Utilizing the CLASS Framework to Develop a Conversational AI Tutor for open-ended problems in ASSISTments

ANONYMOUS AUTHOR(S)

We present a conversational AI tutor (CAIT) for the purpose of aiding students on middle school math problems. CAIT was created utilizing the CLASS framework, and it is an LLM fine-tuned on Vicuna using a conversational dataset created by prompting ChatGPT using problems and explanations in ASSISTments. CAIT is trained to generate scaffolding questions, provide hints, and correct mistakes on math problems. We find that CAIT identifies 60% of correct answers as correct, generates effective sub-problems 33% of the time, and has a positive sentiment 72% of the time, with the remaining 28% of interactions being neutral. This paper discusses the hurdles to further implementation of CAIT into ASSISTments, namely improved accuracy and efficacy of sub-problems, and establishes CAIT as a proof of concept that the CLASS framework can be applied to create an effective mathematics tutorbot.

CCS Concepts: • **Applied computing** → **Computer-assisted instruction**; *Interactive learning environments*; *E-learning*.

Additional Key Words and Phrases: Mathematics Education, ChatBots, Intelligent-tutoring systems, Large Language Models, GPT, Llama 2, Educational Technology

ACM Reference Format:
Anonymous Author(s). 2024. Utilizing the CLASS Framework to Develop a Conversational AI Tutor for open-ended problems in ASSISTments. In . ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Large Language Models have seen a recent surge in popularity in education literature [7–9], with a significant amount of emphasis on Intelligent Tutoring Systems (ITSs). GPT-4's wide domain knowledge and conversational abilities have allowed it to provide feedback and tutoring in a variety of contexts [10]. In the past, these models have been used for a variety of services [13], and with recent advances, we can now produce tutor bots that provide far more advanced, accurate, and reliable messages. Being able to solve problems step-by-step is both difficult for LLMs but also provides students with a greater understanding of problems and mathematical abilities [4].

Our approach was to build off the work established in CLASS meet SPOCK [12] and train an open-source large language model (LLM) for the purpose of being a math tutor chatbot. The LLM, Vicuna [15], is a chat assistant trained by fine-tuning LLaMA 2 on user-shared conversations collected from ShareGPT. We have chosen not to pretrain the model with a mathematics textbook and rather to fine-tune it using open-response problems along with correct responses. The two core functionalities we hope to impart in our model are: 1) Being able to break down any mathematics word problem into easily digestible steps (scaffolding) with appropriate logic and 2) Help students when they need hints or make errors. We analyze the performance of our model to explore whether it is safe to evaluate with students, and whether it would be helpful for students.

Manuscript submitted to ACM

2 BACKGROUND

2.1 Intelligent Tutoring Systems

Intelligent tutoring systems have become increasingly prolific during the COVID-19 pandemic, with large numbers of classes switching to remote learning [1]. The most prominent style of ITS use natural language feedback to explain concepts and misconceptions, while other ITS focus on the constraint-based modeling scaffolding approach, with a set of problems and solutions on very specific conditions which allow for feedback accounting for all student response types [2]. Our model uses the scaffolding approach to build subproblems around mathematics word problems with a natural language interface. The scaffolding approach has historically been very labor-intensive, as an instructor would have to go through hundreds of individual problems and develop explanations for subproblems and consider the students' line of reasoning at each step. Now, with the power of large language models, we can compute this automatically and with no teacher assistance, saving time for teachers.

2.2 ASSISTments

ASSISTments is an online learning and research platform developed by the ASSISTments foundation [3]. The key innovation that differentiates ASSISTments from prior attempts at Intelligent Tutoring Systems is its ability to effortlessly collect student performance and usage data and to run experiments within the ASSISTments platform. To date, ASSISTments has been used in over 50 studies to better understanding learning science and student interaction, and is widely cited in the literature [11]. Our work does not directly leverage the ASSISTment platform in an experimental manner, but rather sources the data of open response mathematics problems and student responses to train our model to accurately scaffold and explain subproblems using natural language.

