LLM-Powered Robots

운영체제의 실제

안인규 (Inkyu An)

Final project!

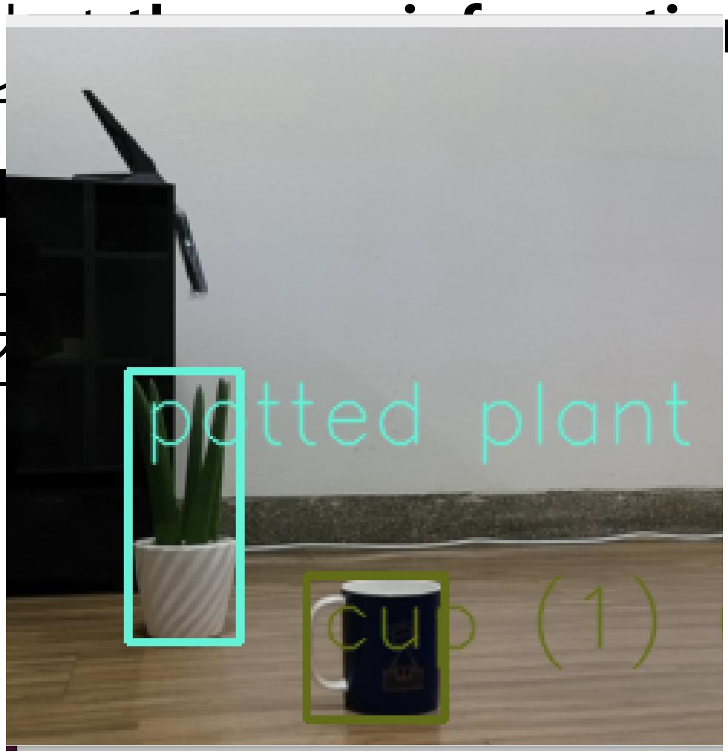
- Objective: Create a semantic map.
  - Automatically build **a map that includes occupied and free space** through autonomous navigation (manual control by a human is not allowed).
  - It is acceptable to use a previously saved map, but **the map information may change**. (장애물의 위치가 변경 및 추가될 수 있다)
  - Detect **specified objects (컵, 화분)** and ~~**mark their locations** on the semantic map~~ (the list of target objects will be provided in advance).
  - Find the objects as quickly as possible. (제한시간: 10분)

- Where? 자율주행스튜디오

Final project!

- Objective: Create a semantic map.
  - Automatically build **a map that includes occupied and free space** through autonomous navigation (manual control by a human is not allowed).
  - It is acceptable to use a previously saved map, **n**
    may change. (장애물의 위치가 변경 및 추가될
  - Detect **specified objects (컵, 화분)** and ~~**mark t**~~
    ~~semantic map~~ (the list of target objects will be
  - Find the objects as quickly as possible. (제한시

- Where? 자율주행스튜디오



Final project!

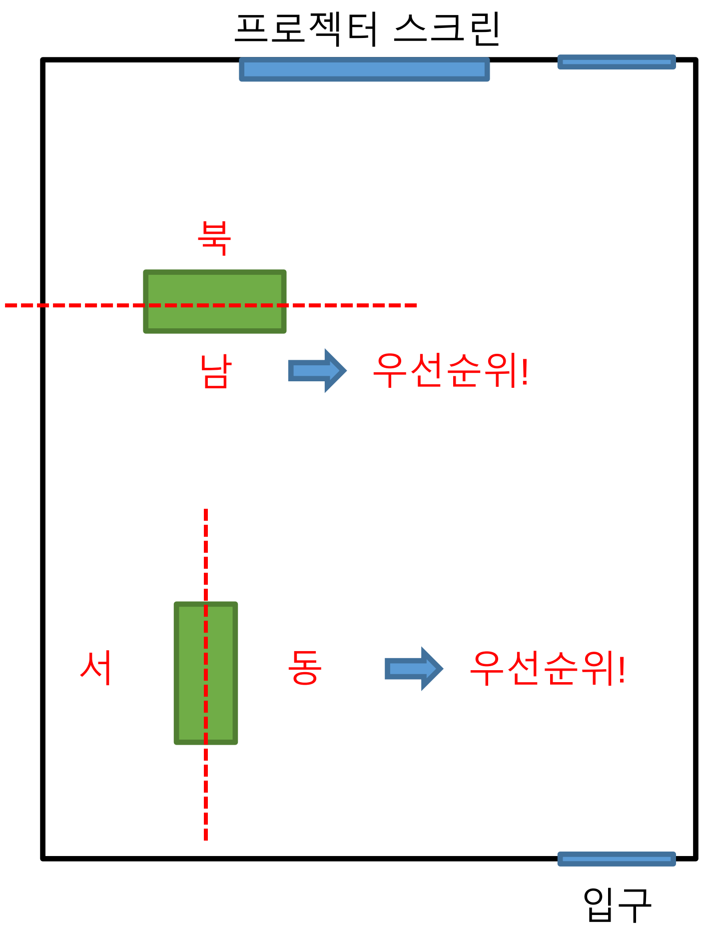
- 프로젝트 진행 순서
  1. Occupancy grid map 업데이트를 위한 경로 생성
     - 사용기술: SLAM + Nav2
     - 필요사항: Occupancy grid map을 생성하기 위한 Nav2의 목적지 생성
  2. Box의 위치를 탐색
     - Box의 위치는 2개 (우체국 박스 4호, 410mm*310mm*280mm)
     - Box의 위치를 모두 찾으면, "1. Occupancy grid map …" 중단
  3. Target object 후보 위치 선정 및 이동
     - 필요기술: Nav2
     - Box의 긴 면에 물체 위치
     - 자주스에서 아래쪽 / 오른쪽에 물체 위치
  4. Target object detection
     - 필요기술: YOLO
     - 보너스 점수 형태

Final project!

- 프로젝트 진행 순서
  1. Occupancy grid map 업데이트를 위한 경로 생성
     - 사용기술: SLAM + Nav2
     - 필요사항: Occupancy grid map을 생성하기 위한 Nav2의 목적지 생성
  2. Box의 위치를 탐색
     - Box의 위치는 2개 (우체국 박스 4호, 410mm*310mm*280mm)
     - Box의 위치를 모두 찾으면, “1. Occupancy grid map …” 중단
  3. Target object 후보 위치 선정 및 이동
     - 필요기술: Nav2
     - Box의 긴 면에 물체 위치
     - 자주스에서 아래쪽 / 오른쪽에 물체 위치
  4. Target object detection
     - 필요기술: YOLO
     - 보너스 점수 형태

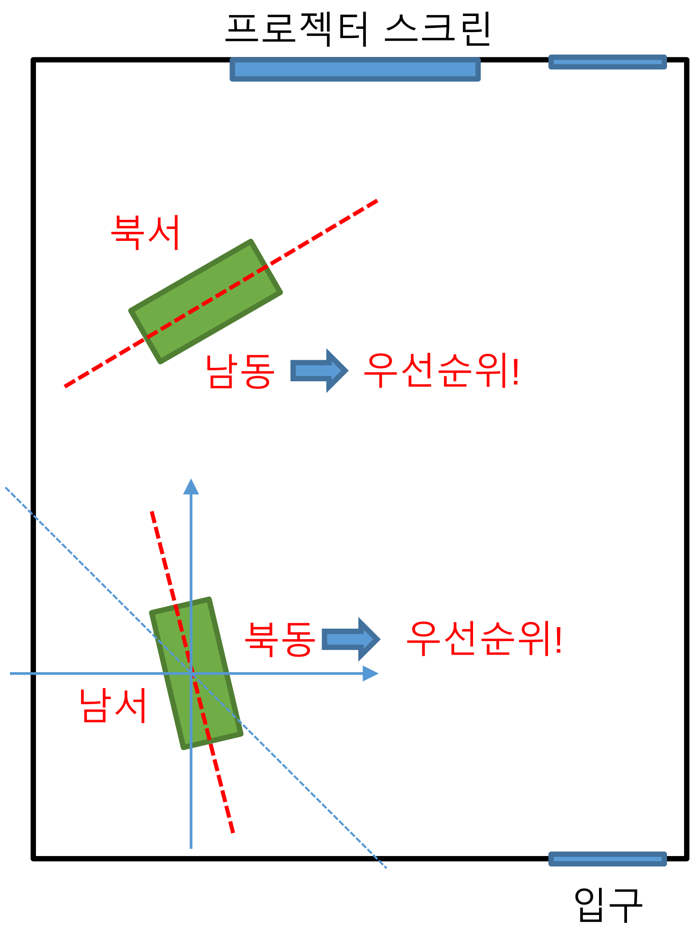
Final project!

