International Journal of Knowledge and Language Processing
Volume 13, Number 2, Dec **2022**

AKLP International ©2010 ISSN 2191-2734
pp. 1—12

Study of ChatGPT and its Comparison with Other Mainstream Large Language Models

Cheng Yuyang
Peking University, School of Software & Microelectronics
No.24 Jinyuan Road, Daxing District, Beijing, China
chengyuyang@stu.pku.edu.cn

ABSTRACT. *This paper aims to investigate ChatGPT, a language model developed by OpenAI. Since OpenAI has not released a paper about ChatGPT, this paper mainly elaborates on the algorithms and related methodologies used by InstructGPT, the sibling model of ChatGPT, to explore the principles and technologies behind ChatGPT. Futhermore, this paper compares ChatGPT with its predecessor model GPT-3, and other two mainstream large language models - Bert and T5 in the following four aspects: dataset, model architecture, training approach, and application, so as to better understand what brings the excellent performance of ChatGPT.*

Keywords: ChatGPT, InstructGPT, Natural Language Processing, Reinforcement Learning

1. Introduction.

ChatGPT is a chatbot model developed by OpenAI that can simulate human speech behavior and interact with users naturally. Its excellent performance in understanding natural language queries and generating coherent and informative responses makes it the quickest platform to reach 100M users[1]. Its name comes from the technology it uses, the GPT-3 architecture, which is the 3rd generation of the generative language model GPT series as well as the core technology of ChatGPT. It simulates human speech behavior by using large amounts of training data and generates logical texts through syntactic and semantic analysis[2]. It can provide accurate and appropriate responses based on contexts, and simulate a wide range of human emotions and tones. This characteristic of ChatGPT provides users with more realistic and natural conversational experience when interacting with the machine.

OpenAI has not yet released a paper about ChatGPT, and only posts an introduction blog and a trial API on the website. ChatGPT is a sibling model of InstructGPT, with same core ideas. Its key capabilities come from three areas: powerful base large model capabilities (InstructGPT), high quality datasets (filtered and rich), and Reinforcement Learning Algorithm[3]. Therefore to understand the technologies behind ChatGPT, InstructGPT and its related methodologies should be first introduced.

2. InstructGPT.

2.1 Training Approach

InstructGPT is a language model proposed by OpenAI in March 2022 in the paper *Training language models to follow instructions with human feedback*. The approach InstructGPT adopted is essentially similar with that of ChatGPT, but differs slightly in data collection, the base model (GPT-3 vs. GPT-3.5), and the third step to initialize the Proximal Policy Optimization (PPO) algorithm.

The training approach of InstructGPT can be summarized as follows. Based on a existing pre-trained model, a prompt distribution for which model is wanted to generate aligned outputs, and a team of trained human labelers, the following three steps are performed[4]:

Step 1: Collect demonstration data and train a supervised policy. the labelers provide prompt demonstration data, and then the pre-trained GPT-3 model is fine-tuned on the datasets using supervised learning.

Step 2: Collect comparison data and train a reward model. Given an input, the model can produce multiple outputs. Labelers will indicate which output they prefer, thus collecting a comparison dataset between the model outputs. Based on this dataset, a reward model is trained to predict the human-preferred outputs.

Step 3: The first two steps provide a fine-tuned GPT-3 model and a reward model respectively. PPO algorithm is then used to fine-tune the supervised policy to optimize reward model.

Steps 2 and 3 can be iterated continuously.

More comparison data is collected on the current optimal policy, which is then used to train a new reward model and a new policy.

