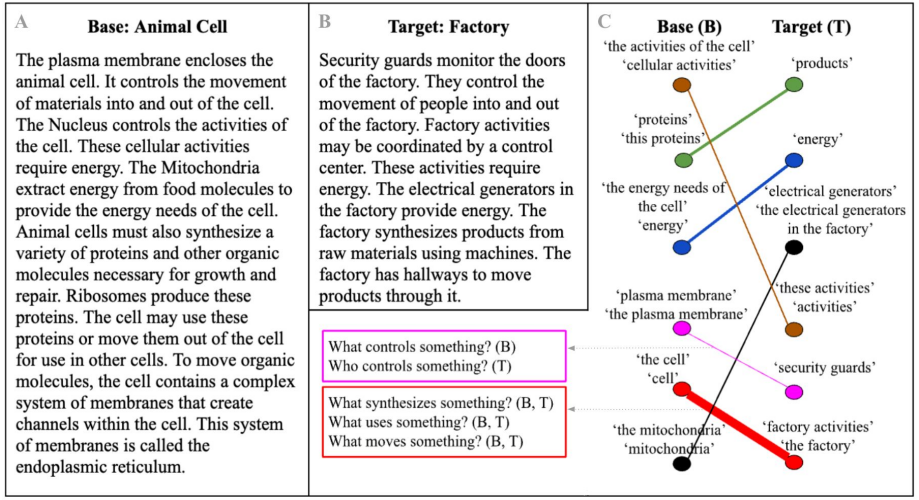
1) Analogical Reasoning over Natural Language Paragraphs (2022-2023)

EMNLP 2022

THE HEBREW UNIVERSITY OF JERUSALEM

Life is a Circus and We are the Clowns: Automatically Finding Analogies between Situations and Processes

Oren Sultan, Dafna Shahaf

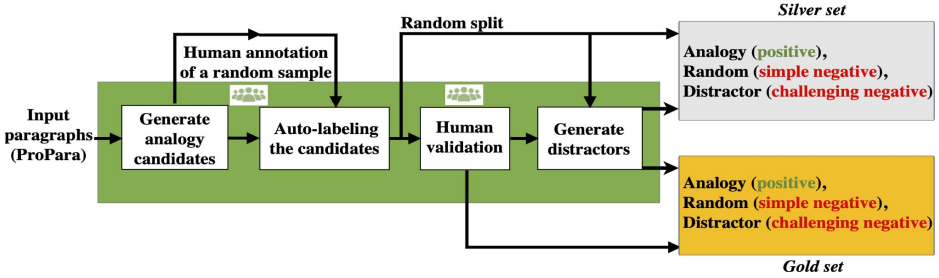


NAACL 2024

THE HEBREW UNIVERSITY OF JERUSALEM

ParallelPARC: A Scalable Pipeline for Generating Natural-Language Analogies

Oren Sultan, Yonatan Bitton, Ron Yosef, Dafna Shahaf



Base	Target	Similar Relations
Title: How does a solar panel work? **Domain**: Engineering **Paragraph**: **solar energy** *powers* an **electric current** within a solar panel. The **photovoltaic cells** within the panel *convert* the **energy** from the sun into electricity. The **electrical wires** then *spread* this **power** throughout the panel. The **electric current** is then used to *power* whatever the panel is connected to.	**Title**: How does photosynthesis occur? **Domain**: Natural Science **Paragraph**: Photosynthesis occurs when **sunlight** *powers* **chemical reactions** within the **chloroplasts** of a plant. The **chloroplasts** are able to *transform* the energy from the **sunlight** into usable energy for the **plant**. This energy is then used to produce **nutrients** for the plant, which are then *distributed* throughout the **plant**.	(**solar energy**, *powers*, **electric current**) (**sunlight**, *powers*, **chemical reactions**) (**photovoltaic cells**, *convert*, **energy**) (**chloroplasts**, *tranform*, **energy**) (**electrical wires**, *spread*, **power**) (**plants**, *distribute*, **nutrients**)

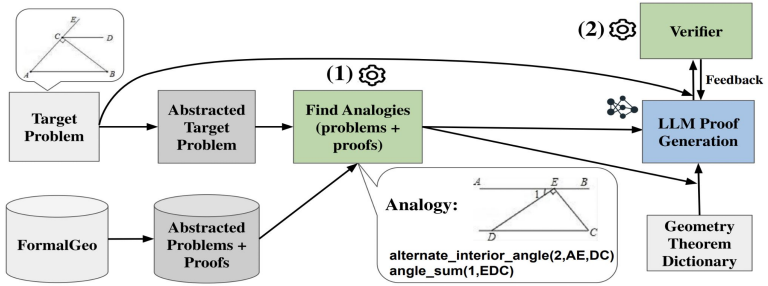
2) Combining LLMs with tools – A neuro-symbolic approach (2024-2025)

