Summary of the paper 'Chain-of-Thought Prompting Elicits Reasoning in Large Language Models'

Chatbots that use large language models (LLMs) reply to the query based on input prompt given post training and the data that has been used during training. Replies generally do not include how they were able to deduce the answer.

This paper presents an idea called 'Chain of thought prompting', which helps LLM chatbots to output their intermediate reasoning steps along with the final answer. These LLMs are trained using examples of chain of thought demonstrations (exemplars) and performance improvement was accomplished for 1.Arithmetic 2. Commonsense and 3.Symbolic reasoning tasks.

When PaLM 540B LLM model was trained with just 8 exemplars, it performed better than 'fine-tuned GPT-3 with a verifier' in solving GSM8K math problems.

Introduction

Scaling up LLMs has shown performance improvement and sample efficiency (learning from few examples) but it does not help when it comes to arithmetic, common sense and symbolic reasoning tasks.

This paper presents a method to unlock reasoning ability of LLMs and this method is motivated by two ideas.

1. Generating natural language reasoning rationale helps in arithmetic reasoning. Previous research work (that is the basis for the current research paper) has helped models to gain the ability to generate reasoning steps in natural language. This was possible by training model from scratch or fine-tuning an already trained model using natural language rationale along with training the model using symbolic reasoning. (Symbolic reasoning uses formal languages that include predicate logic but not natural language)

Example of predicate logic:

a. P(x) be the predicate "x is a human."
b. The statement ∀x P(x) means "For all x, x is a human."
c. "All humans are mortal" can be expressed as ∀x (P(x)→Q(x)), Q(x) is "x is mortal.".

2. In-context few-shot learning via prompting: It means that a pre-trained LLM model is not fine-tuned with a supervised dataset .Rather, the input and output data samples/exemplars are used in prompts. These data samples do not change the model parameters but help model to predict an output for the given input in prompt

Example of a prompt:

Translate to French:
a. Hello → Bonjour
b. How are you? → Comment ça va?
c. I love programming →

The above 2 methods have limitations:

1. Normal machine learning uses input and output pairs. Rationale augmented training is a method where training is done for a model from scratch or a model is fine-tuned using set of rationales. This is complicated since generating such rationale(set of reasoning steps) is costly

2. Models trained by few shot learning methods show poor reasoning ability even though the scale of model is increased

The current paper tries to gain strengths and overcome these two limitations from these two methods. It augments few shot learning method by adding one more parameter called 'chain of thought' in the input
1. Input, chain of thought, Output

Chain of thought is intermediate language reasoning steps that help to achieve the final answer.

The paper shows evaluations proving that 'chain of thought prompting' outperforms 'standard prompting'

Without losing the ability to generalize, LLMs can learn patterns using only a few exemplars with 'prompting only' approach
This method is important since it does not use a large dataset and helps to learn patterns that are typically hidden within a large input output dataset

Chain of thought prompting:

When solving a complicated math problem, it is easier to solve by decomposing it into smaller problems and solve each of them. Chain of thought helps in such a decomposition.

The goal of this paper is to give the LLMs the ability of generating chain of thought.

This paper shows that when sufficiently large language models are given chain of thought exemplars as input as part of few shot prompting, then they can generate chain of thought.

Properties of Chain of thought:

1. It decomposes a problem into intermediate reasoning steps. For a complicated problem, there will be more number of reasoning steps and hence more computation can be allocated.
2. If there is an incorrect prediction, and a chain of thought is displayed for this prediction, it is easier to understand where the reasoning path has gone wrong but getting an understanding of a model's computations for a prediction is still an open question
3. It helps in solving arithmetic reasoning, common sense reasoning and symbolic reasoning problems and any other problem that can be solved using natural language
4. In LLMs(sufficiently larger) that are previously trained and currently in use, chain of thought reasoning can be elicited by using chain of thought reasoning steps in exemplars of few shot prompting

3 Arithmetic reasoning:
When chain of thought prompting is used by 540B parameter LLM , it could perform comparably with task specific fine tuned models and it set a new benchmark challenging even GSM8K benchmark

3.1 Experimental setup

Benchmarks : Five benchmarks were considered

a. GSM8K math world problems
b. SVAMPdataset of math world problems with different structures
c. ASDiv dataset of diverse math world problems
d. AQuA dataset of algebraic world problems
e. MAWPS benchmark

Standard Prompting : Few shot learning benchmarks from the paper 'Language Models are few shot learners' are considered as standard prompting<input as question, output as answer> baseline for comparison with the current paper's arithmetic chain of thought
prompting <input, chain of thought, Final output>performance

Chain of thought Prompting: Each exemplar in few short learning is augmented with chain of thought. Eight such exemplars are used for prompting. These exemplars are manually composed.

