Grounding Language Models to Real-World Environments

Yu Su
The Ohio State University

Slides credit to my amazing student Yu Gu
Language Models Nail Everything?

Natural language as a universal device for human-machine interaction?
Language Models: What’s Missing?

- **Cannot take actions in the real world**
  
  **su** Book a meeting with Tommy tomorrow at 3 pm

  I’m sorry, but I am a text-based AI assistant and don’t have the ability to book meetings. I’m not connected to the internet, so I don’t have access to your calendar or any other information that would be necessary to book a meeting. I can provide general information and answer questions, but I’m not able to perform tasks like booking meetings. Is there something else I can help you with?

- **Cannot handle more specific information needs**
  
  **su** Marc Bulger had the most yards rushing in what season?

  As a text-based AI assistant, I don’t have access to specific information about individuals or events. I don’t have the ability to browse the internet or access other sources of information, so I can’t provide you with specific answers to questions like this. I’m trained to provide general information and answer questions to the best of my ability based on the information I have been trained on, but I don’t have the ability to look up specific facts or provide detailed information about specific individuals or events. Is there something else I can help you with?
Grounded Language Understanding: What and Why?
Grounded Language Understanding: Formal Definition

Given a natural language utterance \( u \) and a target environment \( E \)

\[
\pi : (u, E) \rightarrow p, \ s.t. \ [u]_E = [p]_E
\]

Where \( p \) is a plan/program in a formal language, and \([\cdot]_E\) is the denotation
Grounded Language Understanding: Formal Definition

Given a natural language utterance $u$ and a target environment $E$

$$\pi: (u, E) \rightarrow p, \text{ s.t. } \mathcal{[u]}_E = \mathcal{[p]}_E$$

Where $p$ is a plan/program in a formal language, and $\mathcal{[\cdot]}_E$ is the denotation

$u$: What is the latest released computer emulator developed in Java?

$p$: (ARGMAX (AND ComputerEmulator (JOIN LanguagesUsed Java) LatestReleaseDate))
Grounded Language Understanding: Formal Definition

Given a natural language utterance $u$ and a target environment $E$

$$\pi: (u, E) \rightarrow p, \text{ s.t. } \llbracket u \rrbracket_E = \llbracket p \rrbracket_E$$

Where $p$ is a plan/program in a formal language, and $\llbracket \cdot \rrbracket_E$ is the denotation

$u$: Bring me a cup of coffee

$p$: [turn left, move forward, pick up cup, turn around, move forward, …, put cup in coffee maker, toggle coffee maker, …]
The Symbol Grounding Problem

Language models are mostly trained with textual corpora

- **BERT**: Wikipedia (2.5B words) + BookCorpus (800M words)
- **T5**: C4 (two orders of magnitude larger)
- **GPT-3**: 45TB text data + others

Key challenge: How to ground textual symbols to different environments/formal languages
Pangu: A Unified Framework for Grounded Language Understanding

Yu Gu, Xiang Deng, Yu Su

The Ohio State University
QUIZ TIME!
Q1 Find the right program over a KB

**Question:** Who has ever coached an ice hockey team in Canada?

**Program:**

A. \[(\text{AND} \text{cricket.cricket_coach} (\text{JOIN} \text{cricket.cricket_team.coach_inv} (\text{JOIN} \text{sports.sports_team.location Canada})))\]

B. \[(\text{AND} \text{ice_hockey.hockey_coach} (\text{JOIN} \text{ice_hockey.hockey_team.coach_inv} (\text{JOIN} \text{sports.sports_team.location Canada})))\]

C. \[(\text{AND} \text{ice_hockey.hockey_team} (\text{JOIN} \text{sports.sports_team.location Canada})))\]

- **Correct**: B
- **Incorrect**: A, C
Q2 Write the corresponding KB program

**Question:** What's the classification of the M10 engine?

**Program:**

\[(\text{AND} \ \text{automotive.engine_type} (\text{JOIN} \ \text{automotive.engine_type.used_in} \ \text{M10}))\]
Why is Q2 harder?

