Generating Natural Language Proofs with Verifier-Guided Search

Kaiyu Yang, Jia Deng, and Danqi Chen
Deductive Reasoning in Natural Language

Assumptions

- homes are buildings
- energy is used for heating buildings
- solar is a kind of energy
- solar is renewable

Conclusion

- solar is a kind of renewable energy for heating homes
Deductive Reasoning in Natural Language

- homes are buildings
- energy is used for heating buildings
- solar is a kind of energy
- solar is renewable

Assumptions: homes are buildings, energy is used for heating buildings, solar is a kind of energy, solar is renewable

Conclusion: solar is a kind of renewable energy for heating homes

• Studied extensively in formal theorem proving. But why natural language?
Deductive Reasoning in Natural Language

- Studied extensively in formal theorem proving. But why natural language?
- Natural language is a suitable vehicle for reasoning
  - Expressive and flexible enough for representing all human knowledge
  - Scalable, no manual formalization needed
  - Interface with humans

Assumptions
- homes are buildings
- energy is used for heating buildings
- solar is a kind of energy
- solar is renewable

Conclusion
- solar is a kind of renewable energy for heating homes

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Reasoning Challenges Posed by Natural Language

• Fuzzy, imprecise, requiring implicit knowledge

• No finite, well-defined inference rules
Plausible Solution: Semantic Parsing

• First convert natural language to formal logic; then apply a theorem prover

\[\text{homes are buildings}\]
\[\text{energy is used for heating buildings}\]
\[\text{solar is a kind of energy}\]
\[\text{solar is renewable}\]

\[\text{solar is a kind of renewable energy for heating homes}\]
Plausible Solution: Semantic Parsing

• First convert natural language to formal logic; then apply a theorem prover

∀x ∀y, energy(x) ∧ building(y) → heats(x, y)

building(Home)

energy(Solar)

theorem prover

renewable(Solar) ∧ energy(Solar) ∧ heats(Solar, Home)

[Mineshima et al. EMNLP 2015]
Plausible Solution: Semantic Parsing

• First convert natural language to formal logic; then apply a theorem prover

\[ \forall x \forall y, \text{energy}(x) \land \text{building}(y) \rightarrow \text{heats}(x, y) \]

\[ \text{energy}(\text{Solar}) \land \text{renewable}(\text{Solar}) \land \text{heats}(\text{Solar}, \text{Home}) \]

• Need an ontology to capture the entire natural language: infeasible

[Mineshima et al. EMNLP 2015]
Reasoning with Large Language Models

- “Soft” reasoning over natural languages
- No formal representations or symbol manipulation

- homes are buildings
- energy is used for heating buildings
- solar is a kind of energy
- solar is renewable
- solar is a kind of renewable energy for heating homes

[Clark et al. IJCAI 2020]
Reasoning with Large Language Models

• “Soft” reasoning over natural languages
• No formal representations or symbol manipulation

homes are buildings
energy is used for heating buildings
solar is a kind of energy
solar is renewable

solar is a kind of renewable energy for heating homes

[Clark et al. IJCAI 2020]

[Vaswani et al. NeurIPS 2017]
Task Formulation of Proof Generation

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

- $sent1$: homes are buildings
- $sent2$: solar is renewable
- $sent3$: wind is a kind of energy
- $sent4$: solar is a kind of energy
- $sent5$: energy is used for heating buildings
- $sent6$: coal is nonrenewable
- ...

Input

[Dalvi et al. EMNLP 2021]
Task Formulation of Proof Generation

Hypothesis \((h)\):

\(h\): solar is a kind of renewable energy for heating homes

Supporting facts \((C)\):

\textit{sent1}: homes are buildings
\textit{sent2}: solar is renewable
\textit{sent3}: wind is a kind of energy
\textit{sent4}: solar is a kind of energy
\textit{sent5}: energy is used for heating buildings
\textit{sent6}: coal is nonrenewable
...

Input

[Dalvi et al. EMNLP 2021]
Task Formulation of Proof Generation

Hypothesis ($h$):
$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):
- $sent1$: homes are buildings
- $sent2$: solar is renewable
- $sent3$: wind is a kind of energy
- $sent4$: solar is a kind of energy
- $sent5$: energy is used for heating buildings
- $sent6$: coal is nonrenewable
...

Proof tree ($T'$):

- $h$: solar is a kind of renewable energy for heating homes
  - $int1$: energy is used for heating homes
    - $sent1$: homes are buildings
    - $sent5$: energy is used for heating buildings
  - $int2$: solar is a kind of renewable energy
    - $sent2$: solar is renewable
    - $sent4$: solar is a kind of energy

[Dalvi et al. EMNLP 2021]
Task Formulation of Proof Generation

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

- **sent1**: homes are buildings
- **sent2**: solar is renewable
- **sent3**: wind is a kind of energy
- **sent4**: solar is a kind of energy
- **sent5**: energy is used for heating buildings
- **sent6**: coal is nonrenewable
  ...

Proof tree ($T'$):

- **int1**: energy is used for heating homes
- **int2**: solar is a kind of renewable energy

- **sent1**: homes are buildings
- **sent5**: energy is used for heating buildings
- **sent4**: solar is a kind of energy
- **sent2**: solar is renewable

[Dalvi et al. EMNLP 2021]
Task Formulation of Proof Generation

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

- sent1: homes are buildings
- sent2: solar is renewable
- sent3: wind is a kind of energy
- sent4: solar is a kind of energy
- sent5: energy is used for heating buildings
- sent6: coal is nonrenewable
- ...

Proof tree ($T'$):

- $h$: solar is a kind of renewable energy for heating homes
  - int1: energy is used for heating homes
    - sent1: homes are buildings
    - sent5: energy is used for heating buildings
  - int2: solar is a kind of renewable energy
    - sent2: solar is renewable
    - sent4: solar is a kind of energy
    - sent6: coal is nonrenewable

Input

Output

[Dalvi et al. EMNLP 2021]
Task Formulation of Proof Generation

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

$sent1$: homes are buildings
$sent2$: solar is renewable
$sent3$: wind is a kind of energy
$sent4$: solar is a kind of energy
$sent5$: energy is used for heating buildings
$sent6$: coal is nonrenewable
...

Proof tree ($T'$):

$h$: solar is a kind of renewable energy for heating homes

int1: energy is used for heating homes

int2: solar is a kind of renewable energy

sent1: homes are buildings
sent5: energy is used for heating buildings
sent4: solar is a kind of energy
sent2: solar is renewable

[Dalvi et al. EMNLP 2021]
Single-Shot Methods

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

$\text{sent1}$: homes are buildings
$\text{sent2}$: solar is renewable
$\text{sent3}$: wind is a kind of energy
$\text{sent4}$: solar is a kind of energy
$\text{sent5}$: energy is used for heating buildings
$\text{sent6}$: coal is nonrenewable
...
...