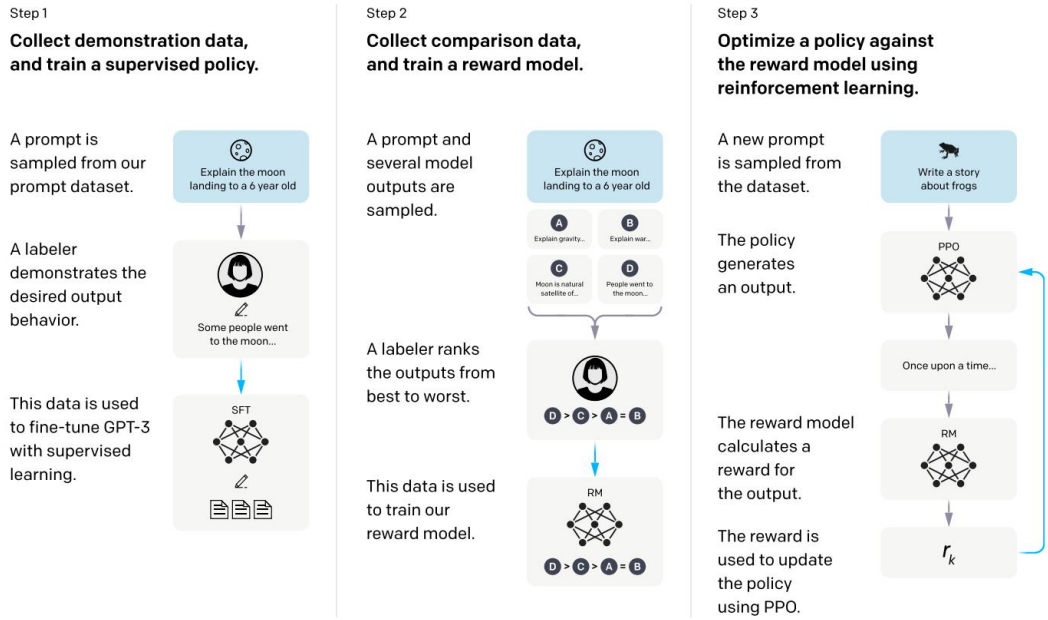

FIGURE 1. A diagram illustrating the three steps of the training method of InstructGPT: (1) supervised fine-tuning(SFT), (2) reward model(RM) training, and (3) reinforcement learning via proximal policy optimization(PPO) on this reward model. Blue arrows indicate that this data is used to train one of the models. In Step 2, boxed A-D are samples from the models that get ranked by labelers[4].

2.2 Models

(1)**Supervised fine-tuning (SFT).** Supervised fine-tuning of GPT-3 is performed based on the dataset provided by labelers. Since GPT-3 is a Prompt Learning-based generative model, the SFT dataset is also a sample of prompt-response pairs. Part of the SFT data comes from users using OpenAI's PlayGround, and part comes from 40 labelers employed by OpenAI[4]. The categories and related information of the datasets are shown in Figure 2 below.

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

FIGURE 2. Illustration of SFT datasets

An example of the manually labeled prompt dataset in this step is given for better understanding. In simple terms, a single sample in the prompt dataset consists of a pair of texts. For example:

Prompt: Use "machine learning" to make a sentence.

Demonstration: Machine Learning is an important direction in the field of computer science and artificial intelligence.

The prompt dataset is used to fine-tune GPT-3 by supervised learning.

(2)**Reward modeling (RM).** Comparison data is collected to train the reward model. In the sampled input statements, forward inference is performed to obtain multiple model outputs, and these outputs are rated by labelers. The reason for using ranking rather than scoring is that scoring is more subjective and difficult to measure compared to ranking. Ultimately these labeled data are used to train the reward model. By providing this reward model with labels, a lower score can be given to generated content that involves bias, therefore encouraging the model to not generate such content that humans do not like, achieving a model that is useful, truthful and harmless[5].

Specifically, for each prompt, InstructGPT randomly generates K outputs ($4 \leq K \leq 9$), and they are then presented in pairs to each labeler, from which the user chooses the output that works better. Such K outputs generate C_K^2 two-by-two comparison. For example, by

maximizing the difference in reward values between the more preferred sample yw and the less preferred sample yl for training, the loss function is obtained as follows[4].

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

In the loss function (1) above, $r_\theta(x, y)$ is the reward value of prompt x and response y in the reward model parameterized by θ, y_w is the response result more preferred by labeler, y_l is the response result less preferred by labeler, and D is the whole training dataset.

During training, InstructGPT treats each prompt's response pair as a batch, and this kind of training approach is less prone to overfitting than the traditional training approach which takes samples as a batch, because each prompt will be taken as the model's input only once.

For example, the labeler needs to rank different outputs generated by the model of the same input:

Prompt: Build a sentence using "machine learning".

Demonstration1: Machine Learning is an important direction in the field of computer science and artificial intelligence.

Demonstration2: Machine learning is a noun.

In the example above, the model generates Demonstration1 and Demonstration2. Obviously, the labeler needs to label Demonstration1 better compared to Demonstration2. The reward model takes the prompt and the results generated by the model as input and then outputs a scalar reward value[4].

