Enhancing Reasoning Path Diversity in Commonsense Question Answering

Sagnik Nandi
23B0905

Harshil Solanki
23B1016

Contents

1 Paper Overview

https://arxiv.org/pdf/2203.11171v4

Paper Title: Self-Consistency Improves Chain of Thought Reasoning in Language Models

The paper investigates methods to enhance the accuracy and performance of Large Language Models (LLMs) through optimized techniques and decoding strategies.

1.1 Key Techniques Highlighted:

- **Chain-of-Thought (CoT) Prompting:** This involves providing illustrative examples to guide the model's reasoning process.
- **Self-Consistency Decoding:** This method generates diverse reasoning paths and selects the most consistent final answer by marginalizing over them.
- CoT and self-consistency are effective for tasks with fixed answer sets and can be adapted for open-text generation using suitable consistency metrics.

1.2 Model Architectures used:

- **Decoder-Only Models (e.g., GPT-3, PaLM, LaMDA):**
  - Well-suited for autoregressive text generation tasks like conversational AI and creative writing.
  - Offer simpler architecture and faster inference times.
- **Encoder-Decoder Models (e.g., T5, BART):**
  - Better suited for sequence-to-sequence tasks such as translation and summarization.
  - Provide stronger input-output alignment but with higher computational demands.

1.3 Sampling Strategies and Self-Consistency:

- Sampling techniques (e.g., temperature sampling, top-k truncation) are crucial for generating diverse reasoning paths.
- Self-consistency decoding outperforms traditional methods like sample-and-rank and beam search in terms of accuracy.
- Self-consistency incurs higher computational costs.
- The authors suggest using a limited number of reasoning paths to balance performance and efficiency.

1.4 Limitations and Future Directions:

- Self-consistency requires significant computational resources.
- Self-consistency may generate nonsensical reasoning paths.
- Future work could focus on fine-tuning models with supervised data generated through self-consistency to improve single-inference accuracy.
- Grounding models' rationale generation remains an open challenge requiring further research for logical and accurate outputs.

2 Model Overview

2.1 Dataset

https://paperswithcode.com/dataset/commonsenseqa

CommonsenseQA 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.

2.2 Pre-trained LLM

We are using the *together.ai* interface to prompt with SOTA Language models like **Llama 4** and **Deepseek R1**.

2.3 Sampling

It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer.

2.4 Output

Comparison of the self-consistency method with greedy decoding and other baseline methods (e.g., beam search) to highlight its advantages.

3 Current Progress and Code

3.1 Analysis of Strategies

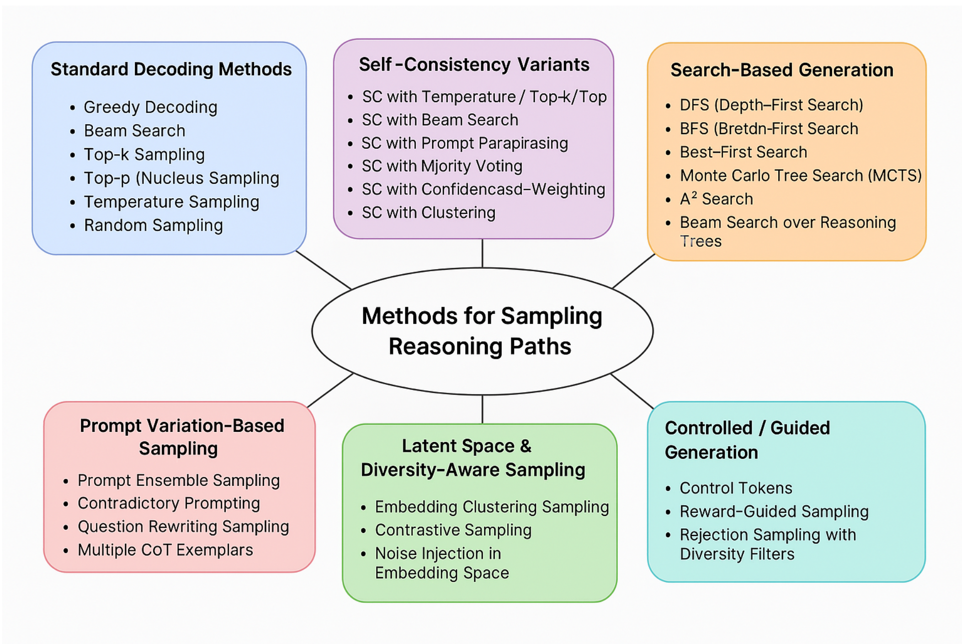

Figure 1: Different Strategies for Sampling

3.2 Dataset Analysis

An example of 'train' looks as follows:


```
{
  'id': '075e483d21c29a511267ef62bedc0461',
  'question': 'The sanctions against the school were a punishing blow, and they seemed to _____
  the efforts the school had made to change?',
  'question_concept': 'punishing',
  'choices': {
    'label': ['A', 'B', 'C', 'D', 'E'],
    'text': ['ignore', 'enforce', 'authoritarian', 'yell at', 'avoid']
  },
  'answerKey': 'A'
}
```


3.3 Self-consistency Implementation

- Ran initial experiments using multiple reasoning paths and beam search
- Choose the most common answer based on 10 responses generated, look at `sc_10_samples.py` in self_implementation

Github Link : https://github.com/HARSHIL2836R/ReasoningLLMs/tree/main/self_implementation

4 Future Plan of Action

4.1 Metrics

Provide metrics like:

- Number of unique reasoning paths.
- Path similarity measures (e.g., cosine similarity).
- Diversity scores (e.g., Shannon entropy) to assess the diversity of generated reasoning paths.

4.2 Strategies

Basic strategies:

- Sample multiple reasoning paths by **varying temperature** and aggregate answers via majority voting
- **Cluster** the sampled responses based on cosine similarity and select most common answer

Advanced strategies:

- Generate different paths by **varying prompts** like paraphrasing the question
- Model reasoning steps as a graph and use search strategies like **BFS**, **DFS** or **A*** to discover paths

4.3 Timeline

- **April 14–20**: Implement basic strategies and different metric computations and compare their performances
- **April 29-30**: Implement advanced strategies and improving the performance
- **May 1–3**: Final analysis, report writing, and polishing of visualizations and results

5 Code References

1. **Codebase:** https://github.com/codelion/optillm/blob/main/optillm/self_consistency.py
2. Drive Link : https://drive.google.com/drive/folders/1Vzzdk7ZRTuls8XM_-3UT0aqaiBEoyV4T?usp=sharing