STaR: Self-Taught Reasoner – Report

Aniket Tiwari
MDS202308
anikett.mds2023@cmi.ac.in

May 20, 2025

Core Methodology

STaR's iterative self-training framework addresses the *rationale bottleneck* in language models through:

1. **Generation Phase**:
   - Models $p_M(r|x)p_M(y|x,r)$ using GPT-J (6B)
   - Initialized with human-written (question, rationale, answer) triples
2. **Verification Phase**:
   - Filters outputs using indicator $\mathbb{I}(y_i = \hat{y}_i)$ by keeping correct answers data points only
   - Retains 78.2% of total training data
3. **Rationalization Phase**:
   - Reverse-engineers explanations via $p_M(r|x,y)$
   - Adds 8.5% high-quality rationales (total 86.7% data utilization)

Technical Implementation

The gradient update rule combines both phases:

$$\nabla \mathcal{J} = \sum_i \mathbb{E}_{r,y} \left[\mathbb{I}(y_i = \hat{y}_i) \cdot \nabla \log p_\theta(\hat{y}_i, \hat{r}_i | x_i) \right]$$

Key Configurations:

- *Batch Processing*: 8 sequences × 1024 tokens (TPU-v3 constraints)
- *Optimization*: Adam ($\eta = 10^{-6}$) with gradient clipping
- *Convergence*: Typically 3–5 iterations of generate–verify–improve

Results Analysis

CommonsenseQA	**GSM8K**
72.5% accuracy (vs GPT-3's 73.0% with 100% human data)	10.7% accuracy (3× better than few-shot)
Human evaluators preferred STaR's rationales 74% of the time	Learned compact solution strategies

Cross-Domain Applications

Initial experiments suggest that STaR-trained LLMs generate more structured reasoning in healthcare and legal tasks (e.g. breaking diagnoses into symptom-test-condition chains). However, while responses appear more logically sound, factual accuracy depends on domain knowledge. Combining STaR with targeted datasets like **MedQA** (for medicine) or **LegalBench** (for law) could bridge this gap, merging improved reasoning with expert-level precision.

Extensions & Future Directions

- **Quiet-STaR**: Token-level rationales improve generalization across tasks
- **Lean-STaR**: Applies informal reasoning to theorem proving in Lean
- **STaR-SQL**: Text-to-SQL generation using self-taught rationales
- **START**: Incorporates external tools for enhanced step-wise reasoning
- **RL-STaR**: Uses reinforcement learning for reasoning policy optimization

References

1. Zelikman, E. et al. (2022). *STaR*
2. Chen et al. (2024). *Quiet-STaR*
3. Bai et al. (2024). *Lean-STaR*
4. Hu et al. (2024). *STaR-SQL*
5. Zhang et al. (2025). *START*