Proof tree ($T'$):

$h$: solar is a kind of renewable energy for heating homes

$\text{int1}$: energy is used for heating homes

$\text{sent1}$: homes are buildings
$\text{sent5}$: energy is used for heating buildings

$\text{int2}$: solar is a kind of renewable energy

$\text{sent4}$: solar is a kind of energy
$\text{sent2}$: solar is renewable

Generate the entire proof altogether

[Saha et al. EMNLP 2020]
[Dalvi et al. EMNLP 2021]
Stepwise Methods

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

$\text{sent1}$: homes are buildings  
$\text{sent2}$: solar is renewable  
$\text{sent3}$: wind is a kind of energy  
$\text{sent4}$: solar is a kind of energy  
$\text{sent5}$: energy is used for heating buildings  
$\text{sent6}$: coal is nonrenewable  
...

Proof tree ($T'$):

Step 1

$\text{int1}$: energy is used for heating homes

$\text{sent1}$: homes are buildings  
$\text{sent5}$: energy is used for heating buildings
Stepwise Methods

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

$sent1$: homes are buildings
$sent2$: solar is renewable
$sent3$: wind is a kind of energy
$sent4$: solar is a kind of energy
$sent5$: energy is used for heating buildings
$sent6$: coal is nonrenewable
...

Proof tree ($T'$):

Step 1

$int1$: energy is used for heating homes

$sent1$: homes are buildings
$sent5$: energy is used for heating buildings

Step 2

$int2$: solar is a kind of renewable energy

$sent2$: solar is renewable
$sent4$: solar is a kind of energy

Stepwise Methods

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

$sent1$: homes are buildings
$sent2$: solar is renewable
$sent3$: wind is a kind of energy
$sent4$: solar is a kind of energy
$sent5$: energy is used for heating buildings
$sent6$: coal is nonrenewable
...
...

Proof tree ($T'$):

$proof$ $tree$ $\rightarrow$

$\begin{align*}
&\text{Step 1} \\
&h: \text{solar is a kind of renewable energy for heating homes} \\
&int1: \text{energy is used for heating homes} \\
&\begin{align*}
&sent1: \text{homes are buildings} \\
&sent4: \text{solar is a kind of energy} \\
&sent5: \text{energy is used for heating buildings} \\
&sent2: \text{solar is renewable}
\end{align*} \\
&\text{Step 2} \\
&int2: \text{solar is a kind of renewable energy} \\
&\begin{align*}
&sent1: \text{homes are buildings} \\
&sent5: \text{energy is used for heating buildings} \\
&sent4: \text{solar is a kind of energy} \\
&sent2: \text{solar is renewable}
\end{align*} \\
&\text{Step 3} \\
&h: \text{solar is a kind of renewable energy for heating homes}
\end{align*}$

[Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen]
Stepwise Methods

Hypothesis ($h$): $h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

- $sent1$: homes are buildings
- $sent2$: solar is renewable
- $sent3$: wind is a kind of energy
- $sent4$: solar is a kind of energy
- $sent5$: energy is used for heating buildings
- $sent6$: coal is nonrenewable

Proof tree ($T'$):

- $h$: solar is a kind of renewable energy for heating homes
  - $int1$: energy is used for heating homes
    - $sent1$: homes are buildings
    - $sent5$: energy is used for heating buildings
  - $int2$: solar is a kind of renewable energy
    - $sent4$: solar is a kind of energy
    - $sent2$: solar is renewable

Generate the proof step by step
- Compositional; easier for the model to learn and generalize
- Limited success on real-world data

[Tafjord et al. Findings of ACL 2021]
[Sanyal et al. ACL 2022]
[Bostrom et al. arXiv 2022]
Stepwise Methods

Hypothesis ($h$):

$h$: solar is a kind of renewable energy for heating homes

Supporting facts ($C$):

sent1: homes are buildings
sent2: solar is renewable
sent3: wind is a kind of energy
sent4: solar is a kind of energy
sent5: energy is used for heating buildings
sent6: coal is nonrenewable
...
...

Proof tree ($T'$):

Generate the proof step by step

- Compositional; easier for the model to learn and generalize
- Limited success on real-world data

Input

Output

[Tafjord et al. Findings of ACL 2021]
[Sanyal et al. ACL 2022]
[Bostrom et al. arXiv 2022]
Relevance vs. Validity

• Many valid steps are irrelevant (not useful for proving the hypothesis)
Relevance vs. Validity

• Many valid steps are irrelevant (not useful for proving the hypothesis)
Relevance vs. Validity

- Many valid steps are irrelevant (not useful for proving the hypothesis)
Relevance vs. Validity

• Many valid steps are irrelevant (not useful for proving the hypothesis)
Relevance vs. Validity

- Many valid steps are irrelevant (not useful for proving the hypothesis)
- The model hallucinates invalid steps

[Sanyal et al. ACL 2022]
[Bostrom et al. arXiv 2022]
Many valid steps are irrelevant (not useful for proving the hypothesis)

The model hallucinates invalid steps

[Sanyal et al. ACL 2022]
[Bostrom et al. arXiv 2022]
Relevance vs. Validity

• Many valid steps are irrelevant (not useful for proving the hypothesis)
• The model hallucinates invalid steps
Relevance vs. Validity

• Many valid steps are irrelevant (not useful for proving the hypothesis)
• The model hallucinates invalid steps
• Existing methods for stepwise proof generation
  • Struggle to generate valid and relevant steps
  • Underperform on real-world datasets

\[ h: \text{solar is a kind of renewable energy for heating homes} \]

\[ \text{int1: energy is used for heating homes} \]

\[ \text{sent1: homes are buildings} \]
\[ \text{sent5: energy is used for heating buildings} \]
\[ \text{sent4: solar is a kind of energy} \]
\[ \text{sent2: solar is renewable} \]
NLProofS: Natural Language Proof Search

• A new method for stepwise proof generation
• To generate relevant steps: Condition on the hypothesis
• To generate valid steps: Train an independent verifier to prevent hallucination
• Effective in generating challenging, human-authored proofs
Overview of NLProofS
Overview of NLProofS

Prover

Verifier

Generate candidate proof steps

Score the validity

Proof search

Training
Overview of NLProofS

Training

Prover

Verifier

Generate candidate proof steps

Score the validity

Inference

Proof search

Search for the final proof with the maximum score
Overview of NLProofS

Training

Prover

Verifier

Generate candidate proof steps

Score the validity

Inference

Proof search

Search for the final proof with the maximum score
Stepwise Prover

[Raffe et al. JMLR 2020] [Tafjord et al. Findings of ACL 2021]

• Finetune a T5 model to predict the next proof step
Stepwise Prover

- Finetune a T5 model to predict the next proof step

\( h: \) solar is a kind of renewable energy for heating homes

\( \text{sent1}: \) homes are buildings
\( \text{sent2}: \) solar is renewable
\( \text{sent3}: \) wind is a kind of energy
\( \text{sent4}: \) solar is a kind of energy
\( \text{sent5}: \) energy is used for heating buildings
\( \text{sent6}: \) coal is nonrenewable
...