2.3 CLASS meet SPOCK

Intelligent tutoring systems have benefitted greatly from the recent improvements of LLMs, such as GPT-4.0 and Llama 2, with studies already demonstrating the potential in education. Moreover, fine-tuning small LLMs on specific tasks has become a cost effective manner to create powerful LLMs for the desired task [5, 6]. One such example is CLASS meet SPOCK [12], which utilized the CLASS framework to fine-tune Vicuna-13B, SPOCK, to create a biology chatbot. We seek to explore if a model can perform equally well in math. The key difference between CAIT and SPOCK is that SPOCK was given information from a biology textbook in both the conversational dataset and when being evaluated to increase accuracy. This allowed SPOCK to generate useful hints and utilize facts from a textbook when conversing with students. In contrast, CAIT's conversational dataset generated by GPT-4.0 was given teacher written explanations to problems when generating a dataset. Additionally, SPOCK was trained on both a scaffolding question dataset and conversational dataset, whereas CAIT was only trained on a conversational dataset. Thirdly, we opted to use Vicuna-7B instead of Vicuna-13B because we hope to show a smaller fine-tuned model will be capable of becoming an accurate and helpful tutorbot. Smaller models also have the benefit of generating text faster, which we believe will be useful as students might get impatient when waiting for text to generate when CAIT is deployed on ASSISTments.

3 METHODOLOGY

3.1 Dataset

Following the CLASS meet SPOCK framework, we prompted GPT-4 to simulate prompts between students and a tutor-bot. GPT-4 simulated conversations where it was both the tutee and tutorbot. As a tutee, GPT-4 would simulate

incorrect answers and correct answers to sub-problems, and statements such as "I don't know". As a tutor, GPT-4 would provide constructive criticism, identify errors in the tutee solution, and provide sub-problems. GPT was also given a teacher authored explanation of each problem to improve accuracy and reduce hallucinations. We gave GPT 800 problems from ASSISTments which had teacher authored explanations. 50% of the problems were from problems tagged as 6th grade problems, 34% were from problems tagged as 7th grade math problems, and 16% were tagged as 8th grade math problems.

Of the dataset, only 753 problems were in order to train CAIT due to 47 having issues reading the JSON. Some of the problems not used were used in the evaluation dataset. Of the 753 conversations used to train CAIT, the conversations had an average of 6.1 messages per conversation. We utilized the following prompt to generate conversations with GPT4.0:

```
   Your goal is to create a mock conversation between Student and a Tutorbot, an AI-powered chatbot designed
    to help Student's with a question:
Question: {problem}

"Student": "Q. {problem}",
"Thoughts of Tutorbot": ".."
"Evaluation of Student Response": ".."
"Action Based on Evaluation": ".."
"Subproblem State": ".."
"Subproblem": ".."
"Tutorbot": "Let's break the problem into subproblems and tackle the subproblems one by one. Let's begin
with the first subproblem...",


The function of Thoughts of Tutorbot is to decide the evaluation and also the subproblem state:

a) Evaluating Incorrect Responses
b) Evaluating Correct Responses
c) Evaluating Partially Correct Responses
d) Evaluating Ambiguous or Unclear or Short Responses
e) Redirecting Off-topic Responses
f) Responding to Student Inquiries
g) N/A

Here is an explanation for the problem:
{explanation}

Tutorbot Actions Based on the Evaluation:
```

If "a" is the evaluation, then:
1) Promptly notify the student about the mistake, Provide constructive feedback to pinpoint the errors, Offer helpful hints
2) Step in to provide a solution if the student is unable to answer even after multiple attempts.

If "b" is the evaluation, then:
3) Confirm the correct answer. Check for completeness for the answer to the subproblem. If solution is incomplete, notify the student to complete the solution.

