Advancing Multi-Tool Usage with Chain-of-Abstraction Reasoning

Abhishek Phaltankar {aphaltankar, **Anushka Yadav** anushkayadav, **Yash Kothekar** ykothekar, **Arushi Misra** arushimisra }@umass.edu

1 Problem statement

One major issue we're observing in LLMs today is their tendency to hallucinate. It's something we've all encountered at some point. Despite being trained on vast amounts of data, LLMs essentially perform next-word prediction. This can lead to inaccuracies, especially in factual information. We're aware that LLMs struggle with mathematical problem-solving. Our proposal is to fine-tune LLMs for Chain-of-Thought reasoning by providing them with a question and enabling them to determine the appropriate tools needed to answer it correctly. We also aim to fine-tune LLMs so they can execute not just single-tool operations but multiple-tool operations, such as conducting a Wikipedia search followed by an arithmetic operation for a given question.

2 What you proposed vs. what you accomplished

Our initial goal was to replicate the performance of Toolformer (14), with added support for multi-tool APIs. During implementation, we came across a better approach using chain of abstraction (CoA) (4), which performed better than toolformer. Hence our goal was to add multi-tool support for Chain of thought abstraction. Here is a step by step list of the things we proposed and it's status:

- ✓Collection and preprocessing dataset: We pre-processed data from three datasets namely GSM8K, HotpotQA, NumGLUE as was proposed by us to get answers for questions involving Single tool API calls, Multi-Tool API calls without Chaining, and Multi-Tool API calls with Chaining.
- ✓Tool execution without blocking decoding: Another feature of Toolformer is that it pauses generation at the API call, waits for the response and then resumes text generation. We avoided this by generating the text continuously (has been referred as C in the forthcoming sections) using variables instead of the actual responses. Once all of the API calls are completed, we populate these variables using their responses. This allowed us to process all the API calls in one batch, preventing decoder blocking.
- ✓Build and train (specific baseline model) on collected dataset and examine its performance: We trained the baseline models (Mistral-7B and Gemma-2B) on the filtered data, and evaluated the performance on the testing set against the gold labels.
- ✓Perform in-depth error analysis to figure out what kinds of examples our approach struggles with: In-depth error analysis can be found in Section 7.

Highlights:

- While the original CoA paper managed to filter only 15% of the synthetic data from LLaMA-70B, our custom Filtering Module achieved a much higher filtration rate of approximately 50%. This significant improvement highlights the efficacy of our data processing approach.
- Unlike the CoA paper, which trained separate models on different datasets, we successfully trained a single model on all datasets. This unified approach not only taught the model to use all tools effectively but also allowed us to experiment with multi-tool chaining, whereas the COA paper employed single tool chaining. This distinction underscores the versatility and robustness of our model.

3 Related work

Since the wide use of LLMs, there has been an ongoing interest in making the model-generated text more reliable, especially when it comes to factual knowledge and mathematical equations. The initial approaches, for eg Tool Augmented Language Model(11) focussed on finetuning the model on certain datasets to improve their performance on them. This also led to creation of specific datasets, for eg: An extensive math word problem dataset (10) and benchmarks, for eg: Knowledge-based benchmark (13). There has also been work done in the first generate and then validate approach, where external tools were used to evaluate and then update the generated text (5) (3).

Most of the recent research in this field steers in the direction of using external tools to either verify or generate certain sections of data. There have even been instances where a model has been finetuned to write optimised API calls (12). The earliest work which inspired this method was Google's Toolformer (14). Here the model automatically decides to call position-wise appropriate APIs to complete the required text. The dataset generation is one of the novel parts of this approach. The initial data is annotated with different API calls using LLMs. The base model has been then finetuned on this dataset. One distinction of the Toolformer approach is that it does API calls when the model encounters API>tag among the top k likely tokens, and stops decoding until it receives the response. Toolformer also doesn't support use of multiple types of APIs for a single query. GPT4Tools (16) supposedly improved upon this approach by leveraging tools based on self instruct from Advanced LLMs. For eg, ChatGPT is used to generate tool-specific instruction data, which is used to tune the base model. This approach still has it's flaws as it doesn't consider the inherent problem in addressing the validity of LLM generated data. Both of the approaches discussed above require model finetuning, and have been experimented to support handful of API's/tools. ToolkenGPT(6) aims to introduce tokens for specific tools. Here each tool is associated with an embedding learnt by the LLM. During generation, when we predict a toolken, the LLM temporarily produces input arguments for the tool to execute, and then insert the output back into the prompt (Like toolformer). The advantage of Toolken is the plug-and-play method, which allows the support for large number of tools, unlike the previous approaches.

The main inspiration behind our project was Meta's Chain of Abstraction-based approach(4). Unlike Toolformer and Toolken the decoding process isn't halted to make API Calls, and we still rely on external APIs, rather than large LLMs for information. This approach uses variables as placeholder for the information, and later fills in the values for them using consecutive API calls. One improvement over Meta's paper we were able to demonstrate was the support for multi-tool chaining. That is when an API output from one tool is used as an input for another tool. Our work is restricted to the usage of following tools: Calculator (Math based), Wiki-Tool(knowledge based). We also refered Multitool COT paper for NumGlue dataset (7).

4 Your dataset

We experimented our approach with three datasets:

1. **GSM8K dataset**: The GSM8K (Grade School Math 8K) dataset comprises 8.5K high-quality grade school math word problems designed for multi-step reasoning. These problems typically involve basic arithmetic operations like addition, subtraction, multiplication, and division, and they can be solved by bright middle school students without advanced algebraic concepts. The text in the dataset is in English. The configuration contains 7473 training instances and 1319 testing instances. Each instance includes a question and its solution with multiple steps of reasoning and calculator annotations. Here's an example from the dataset:

   Input (question): "Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?"

   Output (answer): "Natalia sold 48/2 = $\ll 48/2 = 24 \gg$ 24 clips in May.
   Natalia sold 48+24 = $\ll 48 + 24 = 72 \gg$ 72 clips altogether in April and May. ### 72"

   Solutions (answers) are provided in natural language, not just mathematical expressions, requiring the model to understand and process textual information. Also, the

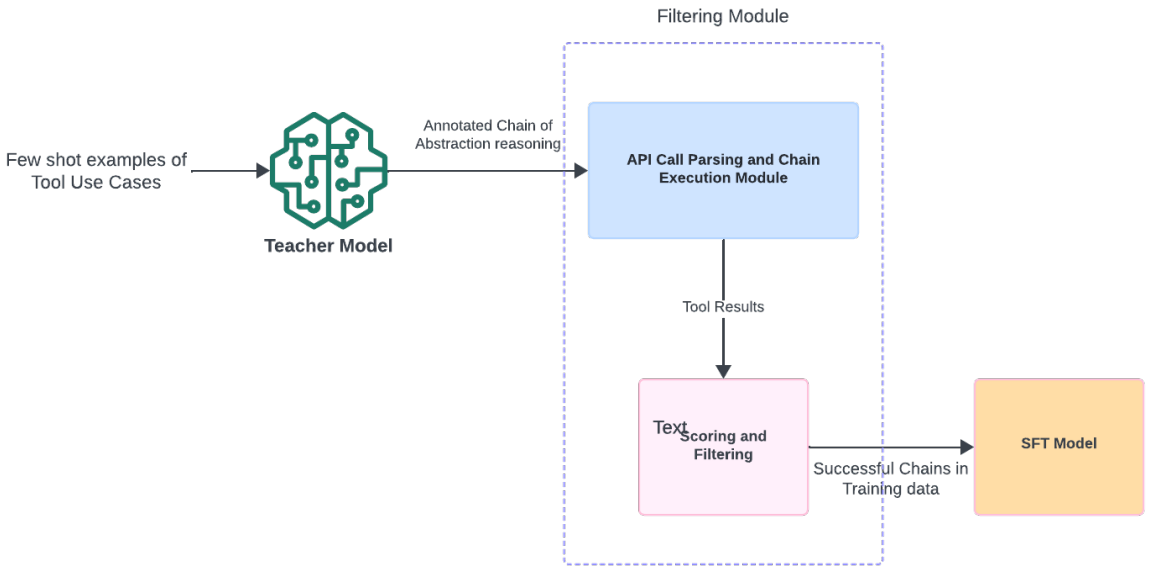

Figure 1: Workflow Diagram of the System

dataset includes problems with various linguistic styles, which adds complexity to the task.