- 프로젝트 진행 순서
  1. Occupancy grid map 업데이트를 위한 경로 생성
     - 사용기술: SLAM + Nav2
     - 필요사항: Occupancy grid map을 생성하기 위한 Nav2의 목적지 생성
  2. Box의 위치를 탐색
     - Box의 위치는 2개 (우체국 박스 4호, 410mm*310mm*280mm)
     - Box의 위치를 모두 찾으면, “1. Occupancy grid map …” 중단
  3. Target object 후보 위치 선정 및 이동
     - 필요기술: Nav2
     - Box의 긴 면에 물체 위치
     - 자주스 (프로젝터 스크린을 북쪽)에 **남동**에 가까운 면에 물체 위치
  4. Target object detection
     - 필요기술: YOLO
     - 보너스 점수 형태



Final project!

- 프로젝트 진행 순서
  1. Occupancy grid map 업데이트를 위한 경로 생성
     - 사용기술: SLAM + Nav2
     - 필요사항: Occupancy grid map을 생성하기 위한 Nav2의 목적지 생성
  2. Box의 위치를 탐색
     - Box의 위치는 2개 (우체국 박스 4호, 410mm*310mm*280mm)
     - Box의 위치를 모두 찾으면, “1. Occupancy grid map …” 중단
  3. Target object 후보 위치 선정 및 이동
     - 필요기술: Nav2
     - Box의 긴 면에 물체 위치
     - 자주스 (프로젝터 스크린을 북쪽)에 **남동**에 가까운 면에 물체 위치
  4. Target object detection
     - 필요기술: YOLO
     - 보너스 점수 형태

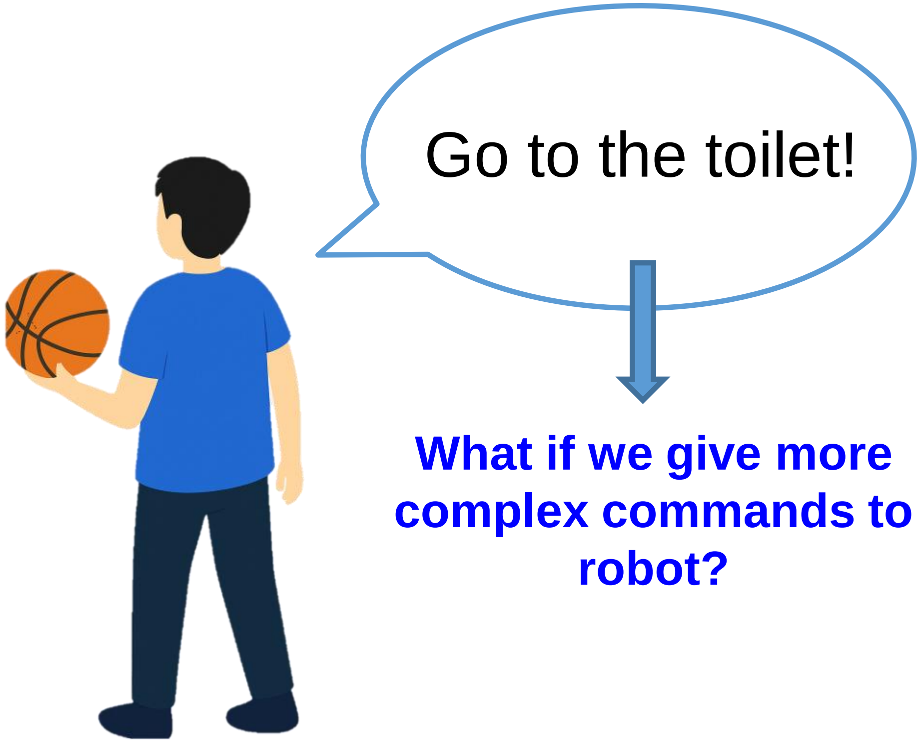


Command using Natural Language

- Isn't it possible to give commands using natural language? (e.g., Go to the toilet!)
  - In traditional navigation methods, you have to input the exact location of the toilet.

Command using Natural Language

- Isn't it possible to give commands using natural language? (e.g., Go to the toilet!)
  - In traditional navigation methods, you have to <u>input the exact location of the toilet.</u>



Command using Natural Language

- Isn't it possible to give <u>more complex commands</u> using natural language?

- Where is the dirty area?
- How should I clean it up?
- Which cleaning tools should I use?
- ...

Clean the house!

Command using Natural Language

- Isn't it possible to give <u>more complex commands</u> using natural language?

- Where is the dirty area?
- How should I clean it up?
- Which cleaning tools should I use?
- ...

LLM can make a plan!

Clean the house!

Powered by LLM

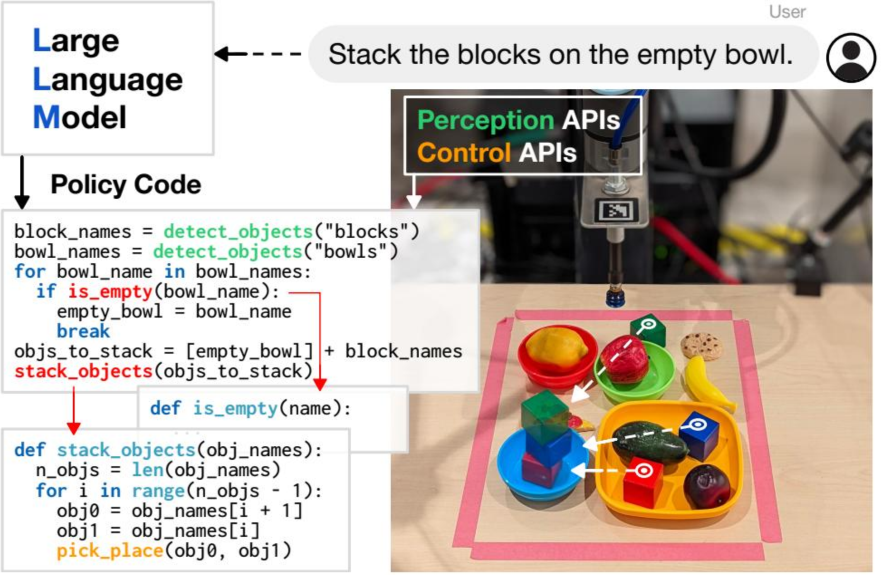
- LLMs are very powerful tools that can understand natural language. LLM can make a plan!
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).

Powered by LLM

- LLMs are very powerful tools that can understand natural language. → LLM can make a plan!
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).