If "c" is the evaluation, then:
4) Acknowledge the accurate parts, Promptly notify the student about the mistake, Provide constructive feedback to pinpoint the errors, Offer helpful hints
5) Step in to provide a solution if the student is unable to answer even after multiple attempts.

If "d" is the evaluation, then:
6) Actively seek clarification through relevant follow-up questions. Request the student to provide more specific information.

If "e" is the evaluation, then:
7) Skillfully redirect the student's attention to the subject matter. Provide guidance on how to approach the question appropriately.

If "f" is the evaluation, then:
8) If student asks for a hint, provide a hint for the current subproblem.
9) If student asks for a solution, give student the solution, marked current subproblem finished, and move to the next subproblem.
10) If student asks to move to previous subproblem, marked current subproblem finished, and move to the previous subproblem.
11) If none apply, prioritize addressing the inquiry. Offer relevant support and guidance to meet the student's specific needs.

If "g" is the evaluation, then:
12) N/A

Function of Subproblem State is to guide through subproblems:
w) N/A
x) One of the subproblems is currently being solved
y) Subproblem finished, moving to next subproblem that is not finished
z) Subproblem finished, no next subproblem, problem finished

Now, let's begin. Your goal is to create a mock conversation between Student and a Tutorbot, an AI-powered

```
chatbot designed to help Student's with a question.

Please create a mock conversation now. Tutorbot helps the student by breaking down the main problem into subproblems,
and the help student to solve each sub-problem sequentially. Tutorbot only provide hints.
Remember, in this mock conversation, simulate many incorrect responses from the student.
Use the following json format:

Put all the output in the following JSON structure
[{{
   "Student": "..",
   "Thoughts of Tutorbot": ".."
   "Evaluation of Student Response": "a,b,c,d,e,f,g"
   "Action Based on Evaluation": "1,2,3,4,5,6,7,8,9,10,11,12"
   "Subproblem State": "w,x,y,z"
   "Subproblem": ".."
   "Tutorbot": "..",
}},
Repeat above N times.
]

Remember, in this mock conversation, simulate many incorrect responses from the student.
```

3.2 Fine-tuning CAIT

We used vicuna-7b-v1.5 as the base model. CAIT was fine-tuned on the conversation dataset aforementioned using the Causal Language Model (CLM) loss. We utilized QloRA in order to efficiently and effectively fine-tune CAIT. CAIT was fine-tuned for 28 epochs which took 3 hours on an A100 GPU to achieve a loss of 0.0092. The cost of generating our dataset was $50.

3.3 Testing Questions

We then determined the efficacy of CAIT using 10 different questions from ASSISTments. Two of the questions were from the 47 which had a incorrect JSON output. The rest were other ASSISTments open-ended questions used by teachers. We utilized questions of varying difficulty. Some of the problems we consider trivial, meaning they have no helpful sub-problems and are generally quite straightforward, while other problems were not trivial and had sub-problems or formulas to recall.

The 6 trivial sub-problems used to evaluate CAIT were:

1. Carlos needs 1.7 meters of wire for one project and 0.8 meter of wire for another project. How many meters of wire does he need total?
2. Helen gave each of her 7 grandchildren an equal amount of money. If she gave a total of $227.50, how much did each grandchild get?
3. Ashley bought a big bag of candy. The bag had 102 blue candies, 100 red candies and 94 green candies.

How many candies were there in total?
4. How many cookies did you sell if you sold 320 chocolate cookies and 270 vanilla cookies?
5. Billy had 2 books at home. He went to the library to take out 2 more books. He then bought 1 book. How many books does Billy have now?
6. There were 3 pizzas in total at the pizza shop. A customer bought 1 pizza. How many pizzas are left?

