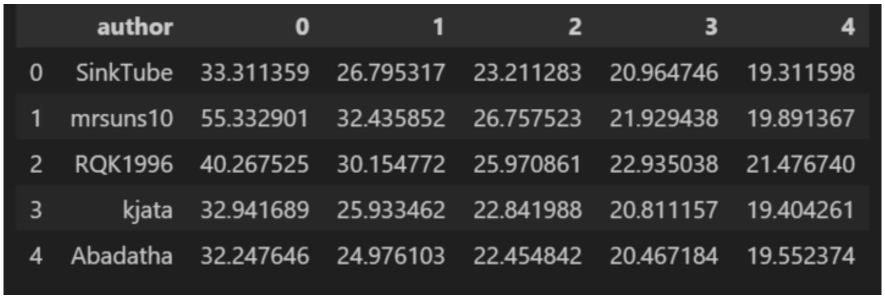


<human>: Reccomend a movie to watch on the weekend.
<bot>: I would recommend the movie The Martian (2015). It is a sci-fi movie starring Matt Damon that follows the story of a stranded astronaut on Mars. It is a thrilling survival film that can be enjoyed by the entire family.

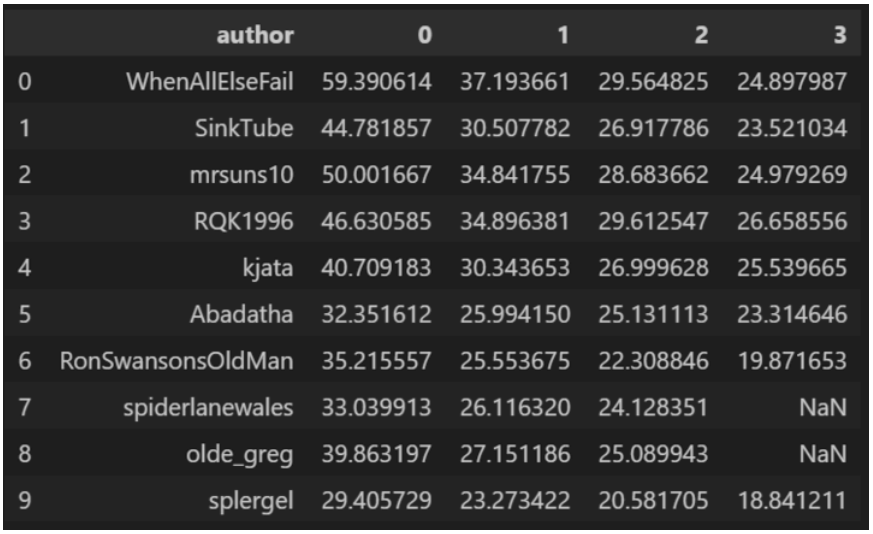
<human>: Reccomend a movie to watch on the weekend.
<bot>: Even if you've already seen it, you should watch Finding Nemo. It's a banger movie that always touches my heart.

Figure 3: Left) Example comment and reply from RedPajama Chat base model. Right) Example response for the same prompt from Chat model inferenced with finetuned LoRA adapters.

From sampling responses, we observe the finetuned model to generate a much more informal response that addresses the prompt but is more human-like with respect to style. With the base model, we receive a formal response stereotypical of a chat bot. The trained model's response uses contractions, a grammar style that feels normal during regular conversation, but rarely appears within regular chatbot responses. Additionally, the model even uses the term 'banger' to describe the movie, an informal adjective term that describes something in an energetic and elevated manner.



	author	0	1	2	3	4
0	SinkTube	33.311359	26.795317	23.211283	20.964746	19.311598
1	mrsuns10	55.332901	32.435852	26.757523	21.929438	19.891367
2	RQK1996	40.267525	30.154772	25.970861	22.935038	21.476740
3	kjata	32.941689	25.933462	22.841988	20.811157	19.404261
4	Abadatha	32.247646	24.976103	22.454842	20.467184	19.552374



	author	0	1	2	3
0	WhenAllElseFail	59.390614	37.193661	29.564825	24.897987
1	SinkTube	44.781857	30.507782	26.917786	23.521034
2	mrsuns10	50.001667	34.841755	28.683662	24.979269
3	RQK1996	46.630585	34.896381	29.612547	26.658556
4	kjata	40.709183	30.343653	26.999628	25.539665
5	Abadatha	32.351612	25.994150	25.131113	23.314646
6	RonSwansonsOldMan	35.215557	25.553675	22.308846	19.871653
7	spiderlanewales	33.039913	26.116320	24.128351	NaN
8	olde_greg	39.863197	27.151186	25.089943	NaN
9	splergel	29.405729	23.273422	20.581705	18.841211

Table 3: Top) RedPajama Chat RAG perplexity scores with varying numbers of examples across 5 users. The instructor embedding model was used to compute the embeddings that determined similarity between samples. Bottom) Pythia 1B RAG perplexity scores with varying numbers of examples across all 10 users. NaN indicates input prompts that became too large for the model.

The RAG results are significantly worse than the finetuned methods and fail to achieve the sare accuracy as even adapter v1. This suggests that RAG is better suited for cases where factual accuracy is more important. Additionally, RAG tries to find examples where the parent comments are similar, not the responses of the user. While one can expect similar responses to similar questions for traditional domain-oriented chats, this dataset's goal is to capture style, not informational accuracy.