To check whether this form of prompting is successful in eliciting successful reasoning for a range of math world problems , these 8 exemplars are used for above benchmark datasets except for AQuA. For AQuA 4 exemplars were used since it is multiple choice Q&A dataset

Language Models:

1. GPT3 (InstructGPT models of 350M, 1.3B, 6.7B, and 175B parameters)
2. LaMDA(has models of 422M, 2B, 8B, 68B, and 137B parameters)
3. PaLM (has models of 8B, 62B, and 540B parameters)
4. UL2 20B
5. Codex

Above five 5 language models are evaluated .

Greedy decoding (a method where next token(word or character) is predicted(the one with highest probability is selected) based on past word or character) is used to get output from each model.

In the follow up work, it proved that chain of prompting can be improved by taking majority vote response from the outputs sampled over multiple generations of a model.

Majority vote response example:

Each model generates one response in each one of the multiple runs

Response in Run1: "The capital of France is Paris."
Response in Run2: "Paris is the capital of France."
Response in Run3: "The capital of France is Berlin."

Count Responses:
Paris: 2 occurrences (from Response in Run1 and Response in Run2)
Berlin: 1 occurrence (from Response in Run3)

Majority vote result: "Paris"

Final output:
"The capital of France is Paris."

For LAMDA, results are averaged over 5 different seeds (each seed contained different ordering of the 8 exemplars). Since shuffling the order using 5 seeds did not show much variance, only one seed was used for all other models.

3.2 Results

Chain of thought prompting yields performance gains only for models with large number of parameters (~100B). Smaller models performed poorly with chain of thought prompting compared to their performance when standard prompting was used.

For models like GPT and PaLM, performance more than doubled when chain of thought prompting was used with complex dataset like GSM8K. But, this prompting yielded negative performance or very small performance gain when these 2 models are prompted with datasets like SingleOp (the easiest subset of MAWPS i.e Math world problems set which require just single step to solve the problem)

Chain of thought prompting yielded good performance for models like GPT3 175B and PaLM 540B when compared to the task specific model which is fine-tuned using a labelled dataset(prior supervised)

PaLM 540B achieved state of the art with chain of thought prompting in solving problems from GSM8K, SVAMP, and MAWPS datasets. It is to be noted that PaLM 540B has surpassed the prior supervised best performance when standard prompting was used for SVAMP dataset. For other 2 datasets AQuA and ASDiv, PaLM 540B, with chain of thought prompting reached a performance that is within 2 percent of the prior best performance

To understand clearly why chain of thought prompting worked , LaMDA 137B model generated chain of thoughts for the dataset GSM8K were manually examined. Among 50(random samples) chain of thoughts generated that led to a correct final answer, two chain of thoughts were incorrect and rest of them were logically and mathematically correct.

The other 50 (random samples) which gave wrong final answer were also manually examined. The summary of analysis:
46% of these were wrong due to symbol mapping error, calculator error or one reasoning step missing. Chain of thoughts were almost correct in these and the errors are minor
54 % of these were wrong due to major errors - incorrect coherence or wrong semantic understanding.

To understand why chain of thought reasoning ability increases with the scale of the model , a similar analysis of errors was made for the model PaLM 62B and whether those errors were corrected when scaling increased to PaLM 540B . It was inferred that a large proportion of one step missing errors and semantic understanding errors were fixed when scale was increased.

3.3 Ablation study

The ablation study aims to verify whether the benefits of chain of thought prompting can be achieved with other forms of prompting. The three variations of chain of thought prompting were studied.

a. Equation Only: Chain of thought prompting helps since it produces equation to be evaluated in its reasoning steps. Model is prompted to output only a equation before giving the final answer. Equation only prompting does not help for GSM8K dataset which helps us to understand that the semantics of GSM8K are very difficult to be translated into an equation without natural language reasoning steps in chain of thought . For datasets with one step or two steps problems performance was improved with equation only prompting since equation could easily be generated from the question.

b. Variable Compute Only: In chain of thought prompting, compute allocated is directly proportional to the number of reasoning steps produced. In order to study this, at certain steps model is prompted to output only dots (…) where number of dots is equal to number of characters in the equation. (compute decreases when only dots are produced instead of equation). Even with this prompting, the performance achieved was comparable to baseline performance. This proves that it is not just the variable compute that impacts the performance but also the natural language reasoning steps (producing dots) has some utility in this situation.

c. Chain of thought after answer: Chain of thought prompting allows the model for better access of the knowledge acquired during pre-training. In this configuration, chain of thought prompt is given after final answer(isolating whether the model actually depends on the produced chain of thought to give the final answer) and a performance compared to baseline performance was achieved. This proves that chain of thought have sequential reasoning embedded within them that helps model beyond just giving the final answer for a question. It could be understood that

Typical CoT(Chain of thought) sequence:

Q1 , Chain of thought prompt1 and answer1

In this configuration ‘Q1, answer1 and Chain of thought prompt1’ sequence could be helping to give answer2 for Q2 correctly

3.4 Robustness of Chain of thought

Varying the permutation order of chain of thought prompt exemplars changed the performance from 54.3%(near chance) to 93.4%(state of the art). Evaluation of robustness of chain of thoughts written by different annotators is done.