2. **HotpotQnA fullwiki dataset**: The HotpotQnA dataset is a comprehensive collection of Wikipedia-based question-answer pairs, totaling 113k pairs, encompassing two distinct configurations: the distractor and fullwiki configurations. The fullwiki configuration that is being used by us includes 90447 training instances, 7405 validation instances, and 7405 test instances. The key feature of this dataset is its emphasis on complex reasoning, as questions often require cross-document reasoning over multiple supporting documents. Moreover, the questions cover a wide range of topics and are not constrained by pre-existing knowledge bases or schemas, making them diverse and challenging.

   In the fullwiki configuration of the HotpotQnA dataset, each instance includes several key fields that provide important information about the question-answer pair and its context. These are the columns that we worked with: id, question, answer, type, level, supporting_facts, context.

3. **MultiTool dataset**: This dataset is made up of two different datasets.

   - **NumGLUE dataset**: It is a multi-task benchmark that evaluates the performance of AI systems on eight different tasks, that at their core require simple arithmetic understanding. Within this dataset, we focused specifically on Type_1 question-answer pairs. Type_1 QA pairs necessitate a level of commonsense understanding and the ability to perform arithmetic reasoning. In total, there are 282 Type_1 QA pairs included in the dataset.

   - **Synthetically generated MultiTool Dataset from HotpotQnA**: The NumGLUE dataset contains fewer data points compared to the HotpotQnA and GSM8k datasets. To address this imbalance, we opted to synthetically generate more data involving the execution of multiple tools. Using HotpotQnA as a foundation, we identified all question-answer pairs resulting in numerical answers, totaling 15,122 out of the initial 90,447 data points. We then prompted Gemini Pro and Llama 3 to inject a mathematical dimension into questions with numerical responses. Gemini Pro 1.0 contributed 2,225 useful data points, while Llama 3 provided 1,993 data points in this enhancement process.

4.1 Data Creation and Preprocessing

Preprocessing steps were applied to all the aforementioned datasets. Let's detail each of them:

GSM8k dataset: In the dataset's example instance, we see that the answer contained content within ≪≫ brackets. Firstly, this content was removed. Additionally, "###" in the answer was replaced with 'The answer is." Commas were removed from numbers in the answer with commas.

After these steps, each instance's answer was replaced with a cleaned-up version. This cleaned-up answer is then used alongside the corresponding question (Q) as input for Gemini Pro 1.0, which generates the rewritten answer as an abstract reasoning chain (C). In particular, we instruct LLM to identify and label the portions of gold answers that represent knowledge operations, such as mathematical derivations. Subsequently, we transform these sentences with labeled sections into fillable CoA traces, where the outcomes of the operations are substituted with abstract placeholders. For instance, in the following example, the derivation is reformulated as "[21 - 15 = y1]."

Table 1: Example of GSM8K dataset

GSM8k Data
Question:There are 15 trees in the grove. Workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees will the grove workers plant today? **Answer**: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21-15=6. The answer is 6. **C**: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been [21 - 15 = y1]. The answer is y1.

We provide our few-shot prompting examples along with the prompt for CoA data re-writing in Appendix A for the GSM8k Dataset.

HotpotQnA dataset: Out of many fields in this dataset, only question, answer, supporting_facts, and context were used for further processing. Deriving the Chain of Abstraction (C) as specified in the (4) was our aim for this dataset. In order to do this, we had to first extract Wikipedia references (W) by referring to supporting facts, and context structured as "article title >related article content". This newly formed W with question, answer was given as an input to Gemini pro 1.0 to generate C which included API calls for Wikipedia/Google(Wiki) and distilbert-base-cased-distilled-squad(QA). For example, you can refer the tbale below for HotpotQnA data. The final answer will be mentioned at the end of the C as "[y1 -QA(when was the movie Knives Out released)->y2]. The final answer is y2.".
We provide our few-shot prompting examples along with the prompt for CoA data re-writing in Appendix B for the HotpotQnA Dataset.

Table 2: Example of HotpotQnA dataset

HotpotQnA Data
Question: Who was once considered the best kick boxer in the world, however he has been involved in a number of controversies relating to his "unsportsmanlike conducts" in the sport and crimes of violence outside of the ring. **Answer:** Badr Hari **W:**Global Fighting Championship >Fighters from around world on the roster include Badr Hari, Peter Aerts, Peter Graham, Dewey Cooper, Zabit Samedov. — Global Fighting Championship >It was considered as one of the biggest kickboxing and MMA promotion in Middle East. — Badr Hari >Badr Hari (Arabic: ; born 8 December 1984) is a Moroccan-Dutch super heavyweight kickboxer from Amsterdam, fighting out of Mike's Gym in Oostzaan. — Badr Hari >Hari has been a prominent figure in the world of kickboxing and was once considered the best kickboxer in the world, however he has been involved in a number of controversies relating to his "unsportsmanlike conducts" in the sport and crimes of violence outside of the ring. **C:** Search [Badr Hari -Wiki->y1]. Then determine [y1 -QA(Who was once considered the best kick boxer in the world?)->y2]. The answer is y2.

MultiTool dataset:

- **NumGLUE dataset**: Given a question and a gold answer, Gemini Pro was prompted to generate the re-writing of answer as abstract reasoning chain (C). C includes API calls to Google SERP/Wikipedia(Wiki), distilbert-base-cased-distilled-squad(QA), and Sympy equation solver for solving equations enclosed in []. The final answer to the question is contained in C as "The answer is .." Here's an example:

Table 3: Example of NumGlue dataset

NumGlue Data
Question: In a parking lot, There are 10 cars and 2 bikes. Find out the number of wheels in that parking lot. **Answer:** 44 **C:** First search [number of wheels a car has -Wiki-> y_1]. A car has [y_1 -QA(How many wheels does a car have?)-> y_2] wheels each. Then search [number of wheels a bike has -Wiki-> y_3]. A bike has [y_3 -QA(How many wheels does a bike have?)-> y_4] wheels each. There are 10 cars, so they have [$10 \times y_2 = y_5$] wheels. And there are 2 bikes, so they have [$2 \times y_4 = y_6$] wheels. Total number of wheels in the parking lot is [$y_5 + y_6 = y_7$]. The answer is y_7.

We provide our few-shot prompting examples along with the prompt for CoA data re-writing in Appendix C for the NumGLUE Dataset.

- **Synthetically generated dataset:** We concentrated on using just the question, output, supporting_facts, and context for additional processing out of all the fields in our dataset. Deriving the Chain of Abstraction (C) according to (4) was our aim. To do this, we first used the context and supporting information from the Wikipedia references (W), which are organised as "article title >related article content."

  Then, a new mathematical layer was added to this newly created data, requiring changes to our original questions and responses. Gemini Pro 1.0 was used to encourage us to create the Chain of Abstraction (C), which comprised API calls to Wikipedia/Google (Wiki) and distilbert-base-cased-distilled-squad (QA), in addition to Sympy for handling polynomial expressions.

  We concentrated on using just the question, response, supporting information, and context for additional processing out of all the fields in our dataset. Deriving the Chain of Abstraction (C) according to (4) was our aim. To do this, we first used the context and supporting information from the Wikipedia references (W), which are organised as "article title >related article content."

  Then, a new mathematical layer was added to this newly created data, requiring changes to our original questions and responses. Gemini Pro 1.0 was used to encourage us to create the Chain of Abstraction (C), which comprised API calls to Wikipedia/Google (Wiki) and distilbert-base-cased-distilled-squad (QA), in addition to Sympy for handling polynomial expressions.