The reward model can be seen as a loss function of the traditional training mechanism. The computation of the reward is more flexible and diverse than the loss function (AlphaGO's reward is the win/loss of the match), the reward of which is not derivable and therefore cannot be directly used for back propagation. The idea of reinforcement learning is to fit the loss function by sampling large amounts of rewards, thus realizing training of the model. Similarly, human feedback is not derivable, then we can also use artificial feedback as a reward for reinforcement learning, which is the core algorithm taken by InstructGPT/ChatGPT, reinforcement learning based on artificial feedback (RLHP)[6].

(3)**Reinforcement learning (RL).** The SFT model is fine-tuned using the PPO algorithm. The reward model in the second step is taken as the optimization target for reinforcement learning. A new sample is randomly selected and the generated responses are scored using the reward model. This score is the overall reward of the responses, which is back-propagated, and the gradient can update the parameter of PPO model. The whole process is iterated several times until the model converges. The reinforcement learning algorithm can modify the model parameters so that the model gets the maximum reward, which means that the responses parameterized by these parameters best match human preferences.

The core idea of RLHF is to use human feedback to modify the responses closest to human behaviors, and training to find the reward function that best explains the human judgment, and then RL is used to learn how to achieve this goal[7]. The data distribution for reinforcement learning changes, as the environment changes and the policy is updated. The PPO algorithm is used to perform the optimization, that is to optimize the objective function $objective(\varphi)$ by stochastic gradient descent. The following objective function is maximized in RL training[4].

$$objective(\phi) = E_{(x,y)\sim D_{\pi_{\phi}^{RL}}}[r_{\theta}(x,y) - \beta log(\pi_{\phi}^{RL}(y \mid x)/\pi^{SFT}(y \mid x))] + \gamma E_{x\sim D_{pertrain}}[log(\pi_{\phi}^{RL}(x))] \quad (2)$$

In formulation (2), π^{SFT} is the SFT model in Step1. $\pi_{\emptyset}^{RL}$ is the model we want to learn, that is the optimal policy of reinforcement learning, Dpertrain is the pre-training distribution, and $r_{\theta}(x, y)$ is the reward model for evaluation of problem x and problem y[4]. The formulation can be divided into three parts: scoring + KL divergence + GPT-3 pre-training.

During training, $\pi_{\emptyset}^{RL}$ will change after each update of the parameters, so y generated by x will also change through $\pi_{\emptyset}^{RL}$. Meanwhile, data of the reward model rθ(x,y) is generated by π^{SFT}. If the difference between π^{SFT} and $\pi_{\emptyset}^{RL}$ is too large, inaccurate score estimation of rθ(x,y) can be resulted, so adding $\beta log(\pi_{\emptyset}^{RL}(y|x)/\pi_{\emptyset}^{SFT}(y|x))$ this KL penalty to ensure that the difference between the output of the PPO model and the output of the SFT will not be large. Since we hope that the difference between the two models can be minimized and the objective function can be maximized, so a minus is added in front of the KL penalty. Here, $\pi_{\emptyset}^{RL}(y|x)$ is the probability that answer y is obtained by question x through $\pi_{\emptyset}^{RL}$. That is, each y prediction and its softmax output are multiplied, $\pi_{\emptyset}^{SFT}(y|x)$ is the probability that is the probability that answer y is obtained by question x through $\pi_{\emptyset}^{SFT}$.

Without the GPT3 pre-training part, the model may end up having good performances for only one task, and perform rather poor on other tasks. So the third part of the formulation adds the original GPT3 objective function, making the first two parts do the fitting on the new dataset while ensuring that the original data is not lost either[8].

3. ChatGPT.

3.1 Training Approach

Similar to Instruct GPT, ChatGPT uses RLHF to train this model. The same algorithm is applied, but with a slight difference in data collection. The initial model is trained using supervised fine-tuning: AI trainers provide the dialogue in which they play both sides - the user and the AI. The trainers are provided with suggestions wrote by the model to help them compose their responses. This new conversation dataset and the InstructGPT dataset are mixed and are transformed into a conversation format.

To create a reward model for reinforcement learning, comparison data should be collected which include two or more model responses ranked by quality. To collect this

data, conversations between AI trainers and ChatGPT are taken. A message written by the model and several alternative responses are randomly selected, and AI trainer are asked rank them. Using these reward models, the model is fine-tuned using proximate policy optimization. This process is iterated several times[2].