1-step partial proof

\( \text{int1}: \) energy is used for heating homes

\( \text{sent1}: \) homes are buildings
\( \text{sent5}: \) energy is used for heating buildings
Stepwise Prover

[Raffe et al. JMLR 2020] [Tajfjord et al. Findings of ACL 2021]

• Finetune a T5 model to predict the next proof step

$h: \text{solar is a kind of renewable energy for heating homes}$

$sent1: \text{homes are buildings}$
$sent2: \text{solar is renewable}$
$sent3: \text{wind is a kind of energy}$
$sent4: \text{solar is a kind of energy}$
$sent5: \text{solar is a kind of energy}$
$sent6: \text{energy is used for heating buildings}$
$...$

1-step partial proof

$int1: \text{energy is used for heating homes}$

$sent1: \text{homes are buildings}$
$sent5: \text{energy is used for heating buildings}$

$hypothesis = \text{solar is a kind of renewable energy for heating homes}$
$facts = \text{sent1: homes are buildings, sent2: solar is renewable...}$
$\text{partial_proof} = \text{sent1 & sent5 --> int1: energy is used for heating homes}$
Stepwise Prover

• Finetune a T5 model to predict the next proof step

$h$: solar is a kind of renewable energy for heating homes

$\text{sent1}$: homes are buildings  
$\text{sent2}$: solar is renewable  
$\text{sent3}$: wind is a kind of energy  
$\text{sent4}$: solar is a kind of energy  
$\text{sent5}$: energy is used for heating buildings  
$\text{sent6}$: coal is nonrenewable  
...
...

$\text{hypothesis} = \text{solar is a kind of renewable energy for heating homes}$  
$\text{facts} = \text{sent1: homes are buildings sent2: solar is renewable} ...$  
$\text{partial\_proof} = \text{sent1} \& \text{sent5} \rightarrow \text{int1: energy is used for heating homes}$

$\text{int1: energy is used for heating homes}$  
$\text{int2: solar is a kind of renewable energy}$

$\text{sent1: homes are buildings}$  
$\text{sent5: energy is used for heating buildings}$  
$\text{sent4: solar is a kind of energy}$  
$\text{sent2: solar is renewable}$

$\text{T5}$  
$\text{sent2} \& \text{sent4} \rightarrow \text{int2: solar is a kind of renewable energy}$
Stepwise Prover

[Raffle et al. JMLR 2020]

• Finetune a T5 model to predict the next proof step

[Tafjord et al. Findings of ACL 2021]

• Generate multiple candidate steps with beam search
• Filter out syntax errors

\( h: \text{solar is a kind of renewable energy for heating homes} \)

\( \text{sent1: homes are buildings} \)
\( \text{sent2: solar is renewable} \)
\( \text{sent3: wind is a kind of energy} \)
\( \text{sent4: solar is a kind of energy} \)
\( \text{sent5: energy is used for heating buildings} \)
\( \text{sent6: coal is nonrenewable} \)

\( \text{int1: energy is used for heating homes} \)
\( \text{int2: solar is a kind of renewable energy} \)

\( \text{sent1: homes are buildings} \)
\( \text{sent5: energy is used for heating buildings} \)
\( \text{sent4: solar is a kind of energy} \)
\( \text{sent2: solar is renewable} \)

\( h\) = solar is a kind of renewable energy for heating homes
\( \text{facts} = \text{sent1: homes are buildings, sent2: solar is renewable, ...} \)
\( \text{partial proof} = \text{sent1 & sent5 -> int1: energy is used for heating homes} \)

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Overview of NLProofS

Training

Prover

Verifier

Inference

Generate candidate proof steps

Score the validity

Proof search

Search for the final proof with the maximum score
Overview of NLProofS

• How to score a single step?
• How to score the entire proof?

Training

Prover

Verifier

Generate candidate proof steps

Score the validity

Inference

Proof search

Search for the final proof with the maximum score
Verifier: Scoring a Proof Step

- **Input**: a proof step (multiple premises and one conclusion)
- **Output**: a continuous validity score in [0, 1]
Verifier: Scoring a Proof Step

- Input: a proof step (multiple premises and one conclusion)
- Output: a continuous validity score in [0, 1]
- Finetune a RoBERTa model to classify steps as valid/invalid

0.9 (valid)

RoBERTa [Liu et al. arXiv 2019]

homes are buildings. energy is used for heating buildings. energy is used for heating homes.

Premises Conclusion
Verifier: Scoring a Proof Step

- Input: a proof step (multiple premises and one conclusion)
- Output: a continuous validity score in $[0, 1]$
- Finetune a RoBERTa model to classify steps as valid/invalid

0.9 (valid)

RoBERTa

[Liu et al. arXiv 2019]

- homes are buildings.
- energy is used for heating buildings.
- energy is used for heating homes.

- How to generate negative examples (invalid proof steps) for training?
Verifier: Scoring a Proof Step

- Input: a proof step (multiple premises and one conclusion)
- Output: a continuous validity score in $[0, 1]$  
- Finetune a RoBERTa model to classify steps as valid/invalid

$0.9$ (valid)

homes are buildings. energy is used for heating buildings. energy is used for heating homes.

- How to generate negative examples (invalid proof steps) for training?
  - Automatically generate **pseudo-negative examples**, without additional annotation efforts

[Liu et al. arXiv 2019]

[Dalvi et al. arXiv 2022]
Pseudo-Negative Examples for Training the Verifier

- homes are buildings
- energy is used for heating homes
- energy is used for heating buildings
Pseudo-Negative Examples for Training the Verifier

- Replace premises

- **Correct Example:**
  - Energy is used for heating homes
  - Homes are buildings
  - Energy is used for heating buildings

- **Incorrect Example:**
  - Solar is a kind of energy
  - Energy is used for heating buildings

---

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Pseudo-Negative Examples for Training the Verifier

- Replace premises
- Remove premises

- energy is used for heating homes
- homes are buildings
- energy is used for heating buildings

- energy is used for heating homes
- solar is a kind of energy
- energy is used for heating buildings

- energy is used for heating homes
- energy is used for heating buildings
Pseudo-Negative Examples for Training the Verifier

- Replace premises
- Remove premises
- Copy premises

- energy is used for heating homes
- homes are buildings
- energy is used for heating buildings

- energy is used for heating homes
- solar is a kind of energy
- energy is used for heating buildings

- energy is used for heating homes
- energy is used for heating buildings
- solar is a kind of energy
- energy is used for heating buildings

- energy is used for heating homes
- energy is used for heating buildings
- solar is a kind of energy
- energy is used for heating buildings