The 4 non-trivial problems used to evaluate CAIT were:

1. Solve for x using the quadratic formula: x^2+ 6x= 3x-5
2. What is 13.5 written as a mixed number?
3. A baby came in for its 6 month appointment. It initially weighed 17.5 pounds. The baby loses 0.5 pounds per month. Write a linear equation for the problem above.
4. A person is 800 meters from shore. They are rowing towards shore at 30 meters per minute. Write an expression that will give their position at time t.

We simulated at least 2 conversations on each question, one where the student provides only correct answers to subproblems and one where the student either gives wrong answers, requests help or hints, or asks for the final answer. We analyzed the correctness of CAIT as well as its sentiment when responding to students.
To improve the reasoning ability of our tutor, we employed Chain-of-Thought prompting since it has been shown to improve performance, particularly on math questions[14]. In order to employ Chain-of-Thought we have our chatbot remember the prompt, prior subproblems and prior student inputs.

4 ANALYSIS AND FINDINGS

4.1 Accuracy

We found issues in 4 problems when students provided the correct answer: trivial problems 1 and 2 and non-trivial 1 and 2. In trivial problem 1, since it was trivial the student was solving the sub-problem:

What is the total amount of wire needed?

That led to this discussion:

Student: Maybe 2.5 meters?
Tutorbot: That's not correct. Let's try to break it down further.
STUDENT: why?
Tutorbot: Because we know the exact amount of wire required for each project separately. We just need to add those amounts together.
STUDENT: So can you give me the answer?
Tutorbot: Sure! For the first project, Carlos needs 1.7 meters of wire, and for the second project, he needs 0.8 meters of wire. So, the total amount
of wire needed for both projects combined is 1.7 + 0.8 = 2.5 meters.
STUDENT: So, my prior answer of 2.5 meters is right.
Tutorbot: I apologize, but your answer is incorrect. According to our calculations, the total amount of wire needed for both projects combined is 2.5 meters.
STUDENT: I said 2.5 meters before

`TUTORBOT: Tutorbot: I apologize for any confusion. You are correct, the total amount of wire needed for both projects combined is indeed 2.5 meters. Good job!`

In trivial question 2, the tutorbot ran into issues due to misunderstanding the problem. The tutorbot agreed with the student when they got $32.50 for each grandchild, but then always proceeded to ask the student to multiply by 7 to find the total amount of money given to grandchildren.

In non-trivial question 1 the tutorbot made an error when solving the sub-problem: Find a, b, and c. The tutorbot found a=1, b=3 and c=-5, when in fact c=5.

In non-trivial question 2, the tutorbot attempted to find 13/5 as a mixed number rather than 13.5. Additionally, the tutorbot found 13/5 = $2\frac{1}{5}$ rather than $2\frac{3}{5}$.

Other questions were generally correct, however, the tutorbot occasionally asked students to verify their answers. For instance, when entering a correct answer, tutorbot would sometimes say "please verify your answer" and would not say "correct" until the student had entered their answer a second time.

When students gave an incorrect answer, the tutorbot identified an incorrect response 5 out of 10 times. For example, when student entered that each grandchild got $10, the tutorbot said "correct" and continued on (the correct answer was $32.50. Additionally, the tutorbot said "correct" to an incorrect answer for trivial problems 2 and 3 and non-trivial problems 1, 2 and 4.

The tutorbot struggled most with generating sub-problems to solve the original problem. In trivial question 2, the tutorbot always asked for the total amount of money given to all of the grandchildren. In non-trivial question 3, the tutorbot would often solve for the baby's weight at 6 months rather than looking for a linear equation to solve the problem. Similarly, for non-trivial question 4, the tutorbot would ask the student to find the time when the person returns to shore, rather than writing an expression for their position.

4.2 Usefulness

When generating sub-problems for math, CAIT was only able to provide useful and actionable information a mere 33% of the time. The remaining 67% of cases were either not helpful or actively leading students astray. At times, CAIT had trouble conceptualizing even simple math problems like adding two numbers. In trivial problems, the tutorbot would typically provide the question itself as the sub-problem, as coming up with a useful sub-problem would be difficult. In non-trivial problems, the sub-problems were either useful or demonstrated a misunderstanding of the problem.

