# Build Multi-Agent Systems with ADK

## Objectives

In this lab, you will:

- Create multiple agents and relate them to one another with parent to sub-agent relationships.
- Build content across multiple turns of conversation and multiple agents by writing to a session's state dictionary.
- Instruct agents to read values from the session state to use as context for their responses.
- Use workflow agents to pass the conversation between agents directly.

```
Running agent parent_and_subagents, type exit to exit.
[user]: สวัสดี
[steering]: สวัสดีครับ! ยินดีต้อนรับสู่บริการช่วยวางแผนการท่องเที่ยวครับ

ไม่ทราบว่าคุณมีจุดหมายปลายทางหรือประเทศที่อยากไปในใจแล้วหรือยังครับ? หรือว่าอยากให้ผมช่วยแนะนำไอเดียสถานที่ท่องเที่ยวดีครับ?
[user]: อยากไปประเทศไทย จังหวัดขอนแก่น
[attractions_planner]: ยินดีด้วยครับ! **จังหวัดขอนแก่น** เป็นจุดหมายปลายทางที่น่าสนใจมาก มีทั้งแหล่งประวัติศาสตร์ วัฒนธรรม และธรรมชาติที่สวยงาม

นี่คือสถานที่ท่องเที่ยวแนะนำในขอนแก่นที่ห้ามพลาดครับ:

1. **พระมหาธาตุแก่นนคร (วัดหนองแวง):** พระธาตุเก่าชั้นที่สวยงามตั้งอยู่ริมบึงแก่นนคร คุณสามารถขึ้นไปชมวิวเมืองขอนแก่นแบบ 360 องศาได้จากชั้นบนสุด และสักการะพระบรมสารีริกธาตุเพื่อความเป็นสิริมงคลครับ
2. **บึงแก่นนคร:** อ่างเก็บน้ำขนาดใหญ่ใจกลางเมือง เหมาะสำหรับการไปเดินเล่นพักผ่อนหย่อนใจ ชมพระอาทิตย์ตกดิน หรือหาของอร่อยๆ ทานรอบๆ บึงในยามเย็น
3. **พิพิธภัณฑ์ไดโนเสาร์ภูเวียง และอุทยานแห่งชาติภูเวียง:** ขอนแก่นขึ้นชื่อว่าเป็นดินแดนไดโนเสาร์ ที่นี่คุณจะได้ชมฟอสซิลไดโนเสาร์ของจริงที่ขุดพบในไทย และเรียนรู้เรื่องราวทางธรณีวิทยาที่น่าตื่นตาตื่นใจ เหมาะมากสำหรับครอบครัวและผู้ที่ชอบประวัติศาสตร์ธรรมชาติ
4. **ตลาดต้นตาล (Ton Tann Market):** ตลาดนัดยามเย็นสุดชิค แหล่งรวมอาหารสตรีทฟู้ดอร่อยๆ เสื้อผ้าแฟชั่น และงานศิลปะแฮนด์เมด พร้อมดนตรีสดฟังสบายๆ เหมาะกับการไปเดินชิลล์ตอนกลางคืนครับ
5. **จุดชมวิวหินช้างสี (อุทยานแห่งชาติน้ำพอง):** จุดชมวิวธรรมชาติที่สวยงาม มีกลุ่มหินขนาดใหญ่รูปร่างแปลกตา และสามารถมองเห็นวิวเขื่อนอุบลรัตน์ได้แบบกว้างไกลสุดสายตา

ไม่ทราบว่าคุณชอบท่องเที่ยวสไตล์ไหนเป็นพิเศษไหมครับ? (เช่น เน้นธรรมชาติ, สายทำบุญไหว้พระ, สายกิน/ช้อปปิ้ง หรืออยากให้จัดแผนการเดินทางกี่วันดีครับ) จะได้ช่วยแนะนำเพิ่มเติมให้ตรงใจที่สุดครับ!
[user]:
```

```
[user]: จริงๆ แล้วไม่รู้จะไปไหน
[travel_brainstormer]: สวัสดีครับ! ไม่ต้องกังวลเลยครับ การเริ่มต้นจากศูนย์แบบนี้แหละสนุกที่สุด เพราะเราสามารถเปิดรับไอเดียใหม่ ๆ ได้อย่างเต็มที่เลยครับ

ผมจะช่วยคุณค้นหาประเทศและรูปแบบการท่องเที่ยวที่เหมาะกับคุณที่สุดเองนะครับ

เพื่อช่วยให้ผมแนะนำได้ตรงใจคุณมากขึ้น ลองบอกผมหน่อยครับว่าคุณชอบแนวไหนในตัวเลือกเหล่านี้ หรือผสมผสานกันได้นะครับ:

1. **สไตล์การเที่ยวที่ชอบ:**
   * เน้นธรรมชาติ ภูเขา ทะเล สูดอากาศบริสุทธิ์ (เช่น สวิตเซอร์แลนด์, นิวซีแลนด์, มัลดีฟส์)
   * เน้นประวัติศาสตร์ วัฒนธรรม สถาปัตยกรรมสวย ๆ (เช่น อิตาลี, ญี่ปุ่น, อียิปต์)
   * เน้นแสงสีเสียง ช้อปปิ้ง เที่ยวในเมืองลำ ๆ (เช่น ฮ่องกง, สิงคโปร์, เกาหลีใต้)
   * เน้นผจญภัย ทำกิจกรรมลุย ๆ ปีนเขา ดำน้ำ (เช่น เนปาล, อินโดนีเซีย)

2. **สภาพอากาศที่อยากเจอ:** ชอบหนาว ๆ มีหิมะ, อากาศเย็นสบายกำลังดี หรือชอบแบบอบอุ่นแดดใส?

3. **เวลาและงบประมาณคร่าว ๆ:** มีเวลาเดินทางประมาณกี่วัน และอยากได้ทริปแบบสบายกระเป๋าหรือเน้นความหรูหราพักผ่อนเต็มที่ครับ?

คุณคิดเห็นอย่างไรกับข้อไหนบ้าง หรือมีภาพในหัวคร่าว ๆ ไหมครับว่าอยากให้ทริปนี้ออกมาเป็นแบบไหน? บอกผมได้เลยนะครับ!
[user]:
```