Pseudo-Negative Examples for Training the Verifier

- Replace premises
- Remove premises
- Copy premises

- energy is used for heating homes
  - homes are buildings
  - energy is used for heating buildings

- energy is used for heating homes
  - energy is used for heating buildings
  - solar is a kind of energy
  - energy is used for heating buildings

- solar is a kind of energy
  - energy is used for heating buildings
  - energy is used for heating buildings
Verifier: Aggregating Scores for the Entire Proof

\[ h: \text{solar is a kind of renewable energy for heating homes} \]

\[ \text{int1: energy is used for heating homes} \]
\[ \text{int2: solar is a kind of renewable energy} \]

\[ \text{sent1: homes are buildings} \]
\[ \text{sent5: energy is used for heating buildings} \]
\[ \text{sent4: solar is a kind of energy} \]
\[ \text{sent2: solar is renewable} \]
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node $u$ with $l$ children $v_1, ..., v_l$

\[
s_{cr}(u) = \min(s_{cr}(s), s_{cr}(v_1), ..., s_{cr}(v_l))
\]

Step score  Children’s node scores

---

$h$: solar is a kind of renewable energy for heating homes

0.85

int1: energy is used for heating homes

0.9

sent1: homes are buildings

1.0

int2: solar is a kind of renewable energy

0.8

sent2: solar is a kind of energy

1.0

sent4: solar is a kind of energy

1.0

sent5: energy is used for heating buildings

1.0

sent2: solar is renewable

1.0
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node \( u \) with \( l \) children \( v_1, \ldots, v_l \)

\[
s_{cr_n}(u) = \min(s_{cr_s}(s), s_{cr_n}(v_1), \ldots, s_{cr_n}(v_l))
\]

Step score  Children’s node scores

\[
s_{cr_i} = 0.9, 0.9, 0.85
\]

\[
s_{cr} = 1.0, 1.0, 1.0, 1.0
\]
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node $u$ with $l$ children $v_1, ..., v_l$

$$\text{scr}_n(u) = \min(\text{scr}_s(s), \text{scr}_n(v_1), ..., \text{scr}_n(v_l))$$

Step score  Children’s node scores

![Diagram of a proof with scores](image)

- **sent1**: homes are buildings
  - **int1**: energy is used for heating homes
  - **h**: solar is a kind of renewable energy for heating homes
    - $0.85$
    - $0.9$

- **sent2**: solar is renewable
  - **int2**: solar is a kind of renewable energy
    - $0.8$

- **sent4**: solar is a kind of energy
  - **sent5**: energy is used for heating buildings
    - $1.0$
  - **sent3**: homes are buildings
    - $1.0$

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node $u$ with $l$ children $v_1, \ldots, v_l$

$$scr_n(u) = \min(scr_s(s), scr_n(v_1), \ldots, scr_n(v_l))$$

**Step score**  
**Children’s node scores**

```
0.85

0.9

0.9

0.8

0.8

h: solar is a kind of renewable energy for heating homes
```

```
0.8

0.9

0.9

0.8

h: solar is a kind of renewable energy for heating homes

0.8

0.9

0.9

0.8

h: solar is a kind of renewable energy for heating homes

```

```
0.9

sent1: homes are buildings

0.9

sent5: energy is used for heating buildings

0.8

sent4: solar is a kind of energy

0.8

sent2: solar is renewable

1.0

1.0

1.0

1.0

1.0

1.0

1.0

1.0

1.0

```
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node $u$ with $l$ children $v_1, ..., v_l$

$$scr_n(u) = \min(scr_s(s), scr_n(v_1), ..., scr_n(v_l))$$

Step score | Children’s node scores

---

0.8

$h$: solar is a kind of renewable energy for heating homes

0.9

$int1$: energy is used for heating homes

0.9

$sent1$: homes are buildings

1.0

0.9

$int2$: solar is a kind of renewable energy

0.8

$sent2$: solar is renewable

1.0

0.8

$sent4$: solar is a kind of energy

1.0

0.8

$sent5$: energy is used for heating buildings

1.0

0.8

$int1$: energy is used for heating homes

0.9

$h$: solar is a kind of renewable energy for heating homes

0.8
Verifier: Aggregating Scores for the Entire Proof

- Leaf nodes have a score of 1
- Internal node $u$ with $l$ children $v_1, ..., v_l$
- The proof score is the root’s node score $\text{scr}_n(h)$

$$\text{scr}_n(u) = \min(\text{scr}_n(s), \text{scr}_n(v_1), ..., \text{scr}_n(v_l))$$

Step score  Children’s node scores

```
0.8
h: solar is a kind of renewable energy for heating homes

0.9
int1: energy is used for heating homes

0.9
sent1: homes are buildings

1.0

0.8
int2: solar is a kind of renewable energy

0.8
sent2: solar is a kind of renewable energy

0.8
sent4: solar is a kind of energy

1.0
sent5: energy is used for heating buildings

1.0
```

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Overview of NLProofS

Training

- Prover
- Verifier

Generate candidate proof steps

Score the validity

Inference

Proof search

Search for the final proof with the maximum score

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
Overview of NLProofS

Training

- Prover
- Verifier

Generate candidate proof steps

Score the validity

Inference

Proof search

Search for the final proof with the maximum score
Proof Search: Overview

• Iteratively expand a graph using proof steps proposed by the prover
• Score the nodes using both prover and the verifier
• Extract the proof with optimal score

**Algorithm 1:** Proof search.

- **Input:** Hypothesis \( h \), supporting facts \( C \), stepwise prover \( \mathcal{P} \), verifier \( V \)
- **Output:** Proof tree \( T \)
Proof Search: Initialization

- Initialize the graph with a greedy proof generated by the prover alone

**Algorithm 1**: Proof search.

**Input**: Hypothesis $h$, supporting facts $C$, stepwise prover $\mathcal{P}$, verifier $\mathcal{V}$

**Output**: Proof tree $T$

1. $\mathcal{G} \leftarrow \text{generate_greedy}(\mathcal{P}, h, C)$
2. $\mathcal{PG} \leftarrow \text{initialize_graph}(\mathcal{G})$
3. explored $\leftarrow \emptyset$
Proof Search: Iteration

• Sample a partial proof from the graph

---

**Algorithm 1: Proof search.**

*Input*: Hypothesis $h$, supporting facts $C$, stepwise prover $\mathcal{P}$, verifier $\mathcal{V}$