1. You need to learn the grammar
2. You need to know the environment specifics
Directly generating plans (programs) may not be the optimal way of using LMs for grounded language understanding.
Pangu:

A unified framework that models grounded language understanding as a discrimination task
The Status Quo

Autoregressive generation with Seq2Seq LMs

Text-to-SQL Parsing
(Wang et al. 2020)

KBQA
(Shu et al. 2022)

Embodied AI
(Shridhar et al. 2019)
The Status Quo

Autoregressive generation can produce invalid plans

Percentage of executable and valid programs for KBQA (Ye et al. 2021)

Percentage of executable plans for embodied AI (Huang et al. 2022)
The Status Quo

A possible fix: constrained decoding

Example Decoding Rules
- The first token must be ‘(’
- The token after ‘(’ can be ‘AND’, ‘JOIN’, ‘ARGMAX’..
- …

Picard (Scholak et al. 2021)
The Status Quo

Constrained decoding can be shortsighted and hard to control

Question: Neil Diamond composed what TV song?

Gold: ( JOIN Composer Neil_Diamond ) ( AND TV_Song #0 )

Predicted: ( JOIN Composer Neil_Diamond ) ( JOIN Song #0 ) ( AND Recording #1 )
Our Proposal: Pangu Framework

Goals:
- Allow LMs to focus on discrimination
- Generic for different tasks

A symbolic agent searches the environment to propose valid candidate plans, while a neural LM scores the plans to guide the search process.
Key Assumptions

1. A complex plan can be expanded from smaller sub-plans incrementally.

2. Valid action space at each step is much smaller compared with decoding.
Our Proposal: Framework

**Algorithm 1: PANGU**

1. **Input:** utterance $q$, initial plans $P_0$, environment $E$
2. $t \leftarrow 1$
3. **while** True **do**
4.  /* AGENT PROPOSES PLANS */
   $C_t \leftarrow \text{Candidate-Plans}(P_{t-1}, E)$
5.  /* LM SCORES AND PRUNES PLANS */
   $P_t \leftarrow \text{Top-K}(q, C_t)$
6.  if Check-Termination() = True then
   7.     return top-scored plan
   8. $t \leftarrow t + 1$

**Initialization of search**

**Enumerate candidate plans from the environment**

**Rank candidate plans using LMs**

**Repeat until the termination condition is met**
Our Proposal: Instantiation

Testbed:
- KBQA

LMs:
- BERT
- T5
- Codex
New SoTA on KBQA

<table>
<thead>
<tr>
<th>Prior Art</th>
<th>78.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pangu w/ BERT-base</td>
<td>79.9</td>
</tr>
<tr>
<td>Pangu w/ T5-base</td>
<td>79.9</td>
</tr>
<tr>
<td>Pangu w/ T5-3B</td>
<td>81.7</td>
</tr>
</tbody>
</table>

F1 on GrailQA
(i.i.d. + non-i.i.d., ~45K training examples)

F1 on GraphQuestions
(non-i.i.d., ~2K training examples)

F1 on WebQSP
(i.i.d., ~3K training examples)

Findings:

1. Particularly strong performance for non-i.i.d. generalization
2. Stable gain from increased model size
In-Context Learning with LLMs

<table>
<thead>
<tr>
<th></th>
<th>Prior Art</th>
<th>Codex 10-shot</th>
<th>Codex 100-shot</th>
<th>Codex 1000-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 on GrailQA</td>
<td>78.7</td>
<td>48.9</td>
<td>53.3</td>
<td>56.4</td>
</tr>
<tr>
<td>(i.i.d. + non-i.i.d., ~45K training examples)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1 on GraphQuestions</td>
<td>34.3</td>
<td>42.8</td>
<td>43.3</td>
<td>44.3</td>
</tr>
<tr>
<td>(non-i.i.d., ~2K training examples)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1 on WebQSP</td>
<td>78.8</td>
<td>45.9</td>
<td>54.5</td>
<td>68.3</td>
</tr>
<tr>
<td>(i.i.d., ~3K training examples)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Findings:

1. SoTA performance on GraphQ with only 10 training examples
2. Marginal gain from more training data for non-i.i.d.
Pangu Improves Sample Efficiency
Autoregressive models tend to overfit seen structures during training.

Highly Skewed
LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, Yu Su
Embodied agents follow language instructions to complete tasks in a physical environment.

- Long-horizon tasks: 50+ steps
- Diverse tasks and environments
- Can LLMs help?
Embodied Agent Planning with LLMs?

**Instruction**: “make me a cup of coffee”

**Low-level Plan**: [turn left, move forward, pick up cup, turn around, move forward, ..., put cup in coffee maker, ...]
Embodied Agent Planning with LLMs?