693040129-3 chawanakon_day2_building_agents.ipynb

```python
# ── ทดสอบ Agent แรก ──
await demo_chat(greeting_agent, [
    'สวัสดีพี่บ่าว',
    'AI มันคือพรั่นพรื้อ',
    'ขอบใจมากนะพี่บ่าว'
])
```

```
...
👤 ผู้ใช้: สวัสดีพี่บ่าว
🤖 Agent: สวัสดีค่ะ 😊

👤 ผู้ใช้: AI มันคือพรั่นพรื้อ
🤖 Agent: AI ย่อมาจากปัญญาประดิษฐ์ค่ะ ✨ เป็นเทคโนโลยีที่ทำให้คอมพิวเตอร์คิดและเรียนรู้ได้เหมือนมนุษย์ค่ะ

👤 ผู้ใช้: ขอบใจมากนะพี่บ่าว
🤖 Agent: ยินดีให้บริการค่ะ! 😊
```

```python
runner = InMemoryRunner(agent=health_agent, app_name=health_agent.name)
try:
    await runner.session_service.create_session(app_name=health_agent.name, user_id='user_1', session_id='test_session')
except Exception:
    pass  # session already exists
# ── ทดสอบ Tool Selection ──
print('=' * 60)
print('🧪 ทดสอบ: Agent เลือก Tool ไหน?')
print('=' * 60)

test_messages = [
    'น้ำหนัก 55 kg สูง 165 cm BMI เท่าไหร่',
    '0 องศาเซลเซียส เท่ากับกี่ฟาเรนไฮต์',
    'kku คืออะไร',  # ไม่ต้องใช้ tool!
]

for msg in test_messages:
    print(f'\n👤 ผู้ใช้: {msg}')
    response = await chat_with_agent(runner, msg, session_id='test_session')
    print(f'🤖 Agent: {response}')
```

```
...
============================================================
🧪 ทดสอบ: Agent เลือก Tool ไหน?
============================================================

👤 ผู้ใช้: น้ำหนัก 55 kg สูง 165 cm BMI เท่าไหร่
  🔧 Tools เรียกใช้: ['calculate_bmi']
🤖 Agent: ค่า BMI ของคุณคือ 20.2 ซึ่งจัดอยู่ในเกณฑ์น้ำหนักปกติ รักษาน้ำหนักนี้ไว้และออกกำลังกายสม่ำเสมอเพื่อสุขภาพที่ดี


👤 ผู้ใช้: 0 องศาเซลเซียส เท่ากับกี่ฟาเรนไฮต์
  🔧 Tools เรียกใช้: ['convert_temperature']
🤖 Agent: 0 องศาเซลเซียส เท่ากับ 32.0 องศาฟาเรนไฮต์

👤 ผู้ใช้: kku คืออะไร
🤖 Agent: KKU ย่อมาจาก มหาวิทยาลัยขอนแก่น (Khon Kaen University) ซึ่งเป็นมหาวิทยาลัยของรัฐที่ตั้งอยู่ในจังหวัดขอนแก่น ประเทศไทย เป็นหนึ่งในมหาวิทยาลัยชั้นนำของประเทศไทยค่ะ
```