  Take a look at the table, for instance. The concluding response is displayed at the Chain of Abstraction's conclusion as "[y2 + 5 = y3]." The answer is y3."

  Here are the details of preprocessing:

  - Identification of Numerical Answers
  - Removal of Missing Values
  - Exclusion of Non-Generated Data
  - Addition of Supporting Texts in Column C
  - Correction of API Calls

  Appendix D for the Sythetic HotpotQnA Dataset contains our few-shot prompting examples as well as the prompt for rewriting CoA data.

4.2 Data filtration

- **GSM8k dataset:** Once C has been created for all question-answer pairs, the equations within C are solved using the Sympy tool (9). The resulting answer from the tool is then checked against the gold answer. If the two exactly answers match, that specific instance is deemed eligible for inclusion in our dataset.

  There were a few cases where C was not generated correctly by the LLM, leading to an

Table 4: Example of Synthetically generated dataset

Synthetically generated Data from Hotpot QnA
Question: What year did the American player soccer franchise based in Carson, California whose president is Chris Klein begin to play, multiplied by 5? **W**: Chris Klein (soccer)>Chris Klein (Jan 4, 1976 in St. Louis, Missouri) is a former American soccer player.. **Answer**: 9900 **C**:First search for [LA Galaxy start -Wiki->y1]. Year the club began play [y1 -QA(which year play start?) $\rightarrow y2$]. Multiply by 5 [y2 * 5 = y3]. The answer is y3.

incorrect final answer. Additionally, we encountered instances where C had formatting issues, such as equations not being properly enclosed within square brackets. These instances were either manually corrected or excluded from the dataset.

After this filtration process, a dataset consisting of **809** samples was prepared for use in fine-tuning the model.

- **HotpotQnA dataset and MultiTool dataset:**

  First step for processing C in both HotpotQnA and MultiTool dataset is to get output for the keywords annotated as Wiki in C, requiring a WikiSearch or a Google Search. In order to achieve this, initially, we attempted to obtain answers to both common-sense queries such as the number of wheels on a car and generic queries such as "Badr Hari" using the Wikimedia (8) API. The process involved several steps:

  1. Fetching the top three relevant articles related to the common-sense question.
  2. Determining whether to utilize the excerpt provided by the Wikimedia API or the entire Wikipedia page based on the semantic similarity score with threshold of 0.5 between the query and the content.
  3. Employing a QA model (`distilbert-base-cased-distilled-squad`) to derive the final answer to the common-sense query, with the excerpt or full Wiki page as the context.

The Wikimedia API worked well for getting relevant information for queries requiring more context to get the output. However, retrieving answers to common-sense questions from Wikipedia proved to be challenging, resulting in a significant number of Q-A-C triplets being filtered out. To simplify this process, we integrated Google's SERP API (ser), which retrieves content displayed on Google's first page in response to the same query search.

Many common-sense questions yielded direct answers in the answer box on Google's first page. For queries without an answer in the answer box, snippets from all links were combined to perform question-answering using the concatenated snippets as context.

The transition to using the SERP API was beneficial. When the SERP API did not provide an answer, we fell back on the Wikipedia search approach.

Once we had the answer to the common-sense question from NumGLUE dataset or obscure questions from Synthetic dataset, the arithmetic equations were solved using Sympy to get to the final answer. The resulting answer from the tool is then checked against the gold answer. If the two answers match, that specific instance is deemed eligible for inclusion in our dataset.

For the HotpotQnA dataset, we first conducted a WikiSearch and subsequently performed Question Answering. Following this, we evaluated the similarity of the final answer with the gold label using one of the following methods:

1. Exact Matching : We consider it a success case if answers exactly match.
2. Fuzzy Matching : We utilize the python based *fuzzywuzzy* library to validate the answers. The Fuzz Partial Ratio method aims to gauge the likeness between two strings by emphasizing the optimal matching substring. This technique proved effective in spotting matches, especially in cases where the predicted response is a subset or prefix of the gold

Table 5: Percentage Filtered data

Datasets	Synthetically Generated Examples	Filtered Examples	Percentage
Gsm8k	1800	809	44.944%
HotpotQnA	1000	521	52.100%
CommonsenseQA	200	115	57.500%
Synthetic HotpotQA	200	91	45.500%

answer.

3. Semantic Matching : We utilise the cosine similarity score between the predicted answer and the gold answer embeddings generated from MiniLM in one case and phrase-bert (15) in another case to decide if it is a successful execution or not. The threshold used was 0.5.

After filtration process, a dataset consisting of following number of samples was prepared for use in fine-tuning the model (5):

1. 115 NumGlUE samples
2. 91 Synthetic data samples
3. 521 HotpotQnA samples

We want to emphasize that our filtration method could retain 52.1% of the data after filtration, specifically for HotpotQA, for fine-tuning purposes, compared to the 15% mentioned in the COA paper (4).

We encountered several challenges during the C generation for this dataset:

1. Incorrect generation of common-sense questions.
2. Formatting issues in the equations/questions to be solved.
3. Answers being contained within C itself.
4. The LLM taking shortcuts to reach the final answer without detailing all necessary steps.

Also, it is important to mention that the dataset being generated was heavily influenced by the API tool being used for fetching the commonsense answer. This dependency on API required us to modify the instruction based approach for dataset generation.

Table 6: Ideal query style for respective APIs

API	Query Example
SERP	How many legs does a dog have?
WikiAPI	dog legs number

5 Baselines

Our baseline models, *Mistral-7B Instruct v2.0* and *Gemma-2B Instruct*, were evaluated using a standard set of parameters: `max_tokens = 500`, `top_k = 30`, `top_p = 0.95`, and `temperature = 0.7`. We assessed the models' performance by administering questions from test sets across various datasets to verify the accuracy of their responses. For rapid LLM inferencing, we employed vLLM, which incorporates efficient management of attention keys and value memory through *PagedAttention* (vll). While both models showed similar levels of performance overall, *Gemma-2B* demonstrated superior results on the *Gsm-8k* dataset, likely due to its pretraining on this particular dataset. The generally low accuracy rates reveal that current LLMs are challenged by these datasets and depend heavily on their stored memory rather than authentic understanding or analytical skills. The results can be seen in 7.

6 Your approach

Finetuned Gemma and Mistral models on 1536 datapoints of Q and C

Libraries and Frameworks Used: Our models were implemented using a suite of libraries, primarily including `transformers`, `datasets`, and `accelerate` from Hugging Face, as well as `torch`, `tqdm`, and `pandas`. For optimized training, we employed `bitsandbytes` and the `trl`'s SFTTrainer. The training code, organized in the Model Training folder, features critical configurations such as `LoraConfig` for LoRA layers and `PeftModel` for enhanced training ef-

Table 7: Performance Comparison of Models on Various Datasets

Model	Gsm8k	HotpotQA	CommonSenseQA	Synthetic HotpotQA
Gemma-2b base	35%	17%	27%	25.90%
Mistral-7b base	26%	17%	22.50%	26.80%
Gemma-2b CoA	11%	19%	18%	18%
Mistral-7b CoA	**29%**	**25%**	**24%**	20%

ficiency. A combination of standard and custom callback functions was used to manage model training and evaluations effectively.

Implementation Details: We did not use any existing implementations as our starting point, nor was the data open-sourced by any tools in the related papers. Our dataset was constructed from scratch, and the models were fine-tuned accordingly: The experiments were conducted on high-performance GPU machines, specifically using NVIDIA V100 and A100 models available through the paid version of Google Colab. The following decoding parameter were used to generate the CoA reasoning: *min tokens=50, max tokens=400, top k=30, top p=0.95, temperature=0.2* We also found the use of DataCollator very beneficial in model training. We defined it at

`###RESPONSE`

. The DataCollatorForCompletionOnlyLM trains the model on the generated prompts only

Figure 2: Training Loss Over Time

Figure 2 shows the training loss over time, indicating the model's learning progress throughout the training phase.
Following hyper-parameters common for both baseline models

- Batch size :4
- lora rank : 32
- lora alpha : 16
- learning rate : 5e-4

Gemma was finetuned for 10 epochs and mistral for 2 epochs.