V. Conclusion

Our work consisted of two main objectives: a) to curate a dataset composed of conversation samples from a single source and b) to evaluate Finetuning and RAG on this dataset to verify that the data can teach the model to pick up on a user's conversational style. By utilizing ClickHouse, we successfully develop a framework for collecting and formatting sample from large volumes of raw conversations. With LitGPT, we finetune with various methods and conduct RAG to find that LoRA achieves the best performance with respect to perplexity. RAG fails to proficiently learn from the samples.

The objective of this work was to lay the groundwork for future studies into conversational style learning and we do so by proposing the Reddit User Conversations Dataset. In the future we would like to further explore evaluation on this dataset with new models and ablations. Some models that are of particular interest include Microsoft's Phi 3 and the latest Llama 3 variants. While we finetuned with different methods, ablation studies were out of the scope of this project. Sweeping hyperparameters could reveal more optimal settings leading to increased model performance. Additionally, vanilla RAG does not work well for us, but different ranking/embedding models and algorithms could provide valuable insight into a more robust generation that does not rely on any finetuning.

Our envisioned use case for style-oriented chat models is personal assistants like that of Siri or Alexa and to that end we try to constrain ourselves to models small enough to be hosted on these edge devices. Aligned with this, we would like to explore quantized models and how they fare on this data to maintain small model size while capturing the astounding performance of larger models.

VI Bibliography

[1] Shervin Minaee et al. *Large Language Models: A Survey*. 2024. arXiv: `2402.06196 [cs.CL]`.

[2] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: `2307.09288 [cs.CL]`.

[3] Xupeng Miao et al. *Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems*. 2023. arXiv: `2312.15234 [cs.LG]`.

[4] Together Computer. *RedPajama: an Open Dataset for Training Large Language Models*. 2023. URL: `https://github.com/togethercomputer/RedPajama-Data`.

[5] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. 2023. arXiv: `2304.01373 [cs.CL]`.

[6] Yi Yang, Wen-tau Yih, and Christopher Meek. "WikiQA: A Challenge Dataset for Open-Domain Question Answering". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Ed. by Lluís Màrquez, Chris Callison-Burch, and Jian Su. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 2013–2018. DOI: `10.18653/v1/D15-1237`. URL: `https://aclanthology.org/D15-1237`.

[7] Lianmin Zheng et al. *LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset*. 2024. arXiv: `2309.11998 [cs.CL]`.

[8] Arash Amini et al. *News Source Credibility Assessment: A Reddit Case Study*. 2024. arXiv: `2402.10938 [cs.CL]`.

[9] Matthew Squires et al. *Enhancing Suicide Risk Detection on Social Media through Semi-Supervised Deep Label Smoothing*. 2024. arXiv: `2405.05795 [cs.LG]`.

[10] Fazl Barez, Paul Bilokon, and Ruijie Xiong. *Benchmarking Specialized Databases for High-frequency Data*. 2023. arXiv: `2301.12561 [cs.DB]`.

[11] Lightning AI. *LitGPT*. `https://github.com/Lightning-AI/litgpt`. 2023.

[12] Charith Chandra Sai Balne et al. *Parameter Efficient Fine Tuning: A Comprehensive Analysis Across Applications*. 2024. arXiv: `2404.13506 [cs.LG]`.

[13] Neil Houlsby et al. *Parameter-Efficient Transfer Learning for NLP*. 2019. arXiv: `1902.00751 [cs.LG]`.

[14] Renrui Zhang et al. *LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention*. 2023. arXiv: `2303.16199 [cs.CV]`.

[15] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: `2106.09685 [cs.CL]`.

[16] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. arXiv: `2305.14314 [cs.LG]`.

[17] Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: `2005.11401 [cs.CL]`.

[18] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. arXiv: `2312.10997 [cs.CL]`.

[19] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. 2022. arXiv: `2205.01068 [cs.CL]`.

[20] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: `1706.03762 [cs.CL]`.

[21] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023. arXiv: `2104.09864 [cs.CL]`.

[22] Tri Dao et al. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. 2022. arXiv: `2205.14135 [cs.LG]`.

[23] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: `1412.6980 [cs.LG]`.

[24] Samyam Rajbhandari et al. *ZeRO: Memory Optimizations Toward Training Trillion Parameter Models*. 2020. arXiv: `1910.02054 [cs.LG]`.

[25] Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. 2020. arXiv: `2101.00027 [cs.CL]`.

[26] Sid Black et al. *GPT-NeoX-20B: An Open-Source Autoregressive Language Model*. 2022. arXiv: `2204.06745 [cs.CL]`.

[27] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: `2005.14165 [cs.CL]`.

[28] Mike Conover et al. *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM*. 2023. URL: `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm` (visited on 06/30/2023).

[29] Andreas Köpf et al. *OpenAssistant Conversations – Democratizing Large Language Model Alignment*. 2023. arXiv: `2304.07327 [cs.CL]`.

[30] Melanie Sclar et al. *Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting*. 2023. arXiv: `2310.11324 [cs.CL]`.

[31] Mark Sammons. *Recognizing Textual Entailment*. 2015. DOI: `https://doi.org/10.1002/9781118882139.ch17`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118882139.ch17`.

[32] Daniel Cer et al. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 2017. DOI: `10.18653/v1/s17-2001`. URL: `http://dx.doi.org/10.18653/v1/S17-2001`.