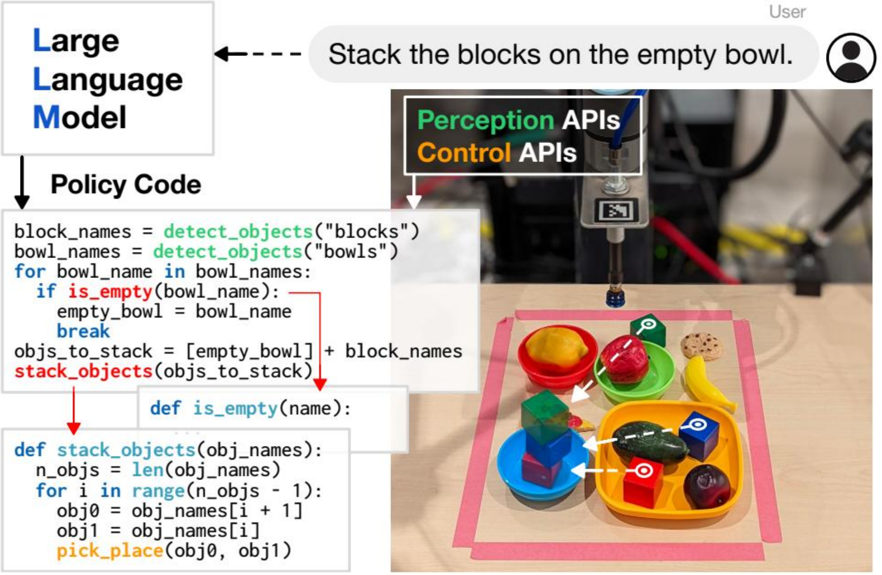
It can only handle predefined motions

- pick and place
- move
- ...



Powered by LLM

- LLMs are very powerful tools that can understand natural language. → LLM can make a plan!
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).



→ It can only handle predefined motions

- pick and place
- move
- ...

→ **What should we do if we need to handle high-dimensional robot motions?**

Powered by LLM

- LLMs are very powerful tools that can understand natural language. → LLM can make a plan!
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).
- LLM cannot control the robot (complex robot control)

→ A method is needed to control robots from LLM-generated commands

Powered by LLM

- LLMs are very powerful tools that can understand natural language. ➡ LLM can make a plan!
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).
- LLM cannot control the robot (complex robot control)

  ➡ A method is needed to control robots from LLM-generated commands

- LLM have not interacted with their environment (LLM does not know about the environment)

  ➡ LLMs need to understand visual information

SayCan, Google

- Do As I Can, Not As I Say: Grounding Language in Robotic Affordances, Arxiv, 2022 (**Robotics at Google**)

SayCan, Google

- **Motivation**: LLMs have not interacted with their environment

Say: LLM을 이용해 계획을 생성!

Can: 시각 정보로부터 로봇이 어떤 행동을 취해야 하는지 결정 (Control)

SayCan, Google

- **User-provided natural language instruction i**
  - "I spilled my drink, can you help?"
- **A set of skills Π, each skill $\pi \in \Pi$ performs a short tasks**
  - e.g., find a particular object, picking up a particular object, …
  - It comes with a short language description l_π: e.g., find a sponge

l_π

SayCan, Google

- **"Say" Language model** provides us with $\boldsymbol{p(l_\pi|i)}$, the probability that a skill's textual label is a valid next step for the user's instruction

"find a sponge"

or

"go to the trash can"

"I spilled my drink, can you help?"

- How the language model can do this?
  - General language model: asked "how would a robot bring me an apple" → respond "a robot could go to a nearby store and purchase an apple for you"

Logically true! but, it is no actionable to an embodied agent

SayCan, Google

- **"Say" Language model** provides us with $\boldsymbol{p(l_\pi|i)}$, the probability that a skill's textual label is a valid next step for the user's instruction

"find a sponge"

or

"go to the trash can"

"I spilled my drink, can you help?"

- How the language model can do this?
  - General language model: asked "how would a robot bring me an apple" → respond "a robot could go to a nearby store and purchase an apple for you"

Logically true! but, it is no actionable to an embodied agent

→ Must inform LLMs that we specifically want the high-level instruction to be broken down into sequences of available low-level skills, How?

SayCan, Google

- **“Say” Language model** provides us with $\boldsymbol{p(l_\pi|i)}$, the probability that a skill’s textual label is a valid next step for the user’s instruction

“find a sponge”

or

“go to the trash can”

“I spilled my drink, can you help?”

- How the language model can do this?
  - General language model: asked “how would a robot bring me an apple” → respond “a robot could go to a nearby store and purchase an apple for you”

Logically true! but, it is no actionable to an emb

Prompt engineering!

→ Must inform LLMs that we specifically want the high-level instruction to be broken down into sequences of available low-level skills, How?