Challenges and Innovations: A significant challenge in our pipeline was its reliance on the quality of synthetic data. Since the annotations were not human-generated but meticulously curated through automated processes, ensuring high data quality was imperative. Unlike the approach in the COA paper (4), which involved training different models on diverse datasets and another fine-tuned QA model specifically for extracting entities from Wikipedia content, our strategy focused on developing***a single, versatile model. This model was designed to handle multi-tool chaining autonomously***, using a generic QA framework tailored for closed-domain questions.

Results and Comparison with Baselines: Our model demonstrated some improvements over the baseline configurations. The results of these experiments are detailed in Table 7. Modifications to training prompts significantly influenced the models' outputs; more structured prompts (E) led to better-formatted responses, including accurate API calls.

Gemma: Specifically, the *Gemma-2B* model showed limited capability in generating CoA tool plans, as evidenced by its lower performance in Table 7—11% for *Gemma-2B CoA* compared to 35% for *Gemma-2B base* on the Gsm8k dataset. In contrast, the *Mistral-7B* model demonstrated enhanced learning of CoA tool planning, where the introduction of tool usage in *Mistral-7B CoA* resulted in an accuracy improvement to 29% from 26% on the same dataset, highlighting its better adaptability to task-specific requirements.

6.1 Testing

Our trained Gemma and Mistral Models will produce C.

We will then process the generated C. First we use Regex to find all the API calls. Then we process the API calls sequentially.

Processing Wiki Calls

Example of Wiki Call:

[What is the capital of France? -Wiki → x]

We search for -Wiki->tag, we then store the query: "What is the capital of France?". It is then passed on as a payload for the request for the google api. We then process the response for highlighted snippets, actual answer and related content. If the google api fails, we query the wiki api, and get the excerpt from the response. The answers are then stored in a dictionary which will be used for subsequent api calls as we are doing multi tool chaining.

Processing Q/A Calls

Example of QA Call:

[x -QA(What is the population?) → y]

We then use the output of previous tool using chaining as context for QA tool, along withe the question within (). The QA tool is basically a fine tuned distillbert. again the answer is stored in the answer dictionary.

Processing Math API Calls

Example of Math Call:

[y/2 = z] [y=400,000]

We are using the sympy libraries to solve multivariable linear equations.Here we use the answer dictionary created to supply the value for previously calculated variables.

7 Error analysis

We spent a lot of time generating synthetic data for our project, and synthetic data comes with its own drawbacks. We found a lot of errors while checking the synthetic data, some of which are summarized below. Subsequently, the trained model introduced some more errors along with what the teacher models already had, which are also summarized here.

Extracting Numbers from Dates

If the answer is 'May 29, 1958', 29, or 1958, our approach is not perfect for this. We have tried hardcoding a few things, but some cases are still missed.

Synthetic Training Data

In many instances, the teacher model uses shortcuts from W and A to create C. C contains the answers from W, and C contains the actual answer and generation of unsolvable equations.

- **Example:** Annette Bening -Wiki→ y1, at y1 -QA (When did she receive a star on the Hollywood Walk of Fame?)→ y2. The Wikipedia page did not consist of any detail about this, despite being in the HotpotQA dataset.

Wikipedia and Google Pages Not Having a Single Answer

- To extract the birthdate of Henry Miller's wife, this person has many wives, and the question was vague, leading to different answers being extracted.
- **Example:** The Dutch-Belgian television series that "House of Anubis" was based on first aired in how many years after 2000? – This show had multiple air dates in different countries.
- Some Wikipedia context given in HotpotQA did not make any sense and sometimes was not relevant.

Answering Yes/No Questions

Our QA module is extractive question answering and generates a snippet from the context. For questions where an abstractive answer is expected, our approach did not work out. One solution for this is to use our fine-tuned model and prompt it to generate an answer as a 7B model is smart. We were not able to integrate this.

Model-Specific Issues

The final fine-tuned model still gets confused about which tool to use for some examples. In math questions, even if the data is present in the question, it still goes and searches it on Wikipedia.

- **Examples:**
  - There are 10 cars and 2 bikes. Find out the number of wheels in that parking lot. Model generates: [Find the Number of cars -Wiki→y1].
  - First search [number of hours Charlotte spends on phone -Wiki→ y1]. Half of

y1 is [y1/2 = y2] spent on social media every day. In a week, there are [7*24 = y3] hours. So, Charlotte spends [y2*y3 = y4] hours on social media in a week. The answer is y4.

- Hallucinated content in between tool generations.
  - **Example:** The car has [Number of wheels in car -Wiki→y1]. It has 4 wheels. ... 4 should have been extracted from the QA module instead of it getting generated on its own.

The baseline models showed poor performance on knowledge based data, specifically Hotpot dataset. For eg:
Who was considered the best kickboxer in the world and has been involved in controversies relating to his "unsportsmanlike conducts" in the sport.
Such obscure and specific questions aren't handled well by both the baseline models. Integrating API calls improves the performance significantly. More importantly, our fine-tuned Mistral outperformed the baseline on all non-synthetic data, including pure arithmetic and commonsense+ arithematic queries.

8 Contributions of group members

- Abhishek: Created artificial data for each activity while working on arithmetic and the Wikipedia multitool prompt development for HotpotQnA. He was instrumental in generating the baseline data for Mistral 7B and Gemma 2B. Abhishek also contributed a great deal of testing and documentation to make sure the project's results were accurate and reliable.

- Anushka: Oversaw the whole math tool implementation, including data generation, prompting, and SymPy toolkit integration. She was also in charge of the whole implementation of Wikipedia tool, which included data production, prompting, and connecting the QA models and Wiki search API. Anushka created data filtering Module for every scenario and customised tool parsing and execution logic for both single and chain multitool operations. She oversaw the SFT training code, final training, model evaluations, testing the generated apis, documentation, and model training for Gemma 2B and Mistral 7B.

- Yash: Created synthetic data for the jobs and concentrated on creating multitool prompts with logical sense. He made a substantial contribution to the testing and documentation procedures, guaranteeing excellent results and smooth module integration. The success of the project was greatly attributed to his work in creating and improving the multitool prompt techniques.

- Arushi: Examined several datasets needing multitool execution, built scoring measures such fuzzy match, semantic match, and exact match, and successfully implemented WikiSearch utilising the WikiMedia API. She also created a words-to-number conversion code, documented her work, and connected the Wikipedia search with the SERP API.

9 Conclusion

Through this project, we were able to implement all aspects of API Based Text generation task. Since we were referencing a few published papers, we came across challenges in replicating some aspects at a significantly smaller scale. Obtaining satisfactory results for C generation proved surprisingly difficult and we had to regularly rehash the dataset. Our baseline models weren't able to generate C in proper format, especially when it came to writing API calls demonstrated in the training data. For future work, it would be interesting to see the model performance on a state of the art LLM generated dataset, and with more parameters. This would likely solve the bottleneck issue of valid C generation, since we had near to perfect accuracy score on syntactically valid C.

10 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, Chat-GPT was used.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.

  In general, we used the following prompts in to grammar check and improve the flow of our content.

  - Improve this paragraph: content
  - Write the following content in latex point format: content
  - Make the following content better : content

- **Free response:** The AI was helpful in providing initial drafts and ideas, which I could then refine and edit to suit my needs. In many cases, it directly gave me a good output, though I occasionally needed to tweak the responses to better fit the context. The AI's output was mostly relevant, with only a few instances of irrelevance or inaccuracy. I used the AI to generate new text, check my own ideas, and rewrite existing text.

  Since the project required extensive prompting on many LLMs, some of the important prompts used are mentioned in the Appendix for data generation.

References

[ser] SerpApi: Google Search API — serpapi.com. https://serpapi.com/. [Accessed 18-05-2024].