*Output*: Proof tree $T$

1. $G \leftarrow \text{generate-greedy}(\mathcal{P}, h, C)$
2. $\mathcal{P}G \leftarrow \text{initialize-graph}(G)$
3. $\text{explored} \leftarrow \emptyset$
4. While true do
   5. $\text{partial_proof} \leftarrow \text{sample-new}(\mathcal{P}G, \text{explored})$
   6. $\text{explored} \leftarrow \text{explored} \cup \{\text{partial_proof}\}$
   7. $\text{steps}, \text{p_scrs} \leftarrow \text{generate}(\mathcal{P}, \text{partial_proof})$
   8. $\text{v_scrs} \leftarrow \text{verify}(\mathcal{V}, \text{steps})$
   9. $\text{scrs} \leftarrow (\text{p_scrs} + \text{v_scrs})/2$
10. $\mathcal{P}G' \leftarrow \text{update}(\mathcal{P}G, \text{steps}, \text{scrs})$
11. If $\mathcal{P}G' = \mathcal{P}G$ then
    12. _break
13. $\mathcal{P}G \leftarrow \mathcal{P}G'$
Proof Search: Iteration

- Use the prover to generate proof steps

Prover score: likelihood scores in beam search

**Algorithm 1**: Proof search.

*Input*: Hypothesis \( h \), supporting facts \( C \), stepwise prover \( \mathcal{P} \), verifier \( \mathcal{V} \)

*Output*: Proof tree \( T \)

1. \( G \leftarrow \text{generate\_greedy}(\mathcal{P}, h, C) \)
2. \( \mathcal{P}G \leftarrow \text{initialize\_graph}(G) \)
3. explored \( \leftarrow \emptyset \)

4. while true do

5. \( \text{partial\_proof} \leftarrow \text{sample\_new}(\mathcal{P}G, \text{explored}) \)

6. explored \( \leftarrow \text{explored} \cup \{\text{partial\_proof}\} \)

7. steps, p_scrs \( \leftarrow \text{generate}(\mathcal{P}, \text{partial\_proof}) \)

8. v_scrs \( \leftarrow \text{verify}(\mathcal{V}, \text{steps}) \)

9. scrs \( \leftarrow (p_scrs + v_scrs)/2 \)

10. \( \mathcal{P}G' \leftarrow \text{update}(\mathcal{P}G, \text{steps}, \text{scrs}) \)

11. if \( \mathcal{P}G' = \mathcal{P}G \) then

12. \( \_ \_ \_ \text{break} \)

13. \( \mathcal{P}G \leftarrow \mathcal{P}G' \)
Proof Search: Iteration

• Use the prover to generate proof steps

Prover score: likelihood scores in beam search

Step score: \((\text{prover score} + \text{verifier score}) / 2\)

---

**Algorithm 1: Proof search.**

**Input**: Hypothesis \(h\), supporting facts \(C\), stepwise prover \(P\), verifier \(V\)

**Output**: Proof tree \(T\)

1. \(G \leftarrow \text{generate_greedy}(P, h, C)\)
2. \(PG \leftarrow \text{initialize_graph}(G)\)
3. \(\text{explored} \leftarrow \emptyset\)

4. while true do

5. \(\text{partial_proof} \leftarrow \text{sample_new}(PG, \text{explored})\)

6. \(\text{explored} \leftarrow \text{explored} \cup \{\text{partial_proof}\}\)

7. \(\text{steps, p_scrs} \leftarrow \text{generate}(P, \text{partial_proof})\)

8. \(v_scrs \leftarrow \text{verify}(V, \text{steps})\)

9. \(\text{scrs} \leftarrow (\text{p_scrs} + v_scrs) / 2\)

10. \(PG' \leftarrow \text{update}(PG, \text{steps, scrs})\)

11. if \(PG' = PG\) then

12. break

13. \(PG \leftarrow PG'\)
Proof Search: Iteration

• Iteratively expand the graph

Prover score: likelihood scores in beam search
Step score: (prover score + verifier score) / 2

```
Algorithm 1: Proof search.

Input: Hypothesis h, supporting facts C, stepwise prover P, verifier V

Output: Proof tree T
1. G ← generate_greedy(P, h, C)
2. PG ← initialize_graph(G)
3. explored ← Ø

4. while true do
   5. partial_proof ← sample_new(PG, explored)
   6. explored ← explored ∪ {partial_proof}
   7. steps, p_scrs ← generate(P, partial_proof)
   8. v_scrs ← verify(V, steps)
   9. scrs ← (p_scrs + v_scrs)/2
10. PG' ← update(PG, steps, scrs)
11. if PG' = PG then
    12. break
13. PG ← PG'
```

Proof Search: Iteration

- Iteratively expand the graph

Prover score: likelihood scores in beam search
Step score: (prover score + verifier score) / 2

**Algorithm 1**: Proof search.

**Input**: Hypothesis $h$, supporting facts $C$, stepwise prover $P$, verifier $V$

**Output**: Proof tree $T$

1. $G \leftarrow$ generate_greedy($P, h, C$)
2. $PG \leftarrow$ initialize_graph($G$)
3. explored $\leftarrow \emptyset$
4. while true do
   5. partial_proof $\leftarrow$ sample_new($PG$, explored)
   6. explored $\leftarrow$ explored $\cup \{\text{partial_proof}\}$
   7. steps, p_scrs $\leftarrow$ generate($P$, partial_proof)
   8. v_scrs $\leftarrow$ verify($V$, steps)
   9. scrs $\leftarrow$ $(p$-scr$s + v$-scr$s)/2$
10. $PG' \leftarrow$ update($PG$, steps, scrs)
11. if $PG' = PG$ then
   12. break
13. $PG \leftarrow PG'$
Proof Search: Iteration

- Iteratively expand the graph

Prover score: likelihood scores in beam search
Step score: (prover score + verifier score) / 2

---

Algorithm 1: Proof search.

**Input**: Hypothesis $h$, supporting facts $C$, stepwise prover $P$, verifier $V$

**Output**: Proof tree $T$

1. $G \leftarrow$ generate_greedy($P$, $h$, $C$)
2. $PG \leftarrow$ initialize_graph($G$)
3. explored $\leftarrow \emptyset$
4. while true do
   5. partial_proof $\leftarrow$ sample_new($PG$, explored)
   6. explored $\leftarrow$ explored $\cup$ {partial_proof}
   7. steps, p_scrs $\leftarrow$ generate($P$, partial_proof)
   8. v_scrs $\leftarrow$ verify($V$, steps)
   9. scrs $\leftarrow$ (p_scrs + v_scrs) / 2
   10. $PG' \leftarrow$ update($PG$, steps, scrs)
   11. if $PG'$ = $PG$ then
      12. break
   13. $PG \leftarrow PG'$
Proof Search: Iteration

* Iteratively expand the graph

Prover score: likelihood scores in beam search
Step score: (prover score + verifier score) / 2

Algorithm 1: Proof search.