# Rag tool

```
# ── Demo: Agent ตัดสินใจเอง ──
print('=' * 60)
print('🧪 Demo: Agent เลือกเมื่อไหร่จะค้นหา เมื่อไหร่จะตอบเอง')
print('=' * 60)

await demo_chat(rag_agent, [
    'สวัสดี',                                  # → ตอบเอง
    'ธนาคารกสิกรไทยมีผลงานอย่างไร',            # → ค้น Qdrant
  #  'RAG มีสถาปัตยกรรมอย่างไร',               # → ค้น Qdrant
    'ขอบคุณครับ',                              # → ตอบเอง
])
```

```
────────────────────────────────────────────────────────────
🧪 Demo: Agent เลือกเมื่อไหร่จะค้นหา เมื่อไหร่จะตอบเอง
────────────────────────────────────────────────────────────

👤 ผู้ใช้: สวัสดี
🤖 Agent: สวัสดีค่ะ มีอะไรให้ช่วยเกี่ยวกับการค้นหาข้อมูล AI ในประเทศไทยไหมคะ?

👤 ผู้ใช้: ธนาคารกสิกรไทยมีผลงานอย่างไร
  🔧 Tools เรียกใช้: ['search_thai_ai_knowledge']
🤖 Agent: ธนาคารกสิกรไทยมีผลงานด้าน AI ที่น่าสนใจดังนี้ค่ะ (อ้างอิงจากฐานความรู้ด้าน Banking):

*   **พัฒนา KADE AI Assistant:** เป็น AI ผู้ช่วยที่ใช้เทคนิค RAG (Retrieval-Augmented Generation) เพื่อดึงข้อมูลผลิตภัณฑ์ทางการเงินมาตอบลูกค้า
*   **ลดระยะเวลาตอบคำถาม:** สามารถลดเวลาในการตอบคำถามลูกค้าจาก 5 นาที เหลือเพียง 30 วินาที
*   **รองรับภาษา:** ระบบสามารถเข้าใจได้ทั้งภาษาไทยและภาษาอังกฤษ
*   **จัดการคำถามหลากหลาย:** สามารถจัดการคำถามเกี่ยวกับสินเชื่อ บัตรเครดิต และการลงทุนได้โดยอัตโนมัติ

👤 ผู้ใช้: ขอบคุณครับ
🤖 Agent: ยินดีค่ะ มีคำถามอื่น ๆ เกี่ยวกับ AI ในประเทศไทยอีกไหมคะ?
```

# LOOP AGENT

```
runner = InMemoryRunner(agent=loop_refiner, app_name=loop_refiner.name)
try:
    await runner.session_service.create_session(app_name=loop_refiner.name, user_id='user_1', session_id='test_session')
except Exception:
    pass  # session already exists
# ── ทดสอบ Loop Agent ──
print('=' * 60)
print('🧪 LoopAgent: เขียน → ตรวจ → แก้ไข → ตรวจ ...')
print('=' * 60)

response = await chat_with_agent(runner, 'สรุปว่า RAG ทำงานอย่างไร พร้อมตัวอย่างใช้จริงในไทย', session_id='test_session')
print(f'\n📋 ผลลัพธ์สุดท้าย (หลังปรับปรุง):')
print(response)
```

```
────────────────────────────────────────────────────────────
🧪 LoopAgent: เขียน → ตรวจ → แก้ไข → ตรวจ ...
────────────────────────────────────────────────────────────

  🔧 Tools เรียกใช้: ['search_thai_ai_knowledge']

📋 ผลลัพธ์สุดท้าย (หลังปรับปรุง):
APPROVED: RAG คือสถาปัตยกรรม AI ที่ช่วยให้ LLM สร้างคำตอบที่แม่นยำและน่าเชื่อถือขึ้น โดย Retriever ค้นหาข้อมูล, Reader ทำความเข้าใจ และ Generator สร้างคำตอบ ช่วยลดปัญหา hallucination ตัวอย่างใ
```