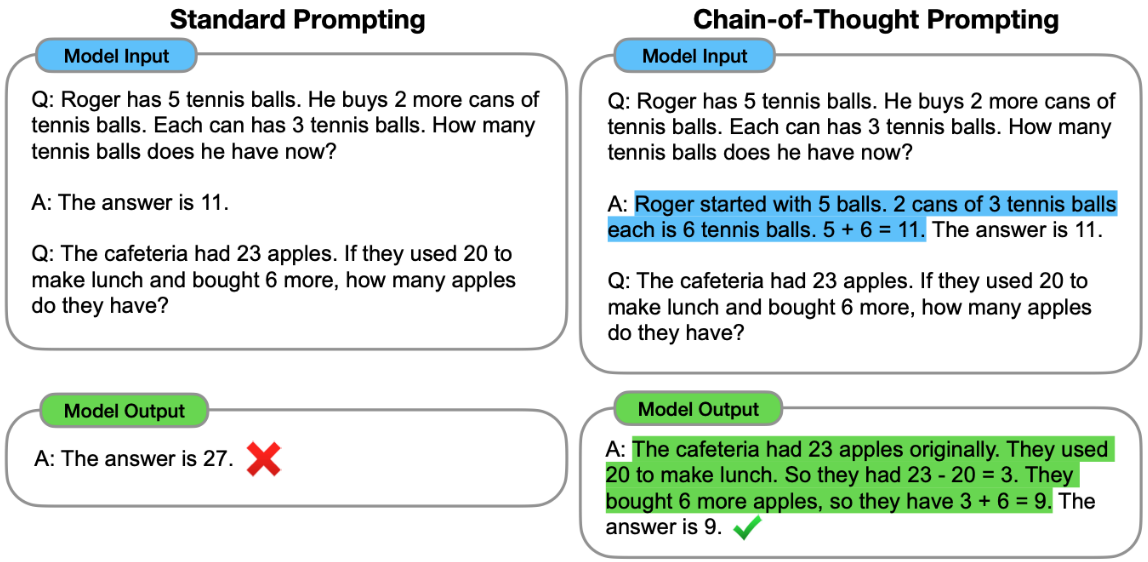
SayCan, Google

- Prompt engineering!
  - It provides examples in the context text ("prompt") for LLMs that specify the task and the response structure which the model will emulate (모방하다)
  - It involves 17 examples
- 응답 형식 통일
  - Human: ... would ...?
  - Robot: 1. [action], 2. [action], ..., done

```
Robot: Hi there, I'm a robot operating in an office kitchen.
Robot: You can ask me to do various tasks and I'll tell you the sequence of actions I would do to
     accomplish your task.
Human: How would you hold the snickers?
Robot: 1. pick up the snickers, 2. done.
Human: How would you put the trailmix here?
Robot: 1. put down the trailmix, 2. done.
Human: How would you put a water bottle and an oatmeal next to the microwave?
Robot: 1. find a water bottle, 2. pick up the water bottle, 3. go to microwave, 4. put down the water
     bottle, 5. find an oatmeal, 6. pick up the oatmeal, 7. go to the microwave, 8. put down the oatmeal,
     9. done.
Human: How would you put a grapefruit from the table into the bowl?
Robot: 1. find a grapefruit, 2. pick up the grapefruit, 3. go to the bowl, 4. put down the grapefruit, 5.
     done.
Human: How would you get a sponge from the counter and put it in the sink?
Robot: 1. find a sponge, 2. pick up the sponge, 3. go to the sink, 4. put down the sponge, 5. done.
Human: How would you move the water bottle from the table to the counter?
Robot: 1. find a water bottle, 2. pick up the water bottle, 3. go to the counter, 4. put down the water
     bottle, 5. done.
Human: How would you bring me some snacks?
Robot: 1. find a twix, 2. pick up the twix, 3. bring it to you, 4. put down the twix, 5. find an dried
     fruit, 6. pick up the dried fruit, 7. bring it to you, 8. put down the dried fruit, 9. done.
Human: How would you bring me an orange?
Robot: 1. find an orange, 2. pick up the orange, 3. bring it to you, 4. put down the orange, 5. done.
Human: How would you deliver a can of coke to me?
Robot: 1. find a coke can, 2. pick up the coke can, 3. bring it to you, 4. put down the coke can, 5. done.
Human: I finished my drink, can you throw it away and bring me a mountain dew?
Robot: 1. find a sprite can, 2. pick up the sprite can, 3. go to the trash can, 4. put down the sprite can
     , 5. find a mountain dew can, 6. pick up the mountain dew can, 7. bring it to you, 8. put down the
     mountain dew can, 9. done.
Human: How would you let go of the banana?
Robot: 1. put down the banana, 2. done.
Human: How would you put the grapes in the bowl and then move the cheese to the table?
Robot: 1. find the grapes, 2. pick up the grapes, 3. go to the bowl, 4. put down the grapes, 5. pick up
     the cheese, 6. go to the table, 7. put down the cheese, 8. tell you I'm done.
Human: How would you pick up the peanuts?
Robot: 1. pick up the peanuts, 2. done.
Human: How would you bring me the peanuts?
Robot: 1. pick up the peanuts, 2. bring it to you, 3. put down the peanuts, 4. done.
Human: How would you throw away a coffee cup?
Robot: 1. find a coffee cup, 2. pick up the coffee cup, 3. go to trash can, 4. put down the coffee cup, 5.
      done.
Human: How would you place a knife and a banana to the table?
Robot: 1. find a knife, 2. pick up the knife, 3. go to the table, 4. put down the knife, 5. find a banana,
      6. pick up the banana, 7. go to the table, 8. put down the banana, 9. done.
```

SayCan, Google

- Prompt engineering!
  - It provides examples in the context text ("prompt") for LLMs that specify the task and the response structure which the model will emulate (모방하다)
  - <u>It involves 17 examples</u>
  - SayCan can be intergrated with recent work improving LLM reasoning (**<u>Chain-of-Thought</u>**)



Few-shot CoT prompting 예시
(Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, NeurIPS 2022)

SayCan, Google

- Prompt engineering!
  - It provides examples in the context text ("prompt") for LLMs that specify the task and the response structure which the model will emulate (모방하다)
  - It involves 17 examples
  - SayCan can be intergrated with recent work improving LLM reasoning (**Chain-of-Thought**)

CHAIN-OF-THOUGHT PLANNING ROLLOUT WITH PALM-SAYCAN.

Human: Can you bring a fruit-flavored drink without caffeine?
Explanation: The user has asked for a drink that is fruit-flavored and does not have caffeine, I will bring the lime soda.
Robot: 1. find a lime soda, 2. pick up the lime soda, 3. bring it to you, 4. put down the lime soda, 5. done

Human: Bring me the spicy chips.
Explanation: The user has asked for chips that are spicy, I will bring the Jalapeno chips.
Robot: 1. find the jalapeno chips, 2. pick up the jalapeno chips, 3. bring it to you, 4. put down the jalapeno chips, 5.done

Human: The snack is too small, bring me something more filling.
Explanation: The user has asked for a snack that is more filling, I will bring the multigrain chips.
Robot: 1. find the multigrain chips, 2. pick up the multigrain chips, 3. bring it to you, 4. put down the multigrain chips, 5. done

SayCan (Google)

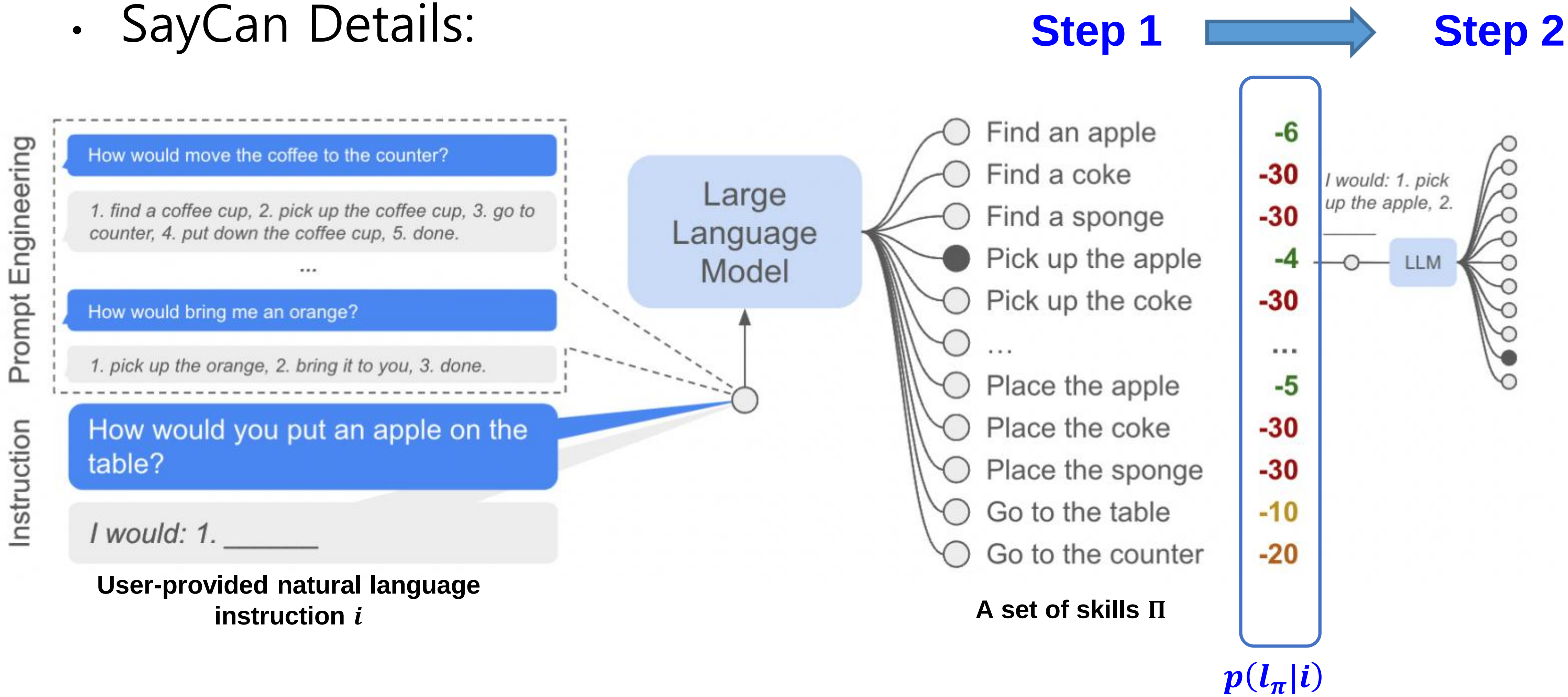
- Prompt engineering!
  - It provides examples in the context text ("prompt") for LLMs that specify the task and the response structure which the model will emulate (모방하다)
  - <u>It involves 17 examples</u>