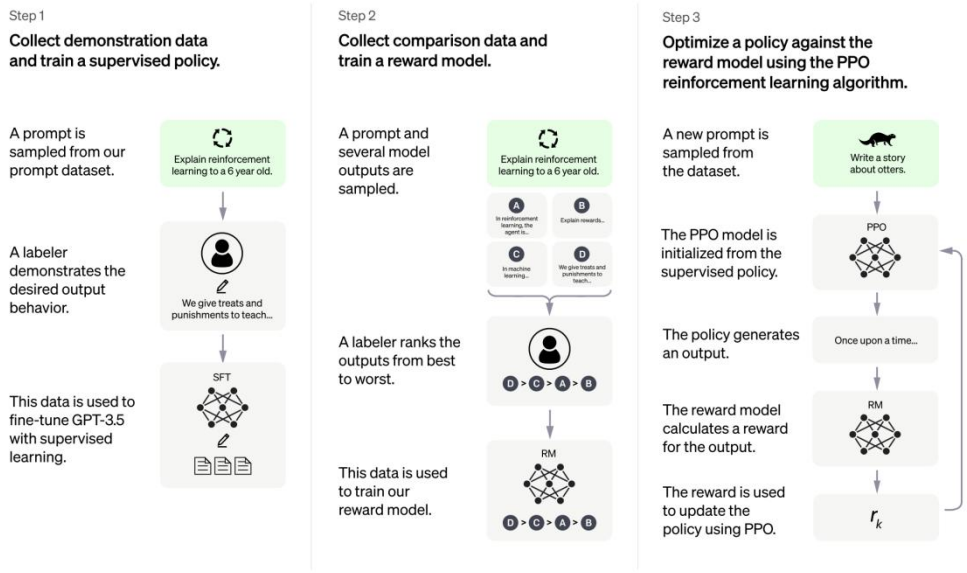

FIGURE 3. The training process of ChatGPT

3.2 Model Comparison

The performance improvements of ChatGPT are certainly transcendent. Compared with other Large Language Models (LLM), the GPT family of models is known for its larger dataset sizes and reinforcement learning algorithms used in the pre-training and fine-tuning process[6]. Taking several mainstream large language models Bert, T5, and ChatGPT's predecessor GPT-3.0 as examples, this report furthermore compares the similarities and differences of these models in terms of datasets, model architectures, training approaches and applications.

(1). Dataset

ChatGPT, GPT-3, Bert and T5 all employ large datasets for pre-training and fine-tuning. ChatGPT has not yet released information about its datasets and parameters. The training data of its sibling model InstructGPT includes but is not limited to: a filtered web crawler dataset (429 billion words), Wikipedia articles (3 billion words), two different book datasets (67 billion words), human evaluation and feedback data collected by GPT-3. ChatGPT adds conversation data provided by AI trainers who play both the role of customer and AI. The number of parameters of the InstructGPT model is 175 billion, and ChatGPT should be roughly similar in terms of parameter size[4].

GPT-3, which is the basis of the ChatGPT model, is trained on a total of 499B tokens, 60% of which come from the filtered Common Crawl. The rest comes from: webtext2 (the corpus used to train GPT-2), Books1, Books2, Wikipedia, and code datasets (e.g. Github Code), etc. GPT-3, with 175 billion parameters[10], has a much larger number of parameters

than Bert and T5, making it difficult to fine-tune.

Bert is pre-trained on a token of about 137B, with datasets from BookCorpus (800M Words) and English Wikipedia (2500M Words), where only Wikipedia article texts were adopted. Tables, lists, and titles were not included in the dataset. Bert-Base has 110M parameters, and Bert- Large has 340 M parameters[11].

T5 was pre-trained on 34B tokens, which filtered the publicly crawled web dataset Common Crawl to remove some duplicates, low-quality, code-looking texts, etc., and finally kept only English texts to obtain dataset C4: the Colossal Clean Crawled Corpus, with a size of about 750GB. T5-Base has 220M parameters, T5-Large has 770M parameters, and T5-11B has 11B parameters[12].

(2). Model architecture

The model architectures of ChatGPT, GPT-3, Bert and T5 are all based on the Transformer, but they are slightly different in specific. GPT-3 uses a autoregressive model with a multi-layer Transformer Decoder and a self-attention mechanism to generate high-quality natural language text[10].

