Generating Natural Language Proofs with Verifier-Guided Search

Kaiyu Yang, Jia Deng, and Danqi Chen

Department of Computer Science, Princeton University

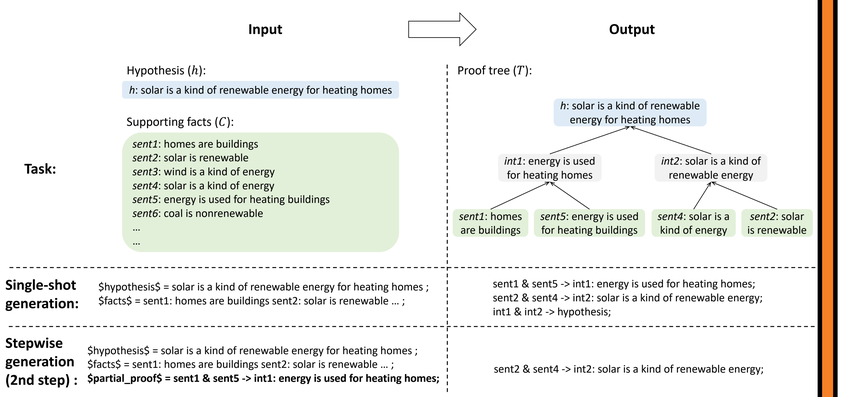
Deductive Reasoning in Natural Language

How to draw conclusions from assumptions in *natural language*?

- **Challenges**
  1. Fuzzy, imprecise, requiring implicit knowledge
  2. No finite, well-defined inference rules as in formal logic
  3. Difficult for large pretrained language models

Generating Natural Language Proofs

- **Task**
  1. Input: a hypothesis h and a set of supporting facts $C = \{sent_1, sent_2, \dots, sent_n\}$ in natural language
  2. Output: a proof tree T for deriving h from a subset of C
     a. The root node is h; the leaf nodes are sentences in C
     b. Others are intermediate conclusions *generated by the model*



- **Single-shot methods**
  1. Generate the entire proof in a single shot
  2. Encode the input/output as text sequences, and map the input to the output by finetuning an encoder-decoder transformer

- **Stepwise methods**
  1. Generate the next step given a partial proof
  2. Compositional; easier for the model to learn and generalize
  3. Limited success on real-world data
  4. Hallucinate invalid steps
  5. Many irrelevant steps

Contributions

- **NLProofS (Natural Language Proof Search)**
  1. A new method for stepwise proof generation
  2. Generate relevant proof steps conditioning on the hypothesis
  3. Train an independent verifier to prevent hallucination

Our Method: NLProofS

- **Overview**
  1. Training: train a stepwise prover for generating candidate steps and a verifier for scoring the validity of steps
  2. Inference: search for proofs with high aggregated proof scores

- **Stepwise prover**
  1. Similar to existing methods for stepwise proof generation
  2. Finetune a T5 model to predict the next step
  3. Generate multiple candidate steps via beam search

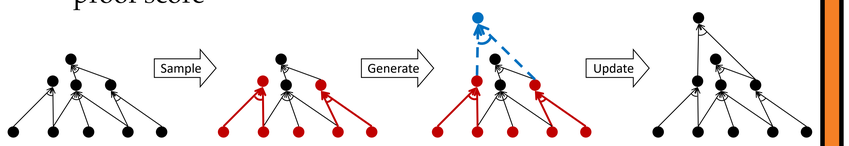
- **Verifier**
  1. An independently trained neural network for checking the validity of proof steps, producing a score in [0, 1]
  2. Finetune a RoBERTa model with positive examples and pseudo-negative examples
  3. Step scores are aggregated to calculate proof scores.

- **Proof search**
  1. Initialize the proof graph with a greedy proof from the prover
  2. Sample a partial proof from the graph
  3. Generate multiple candidate steps using the prover
  4. Execute them to update the graph and keep track of scores produced by the prover/verifier
  5. Return the proof with the maximum proof score

Algorithm 1: Proof search.

Input : Hypothesis h, supporting facts C, stepwise prover $\mathcal{P}$, verifier $\mathcal{V}$
Output : Proof tree T

```
G ← generate_greedy(P, h, C)
PG ← initialize_graph(G)
explored ← ∅
while true do
    partial_proof ← sample_new(PG, explored)
    explored ← explored ∪ {partial_proof}
    steps, p_scrs ← generate(P, partial_proof)
    v_scrs ← verify(V, steps)
    scrs ← (p_scrs + v_scrs)/2
    PG' ← update(PG, steps, scrs)
    if PG' = PG then
        break
    PG ← PG'
return extract_proof(PG)
```



Experiments

- **EntailmentBank[2]**
  1. Challenging, human-authored proofs
  2. Four evaluation metrics: Leaves, Steps, Intermediates, Overall
  3. We outperform existing methods
  4. Verifier-guided search is important for the improvement

Method	Leaves	Steps	Intermediates	Overall
EntailmentWriter[2]	35.6	22.9	28.5	20.9
IRGR[3] (concurrent work)	23.8	22.3	26.5	22.0
MetGen[4] (concurrent work)	48.6	30.4	32.7	28.0
NLProofS (ours)	**58.8**	**34.4**	**37.8**	**33.3**
w/o search	56.5	33.7	36.4	31.8
w/o search w/o stepwise	45.6	29.7	32.2	27.1
w/o verifer score	55.8	33.8	36.1	31.9

- **The verifier helps**
  1. Mitigate hallucination
  2. Avoid copying premises as the conclusion

Hypothesis	Premises	Conclusions generated by the model
The next new moon will occur on June 30.	1. A new moon is a kind of phase of the moon. 2. A moon phase occurs 28 days after the last time it occurs.	**EntailmentWriter[2]**: The next new moon will occur 28 days after June 2.
		NLProofS (ours): The next new moon will occur 28 days after the last new moon.
Planting trees prevents soil from washing away.	1. Planting trees increases the amount of trees in an environment. 2. Tree roots decrease / reduce soil erosion.	**EntailmentWriter[2]**: Plants trees increases the amount of trees in an environment.
		NLProofS (ours): Planting trees decreases soil erosion.

- **RuleTaker[1]**
  1. Simple, synthetic proofs generated by templates
  2. We perform competitively with existing methods

Method	Answer	Proof
FaiRR[5]	99.2	98.8
ProofWriter[1]	99.8	99.7
NLProofS (ours)	99.3	99.2

- **Other observations**
  1. The prover might be the main bottleneck
  2. Long proofs remain challenging

1. Tafjord et al. "ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language." Findings of ACL 2021
2. Dalvi et al. "Explaining Answers with Entailment Trees." EMNLP 2021
3. Ribeiro et al. "Entailment Tree Explanations via Iterative Retrieval-Generation Reasoner." NAACL 2022
4. Hong et al. "METGEN: A Module-Based Entailment Tree Generation Framework for Answer Explanation." NAACL 2022
5. Sanyal et al. "FaiRR: Faithful and Robust Deductive Reasoning over Natural Language." ACL 2022