Num Examples	Require Termination	No Termination Required
0	10%	52%
1	64%	74%
2	68%	76%
4	82%	84%
8	80%	80%
Full Prompt (17)	88%	88%

```
Robot: Hi there, I'm a robot operating in an office kitchen.
Robot: You can ask me to do various tasks and I'll tell you the sequence of actions I would do to
     accomplish your task.
Human: How would you hold the snickers?
Robot: 1. pick up the snickers, 2. done.
Human: How would you put the trailmix here?
Robot: 1. put down the trailmix, 2. done.
Human: How would you put a water bottle and an oatmeal next to the microwave?
Robot: 1. find a water bottle, 2. pick up the water bottle, 3. go to microwave, 4. put down the water
     bottle, 5. find an oatmeal, 6. pick up the oatmeal, 7. go to the microwave, 8. put down the oatmeal,
     9. done.
Human: How would you put a grapefruit from the table into the bowl?
Robot: 1. find a grapefruit, 2. pick up the grapefruit, 3. go to the bowl, 4. put down the grapefruit, 5.
     done.
Human: How would you get a sponge from the counter and put it in the sink?
Robot: 1. find a sponge, 2. pick up the sponge, 3. go to the sink, 4. put down the sponge, 5. done.
Human: How would you move the water bottle from the table to the counter?
Robot: 1. find a water bottle, 2. pick up the water bottle, 3. go to the counter, 4. put down the water
     bottle, 5. done.
Human: How would you bring me some snacks?
Robot: 1. find a twix, 2. pick up the twix, 3. bring it to you, 4. put down the twix, 5. find an dried
     fruit, 6. pick up the dried fruit, 7. bring it to you, 8. put down the dried fruit, 9. done.
Human: How would you bring me an orange?
Robot: 1. find an orange, 2. pick up the orange, 3. bring it to you, 4. put down the orange, 5. done.
Human: How would you deliver a can of coke to me?
Robot: 1. find a coke can, 2. pick up the coke can, 3. bring it to you, 4. put down the coke can, 5. done.
Human: I finished my drink, can you throw it away and bring me a mountain dew?
Robot: 1. find a sprite can, 2. pick up the sprite can, 3. go to the trash can, 4. put down the sprite can
     , 5. find a mountain dew can, 6. pick up the mountain dew can, 7. bring it to you, 8. put down the
     mountain dew can, 9. done.
Human: How would you let go of the banana?
Robot: 1. put down the banana, 2. done.
Human: How would you put the grapes in the bowl and then move the cheese to the table?
Robot: 1. find the grapes, 2. pick up the grapes, 3. go to the bowl, 4. put down the grapes, 5. pick up
     the cheese, 6. go to the table, 7. put down the cheese, 8. tell you I'm done.
Human: How would you pick up the peanuts?
Robot: 1. pick up the peanuts, 2. done.
Human: How would you bring me the peanuts?
Robot: 1. pick up the peanuts, 2. bring it to you, 3. put down the peanuts, 4. done.
Human: How would you throw away a coffee cup?
Robot: 1. find a coffee cup, 2. pick up the coffee cup, 3. go to trash can, 4. put down the coffee cup, 5.
      done.
Human: How would you place a knife and a banana to the table?
Robot: 1. find a knife, 2. pick up the knife, 3. go to the table, 4. put down the knife, 5. find a banana,
      6. pick up the banana, 7. go to the table, 8. put down the banana, 9. done.
```

SayCan (Google)

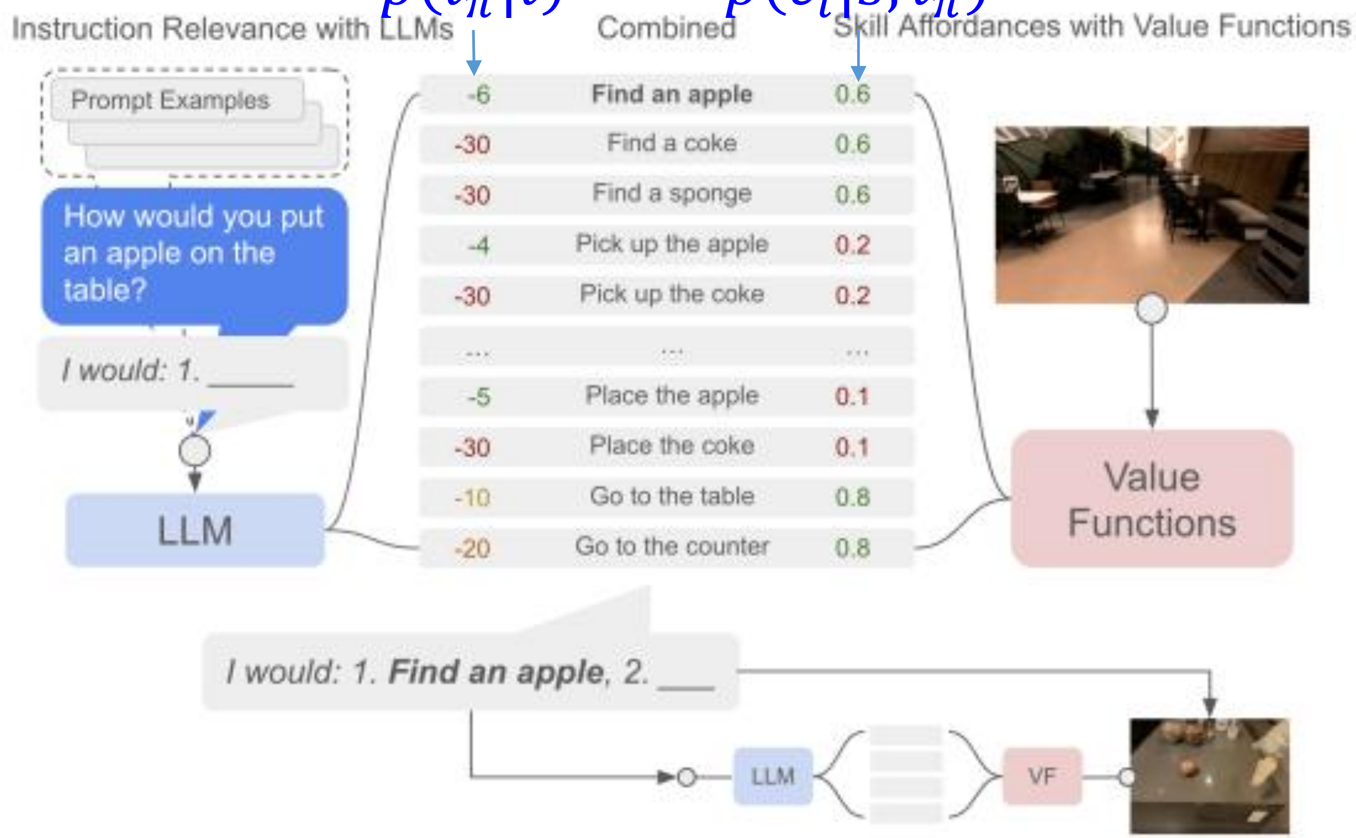
- SayCan Details:



SayCan (Google)

- **Skill Affordances (i.e., 환경이 특정 행동을 가능하게 하는 성질) with Value Functions**:
  - E.g., “컵을 집기” skill은 컵이 로봇 팔에 닿는 위치에 있는 경우에만 높은 affordance를 가짐

$p(l_\pi|i)$ $p(c_i|s,l_\pi)$



- $p(c_i|s,l_\pi)$: value function
- s: state (observation에서)
- c_i: 스킬이 성공적으로 완료됨을 나타내는

$$c_\pi = \begin{cases} 1 & \text{스킬 } \pi\text{가 성공적으로 완료됨} \\ 0 & \text{스킬 } \pi\text{가 실패함} \end{cases}$$