Bert uses a multi-layer bidirectional Transformer architecture, which has only an encoder but no decoder. The term “bidirectional” means that Bert processes a word in such a way that it utilizes both the preceding words and the following words. The source of this bidirectionality is that Bert, unlike traditional language models, does not predict the most likely word to appear given all preceding words, but masks some words at random and uses all unmasked words for prediction[11].

Unlike the Bert and GPT families that use only a part of the Transformer architecture, T5 uses an encoder-decoder system with the full Transformer architecture, comparable in size to Bert-Base (12 layers), but with layer normalization simplified, the activations are only rescaled and no additive bias is applied. Dropout is applied within the feed-forward network, on the attention weights, and at the input and output of the entire stack, etc.[12] T5 can encode and decode input texts for a variety of conversion tasks, and its basic idea is to treat each NLP task as a “text-to-text” problem, that is to take text as input and produce new text as output.

(3). Training approach

The training approach of ChatGPT, Bert, GPT-3 and T5 also differ. ChatGPT and GPT-3 are pre-trained using self-supervised learning, while GPT-3 using autoregressive pre-training, ChatGPT is pre-trained using a masked language model.

The training steps of ChatGPT have been described in previous chapter, and the steps can be briefly summarized as: supervised fine-tuning based on the GPT-3.5 model, then building a reward model, ranking the answers generated by the model through human labeling, then fine-tuning parameters of the model using a reinforcement learning algorithm, using the reward model as the optimization target for reinforcement learning.

In contrast to Bert and T5, which pre-trained the model first and then fine-tuned it on downstream tasks, GPT-3 does not use fine-tuning but proposes “In-context learning”. The cost of fine-tuning GPT-3 is too high due to its huge number of parameters. The approach can be divided into Zero-Shot/One-Shot/Few-Shot Prompting. Specifically, users give

prompts such as “Translate English to French”, (One-Shot/Few-Shot will still give one or a small number of examples as input) as an indication, and then the model gives the corresponding answer. In this process there is no gradient descent and back propagation as in traditional training process, and no fine-tuning and additional training is required for the user to use for downstream tasks[10].

Bert pre-training is done using a multi-task approach, including Masked Language Model (MLM) and Next Sentence Prediction (NSP). In which, the MLM is trained by Masking off some words randomly from the input sentences and then predicting the word by the context. When training the model, a sentence will be fed to the model several times for parameter learning, but Google does not mask off these words in each time, but after determining the word to be Masked off, 80% of the time it will be directly replaced with [Mask], 10% of the time it will be replaced with other arbitrary words, and 10% of the time it will keep the original token[11]. The task of NSP is to determine whether sentence B is the following of sentence A. If it is, output “IsNext”, otherwise output “NotNext”. The training data is generated by randomly extracting two consecutive sentences from the parallel corpus, where 50% of the extracted two sentences are retained and they conform to the “IsNext” relation, and the other 50% of the second sentence is randomly extracted from the corpus and their relation is “NotNext”[13].

The T5 model unifies translation, classification, regression, and summary generation tasks into Text-to-Text tasks, thus enabling these tasks to use the same objective function during training (pre-training and fine-tuning) and the same decoding process during testing. The text-to-text framework of T5 allows the same model architecture, objective, training process, and decoding process directly being applied to each task in the experiment. Tasks including translation, question and answer, and classification are converted into text for the input model, and then the model is trained to generate the target text. This ensures that the same model, loss function, hyperparameters, etc. are used for different tasks. A multi-task pre-training approach is used (unsupervised and supervised tasks are pre-trained together). In pre-training, some unique special symbols <X>, <Y> are used to denote the span or token that is randomly masked in the original sample, while the target sample is a sequence of masked spans or tokens, separated by the special symbols <X>, <Y> at the corresponding positions in the input sample, and finally a special symbol <Z> is added to indicate the end of the sequence[12].

(4). Applications

ChatGPT, Bert, GPT-3 and T5 can be applied to various tasks in the field of natural language processing, but their application directions are different. ChatGPT is currently used for tasks such as chatting systems, text generation and automated Q&A system. Bert is mainly used for tasks such as text classification, sequence annotation and Q&A systems. GPT-3 is suitable for tasks such as text generation, dialogue generation and translation. T5 can be used for various text conversion tasks.