Considering the different chain of thoughts written for same exemplars written by
1. Annotator A(chain of thoughts were more concise) 2. Annotator B and 3. Annotator C , results were studied for LaMDA 137B on GSM 8K dataset and MAWPS dataset . It was observed that there is a variance in solve rate for CoTs from different annotators which was expected for CoT prompting. However, all CoTs outperformed standard prompting by large margin. This result implies successful use of CoT does not depend on linguistic style.

To confirm that CoT works for other sets of exemplars, experiments were done with 3 sets of 8 exemplars from GSM8K dataset, an independent source(examples in this dataset already included CoT) . These prompts performed well that is comparable to the performance of other manually written CoT exemplars, outperforming standard prompting.

In Arithmetic reasoning it is concluded that CoT prompting is robust to a. linguistic style b. independently written CoT exemplars c. other datasets e. exemplar orders and f. varying number of exemplars .
This conclusion about robustness was made by checking that performance of CoT is always better when compared to performance of standard prompting

4 Commonsense reasoning

CoT is mainy used for arithmetic reasoning. But, its natural language based reasoning steps help in solving wide range of commonsense reasoning problems as well.
Commonsense reasoning problems include human and physical interactions. Better commonsense reasoning helps in better interaction with natural world. Current natural language understanding systems do not help much in commonsense reasoning

Benchmarks:
Five datasets specific for commonsense reasoning were considered.
a. CSQA : This dataset asks common sense reasoning questions about world having complex semantics which needs prior knowledge
b. StrategyQA: This required model to infer multi hop strategy to answer questions.
Two specialized evaluation sets are from BIG bench effort are chosen:
c. Date understanding: Inferring a date from a given context
d. Sports understanding: Determining whether a sentence related to sports is plausible or not
e. SayCan dataset: Mapping a natural language instruction steps to a sequence of robot actions from a discrete set

Prompts:

The experimental setup used for arithmetic reasoning was followed for commonsense reasoning as well.
For CSQA and StrategyQA datasets, the samples are randomly drawn from training sets and chain of thoughts were manually written.

BIG-bench tasks do not have training dataset but only evaluation datasets. First 10 samples are selected from each evaluation set and chain of thought exemplars were created using them . Thereafter, remaining samples in evaluation datasets were used to test output

From SayCaN dataset, 6 samples which were used in an earlier research paper are used in this paper and CoTs were manually composed using them.

Results:

For each of the above 5 datasets mentioned above

- performance improved with scale in both standard prompting and CoT prompting
- PaLM showed highest performance for 540B parameter variant
- PaLM 540B outperformed prior state of the art (75.6% vs 69.4%)
- PaLM 540B outperformed an unaided sports enthusiast on sports understanding dataset (95.4% vs 84%).

These results prove that CoT prompting improved performance in common sense reasoning tasks.
It is to be noted that performance gain was minimal in CSQA dataset

5 Symbolic reasoning

Symbolic reasoning is a type of reasoning that involves the manipulation of symbols to represent concepts, ideas, or relationships. It is commonly used in fields such as mathematics, computer science, and philosophy

Symbolic reasoning is easy for humans but difficult for language models.

CoT (Chain of Thought) helps language models to do symbolic reasoning that is difficult to do using standard prompting. Not only that, it helps in length generalization (which means the output during inference can be longer than the length of the outputs in the few shot exemplars)

Tasks

Last letter concatenation:
This task asks the model to concatenate last letters of two words in a name. This task is challenging version of first letter concatenation task, which language models can perform without CoT.
The full names are generated by randomly concatenating names from the top one-thousand first and last names from name census data.