<table>
<thead>
<tr>
<th>Input</th>
<th>: Hypothesis h, supporting facts C, stepwise prover P, verifier V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>: Proof tree T</td>
</tr>
</tbody>
</table>

1. $G \leftarrow$ generate_greedy($P, h, C$)
2. $PG \leftarrow$ initialize_graph($G$)
3. explored $\leftarrow$ $\varnothing$
4. while true do
   5. partial_proof $\leftarrow$ sample_new($PG$, explored)
   6. explored $\leftarrow$ explored $\cup$ {partial_proof}
   7. steps, p_scrs $\leftarrow$ generate($P$, partial_proof)
   8. v_scrs $\leftarrow$ verify($V$, steps)
   9. scrs $\leftarrow$ ($p$_scrs + v_scrs)/2
10. $PG' \leftarrow$ update($PG$, steps, scrs)
11. if $PG' = PG$ then
    12. break
13. $PG \leftarrow PG'$
Proof Search: Iteration

- Iteratively expand the graph

Prover score: likelihood scores in beam search
Step score: (prover score + verifier score) / 2
Proof Search: Extraction

- Extract a proof tree with optimal score

Algorithm 1: Proof search.

Input: Hypothesis $h$, supporting facts $C$, stepwise prover $P$, verifier $V$

Output: Proof tree $T$

1. $G \leftarrow$ generate_greedy($P, h, C$)
2. $PG \leftarrow$ initialize_graph($G$)
3. explored $\leftarrow \emptyset$
4. while true do
   5. partial_proof $\leftarrow$ sample_new($PG$, explored)
   6. explored $\leftarrow$ explored $\cup$ {partial_proof}
   7. steps, p_scrs $\leftarrow$ generate($P$, partial_proof)
   8. v_scrs $\leftarrow$ verify($V$, steps)
   9. scrs $\leftarrow$ ($p$_scr$ + v$_scr$)/2$
   10. $PG' \leftarrow$ update($PG$, steps, scrs)
   11. if $PG' = PG$ then
       12. break
   13. $PG \leftarrow PG'$
14. return extract_proof($PG$)
Overview of NLProofS

- **Training**
  - Prover
  - Verifier

- **Inference**
  - Generate candidate proof steps
  - Score the validity
  - Proof search
    - Search for the final proof with the maximum score
Experiments
Experiments Overview

• Evaluate on two benchmarks
  • **RuleTaker**: Simple, synthetic proofs  [Dalvi et al. EMNLP 2021]
• State-of-the-art results on both
• Ablations highlight the importance of the verifier
Experiments Overview

• Evaluate on two benchmarks
  • **RuleTaker**: Simple, synthetic proofs [Dalvi et al. EMNLP 2021]
• State-of-the-art results on both
• Ablations highlight the importance of the verifier
EntailmentBank

• 1,840 proof trees constructed by experts: 1,313/187/340 for training/validation/testing
• How to compare a predicted proof tree against the ground truth?
  • Four official evaluation: Leaves, Steps, Intermediates, Overall
EntailmentBank: Evaluation Metrics

Predicted proof tree

- **sent1**: homes are buildings
- **sent5**: energy is used for heating buildings
- **sent4**: solar is a kind of energy

**h**: solar is a kind of renewable energy for heating homes

**int1**: energy can heat homes

Ground truth proof tree

- **sent1**: homes are buildings
- **sent5**: energy is used for heating buildings
- **sent4**: solar is a kind of energy

**h**: solar is a kind of renewable energy for heating homes

**int1**: energy is used for heating homes

**int2**: solar is a kind of renewable energy

[Dalvi et al. EMNLP 2021]
EntailmentBank: Evaluation Metrics

Predicted proof tree

Ground truth proof tree

h: solar is a kind of renewable energy for heating homes

int1: energy can heat homes

sent1: homes are buildings

sent5: energy is used for heating buildings

sent4: solar is a kind of energy

h: solar is a kind of renewable energy for heating homes

int1: energy is used for heating homes

sent1: homes are buildings

sent5: energy is used for heating buildings

sent4: solar is a kind of energy

sent2: solar is renewable

Sentences:
- homes are buildings
- solar is a kind of renewable energy for heating homes
- energy is used for heating homes
- solar is a kind of energy
- solar is renewable

Interventions:
- energy can heat homes
- energy is used for heating homes

[Dalvi et al. EMNLP 2021]
EntailmentBank: Evaluation Metrics

- Leaves: Accuracy and F1 score of the predicted leaf nodes

\[
\text{accuracy} = 0 \quad \text{precision} = \frac{3}{3} = 1 \quad \text{recall} = \frac{3}{4} = 0.75 \quad F1 = 1/ \frac{1 + \frac{4}{3}}{2} \approx 0.86
\]
EntailmentBank: Evaluation Metrics

- **Leaves**: Accuracy and F1 score of the predicted leaf nodes
- **Steps**: Proof steps are "structurally correct"

[Dalvi et al. EMNLP 2021]
EntailmentBank: Evaluation Metrics

- **Leaves**: Accuracy and F1 score of the predicted leaf nodes
- **Steps**: Proof steps are “structurally correct”
- **Intermediates**: Intermediate conclusions match, judged by the BLEURT score

[Dalvi et al. EMNLP 2021]

[Sellam et al. ACL 2020]
EntailmentBank: Evaluation Metrics

[Dalvi et al. EMNLP 2021]

- Leaves: Accuracy and F1 score of the predicted leaf nodes
- Steps: Proof steps are “structurally correct”
- Intermediates: Intermediate conclusions match, judged by the BLEURT score
- **Overall: The entire proof is correct**

[Sellam et al. ACL 2020]
EntailmentBank: Evaluation Metrics

Predicted proof tree

1. **h**: solar is a kind of renewable energy for heating homes
2. **int1**: energy can heat homes
3. **sent1**: homes are buildings
4. **sent4**: solar is a kind of energy

Ground truth proof tree

1. **h**: solar is a kind of renewable energy for heating homes
2. **int1**: energy is used for heating homes
3. **sent1**: homes are buildings
4. **sent5**: energy is used for heating buildings
5. **sent4**: solar is a kind of energy
6. **sent2**: solar is renewable

- Limitations: Does not handle correct predictions that are different from the ground truth
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

- **EntailmentWriter** [Dalvi et al. EMNLP 2021]
- **IGRG (concurrent)** [Ribeiro et al. NAACL 2022]
- **MetGen (concurrent)** [Hong et al. NAACL 2022]
- **NLProofS (ours)**

<table>
<thead>
<tr>
<th>Component</th>
<th>EntailmentWriter</th>
<th>IGRG (concurrent)</th>
<th>MetGen (concurrent)</th>
<th>NLProofS (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves</td>
<td>35.6</td>
<td>48.6</td>
<td>22.9</td>
<td>58.8</td>
</tr>
<tr>
<td>Steps</td>
<td>22.9</td>
<td>22.3</td>
<td>30.4</td>
<td>34.4</td>
</tr>
<tr>
<td>Intermediates</td>
<td>28.5</td>
<td>26.5</td>
<td>32.7</td>
<td>37.8</td>
</tr>
<tr>
<td>Overall</td>
<td>20.9</td>
<td>22</td>
<td>28</td>
<td>33.3</td>
</tr>
</tbody>
</table>