3.3 Trial Feedback

ChatGPT is a very powerful chatting robot that can write codes and conduct other

operations in computer science field, and it can generate long texts that are clear and logical. ChatGPT is undoubtedly very smart and practical compared with other conversation models, such as GPT-3, Baidu Ernie Bot, etc.

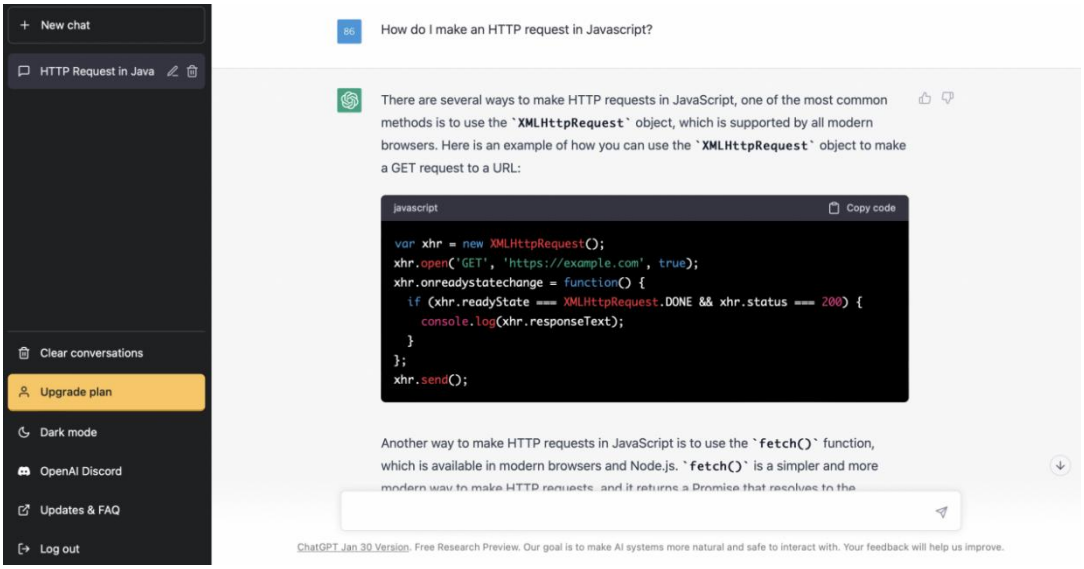

FIGURE 4. Asking ChatGPT to write Codes

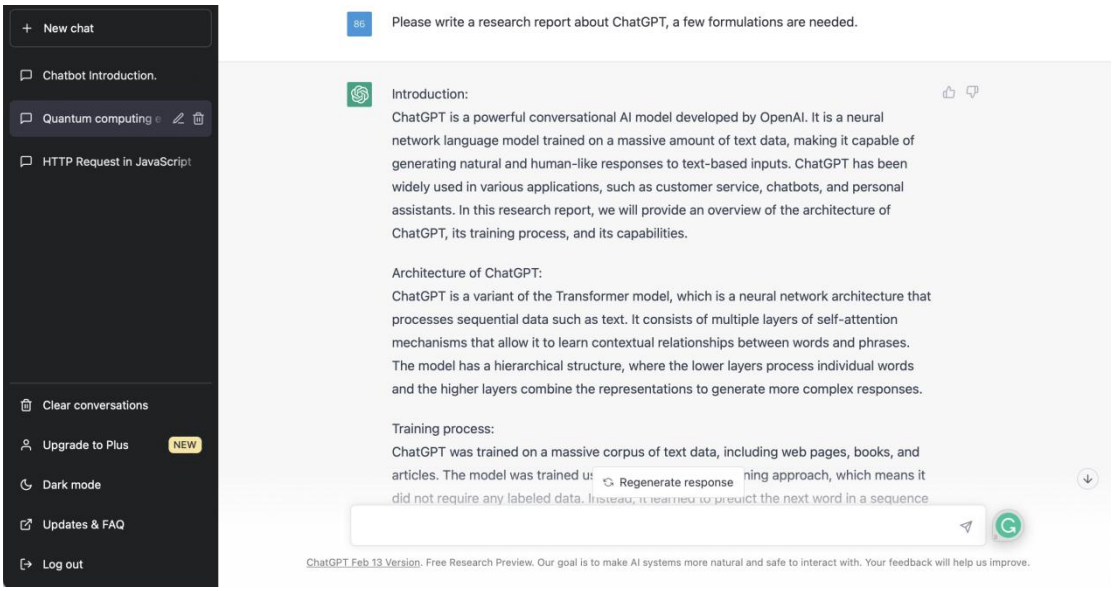