Coin flip.
This task asks the model to answer whether a coin is still heads up after people either flip or don't flip the coin (e.g., "A coin is heads up. Phoebe flips the coin. Osvaldo does not flip the coin. Is the coin still heads up?" → "no")

Since these symbolic tasks are well defined, an in-domain test set for such tasks is constructed in such a way that each sample in this set has same number of reasoning steps as an exemplar in the training set
There is an out domain test set where each sample has reasoning steps more than the exemplar's reasoning steps
In training exemplars
1, Input : Amy Brown , output : yn

In Out of domain test exemplars
1. Input: Amy Brown Vishnesky , output : yny

Out of domain test exemplars helps the model to test if it can concatenate last letters from three or four words

Similary, out of domain test is created for the number of flips in coin flip task as well.

Experimental set up for symbolic reasoning uses the same methods and models as in Commonsense reasoning and Arithmetic reasoning experiments.

Few shot exemplars with Chain of thoughts are again manually composed same as in Commonsense reasoning and Arithmetic reasoning experiments.

Results:
With CoT prompting in PaLM 540B model , 100% solve rate is achieved.

It is to be noted that standard prompting solves Coin flip task with PaLM 540 B model but not with LaMDA 137B model

In domain evaluations are toy tasks since the solution structure is already provided in the exemplars. Model has to repeat the same steps as in the exemplar provided ,changing symbols during test. But , even such a toy task cannot be performed by the small models.

The ability to do symbolic reasoning for these tasks listed above (letter concatenation in domain , coin flip

in domain) comes up when the model scale is 100B parameters

For OOD (out of domain tasks), standard prompting fails for both tasks.
With CoT and increasing scale in language models, solve rate performance for OOD tasks increases but this performance is lower than performance of the In domain tasks solved by CoT

Hence, CoT prompting helps in length generalization beyond seen chains of thought for language models of sufficient scale

Discussion:

- Chain of Thought(CoT) prompting is useful in eliciting in multistep reasoning in large language models
- CoT improves performance of large models for arithmetic , common sense and symbolic reasoning tasks.
  These improvements are stronger than ablations, robust to different annotations style/linguistic style , different language models and exemplars
- Experiments showed how linguistic nature of CoT makes it suitable for commonsense reasoning tasks
- CoT helps in output sequence length generalization for OOD test samples in symbolic reasoning tasks
- CoT reasoning elicitation in off the shelf(pre trained) models is obtained by just prompting
- No other language model is fine-tuned as part of the process of writing this paper
- The improvement of CoT reasoning with model scale is a prevalent observation (from other research work)
- For many reasoning tasks where standard prompting did not show much performance, CoT prompting helped to achieve it
- In many of the tasks CoT overshadowed standard prompting. This put 2 two questions into picture
  1. How much more the reasoning ability of the LLMs can increase with the increase in its scale ?
  2. Like CoT, what other prompting methods can increase the capabilities of LLMs ?

Limitations
- Though the authors of paper qualify that CoT emulates the reasoning behavior of humans, it is still not clear whether neural network actually performs the reasoning
- The cost of augmenting few shot exemplars with CoT is minimal but when it comes to augmenting datasets with CoT for fine tuning the model , the costs might be higher . But this problem could be solved by generating synthetic data which includes CoT or zero shot generalization.
  Zero shot generalization: When a model is trained with large data, based on the features it learnt from the data, it could predict correctly for unseen test data . E.g.: Model which is trained on Cats and Dogs data could predict whether horse exists in a picture
- There is no guarantee for correct reasoning paths which can lead to correct or incorrect answers
  Improving factual/correct generations of LLMs is open for future work
- Since CoT works only with large models and large models are costly. Further research shall explore more on how to make smaller models reason correctly

Related Work :
Two directions inspired the current paper

1. Reasoning steps:
   - The use of natural language reasoning steps to solve a problem instead of using the formal language (pioneered by Ling et al. (2017))
     Before this, formal languages were used to solve problems instead of natural language reasoning.
   - Work by Cobbe et al(2021 who extended work of Ling by creating a large dataset that was used in fine tuning a model instead of training from scratch.
   - Nye et al. (2021) showed that generating programs step by step prediction of python programs is better than generating the whole final output at once

2. Few shot prompting and the other types of similar prompting:
   - Few shot prompting idea given by Brown et al (2020). Based on that several other methods improved prompting ability of models which are automatic learning of prompts, giving step by step instructions to models
     These approaches improve the input part of the prompt

The current work follows the idea of step by step reasoning but the reasoning is included in output of the exemplar unlike the augmentation of the input part of prompt mentioned in point2

Conclusions

1. Explored that the method Chain of Thought reasoning is simply and broadly applicable to LLMs to improve their reasoning ability
2. From experiments on arithmetic, commonsense and symbolic reasoning tasks it is understood that Chain of Thought reasoning performance improves with the scale of LLMs
3. Current work might inspire further work on natural language methods to improve reasoning of LLMS

References:

1. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models paper
Link : https://arxiv.org/pdf/2201.11903