[Dalvi et al. EMNLP 2021]
[Ribeiro et al. NAACL 2022]
[Hong et al. NAACL 2022]
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

- **EntailmentWriter** [Dalvi et al. EMNLP 2021]
- **IGRG (concurrent)** [Ribeiro et al. NAACL 2022]
- **MetGen (concurrent)** [Hong et al. NAACL 2022]
- **NLProofS (ours)**

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntailmentWriter</td>
<td>35.6</td>
<td>22.9</td>
<td>28.5</td>
<td>20.9</td>
</tr>
<tr>
<td>IGRG (concurrent)</td>
<td>48.6</td>
<td>30.4</td>
<td>32.7</td>
<td>22</td>
</tr>
<tr>
<td>MetGen (concurrent)</td>
<td>58.8</td>
<td>34.4</td>
<td>37.8</td>
<td>28</td>
</tr>
<tr>
<td>NLProofS (ours)</td>
<td>46.8</td>
<td>26.5</td>
<td>33.3</td>
<td>28</td>
</tr>
</tbody>
</table>

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntailmentWriter</td>
<td>35.6</td>
<td>22.9</td>
<td>28.5</td>
<td>20.9</td>
</tr>
<tr>
<td>IGRG (concurrent)</td>
<td>48.6</td>
<td>22.3</td>
<td>32.7</td>
<td>22.0</td>
</tr>
<tr>
<td>MetGen (concurrent)</td>
<td>30.4</td>
<td>34.4</td>
<td>37.8</td>
<td>33.3</td>
</tr>
<tr>
<td>NLProofS (ours)</td>
<td>58.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Dalvi et al. EMNLP 2021] [Ribeiro et al. NAACL 2022] [Hong et al. NAACL 2022]
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

- **EntailmentWriter** [Dalvi et al. EMNLP 2021]
- **IGRG (concurrent)** [Ribeiro et al. NAACL 2022]
- **MetGen (concurrent)** [Hong et al. NAACL 2022]
- **NLProofS (ours)**

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntailmentWriter</td>
<td>35.6</td>
<td>22.9</td>
<td>28.5</td>
<td>20.9</td>
</tr>
<tr>
<td>IGRG (concurrent)</td>
<td>48.6</td>
<td>22.3</td>
<td>32.7</td>
<td>22</td>
</tr>
<tr>
<td>MetGen (concurrent)</td>
<td>58.8</td>
<td>30.4</td>
<td>37.8</td>
<td>33.3</td>
</tr>
<tr>
<td>NLProofS (ours)</td>
<td>30.4</td>
<td>26.5</td>
<td>34.4</td>
<td>33.3</td>
</tr>
</tbody>
</table>
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntailmentWriter</td>
<td>35.6</td>
<td>22.9</td>
<td>22.9</td>
<td>20.9</td>
</tr>
<tr>
<td>IGRG (concurrent)</td>
<td>48.6</td>
<td>22.3</td>
<td>30.4</td>
<td>32.7</td>
</tr>
<tr>
<td>MetGen (concurrent)</td>
<td>58.8</td>
<td>34.4</td>
<td>28.5</td>
<td>37.8</td>
</tr>
<tr>
<td>NLProofS (ours)</td>
<td>32.7</td>
<td>26.5</td>
<td>28</td>
<td>33.3</td>
</tr>
</tbody>
</table>

[Dalvi et al. EMNLP 2021] [Ribeiro et al. NAACL 2022] [Hong et al. NAACL 2022]
## State-of-the-art Performance on EntailmentBank

### Accuracies on Task 2 of EntailmentBank

<table>
<thead>
<tr>
<th></th>
<th>EntailmentWriter</th>
<th>IGRG (concurrent)</th>
<th>MetGen (concurrent)</th>
<th>NLProofS (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves</td>
<td>35.6</td>
<td>48.6</td>
<td>30.4</td>
<td>23.2</td>
</tr>
<tr>
<td>Steps</td>
<td>22.9</td>
<td>22.9</td>
<td>28.5</td>
<td>20.9</td>
</tr>
<tr>
<td>Intermediates</td>
<td>34.4</td>
<td>26.5</td>
<td>32.7</td>
<td>22</td>
</tr>
<tr>
<td>Overall</td>
<td>37.8</td>
<td>37.8</td>
<td>33.3</td>
<td>33.3</td>
</tr>
</tbody>
</table>

*Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen*
State-of-the-art Performance on EntailmentBank

Accuracies on Task 2 of EntailmentBank

- EntailmentWriter [Dalvi et al. EMNLP 2021]
- IGRG (concurrent) [Ribeiro et al. NAACL 2022]
- MetGen (concurrent) [Hong et al. NAACL 2022]
- NLProofS (ours)

- Leaves: 35.6, 22.9, 48.6, 58.8, 10.2
- Steps: 30.4, 22.9, 22.3, 34.4, 4.0
- Intermediates: 28.5, 26.5, 32.7, 37.8, 5.1
- Overall: 20.9, 22, 28, 33.3, 5.3
Ablations

Accuracies on Task 2 of EntailmentBank

- NLProofS (full)
- w/o search
- w/o search w/o stepwise
- w/o verifier score

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o search</td>
<td>58.8</td>
<td>34.4</td>
<td>37.8</td>
<td>33.3</td>
</tr>
<tr>
<td>w/o verifier score</td>
<td>56.5</td>
<td>33.7</td>
<td>36.4</td>
<td>31.8</td>
</tr>
<tr>
<td>w/o stepwise</td>
<td>55.8</td>
<td>29.7</td>
<td>32.2</td>
<td>27.1</td>
</tr>
<tr>
<td>Overall</td>
<td>55.8</td>
<td>33.8</td>
<td>36.1</td>
<td>31.9</td>
</tr>
</tbody>
</table>
Ablations

Accuracies on Task 2 of EntailmentBank

- **NLProofS (full)**
- w/o search
- w/o search w/o stepwise
- w/o verifier score

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLProofS (full)</td>
<td>58.8</td>
<td>34.4</td>
<td>37.8</td>
<td>33.3</td>
</tr>
<tr>
<td>w/o search</td>
<td>56.5</td>
<td>33.7</td>
<td>36.4</td>
<td>31.8</td>
</tr>
<tr>
<td>w/o search w/o stepwise</td>
<td>55.8</td>
<td>29.7</td>
<td>32.2</td>
<td>27.1</td>
</tr>
<tr>
<td>w/o verifier score</td>
<td>45.6</td>
<td>33.8</td>
<td>36.1</td>
<td>31.9</td>
</tr>
</tbody>
</table>
Verifier-Guided Proof Search Is Helpful