4.3 Sentiment

We found CAIT to present a positive sentiment 72% of the time, stating things such as "excellent!", "sure!", and "You're welcome! I'm glad I could help. If you have any more questions or need further assistance, feel free to ask.". The other 28% responses were neutral, lacking any supportive or positive language but also not having any demeaning or unsupportive language. There were no negative sentiment responses.

5 DISCUSSION

5.1 CAIT

CAIT is a proof-of-concept tutorbot which requires improvements before implementation in tools such as ASSISTments. CAIT often gave correct answers and always gave answers which were written in fluent English. Where CAIT struggles

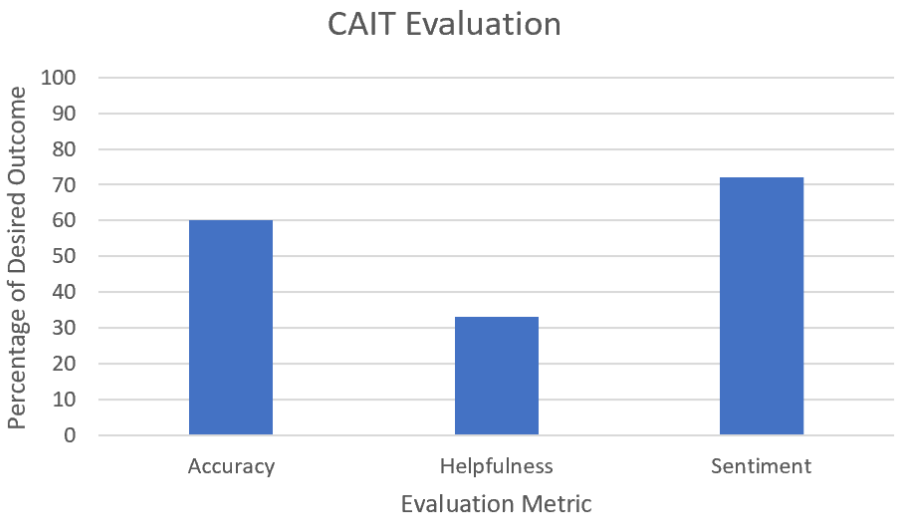

Fig. 1. Different Metrics for Evaluating our Model

most is recognizing and correcting incorrect responses. One potential improvement is to provide CAIT extra relevant information when a student asked a question, as that is one technique which CLASS meet SPOCK found improved performance. We believe providing CAIT with additional information from a textbook or relevant equations would likely improve the accuracy. Additionally, we found CAIT to display a low level of helpfulness in generating useful subproblems. This may be due to its inability to parse out basic details from math word problems. Moreover, not all of our problems were supportive of subproblems which likely contributed to the difficulty coming up with subproblems. Sentiment-wise, CAIT was broadly positive in 72% of cases, with the remaining responses being neutral. We believe CAIT was appropriately happy and appropriately neutral and is safe to be deployed if all we cared about was sentiment.

5.2 Limitations

We recognize that we did not behave exactly as typical ASSISTments users would, as we are not middle school students in math class. We also had a relatively small sample size of mock conversations between CAIT and a student due to the variety of answers CAIT could give to the same prompt. More mock conversations would be useful to determine the actual effectiveness of CAIT. Additionally, we did not explore every type of input a student could give. Notably, we did not test off-topic responses from students to see if CAIT would appropriately redirect.

6 FUTURE WORK

We intend to refine and improve CAIT, with a key future direction providing relevant information when a student asks a question. Following improvements and additional evaluations, including safety, we plan to deploy CAIT into ASSISTments for a randomized control trial. Select students in the trial will be able to ask CAIT questions and receive answers. Students will then be able to rate the responses of CAIT. We will then use student ratings for Reinforcement Learning with Human Feedback to iteratively improve CAIT. Moreover, we intend to perform a randomized control

trial to see if students who interact with CAIT will perform better on questions than students who did not have access to CAIT.