[vll] vllm quickstart. https://docs.vllm.ai/en/latest/getting_started/quickstart.html. Accessed: 2024-05-18.

[3] Du, Y., Wei, F., and Zhang, H. (2024). Anytool: Self-reflective, hierarchical agents for large-scale api calls.

[4] Gao, S., Dwivedi-Yu, J., Yu, P., Tan, X. E., Pasunuru, R., Golovneva, O., Sinha, K., Celikyilmaz, A., Bosselut, A., and Wang, T. (2024). Efficient tool use with chain-of-abstraction reasoning.

[5] Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. (2024). Critic: Large language models can self-correct with tool-interactive critiquing.

[6] Hao, S., Liu, T., Wang, Z., and Hu, Z. (2024). Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings.

[7] Inaba, T., Kiyomaru, H., Cheng, F., and Kurohashi, S. (2023). MultiTool-CoT: GPT-3 can use multiple external tools with chain of thought prompting. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1522–1532, Toronto, Canada. Association for Computational Linguistics.

[8] MediaWiki (2024). Api:main page — mediawiki,. [Online; accessed 18-May-2024].

[9] Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., and Scopatz, A. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

[10] Miao, S.-y., Liang, C.-C., and Su, K.-Y. (2020). A diverse corpus for evaluating and developing English math word problem solvers. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

[11] Parisi, A., Zhao, Y., and Fiedel, N. (2022). Talm: Tool augmented language models.

[12] Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. (2023). Gorilla: Large language model connected with massive apis.

[13] Petroni, F., Piktus, A., Fan, A., Lewis, P., Yazdani, M., De Cao, N., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J., Plachouras, V., Rocktäschel, T., and Riedel, S. (2021). KILT: a benchmark for knowledge intensive language tasks. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

[14] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools.

[15] Wang, S., Thompson, L., and Iyyer, M. (2021). Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. In *Empirical Methods in Natural Language Processing*.

[16] Yang, R., Song, L., Li, Y., Zhao, S., Ge, Y., Li, X., and Shan, Y. (2023). Gpt4tools: Teaching large language model to use tools via self-instruction.

Appendix

A Prompts for GSM8K:

Examples to generate Chain of Equations(C) given Q and A are:

Example 1
Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees will the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21-15=6. The answer is 6.
C: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been [21 - 15 = y1]. The answer is y1.
Example 2
Q: The flowers cost $9, the clay pot costs $20 more than the flower, and the bag of soil costs $2 less than the flower. How much does it cost to plant the flowers?
A: The clay pot costs $20 + $9 = $29. The bag of soil costs $9 - $2 = $7. The cost to plant the flowers is $9 + $29 + $7 = $45. The answer is 45.
C: The clay pot costs [20 + 9 = y1]. The bag of soil costs [9 - 2 = y2]. The cost to plant the flowers is [9 + y1 + y2 = y3]. The answer is y3.
Example 3
Q: From March to August, Sam made $460 doing 23 hours of yard work. However, from September to February, Sam was only able to work for 8 hours. If Sam is saving up to buy a video game console that costs $600 and has already spent $340 to fix his car, how many more hours does he need to work before he can buy the video game console?
A: Sam makes $460 / 23 hrs = $20/hr. From September to February, Sam made 8hrs x $20/hr = $160. From March to February, Sam made a total of $460 + $160 = $620. After fixing his car, he was left with $620 - $340 = $280. Sam needs another $600 - $280 = $320. Sam needs to work another $320 / $20/hr = 16 hours. The answer is 16.
C: Sam makes [460 / 23 = y1] dollars per hour. From September to February, Sam made [8 * y1 = y2] dollars. From March to February, Sam made a total of [460 + y2 = y3] dollars. After fixing his car, he was left with [y3 - 340 = y4]. Sam needs another [600 - y4 = y5] dollars. Sam needs to work another [y5 / y1 = y6] hours. The answer is y6.

Example 4
Q: There were nine computers in the server room. Five more computers were installed each day, from Monday to Thursday. How many computers are now in the server room?
A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29.
C: There were originally 9 computers. For each of 4 days, 5 more computers were added. So [5 * 4 = y1] computers were added. [9 + y1 = y2]. The answer is y2.

Now generate C for the test examples below. Please Write ”DONE” when you have completed generating C for all of the examples below.
Test Example 1
Q: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
A: Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May. The answer is 72
C:
.
.
.

Respond only with C for the test examples. Use this format >>Test Example >numberC : >generated C
Chain of Equations C for the above test examples are:

B Prompts for Wiki:

You are an agent that converts Questions, Answers, and corresponding Wikipedia Knowledge to a Chain of Abstractions. The intuition is to get to the Answer using the titles searched in the Wikipedia knowledge.

Rules to generate C:
1. You have 2 tools, Wiki and QA. Use **BOTH tools** to derive C. Tool Explanations are:
Wiki Tool to get relevant articles from Wikipedia. Format: [search query -Wiki->search query output]
QA tool to get the focused answer from the Wikipedia articles. Format: [input context -QA(question)->output]
2. W is formatted to put the title before '>' and content after '>' and separate articles using '—'.
3. Use the outputs from Wiki tool as input context for QA tool. The final answer is always output of QA tool. **This is crucial**
4. You can use page titles but cannot use page content information given in W to form Wiki search Queries.
5. You cannot use answer (A) in the tool queries directly.
6. You can only use first wikipedia title in the tool queries directly and cannot use any subsequent titles directly in tool queries
7. Utilize the Chain of Thought process from the Wikipedia Knowledge given. Learn what titles need to be searched in the Wiki tool and asked in the QA tool to get to the final answer. Conclude by ** The answer is variable$_name$>. * *

Your task to generate Abstractions(C) for the given Question (Q) using both the Wiki tool and the QA tool using the rules.

Examples to generate Chain of Abstraction(C) given Q, A and W are:
EXAMPLES
Example 1
Q: Fritz von Brodowski was killed during what global war that lasted from 1939 to 1945?
A: The answer is World War II.
W: Fritz von Brodowski >Friedrich Wilhelm Konrad von Brodowski was controversially killed while in French custody during World War II.
C: Find the [war in which Fritz von Brodowski was killed -Wiki->y1]. Fritz von Brodowski was killed in [y1 -QA(Fritz von Brodowski was killed in which war?)->y2]. The answer is y2.

Example 2
Q: Which tennis player won more Grand Slam titles, Henri Leconte or Jonathan Stark?
A: The answer is Jonathan Stark.
W: Henri Leconte >He won the French Open men's doubles title in 1984. — Jonathan Stark (tennis) >During his career he won two Grand Slam doubles titles.
C: First Search [Henri Leconte Grand Slam titles -Wiki->y1]. Then Search [Jonathan Stark Grand Slam titles -Wiki->y2]. [y1+y2 -QA(Which tennis player won more Grand Slam titles, Henri Leconte or Jonathan Stark?)->y3]. The answer is y3.

Example 3
Q: The director of the romantic comedy "Big Stone Gap" is based out of which part of New York city?
A: The answer is Greenwich Village.
W: Big Stone Gap (film) >Big Stone Gap is a 2014 American romantic comedy film directed by Adriana Trigiani. — Adriana Trigiani >Adriana Trigiani is an Italian American film director based in Greenwich Village.
C: First search the [Big Stone Gap director -Wiki->y1]. The name of this film's director is [y1 -QA(Who is the director of Big Stone Gap?)->y2]. Then determine [y2 is based out of which city -Wiki->y3]. The city is [y3 -QA(Where is y2 based out of?)->y4]. Answer is y4.

Example 4
Q: Are Randal Kleiser and Kyle Schickner of the same nationality?
A: The answer is yes.
W: Randal Kleiser >John Randal Kleiser (born July 20, 1946) is an American film director and producer. — Kyle Schickner >Kyle Schickner is an American film producer, writer, director, actor.
C: First find out the [Randal Kleiser nationality -Wiki->y1]. Then figure out the [Kyle Schickner nationality -Wiki->y2]. Then check [y1+y2 -QA(Are Randal Kleiser and Kyle Schickner of the same nationality?)->y3]. the answer is y3.