Accuracies on Task 2 of EntailmentBank

- NLProofS (full)
- w/o search
- w/o search w/o stepwise
- w/o verifier score
Stepwise Generation Is Helpful

Accuracies on Task 2 of EntailmentBank

<table>
<thead>
<tr>
<th></th>
<th>NLProofS (full)</th>
<th>w/o search</th>
<th>w/o search w/o stepwise</th>
<th>w/o verifier score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaves</td>
<td>58.8</td>
<td>56.5</td>
<td>55.8</td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td>34.4</td>
<td>33.7</td>
<td>29.7</td>
<td>32.2</td>
</tr>
<tr>
<td>Intermediates</td>
<td>37.8</td>
<td>36.4</td>
<td>36.1</td>
<td>31.8</td>
</tr>
<tr>
<td>Overall</td>
<td>33.3</td>
<td>31.8</td>
<td>27.1</td>
<td>31.9</td>
</tr>
</tbody>
</table>

Generating Natural Language Proofs with Verifier-Guided Search - Kaiyu Yang, Jia Deng, and Danqi Chen
The Verifier Is Necessary for Proof Search

Accuracies on Task 2 of EntailmentBank

<table>
<thead>
<tr>
<th></th>
<th>Leaves</th>
<th>Steps</th>
<th>Intermediates</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLProofS (full)</td>
<td>58.8</td>
<td>34.4</td>
<td>37.8</td>
<td>33.3</td>
</tr>
<tr>
<td>w/o search</td>
<td>56.5</td>
<td>33.7</td>
<td>36.4</td>
<td>31.8</td>
</tr>
<tr>
<td>w/o search w/o stepwise</td>
<td>45.6</td>
<td>29.7</td>
<td>32.2</td>
<td>27.1</td>
</tr>
<tr>
<td>w/o verifier score</td>
<td>55.8</td>
<td>33.8</td>
<td>36.1</td>
<td>31.9</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
</table>
| 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**: 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. |
## Examples

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next new moon will occur on June 30.</td>
<td>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.</td>
<td><strong>EntailmentWriter</strong>: The next new moon will occur 28 days after June 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NLProofS (ours)</strong>: The next new moon will occur 28 days after the last new moon.</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The next new moon will occur on June 30.</strong></td>
<td><strong>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.</strong></td>
<td><strong>EntailmentWriter</strong>: The next new moon will occur 28 days after June 2. <strong>NLProofS (ours)</strong>: The next new moon will occur 28 days after the last new moon.</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next new moon will occur on June 30.</td>
<td>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.</td>
<td><strong>EntailmentWriter</strong>: The next new moon will occur 28 days after June 2. <strong>NLProofS (ours)</strong>: The next new moon will occur 28 days after the last new moon.</td>
</tr>
</tbody>
</table>
Examples: Hallucination

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
</table>
| 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**: The next new moon will occur 28 days after June 2.  
**NLPof**S **(ours)**: The next new moon will occur 28 days after the last new moon. |
### Examples: Copying from Premises

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Premises</th>
<th>Conclusions generated by the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next new moon will occur on June 30.</td>
<td>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.</td>
<td><strong>EntailmentWriter</strong>: The next new moon will occur 28 days after June 2.</td>
</tr>
<tr>
<td>Planting trees prevents soil from washing away.</td>
<td>1. <strong>Planting trees increases the amount of trees in an environment.</strong> 2. Tree roots decrease / reduce soil erosion.</td>
<td><strong>EntailmentWriter</strong>: Plants trees increases the amount of trees in an environment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NLP proofs (ours)</strong>: Planting trees decreases soil erosion.</td>
</tr>
</tbody>
</table>
Conclusion and Future Work
Key Takeaways

• The verifier is important
  • Prevent hallucinated generations
Key Takeaways

• The verifier is important
  • Prevent hallucinated generations
  • Also explored in other contexts, e.g., math word problems, code generation

Key Takeaways

• The verifier is important
  • Prevent hallucinated generations
  • Also explored in other contexts, e.g., math word problems, code generation


• Mitigate the problem that language models do not “understand” their own generations

  [Austin and Odena et al. "Program Synthesis with Large Language Models"]
Limitations and Future Directions

Input

- **h**: solar is a kind of renewable energy for heating homes
- **sent1**: homes are buildings
- **sent2**: solar is renewable
- **sent3**: wind is a kind of energy
- **sent4**: solar is a kind of energy
- **sent5**: energy is used for heating buildings
- **sent6**: coal is nonrenewable

...
Limitations and Future Directions

Input

$h$: solar is a kind of renewable energy for heating homes

$sent1$: homes are buildings
$sent2$: solar is renewable
$sent3$: wind is a kind of energy
$sent4$: solar is a kind of energy
$sent5$: energy is used for heating buildings
$sent6$: coal is nonrenewable
...

Concatenate

$hypothesis$ = solar is a kind of renewable energy for heating homes;
$facts$ = sent1: homes are buildings sent2: solar is renewable ...;
$partial_proof$ = ...
Limitations and Future Directions

Input

- $hypothesis$ = solar is a kind of renewable energy for heating homes
- $facts$ = sent1: homes are buildings
  - sent2: solar is renewable
  - sent3: wind is a kind of energy
  - sent4: solar is a kind of energy
  - sent5: energy is used for heating buildings
  - sent6: coal is nonrenewable
  ...

$partial\_proof$ = ...

\[ Concatenate \]

- Scale poorly w.r.t. the number of supporting facts
Limitations and Future Directions

Input

$h$: solar is a kind of renewable energy for heating homes

$\text{sent1}$: homes are buildings
$\text{sent2}$: solar is renewable
$\text{sent3}$: wind is a kind of energy
$\text{sent4}$: solar is a kind of energy
$\text{sent5}$: energy is used for heating buildings
$\text{sent6}$: coal is nonrenewable
...

Concatenate

$h$: solar is a kind of renewable energy for heating homes
$\text{sent1}$: homes are buildings
$\text{sent2}$: solar is renewable
$\text{sent3}$: wind is a kind of energy
$\text{sent4}$: solar is a kind of energy
$\text{sent5}$: energy is used for heating buildings
$\text{sent6}$: coal is nonrenewable
...

$\text{hypothesis} = \text{solar is a kind of renewable energy for heating homes}$;
$\text{facts} = \text{sent1: homes are buildings sent2: solar is renewable } ...$
$\text{partial proof} = ...$

• Scale poorly w.r.t. the number of supporting facts
• Practically applications may have thousands or millions of supporting facts

• **Retrievers or memory-augmented transformers may help**

[Raspukhin et al. EMNLP 2020]
[Ribeiro et al. NAACL 2022]
[Wu et al. ICLR 2022]
Generating Natural Language Proofs with Verifier-Guided Search

Kaiyu Yang, Jia Deng, and Danqi Chen