SayCan (Google)

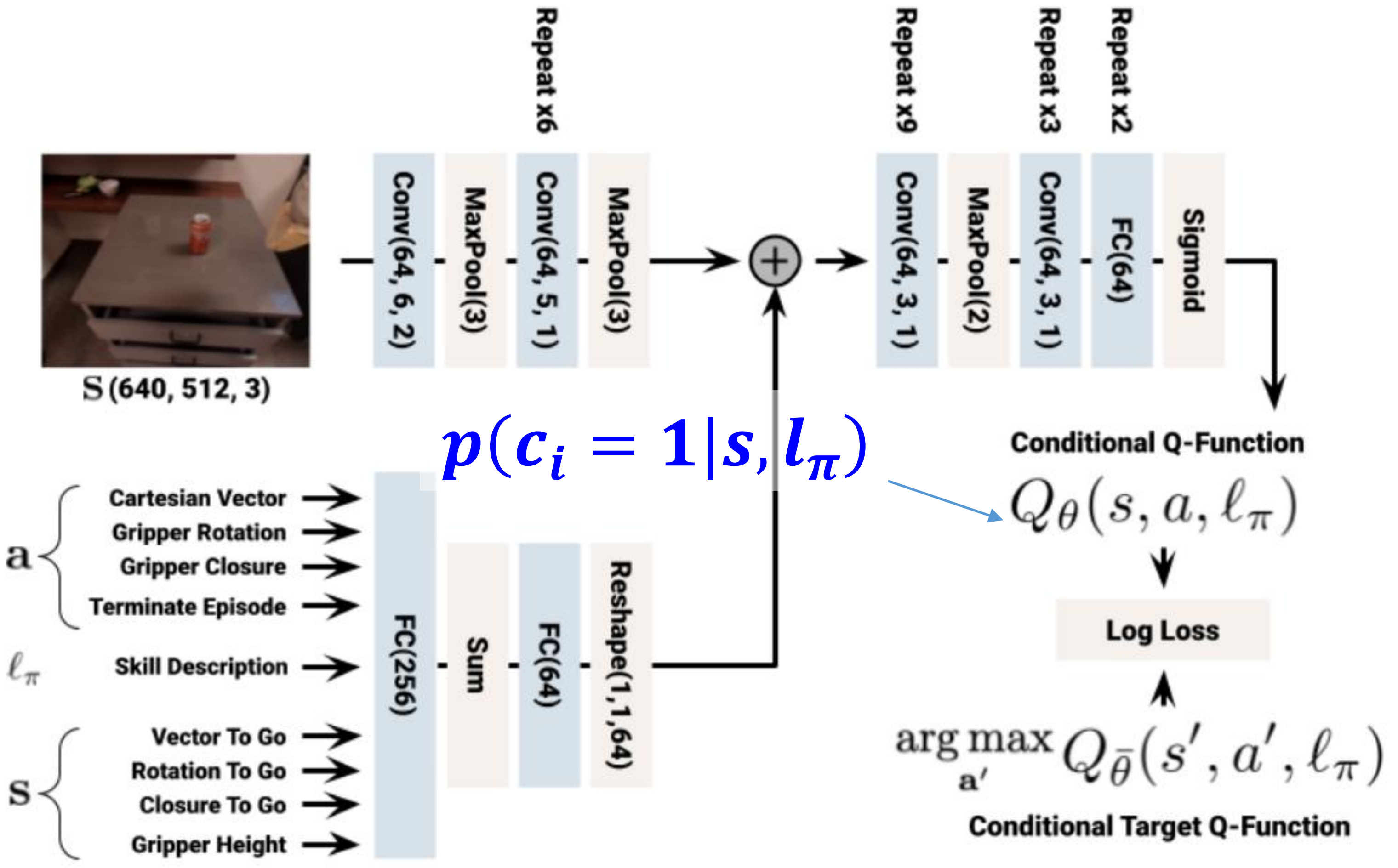
- Value function $p(c_i|s, l_\pi)$:
  - s: state (observation에서)
  - c_i: 스킬이 성공적으로 완료됨을 나타내는 Bernoulli random variable

$$c_\pi = \begin{cases} 1 & \text{스킬 } \pi \text{가 성공적으로 완료됨} \\ 0 & \text{스킬 } \pi \text{가 실패함} \end{cases}$$

  - **즉, 특정 상태 s (특정 시점 observation일 때 state)에서, 어떠한 skill l_π을, 성공적으로 완료할 수 있는 확률**

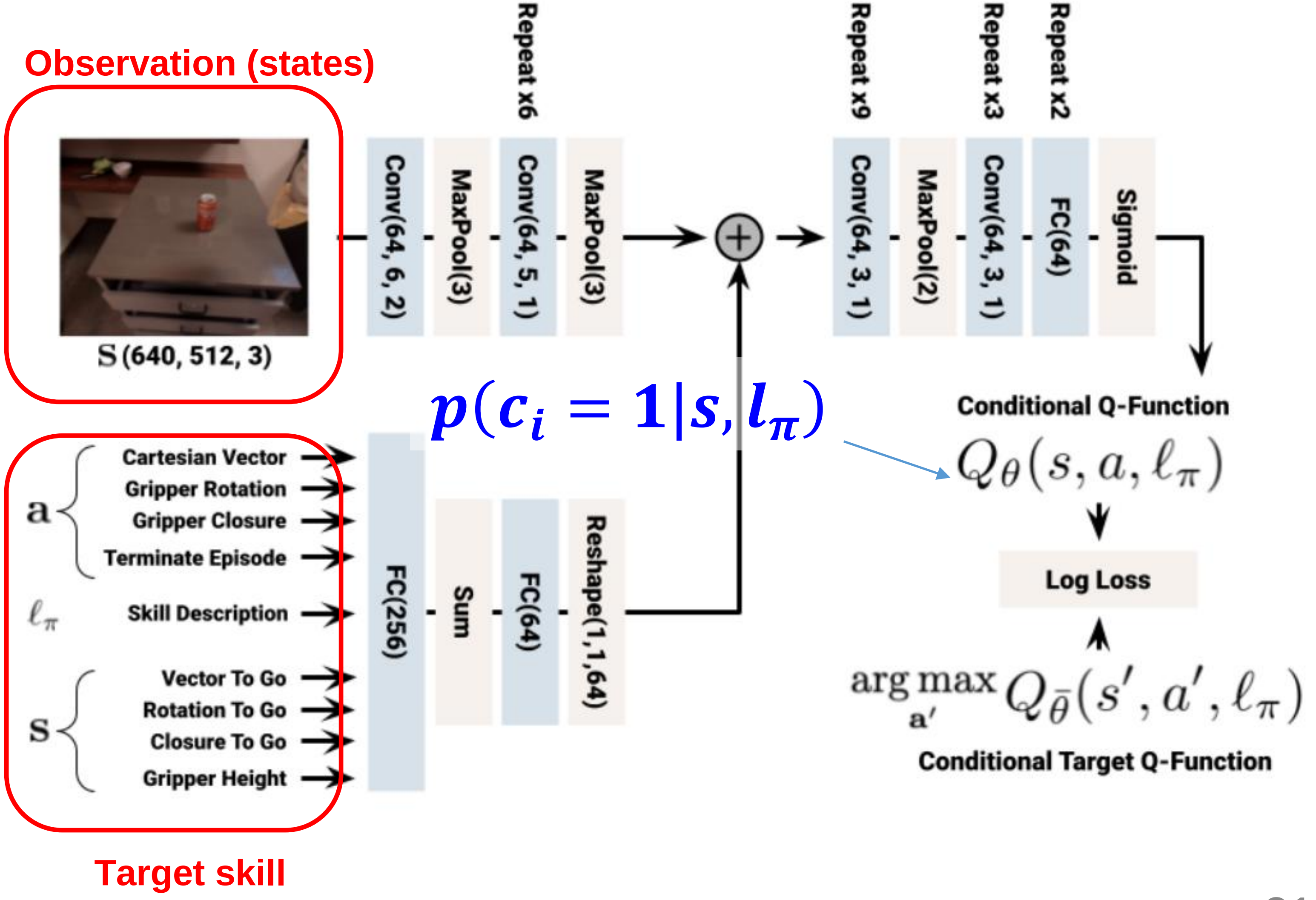
SayCan (Google)