FIGURE 5. ChatGPT generating long texts

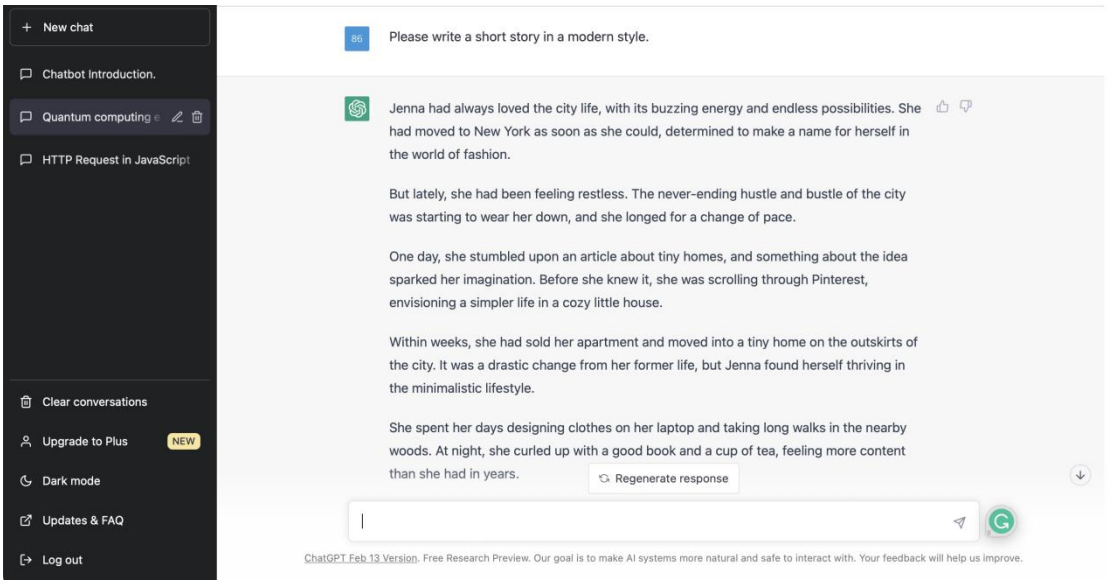

FIGURE 6. ChatGPT writing short stories

However, ChatGPT is prone to making things up when it comes to questions that it does not really "know". StackOverFlow has therefore banned answers generated by ChatGPT.

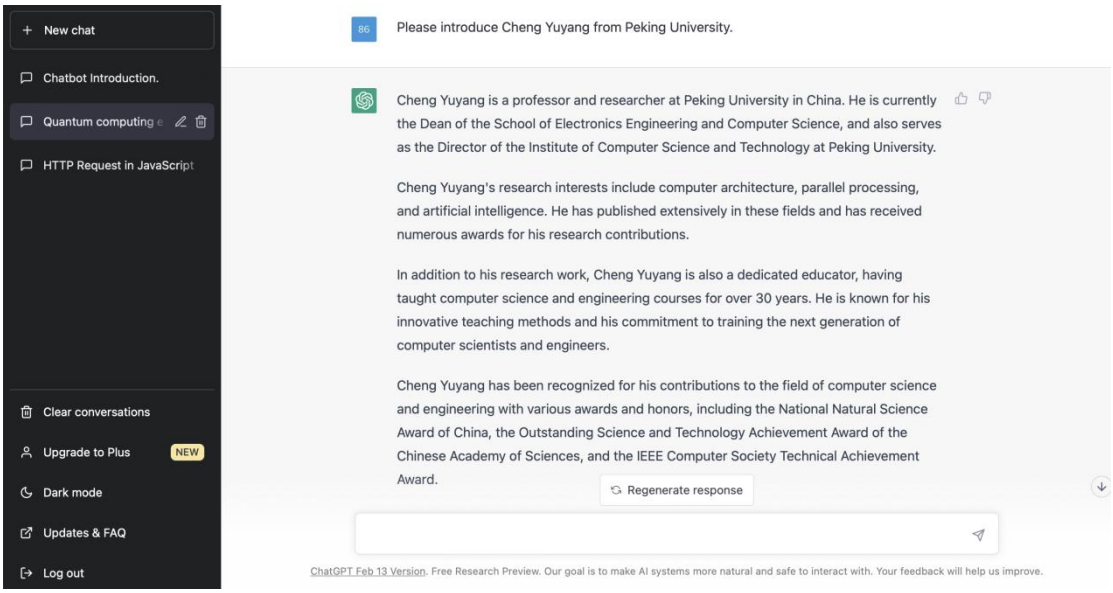

FIGURE 7. ChatGPT making things up

OpenAI has also posted blog explaining the shortcomings of ChatGPT as follows[3].