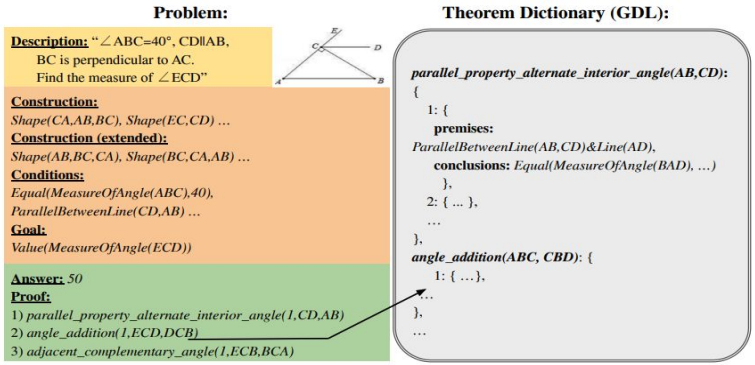
ACL 2026
submission

THE HEBREW UNIVERSITY OF JERUSALEM

Towards Reliable Proof Generation with LLMs: A Neuro-Symbolic Approach

Oren Sultan, Eitan Stern, Dafna Shahaf



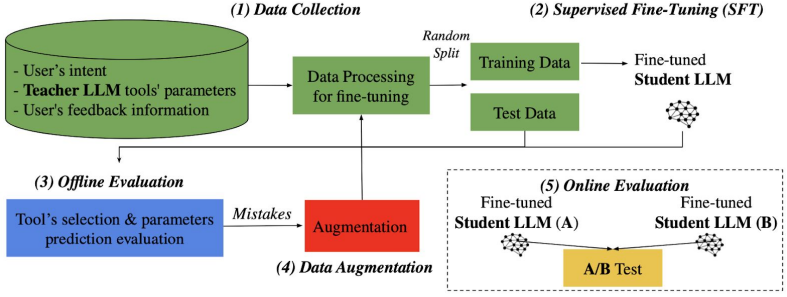


2) Combining LLMs with tools – A neuro-symbolic approach (2024)

EMNLP 2024 Lightricks

🎥🔧🔗 Visual Editing with LLM-based Tool Chaining: An Efficient Distillation Approach for Real-Time Applications

Oren Sultan, Alex Khasin, Guy Shiran, Asnat Greenstein-Messica, Dafna Shahaf



Adjust

```
{
 "exposure": 0,
 "contrast": 10,
 "brightness": 10,
 "highlights": 20,
 "shadows": -10,
 "saturation": 15,
 "vibrance": 15,
 "temperature": 30,
 "tint": 10,
 "hue": 0,
 "bloom": 0,
 "sharpen": 0,
 "structure": 0,
 "linearOffset": 0
}
```

Selective adjust

```
{
 "red": {"saturation": 20, "luminance": 10},
 "orange": {"saturation": 30, "luminance": 20},
 "yellow": {"saturation": 40, "luminance": 30},
 "green": {"saturation": -20, "luminance": 0},
 "cyan": {"saturation": -20, "luminance": 0},
 "blue": {"saturation": 0, "luminance": 0}
}
```

Filter

```
{
 "name": "faded_HighNoon",
 "intensity": 40
}
```

"Golden hour"

3) Code World Models (CWM) and the Halting Problem (2025)

CWM: An Open-Weights LLM for Research on Code Generation with World Models

Meta FAIR CodeGen Team, **Oren Sultan**

<|trace_context_start|>

```
def count(s, t):
    n = 0
    for c in s:
        n += int(c == t)
    return n


count("strawberry", "r")  # << START_OF_TRACE
```

<|frame_sep|>

<|call_sep|>	{"s": "'strawberry'", "t": "'r'"}	<|action_sep|>	def count(s, t):
<|frame_sep|>			
<|line_sep|>	{"s": "..", "t": ".."}	<|action_sep|>	n = 0
<|frame_sep|>			
<|line_sep|>	{"s": "..", "t": "..", "n": "0"}	<|action_sep|>	for c in s:
<|frame_sep|>			
<|line_sep|>	{"s": "..", "t": "..", "n": "..", "c": "'s'"}	<|action_sep|>	n += int(c == t)
...			
<|frame_sep|>			
<|return_sep|>	<|action_sep|> return n	<|arg_sep|>	"3"
<|frame_sep|>			

ACL 2026 submission

LLMs versus the Halting Problem: Revisiting Program Termination Prediction

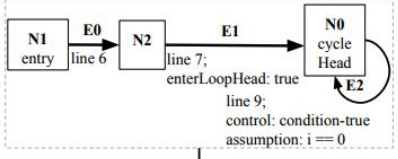
Oren Sultan, Jordi Armengol-Estapé, Pascal Kesseli, Julien Vanegue, Dafna Shahaf, Peter O'Hearn, Yossi Adi

C code example:

```
1:  typedef enum {false,true} bool;
2:
3:  extern int __VERIFIER_nondet_int(void);
4:
5:  int main() {
6:  int i;
7:  i = __VERIFIER_nondet_int();
8:
9:  while (i >= -5 && i <= 5) {
10:   if (i > 0) {
11:     i = i-1;
12:   }
13:   if (i < 0) {
14:     i = i+1;
15:   }
16: }
17:
18:  return 0;
19: }
```

Verdict: Non-termination

LLM witness prediction:



UAutomizer (Witness Validator)