Example 5
Q: Extras was created, written, and directed by Ricky Dene Gervais, an English comedian, actor, writer, producer, director, singer, and musician, born on which date?
A: The answer is 25 June 1961.
W: Ricky Gervais >Ricky Dene Gervais (born 25 June 1961) is an English comedian, actor, writer, producer, director, singer, and musician.
C: Search [when Ricky Dene Gervais was born -Wiki->y1]. Then determine [y1 -QA(When Ricky Dene Gervais was born?)->y2]. The answer is y2.

Example 6
Q: Sameera Perera is a cricketer from what island country located southeast of the Republic of India and northeast of the Maldives? A: The answer is Sri Lanka.
W: Sameera Perera >Sameera Perera (born 20 August 1988) is a Sri Lankan cricketer.
C: Search [Sameera Perera -Wiki->y1]. Then find [y1 -QA(Where is Sameera Perera from?)->y2]. The answer is y2.

Example 7
Q: What screenwriter with credits for “Evolution” co-wrote a film starring Nicolas Cage and Téa Leoni?
A: The answer is David Weissman.
W: The Family Man >The Family Man is a 2000 American romantic comedy-drama film starring Nicolas Cage and Téa Leoni. — David Weissman >His film credits include “The Family Man” (2000), “Evolution” (2001), and “When in Rome” (2010).
C: First figure out the [film of Nicolas Cage and Téa Leoni -Wiki->y1]. The name of this film is [y1 -QA(In which film did Nicolas Cage and Téa Leoni star together?)->y2]. Then find out [who wrote y2 with credits for “Evolution” -Wiki->y3]. The name is screenwriter is [y3 -QA(Who wrote y2 with credits for Evolution?)->y4]. The answer is y4.
Example 8
Q: Ralph Hefferline was a psychology professor at a university that is located in what city?
A: The answer is New York City.
W: Ralph Hefferline >Ralph Franklin Hefferline was a psychology professor at Columbia University. — Columbia University >Columbia University is a private Ivy League research university in Upper Manhattan, New York City.
C: First identify the [university of psychology professor Ralph Hefferline -Wiki->y1]. The university of this professor is [y1 -QA(Ralph Hefferlin was a psychology professor at which university?)->y2]. Then figure out [y2 in which city -Wiki->y3]. The name is city is [y3 -QA(y2 is in which city?)->y4]. The answer is y4.

TASK
Now generate C for the test examples below. Please Write ’DONE’ when you have completed generating C for all of the examples below. Respond in this format only:
Test Example example number>C:

Test Example 1
Q: Which genus has more species, Fir or Chelone?
A: Firs
W: Fir >Firs (Abies) are a genus of 48–56 species of evergreen coniferous trees in the family Pinaceae. — Chelone (plant) >Chelone is a genus of four species of perennial herbaceous plants native to eastern North America.
.
.
.
RESPONSE

C Prompt for NumGLUE

Example Data:
”question”: ”An garage has 14 bike wheels. How many bikes he can assemble using 14 wheels?”,
”answer”: 7,
”C”: ”First search [number of wheels a bike has -Wiki->y1]. The number of wheels on a single bike are [y1 -QA(How many wheels does a bike have?)->y2].So, [14/y2 = y3] bikes will be assembled. The answer is y3.”,

”question”: ”Leonardo wants to buy a chocolate. The cost of chocolate is 5 dollars and Leonardo has only 4 dollars in his pocket. He borrowed 59 cents from his friend. How many more cents Leonardo needs now to purchase a chocolate?”,
”answer”: 41,
”C”: ”First search [cents in a dollar -Wiki->y1]. The number of cents in a dollar are[y1 -QA(How many cents in a dollar?)->y2]. So, choclate costs [5*y2 = y3] cents. He has[4*y2 +59 = y4] cents. Leonardo needs [y3-y4 = y5] cents. The answer is y5.”,

”question”: ”A 6 minutes scene from an hour long movie was cut due to copyright issues. What is the final length of the movie (in minutes)?”,
”answer”: 54,
”C”: ”First search [number of minutes in an hour -Wiki->y1]. The number of minutes in an hour are[y1 -QA(How many minutes in an hour?)->y2].
So, The movie was of [1*y2 = y3] minutes. Now the total movie length is [y3 - 6 = y4] minutes. The answer is y4.”,

”question”: ”For an entire month of June, Alice purchased 4 burgers each day. If each burger costs 13 dollars. How much did Alice spend on burgers in June”,
”answer”: 1560,
”C”: ”First search [number of days in June -Wiki->y1]. The number of days in June are[y1 -QA(How many days in June?) ->y2]. So, The number of burgers Alice purchased in june was [4*y2 = y3] minutes. The total amount of money alice spent is [y3*13 = y4] dollars. The answer is y4.”,

”question”: ”Hazel runs at a constant speed of 59 m/s. How many kms can he cover in 5 minutes”,
”answer”: ”17.7”,
”C”: ”First search [number of seconds in a minute -Wiki->y1]. The number of seconds in a minute are[y1 -QA(How many seconds in a minute?)->y2]. Then search [Kilometer to metre -Wiki->y3]. A Kilometer has [y3 -QA(A kilometer has how many metres?)->y4]. Hazel has [y2*5 = y5] seconds. She can run [y5*59 = y6] meters. So, hazel can cover [y5/y4 = y6] kilometers. So, The number of burgers Alice purchased in june was [4*y2 = y3] minutes. The total amount of money alice spent is [y3*13 = y4] dollars. The answer is y4.”,

TASKS:
1. You will be give data in the following format:
”question”:”question data”,
”answer”:”answer data”
2. You should generate C (as mentioned in the example) for the given question and answer.
3. While generating C, be clear to mention ”-Wiki->” , ”-QA()->” and ”=” tags wherever necessary.
4. These tags should be in following format: [sampledata -tag output]
5. C shouldn’ contain any data from the answer.
6. C should be in such a way that it leads to answer.
7. last line in c should be of the fomate: the answer is ...

What output am i expecting you to give:

For the Following input format:
”question”: ”Anthony spends 6 hours at work, 6 hours on other daily chores and sleeps for the remaining time of the day. For how many hours does Anthony sleep in a day?”,
”answer”: 12,
I am expecting you to give an update version with C, with the appropriate format and data.
Note: Keep the formatting same as described below:
Eg:
”question”: ”Anthony spends 6 hours at work, 6 hours on other daily chores and sleeps for the remaining time of the day. For how many hours does Anthony sleep in a day?”,
”answer”: 12,
”C”:”First search [number of hours in a day -Wiki->y1]. A day has [y1 -QA(How many hours in a day?)->y2] hours. Anthony spends 6 hours at work. That leaves us with [y2-6 = y3]hours. Then he spends 6 hours in chores, that leaves us with [y3-6 = y4]hours. The answer is y4.”

MY DATA section has a number of comma seperated examples of Question answer Pairs.
You are supposed generate C for every question answer example in MY DATA section.