ChatGPT sometimes writes answers that sound plausible but are actually incorrect or nonsensical. Solving this problem is challenging because there is no absolutely correct source of truth in RL training; training the ChatGPT model makes it more cautious, causing ChatGPT to reject questions it can answer correctly; and supervised training misleads the model because the ideal answer depends on what the model knows, rather than what human

knows.

ChatGPT is sensitive to adjustments in the wording of input or multiple attempts with the same prompt. For example, given the wording of a question, the model can claim not to know the answer, but if applied with some minor adjustments, ChatGPT can answer correctly.

ChatGPT tends to be wordy and overuses certain phrases, such as reiterating that it is a language model trained by OpenAI. These problems arise from biases in the training data (AI trainers prefer longer answers that seem more comprehensive) and over-optimization issues.

Ideally, the model would ask clarified questions when the user provides an ambiguous prompt. But instead, current ChatGPT typically guesses the user's intent.

Although OpenAI has worked hard to make the model reject inappropriate requests, ChatGPT continues to sometimes respond to harmful instructions or exhibit biased behavior.

5. Conclusions.

ChatGPT is widely used in various natural language processing tasks, and one of its most common and well-known applications is dialogue system. It can generate responses that are natural and fluent, covering a wide range of intellectual content. In addition, ChatGPT can be used for various language tasks such as machine translation, text summarization, language modeling and text generation, and performs quite well on many tasks.

ChatGPT is an excellent work standing on the shoulders of InstructGPT and algorithms such as RLHP and PPO. Its combination of LLM (Large Language Model), PTM (Pretrain Language Model) and RL (Reinforcement Learning) provides possible development directions for performance improvement of natural language processing tasks. The use of supervised learning and high-quality human annotated datasets contributes to the excellent performance improvement of ChatGPT.

Acknowledgment. This work is partially supported by my supervisor Professor Liu Yao, and I want to express my sincere gratitude for his guidance. I also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] B. Reed, ChatGPT reaches 100 millions users 2 months after launch, https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app, 2023.

[2] Introducing ChatGPT, https://openai.com/blog/chatgpt/, 2023.

[3] Q. Zhang, T. Gui X. Huang, *Introduction to Natural Language Processing*, https://intro-nlp.github.io/, Shanghai, 2023.

[4] O. Long, et al, Training language models to follow instructions with human feedback, In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Proc. of NeurIPS*, New Orleans, pp.1-18, 2022.

[5] Aligning language models to follow instructions, https://openai.com/research/instruction-following, 2023.
[6] Illustrating Reinforcement Learning from Human Feedback (RLHF) https://huggingface.co/blog /rlhf, 2022.
[7] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, Deep reinforcement learning from human preferences, doi:10.48550/arXiv.1706.03741, pp.1-3, 2017.
[8] D. Go, T. Korbak, G. Kruszewski, J. Rozen, N. Ryu, and M. Dymetman, Aligning Language Models with Preferences through f-divergence Minimization, doi:10.48550 /arXiv.2302.08215, pp.379-380, 2023.
[9] K. Sun, X. Luo, Y. Luo. A Review of Applications of Pre-trained Language Models. *Computer Science*, vol.50, no.01, pp.177-178, 2023.
[10] T. Brown et al., Language Models are Few-Shot Learners, doi:10.48550/arXi v.2005.14165, pp.1-10, 2020.
[11] J. Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, doi:10.48550/arXiv.1810.04805, pp.1-7, 2018.
[12] C. Raffel et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, doi:10.48550/arXiv.1910.10683, pp.2-5, 2020.
[13] Z. Yue, X. Ye, R. Liu, A Review of Research on Pre-training Techniques Based On Language Models. *Journal of Chinese Information Processing*, vol.35, no.09, pp.16-20, 2021.