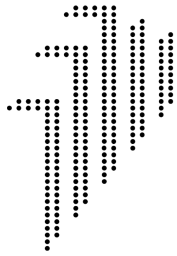

h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Exposé

Lorenc Zhuka
Hochschule Darmstadt
Fachbereich Mathematik und Naturwissenschaften
Data Science

15.03.2023

Chain of thought prompting in NLP with Large Language Models

Research Problem

Chain of thought prompting is a method for guiding large language models to generate coherent and meaningful text in response to a given prompt or input. This approach aims to improve the consistency and coherence of the model's responses by providing a series of prompts or cues that build on each other to create a cohesive narrative. The use of large language models, such as GPT-3, has revolutionized natural language processing (NLP) by enabling machines to generate high-quality text across a wide range of domains. However, these models can sometimes produce nonsensical or irrelevant text in response to prompts, which limits their usefulness in practical applications. Chain of thought prompting seeks to address this issue by providing the model with a series of related prompts that guide it towards generating relevant and coherent text. These prompts can be based on previous responses from the model, context from the input text, or other relevant information.

In this thesis I will explore, how generating a chain of thought -- a series of intermediate reasoning steps -- significantly improves the ability of large language models to perform complex reasoning. In particular, I will show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called chain of thought prompting, where a few chain of thought demonstrations are provided as exemplars in

prompting and explore on which tasks this technique works the best like arithmetic, commonsense, and symbolic reasoning tasks. However, the implementation of chain of thought prompting also poses significant challenges, including selecting relevant prompts, manual design of the demonstrations coherence of generated text and avoiding bias. These challenges must be addressed to realize the full potential of chain of thought prompting with large language models.

Research questions and objectives

- ➔ On which tasks or problems does CoT prompting work the best?
- ➔ Perform Error Analysis in order to better understand why CoT works. Find out when the model fails and when it performs well.
- ➔ Try out different model sizes to find out how it affects the performance of CoT.
- ➔ Run several experiments using different CoT prompting techniques like Standard Prompting, Zero-shot CoT, Few-shot CoT using manual design of prompts, fine-tuned large GPT on different benchmarks like GSM8K, SVAMP, ASDiv.
- ➔ Compare the performance of these techniques and find out the advantages and limitations of them.
- ➔ Investigate the robustness of CoT prompting across different models/architectures, different order of few-shot examples, different annotators (for same few-shot demonstration) , different demonstrations and different number of demonstrations.
- ➔ Run several experiments using Automatic CoT techniques like Retrieval CoT, Random CoT, Auto CoT and compare them with CoT with manual design.
- ➔ Investigate the effect of wrong demonstrations. Are Auto CoT with diversity sampling and Retrieval CoT affected by wrong demonstrations? How can we reduce the generation of wrong demonstration?
- ➔ Investige in which cases or NLP tasks adding CoT to Standard Prompting hurts performance.
- ➔ Conduct experiments to compare self consistency with naive greedy decoding used in CoT prompting.
- ➔ Test the robustness of self-consistency method with respect to sampling strategies and scaling as well as to wrong demonstrations/imperfect prompts.
- ➔ Investigate more sophisticated strategies like React and compare with CoT.
- ➔ Can we improve the CoT/React performance using human-in-the-loop behavior correction? How can we correct wrong explanations using humans as annotators?
- ➔ Can we achieve higher performance and truthfulness by fine-tuning a language model using high quality CoT prompts (correct explanations)?

Methodology

Data collection

The research involves collecting datasets for different NLP tasks like arithmetic reasoning, commonsense reasoning, question answering and fact verification.

For arithmetic reasoning we will use the following datasets:

- GSM8K: The GSM8K dataset contains 8.5K high quality linguistically diverse grade school math word problems.
- SVAMP: The SVAMP dataset contains math word problems with multiple arithmetic operations, covering various arithmetic concepts such as ratios, percentages and algebraic expressions.
- AQuA : The AQuA dataset consists of about 100,000 algebraic word problems with natural language rationales.
- ASDiv : The ASDiv dataset contains math word problems and is designed to evaluate the ability of models to perform arithmetic reasoning.

For commonsense reasoning we will use the following datasets:

- CSQA: The CSQA (CommonsenseQA) dataset is a new multiple-choice question answering dataset that requires different types of commonsense knowledge to predict the correct answers . It contains 12,102 questions with one correct answer and four distractor answers.
- StrategyQA : The StrategyQA dataset is a question-answering benchmark focusing on open-domain questions where the required reasoning steps are implicit in the question and should be inferred using a strategy.

For question answering we will use the following datasets:

- HotpotQA is a question answering dataset featuring natural, multi-hop questions, with strong supervision for supporting facts to enable more explainable question answering systems.

For Fact Verification we will use the following datasets:

- FEVER : The FEVER dataset consists of 185,445 claims manually verified against the introductory sections of Wikipedia pages and classified as SUPPORTED, REFUTED or NOTENOUGHINFO. For the first two classes, systems and annotators need to also return the combination of sentences forming the necessary evidence supporting or refuting the claim.

Models and Metrics

- The type of language models used for CoT prompting varies depending on the specific task and context. However, in general language models used for CoT prompting are typically large-scale neural network models, such as transformer-based models like GPT-3, that have been pre-trained on large amounts of text data.
- To run the experiments different large language models will be used like : UL2, GPT-3 (text-davinci-002) and Codex (code-davinci-002). Authors have also made use of LaMDA-137B and PaLM-540B but these models are not publicly available.
- To evaluate the performance of these models accuracy (solve rate) will be used as performance metric.
- To answer the main part of the research questions, chain-of-thought reasoning will be elicited simply by prompting an off-the-shelf language model. No language models will get fine-tuned.

References

For this research work, I will refer to the following papers :

➔ Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
➔ Automatic Chain of Thought Prompting in Large Language Models
➔ Self-consistency improves chain of thought reasoning in language models
➔ ReAct: Synergizing Reasoning and Acting in Language Models