- **Value function** $\boldsymbol{p(c_i|s, l_\pi)}$: Temporal-Difference (TD) RL을 사용해서 value function을 학습
  - 16 TPUv3 chips for about 100 hours



SayCan (Google)

- **Value function** $p(c_i|s, l_\pi)$: Temporal-Difference (TD) RL을 사용해서 value function을 학습
  - 16 TPUv3 chips for about 100 hours



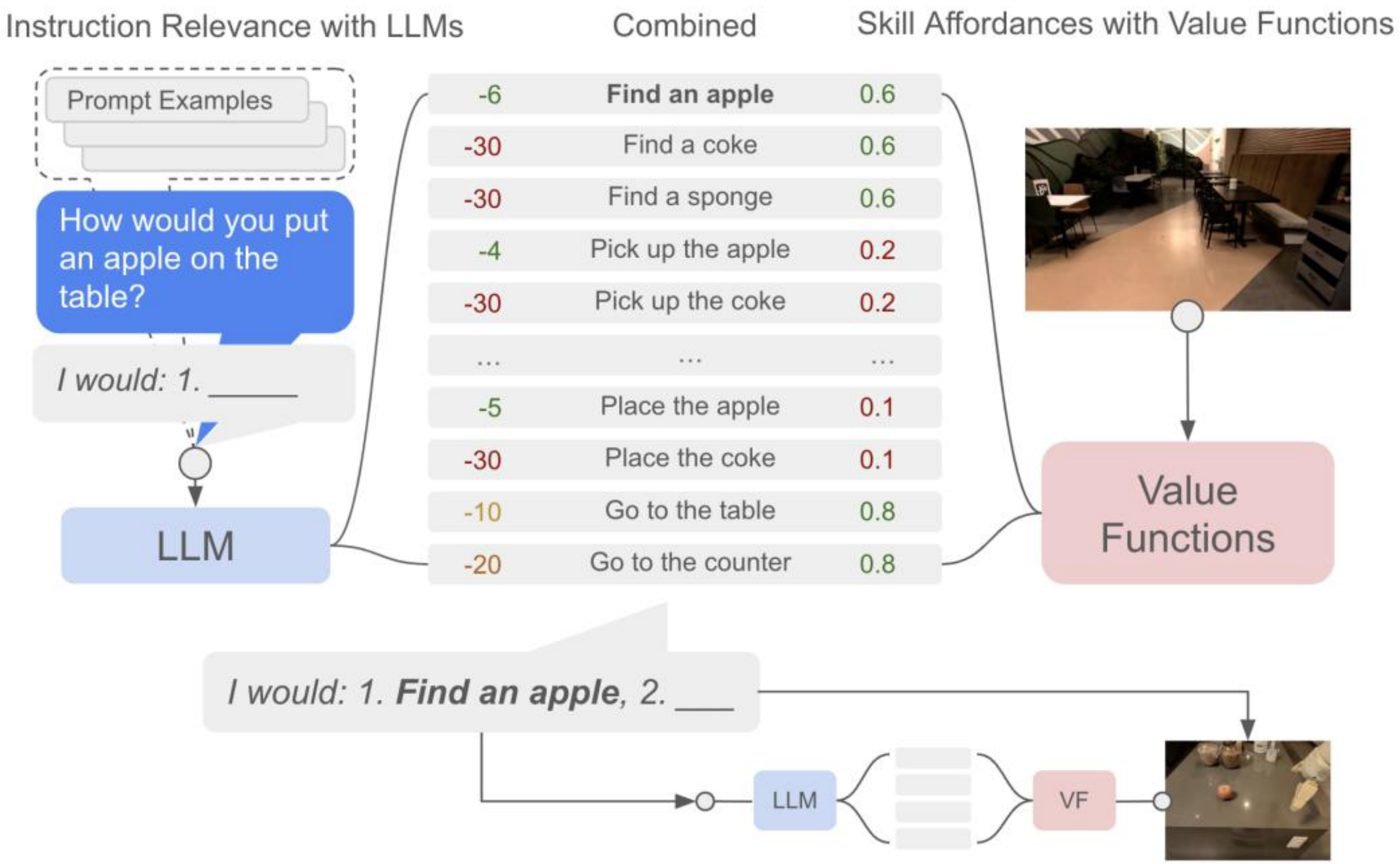
SayCan (Google)

Algorithm 1 SayCan

Given: A high level instruction i, state s_0, and a set of skills Π and their language descriptions ℓ_Π

```
n = 0, π = ∅
while ℓ_{π_{n-1}} ≠ "done" do
    C = ∅
    for π ∈ Π and ℓ_π ∈ ℓ_Π do
        p_π^LLM = p(ℓ_π | i, ℓ_{π_{n-1}}, ..., ℓ_{π_0})            ▷ Evaluate scoring of LLM
        p_π^affordance = p(c_π | s_n, ℓ_π)                          ▷ Evaluate affordance function
        p_π^combined = p_π^affordance p_π^LLM
        C = C ∪ p_π^combined
    end for
    π_n = arg max_{π∈Π} C
    Execute π_n(s_n) in the environment, updating state s_{n+1}
    n = n + 1
end while
```

SayCan (Google)