We planned to evaluate our model output with actual math teachers. Future work will analyze our model from a learning perspective and gauge how willing teachers would be to recommend responses, with a standardized rubric for evaluation of accuracy, sentiment and helpfulness. We also intend to use more difficult and complicated problems to determine the performance of CAIT on more types of math problems.

We also planned to fine-tune a LLM that is able to take long conversations as input. The lengths of more than a half of training dataset are over 6500, while the context length of the used model is only 4096 tokens. While a token is approximately 4 characters, it is possible that a few conversations were more than 4096 tokens. If a conversation were over 4096 tokens, the model would automatically truncate the data to the length of 4096, which would decrease the performance of the model.

7 CONCLUSION

In this paper, we discussed the implementation of a new intelligent tutoring bot, CAIT, and how it performed in simulate conversations between itself and a student. We built our bot using the ideas presented in [12]. We created training data using simulated conversations between a tutorbot and a student in ChatGPT which we intend tor release publically. We then fine-tuned Vicuna on this data to create CAIT. Finally, we evaluated CAIT on open ended math questions to analyze its accuracy, helpfulness and positivity.

8 APPENDICES

REFERENCES

[1] Jijuan Cao, Ting Yang, Ivan Ka-Wai Lai, and Jun Wu. 2021. Student acceptance of intelligent tutoring systems during COVID-19: The effect of political influence. *The International Journal of Electrical Engineering &amp Education* (March 2021), 002072092110032. https://doi.org/10.1177/00207209211003270

[2] Arthur C. Graesser, Mark W. Conley, and Andrew Olney. 2012. Intelligent tutoring systems. In *APA educational psychology handbook, Vol 3: Application to learning and teaching*. American Psychological Association, 451–473. https://doi.org/10.1037/13275-018

[3] Neil T. Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (Sept. 2014), 470–497. https://doi.org/10.1007/s40593-014-0024-x

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. https://doi.org/10.48550/ARXIV.2103.03874

[5] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. arXiv:2305.02301 [cs.CL]

[6] Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, Xing Xu, and Soujanya Poria. 2023. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. arXiv:2304.01933 [cs.CL]

[7] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* 103 (2023), 102274. https://doi.org/10.1016/j.lindif.2023.102274

[8] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLoS digital health* 2, 2 (2023), e0000198.

[9] Huy A. Nguyen, Hayden Stec, Xinying Hou, Sarah Di, and Bruce M. McLaren. 2023. Evaluating ChatGPT's Decimal Skills and Feedback Generation in a Digital Learning Game. In *Responsive and Sustainable Educational Futures*, Olga Viberg, Ioana Jivet, Pedro J. Muñoz-Merino, Maria Perifanou, and Tina Papathoma (Eds.). Vol. 14200. Springer Nature Switzerland, Cham, 278–293. https://doi.org/10.1007/978-3-031-42682-7_19 Series Title: Lecture Notes in Computer Science.

[10] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[11] Ethan Prithar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. [n. d.]. Exploring Common Trends in Online Educational Experiments. *Proceedings of the 15th International Conference on Educational Data Mining* ([n. d.]). https://par.nsf.gov/biblio/10354096

[12] Shashank Sonkar, Lucy Liu, Debshila Basu Mallick, and Richard G. Baraniuk. 2023. CLASS Meet SPOCK: An Education Tutoring Chatbot based on Learning Science Principles. arXiv:2305.13272 [cs.CL]

[13] Rainer Winkler and Matthias Söllner. 2018. Unleashing the potential of chatbots in education: A state-of-the-art analysis. In *Academy of Management Proceedings*, Vol. 2018. Academy of Management Briarcliff Manor, NY 10510, 15903.

[14] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493 [cs.CL]

[15] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685 [cs.CL]