**Instruction:** “make me a cup of coffee”

**High-level Plan:** [navigation cup, pick up cup, navigation coffee machine, …]

**Low-level Plan:** [turn left, move forward, pick up cup, turn around, move forward, …, put cup in coffee maker, …]
Dynamic Grounded Planning

**Instruction:** “make me a cup of coffee”

**Low-level Plan:** [Turn left, move forward, pick up cup, turn around, move forward, …, put cup in coffee maker, …]

**High-level Plan:** [navigation cup, pick up cup, navigation coffee machine, …]

**Low-level Planner**

![Diagram](image)

Figure 1. An illustration of LLM-Planner for high-level planning. After receiving the natural language instruction ($t = 0$), LLM-Planner first generates a high-level plan by prompting a large language model (e.g., GPT-3). When the embodied agent gets stuck during the execution of the current plan ($t = 5$ and 20), LLM-Planner re-plans based on observations from the environment to generate a more grounded plan, which may help the agent get unstuck. The commonsense knowledge in the LLM (e.g., food is often stored in a fridge) allows it to produce plausible high-level plans and re-plan based on new information from the environment.
Cook the potato and put it into the recycle bin.

Create a high-level plan for completing a household task using the allowed actions and visible objects.

**Allowed actions:** OpenObject, CloseObject, PickupObject, PutObject, ToggleObjectOn, ToggleObjectOff, SliceObject, Navigation

**In-context Examples**

**Task description:** Cook the potato and put it into the recycle bin.

**Completed plans:**

Visible objects are microwave, fridge, garbagecan, chair.

Next Plans:

Plan: Navigation potato, PickupObject potato, ...
Evaluation on ALFRED

- LLM-Planner achieves competitive performance with only 100 training examples
- Existing methods can barely complete any task under the same low-data setting

<table>
<thead>
<tr>
<th>Model</th>
<th>SR</th>
<th>GC</th>
<th>HLP ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-data setting: 21,023 (instruction, trajectory) pairs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.T. [27]</td>
<td>8.57</td>
<td>18.56</td>
<td>–</td>
</tr>
<tr>
<td>HiTUT [40]</td>
<td>13.87</td>
<td>20.31</td>
<td>–</td>
</tr>
<tr>
<td>M-TRACK [36]</td>
<td>16.29</td>
<td>22.60</td>
<td>–</td>
</tr>
<tr>
<td>FILM [26]</td>
<td>27.80</td>
<td>38.52</td>
<td>–</td>
</tr>
<tr>
<td>LEBP [18]</td>
<td>28.30</td>
<td>36.79</td>
<td>–</td>
</tr>
<tr>
<td>Few-shot setting: 100 (instruction, high-level plan) pairs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLSM [3]</td>
<td>0.61</td>
<td>3.72</td>
<td>0.00</td>
</tr>
<tr>
<td>FILM [26]</td>
<td>0.20</td>
<td>6.71</td>
<td>0.00</td>
</tr>
<tr>
<td>SayCan [1]</td>
<td>9.88</td>
<td>22.54</td>
<td>37.57</td>
</tr>
<tr>
<td>LLM-Planner (Static) + HLSM</td>
<td>15.83</td>
<td>20.99</td>
<td>43.24</td>
</tr>
<tr>
<td>LLM-Planner + HLSM</td>
<td><strong>16.42</strong></td>
<td><strong>23.37</strong></td>
<td><strong>46.59</strong> – <strong>68.31</strong></td>
</tr>
</tbody>
</table>

SR: Success Rate, GC: Goal Completion Rate, HLP ACC: High-level Planning Accuracy
What’s the journey ahead of us?

• Is NLP dead?

• Absolutely not. It’s the most exciting time for NLP ever!

• However, instead of natural language processing, perhaps we should focus on natural language programming next
Natural Language Programming

- When is my flight to Seattle?
  - Tomorrow at 5:00 pm.
- How long will it take to get to the airport?
  - It will take 20 minutes according to Google Maps.
- Book a Uber 1.5 hours before that.
  - Sure. Booked an Uber for 3:30 pm tomorrow to the Columbus airport.
- Any good Chinese restaurants close to my hotel?
  - According to Yelp, Haidilao has 4.5 stars and is 2-min walk from Hyatt.

Foundation Model
Thanks &