MY DATA

D Prompt for Multi-tool Synthetic

Before Data:

This is the data in the dataset that i have:
Example 1 :
Q: The Dutch-Belgian television series that "House of Anubis" was based on first aired in what year?
A: 2006
W: House of Anubis >House of Anubis is a mystery television series developed for Nickelodeon based on the Dutch-Belgian television series "Het Huis Anubis". — Het Huis Anubis >It first aired in September 2006 and the last episode was broadcast on December 4, 2009.
Example 2 :
Q: What is the length of the track where the 2013 Liqui Moly Bathurst 12 Hour was staged?
A: 6.213 km long
W: 2013 Liqui Moly Bathurst 12 Hour >The 2013 Liqui Moly Bathurst 12 Hour was an endurance race for a variety of GT and touring car classes, including: GT3 cars, GT4 cars, Group 3E Series Production Cars and Dubai 24 Hour cars. — 2013 Liqui Moly Bathurst 12 Hour >The event, which was staged at the Mount Panorama Circuit, near Bathurst, in New South Wales, Australia on 10 February 2013, was the eleventh running of the Bathurst 12 Hour. — Mount Panorama Circuit >Mount Panorama Circuit is a motor racing track located in Bathurst, New South Wales, Australia. — Mount Panorama Circuit >The 6.213 km long track is technically a street circuit, and is a public road, with normal speed restrictions, when no racing events are being run, and there are many residences which can only be accessed from the circuit.
Example 3 :
Q: The 1988 American comedy film, The Great Outdoors, starred a four-time Academy Award nominee, who received a star on the Hollywood Walk of Fame in what year?
A: 2006
W: The Great Outdoors (film) >The Great Outdoors is a 1988 American comedy film directed by Howard Deutch, and written and produced by John Hughes. — The Great Outdoors (film) >It stars Dan Aykroyd, John Candy, Stephanie Faracy and Annette Bening in her film debut. — Annette Bening >Annette Carol Bening (born May 29, 1958) is an American actress. — Annette Bening >She is a four-time Academy Award nominee; for "The Grifters" (1990), "American Beauty" (1999), "Being Julia" (2004) and "The Kids Are All Right" (2010). — Annette Bening >In 2006, she received a star on the Hollywood Walk of Fame.
Example 4 :
Q: Dua Lipa, an English singer, songwriter and model, the album spawned the number-one single "New Rules" is a song by English singer Dua Lipa from her eponymous debut studio album, released in what year?
A: 2017
W: Dua Lipa >Her self-titled debut studio album was released on 2 June 2017. — New Rules (song) >"New Rules" is a song by English singer Dua Lipa from her eponymous debut studio album (2017).
Example 5 :
Q: How old is the female main protagonist of Catching Fire?
A: 16-year-old
W: Catching Fire >Catching Fire is a 2009 science fiction young adult novel by the American novelist Suzanne Collins, the second book in "The Hunger Games trilogy". — The Hunger Games (novel) >It is written in the voice of 16-year-old Katniss Everdeen, who lives in the future, post-apocalyptic nation of Panem in North America.
Example 6 :
Q: In what year was the creator of the current arrangement of the "Simpson's Theme" born?
A: March 28, 1941
W: The Simpsons Theme >"The Simpsons" Theme", also referred to as "The Simpsons" Main Title Theme" in album releases, is the theme music of the animated television series "The Simpsons". — The Simpsons Theme >The theme, as used for the opening sequence, was re-arranged during season 2, and the current arrangement by Alf Clausen was introduced at the beginning of the third season. — Alf Clausen >Alf Heiberg Clausen (born March 28, 1941) is an American film and television composer.
Example 7 :
Q: The American Pre-Code comedy film featuring an American actress, dancer, and singer, widely known for performing in films and RKO's musical films, was released in what year?
A: 1932
W: Hat Check Girl >Hat Check Girl is a 1932 American Pre-Code comedy film directed by Sidney Lanfield and written by Barry Conners and Philip Klein. — Hat Check Girl >The film stars Sally Eilers, Ben Lyon, Ginger Rogers and Monroe Owsley. — Ginger Rogers >Ginger Rogers (born Virginia Katherine McMath; July 16, 1911 – April 25, 1995) was an American actress, dancer, and singer, widely known for performing in films and RKO's musical films, partnered with Fred Astaire.
Example 8 :
Q: What year was the winner of the 2016 Marrakesh ePrix born?
A: 1988
W: 2016 Marrakesh ePrix >The 33-lap race was won by e.Dams-Renault driver Sébastien Buemi, who started from the seventh position. — Sébastien Buemi >Sébastien Olivier Buemi (born 31 October 1988) is a Swiss professional racing driver, who formerly competed for Scuderia Toro Rosso in Formula One.

After data:

1.

Q: The Dutch-Belgian television series that "House of Anubis" was based on first aired in how many years after 2000?
A: 6
W: House of Anubis >House of Anubis is a mystery television series developed for Nickelodeon based on the Dutch-Belgian television series "Het Huis Anubis". — Het Huis Anubis >It first aired in September 2006 and the last episode was broadcast on December 4, 2009.
C: First search for [Het Huis Anubis first aired -Wiki->y1]. First episode released in [y1 -QA(Which year was the fist episode aired ?)->y2]. Number of years it aired after 2000 [y2 - 2000 = y3]. The answer is y3.

2.
Q: What would be the length of the track where the 2013 Liqui Moly Bathurst 12 Hour was staged, if it would have been 4km longer?
A: 10.213
W: 2013 Liqui Moly Bathurst 12 Hour >The 2013 Liqui Moly Bathurst 12 Hour was an endurance race for a variety of GT and touring car classes, including: GT3 cars, GT4 cars, Group 3E Series Production Cars and Dubai 24 Hour cars. — 2013 Liqui Moly Bathurst 12 Hour >The event, which was staged at the Mount Panorama Circuit, near Bathurst, in New South Wales, Australia on 10 February 2013, was the eleventh running of the Bathurst 12 Hour. — Mount Panorama Circuit >Mount Panorama Circuit is a motor racing track located in Bathurst, New South Wales, Australia. — Mount Panorama Circuit >The 6.213 km long track is technically a street circuit, and is a public road, with normal speed restrictions, when no racing events are being run, and there are many residences which can only be accessed from the circuit. C: First search [Mount Panorama Circuit -Wiki->y1]. Length of circuit is [y1 -QA(what is the length of Mount Panorama Circuit ?)->y2]. Length after adding 4km will be [4 + y2 = y3]. The answer is y3.

3.
Q: The 1988 American comedy film, The Great Outdoors, starred a four-time Academy Award nominee, who received a star on the Hollywood Walk of Fame in how many years before 2010?
A: 4
W: The Great Outdoors (film) >The Great Outdoors is a 1988 American comedy film directed by Howard Deutch, and written and produced by John Hughes. — The Great Outdoors (film) >It stars Dan Aykroyd, John Candy, Stephanie Faracy and Annette Bening in her film debut. — Annette Bening >Annette Carol Bening (born May 29, 1958) is an American actress. — Annette Bening >She is a four-time Academy Award nominee; for "The Grifters" (1990), "American Beauty" (1999), "Being Julia" (2004) and "The Kids Are All Right" (2010). — Annette Bening >In 2006, she received a star on the Hollywood Walk of Fame.
C: First Search [Annette Bening -Wiki->y1]. She received star on the Hollywood Walk of Fame on [y1 -QA(When did she recieved star on Hollywood Walk of Fame ?)->y2]. Number of years before 2010 [2010 - y2 = y3]. The answer is y3.

4.
Q: Dua Lipa, an English singer, songwriter and model, the album spawned the number-one single "New Rules" is a song by English singer Dua Lipa from her eponymous debut studio album, released in how many years after 2000?
A: 17
W: Dua Lipa >Her self-titled debut studio album was released on 2 June 2017. — New Rules (song) >"New Rules" is a song by English singer Dua Lipa from her eponymous debut studio album (2017).
C: First Search [Dua Lipa New Rules release -Wiki->y1]. New rules was released in [y1 -QA(In which year was New Rules song released ?)->y2]. Number of years from 2000 after its released is [y2 - 2000 = y3]. The answer is y3.

5.
Q: How old would be the female main protagonist of Catching Fire, if she was born 6 years before?
A: 22
W: Catching Fire >Catching Fire is a 2009 science fiction young adult novel by the American novelist Suzanne Collins, the second book in "The Hunger Games trilogy". — The Hunger Games (novel) >It is written in the voice of 16-year-old Katniss Everdeen, who lives in the future, post-apocalyptic nation of Panem in North America.
C: First Search for [Catching Fire Suzanne Collins -Wiki->y1]. Search for her age at that time [y1 -QA(What was the age of Katniss Everdeen at that time ?)->y2]. She would how many years old is she was born 6 years before [y2 + 6 = y3]. The answer is y3.