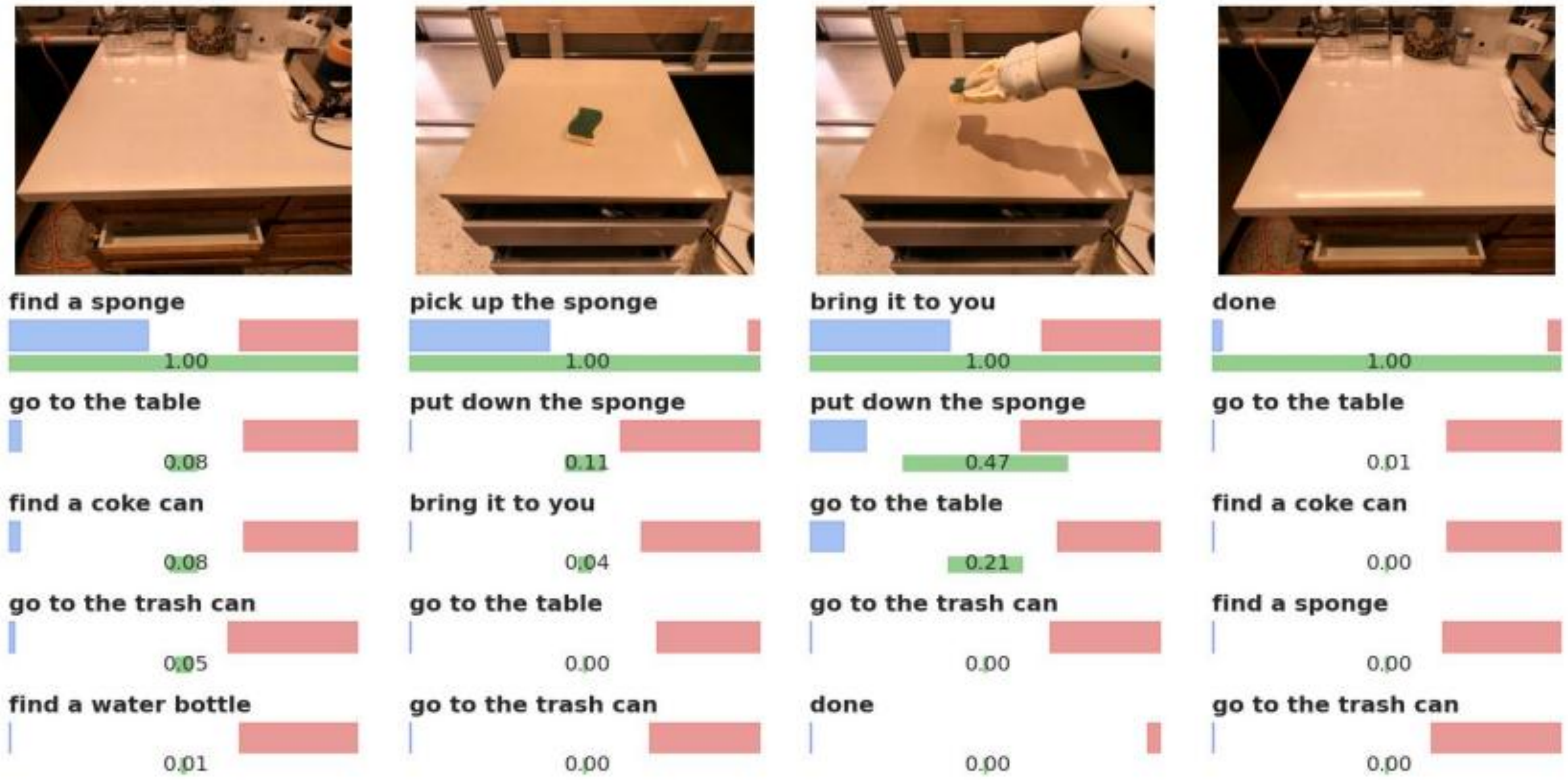
SayCan (Google)

- Do As I Can, Not As I Say: Grounding Language in Robotic Affordances, Arxiv, 2022 (**Robotics at Google**)

SayCan (Google)

Human: I spilled my coke, can you bring me something to clean it up?

Robot: I would
1. Find a sponge
2. Pick up the sponge
3. Bring it to you
4. Done

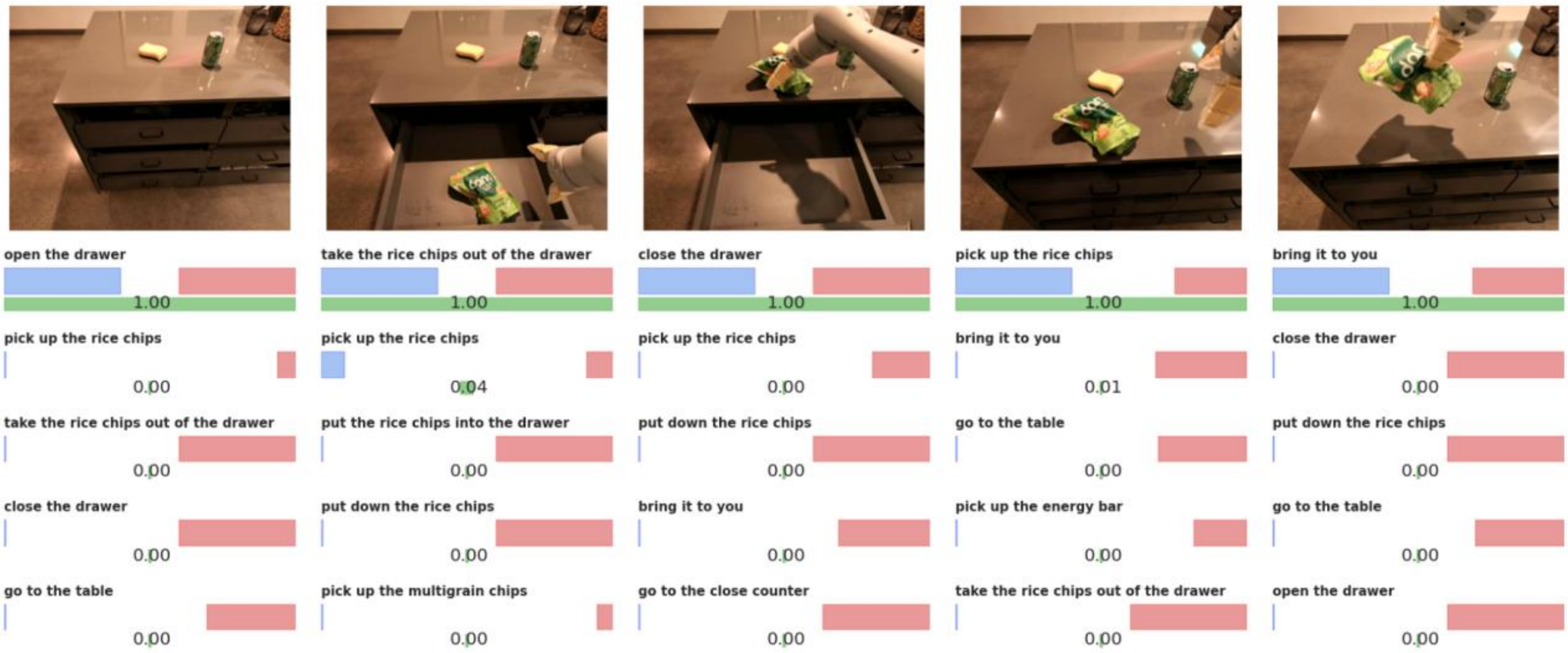


Language × Affordance
Combined Score

SayCan (Google)

Human: Bring me the rice chips from the drawer.

Robot: I would
1. Go to the drawer
2. Open the drawer
3. Take the rice chips out of the drawer
4. Close the drawer
5. Pick up the rice chips
6. Bring it to you
7. Put down the rice chips
8. Done



SayCan (Google)

No Value Function

Generative LLM
(LLM이 모든 sequence
직접 생성)

Behavior Cloning Natural
Language

Behavior Cloning
Universal Sentence
Encoder

		Mock Kitchen		Kitchen		No Affordance		No LLM	
		PaLM-SayCan	**PaLM-SayCan**	**PaLM-SayCan**	**PaLM-SayCan**	**No VF**	**Gen.**	**BC NL**	**BC USE**
Family	**Num**	Plan	Execute	Plan	Execute	Plan	Plan	Execute	Execute
NL Single	15	100%	100%	93%	87%	73%	87%	0%	60%
NL Nouns	15	67%	47%	60%	40%	53%	53%	0%	0%
NL Verbs	15	100%	93%	93%	73%	87%	93%	0%	0%
Structured	15	93%	87%	93%	47%	93%	100%	0%	0%
Embodiment	11	64%	55%	64%	55%	18%	36%	0%	0%
Crowd Sourced	15	87%	87%	73%	60%	67%	80%	0%	0%
Long-Horizon	15	73%	47%	73%	47%	67%	60%	0%	0%
Total	101	84%	74%	81%	60%	67%	74%	0%	9%

Table 2: Success rates of instructions by family. PaLM-SayCan achieves a planning success rate of 84% and execution success rate of 74% in the training environment and 81% planning and 60% execution in a real kitchen. *No VF* uses the maximum score skill from the LLM, *Generative (Gen.)* uses a generative LLM and then projects to the nearest skill via USE embeddings, *BC NL* uses the policy with the natural language instruction, and *BC USE* uses the policy with the natural language instruction projected to the nearest skill via USE embeddings.