6.
Q: How many years before 1947 was the creator of the current arrangement of the "Simpson's Theme" born?
A: 6
W: The Simpsons Theme >"The Simpsons" Theme", also referred to as "The Simpsons" Main Title Theme" in album releases, is the theme music of the animated television series "The Simpsons". — The Simpsons Theme >The theme, as used for the opening sequence, was re-arranged during season 2, and the current arrangement by Alf Clausen was introduced at the beginning of the third season. — Alf Clausen >Alf Heiberg Clausen (born March 28, 1941) is an American film and television composer.
C: First find [Simpson's Theme creator -Wiki->y1]. Now search for when was he born [y1 -QA(When was Alf Heiberg Clausen born ?)->y2]. Now find how many years before 1947 he was born [1947 - y2 = y3]. The final answer is y3.

7.
Q: The American Pre-Code comedy film featuring an American actress, dancer, and singer, widely known for performing in films and RKO's musical films, was released in how many years after 1925?
A: 7
W: Hat Check Girl >Hat Check Girl is a 1932 American Pre-Code comedy film directed by Sidney Lanfield and written

by Barry Conners and Philip Klein. — Hat Check Girl >The film stars Sally Eilers, Ben Lyon, Ginger Rogers and Monroe Owsley. — Ginger Rogers >Ginger Rogers (born Virginia Katherine McMath; July 16, 1911 – April 25, 1995) was an American actress, dancer, and singer, widely known for performing in films and RKO's musical films, partnered with Fred Astaire.
C: First Search for [American Pre-Code comedy RKO musical film -Wiki->y1]. Now search for year it was released [y1 -QA(When was Hat Check Girl Released ?)->y2]. Now find number of years after 1925 it was released [y2 - 1925 = y3]. The answer is y3.

8.
Q: How many years after 1950 was the winner of the 2016 Marrakesh ePrix born?
A: 38
W: 2016 Marrakesh ePrix >The 33-lap race was won by e.Dams-Renault driver Sébastien Buemi, who started from the seventh position. — Sébastien Buemi >Sébastien Olivier Buemi (born 31 October 1988) is a Swiss professional racing driver, who formerly competed for Scuderia Toro Rosso in Formula One.
C: First search for [2016 Marrakesh ePrix winner -Wiki->y1]. Now search for when he was born [y1 -QA(When was Sébastien Buemi born ?)->y2]. Now search for how many years after 1950 he was born so subtract [y2 - 1950 = y3]. The answer is y3.

For this task you have to work on "My data" Given at the End of the prompt a set of 10 rows on following tasks:
1] Change the question (Q) such that we also include a numerical calculation to the answer. While change make changes like multply, divide, add, subtract in clude the math element.
2] Change the answer (A) according to the question. Remember answer should only be in integer.Eg 4,5,6,989,4000, Do not even include any string in the A. dont even include the metrics like 5km , 5m, 5 month. I want just the number. I want the answer in datatype integer!!!
3] then by referring all of these Q(question), A(answer) and W(wikipedia data) genereate C (Chain of Abstarction) for all the My Data QAW pairs. Make sure you generate C in the same format as in the given above examples.

Format for api:

Wikipedia "[keywords -Wiki->variable]" , Here use keywords using which we can search articles on wikipedia and include all important keywords necessary
QA "[variable -QA(question)->variable]"
Mathematical "[polynomial expression ending with = final variable]"
And Keep it in order - Wikipedia, QA then mathematical.

TASKS:

1. You will be give data in the following format:

"Q":"question data",
"A":"answer data"
"W":"Wikipedia Extracts"

2. You should alter Q and A such that include a numerical calculation to the question and recieve a numerical A value as done in the after data from the before data.
3. You should generate C (as mentioned in the example) for My Data from data Q, A and W.
4. Ater the Q af if the answer is some number, it asks to do some random mathematical calculations over that number. and accordingly adjust A.
5. While generating C, be clear to mention "-Wiki->" , "-QA()->" and "=" tags wherever necessary.
6. These tags should be in following format: [sampledata -tag output]
7. Change Q as it is donw in the before to after data, where an extra mathematical calculation is done to get an extra polynomial expression in C inside [].
8. A should only contain an integer value, no string values should be included
9. C shouldn' contain any data from the answer.
10. C should be in such a way that it leads to answer.
11. Here Wiki Api should have keywords to search articles with, QA api has question which lead to what value we need from the article

What output am i expecting you to give:

For the Following input format:
Example 5 :
Q: How old is the female main protagonist of Catching Fire?
A: 16-year-old
I am expecting you to give an update version of Q , A with C, with the appropriate format and data.
Note: Keep the formatting same as described below:
Q: How old would be the female main protagonist of Catching Fire, if she was born 6 years before?
A: 22
W: Catching Fire >Catching Fire is a 2009 science fiction young adult novel by the American novelist Suzanne Collins, the second book in "The Hunger Games trilogy". — The Hunger Games (novel) >It is written in the voice of 16-year-old Katniss Everdeen, who lives in the future, post-apocalyptic nation of Panem in North America.

C: First Search for [Catching Fire Suzanne Collins -Wiki->y1]. Search for her age at that time [y1 -QA(What was the age of Katniss Everdeen at that time ?)->y2]. She would how many years old is she was born 6 years before [y2 + 6 = y3]. The answer is y3.

So return Q, A, W and C.
MY Data section has a number of examples of Question answer Pairs. You are supposed to generate C for every question answer example in MY Data section.

Now follow the instructions for the My Data and Change Q and A accordingly and generate C for it and keep W as it is.

Improvements over your last results:
Do not include the final answer in C, We are only defining how are we generating C using only variables, we just want the polynomial expression.
Do not reprint my give Data
Do not refer to current date as it cannot be found

My Data:

Example 11 :
Q: According to the 2006 census what is the population of the city in which James Iroha Uchechukwu was born ?
A: 722,664
W: Uche James-Iroha >He was born in 1972 in Enugu. — Enugu >The city had a population of 722,664 according to the 2006 Nigerian census.
.
.
.

E Prompt for Training

INSTRUCTION
Your task is to generate a chain of abstractions (C) for the given question (Q) using the available tools: Wiki, QA, and Mathematical. You can use a single tool or a combination of tools to derive the answer (C). Follow the rules and formats provided for each tool:
Tools:*
1. *Wiki Tool:* Retrieves relevant articles from Wikipedia. * *Format:* [search query -Wiki->search query output]
2. *QA Tool:* Extracts focused answers from Wikipedia articles. * *Format:* [input context -QA(question)->output]
3. *Mathematical Tool:* Solves mathematical computations based on information returned from the QA tool. * *Format:* [polynomial expression] (e.g., [y1 + 20 = y2])

See examples below on how to decide which tools to use and their usage to generate C.
EXAMPLES

Example 1 : Only Math tool used
Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees will the grove workers plant today?
C: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been [21 - 15 = y1]. The answer is y1.

Example 2 : Wiki tool and QA tool used
Q: Fritz von Brodowski was killed during what global war that lasted from 1939 to 1945?
C: Find the [war in which Fritz von Brodowski was killed -Wiki->y1]. Fritz von Brodowski was killed in [y1 -QA(Fritz von Brodowski was killed in which war?)->y2]. The answer is y2.

Example 3 : Wiki tool ,QA tool and Math tool used
Q: What would be the length of the track where the 2013 Liqui Moly Bathurst 12 Hour was staged, if it would have been 4km longer?
C: First search for [Mount Panorama Circuit -Wiki->y1]. Length of circuit is [y1 -QA(what is the length of Mount Panorama Circuit ?)->y2]. Length after adding 4km will be [4 + y2 = y3]. The answer is y3.

Now Generate C for the following Q. Respond in following format
C: of abstractions for Q>

QUESTION
Q: prompt$_q$

RESPONSE
prompt$_t ool$