## Session & Memory

```
[34]
8s
        final_response = ''
        async for event in runner.run_async(
            user_id='student_demo',
            session_id='memory_demo',
            new_message=content
        ):
            if event.content and event.content.parts:
                for part in event.content.parts:
                    if part.text:
                        final_response = part.text

        print(f'🤖 Agent: {final_response}')

# ทดสอบ
await conversation_with_memory(greeting_agent, [
    'สวัสดีครับ ผมชื่อสมชาย',
    'ผมสนใจเรื่อง AI ในวงการสุขภาพ',
    'แล้วในวงการการเงินล่ะ',
    'เปรียบเทียบสองวงการนี้ให้หน่อย',
    'จำได้ไหมว่าผมชื่ออะไร?',
])

...
👤 ผู้ใช้: สวัสดีครับ ผมชื่อสมชาย
🤖 Agent: สวัสดีครับคุณสมชาย ยินดีที่ได้รู้จักนะครับ! 😊

👤 ผู้ใช้: ผมสนใจเรื่อง AI ในวงการสุขภาพ
🤖 Agent: น่าสนใจมากครับ! 🤖🏥 อยากให้ผมช่วยหาข้อมูลหรือตอบคำถามอะไรเกี่ยวกับเรื่องนี้ไหมครับ?

👤 ผู้ใช้: แล้วในวงการการเงินล่ะ
🤖 Agent: ในวงการการเงิน AI ก็มีบทบาทสำคัญมากครับ! 🏦 ใช้ในการวิเคราะห์ข้อมูล, ป้องกันการฉ้อโกง และปรับปรุงบริการลูกค้าครับ สนใจให้ผมหาข้อมูลเพิ่มเติมไหมครับ?

👤 ผู้ใช้: เปรียบเทียบสองวงการนี้ให้หน่อย
🤖 Agent: ทั้งสองวงการใช้ AI เพื่อวิเคราะห์ข้อมูลและตัดสินใจคล้ายกันครับ 📈 แต่ในสุขภาพจะเน้นการวินิจฉัยและรักษาผู้ป่วย 👨‍⚕️ ส่วนการเงินเน้นการวิเคราะห์ตลาด, ป้องกันการฉ้อโกง และบริหารความเสี่ยงครับ 💰

👤 ผู้ใช้: จำได้ไหมว่าผมชื่ออะไร?
🤖 Agent: จำได้ครับ คุณชื่อสมชายครับ! 😊

💡 สังเกต
```

## Agentic Rag Multi Agent

```
print('🎓 Demo: Study Assistant (Agentic RAG เต็มรูปแบบ)')
print('-' * 60)

await conversation_with_memory(study_assistant, [
    'ช่วยอธิบายว่า RAG ทำงานยังไง',
    'มีตัวอย่างใช้จริงในไทยมั้ย',
    'สรุปทั้งหมดให้หน่อย',
    'แปลสรุปเป็นภาษาลาว',
])

...
────────────────────────────────────────────────────────────
🎓 Demo: Study Assistant (Agentic RAG เต็มรูปแบบ)
────────────────────────────────────────────────────────────

👤 ผู้ใช้: ช่วยอธิบายว่า RAG ทำงานยังไง
🤖 Agent: RAG (Retrieval-Augmented Generation) ทำงานโดยมีส่วนประกอบหลัก 3 ส่วน ได้แก่:

1.  **Retriever**: ทำหน้าที่ค้นหาเอกสารที่เกี่ยวข้องจากฐานข้อมูล (VectorDB)
2.  **Reader**: ทำหน้าที่อ่านและทำความเข้าใจเนื้อหาจากเอกสารที่ Retriever ค้นพบ
3.  **Generator**: สร้างคำตอบโดยใช้ข้อมูลที่ Reader ได้ทำความเข้าใจ

ระบบ RAG ช่วยลดปัญหาการสร้างข้อมูลที่ผิดพลาด (hallucination) ของโมเดลภาษาขนาดใหญ่ (LLM)

👤 ผู้ใช้: มีตัวอย่างใช้จริงในไทยมั้ย
🤖 Agent: มีตัวอย่างการใช้งาน RAG ในประเทศไทยจริงดังนี้:

*   **ด้านการศึกษา**: จุฬาลงกรณ์มหาวิทยาลัยได้สร้างระบบ RAG ถาม-ตอบจากตำราเรียน โดยแปลงไฟล์ PDF มากกว่า 500 เล่มให้เป็น vector embeddings และใช้ multilingual model ที่รองรับภาษาไทย ทำให้นัก
*   **ด้านการธนาคาร**: ธนาคารกสิกรไทยได้พัฒนา KADE AI Assistant ซึ่งใช้ RAG ในการดึงข้อมูลผลิตภัณฑ์ทางการเงินมาตอบลูกค้า ช่วยลดเวลาในการตอบคำถามจาก 5 นาทีเหลือเพียง 30 วินาที ระบบนี้สามารถเข้าใจ

👤 ผู้ใช้: สรุปทั้งหมดให้หน่อย
🤖 Agent: *   🧠 **RAG (Retrieval-Augmented Generation)** ทำงานโดยมี 3 ส่วนหลัก: Retriever ค้นข้อมูลจาก VectorDB, Reader ทำความเข้าใจเอกสาร, และ Generator สร้างคำตอบจากข้อมูลที่พบ
*   🚫 ระบบนี้ช่วย **ลดปัญหาการสร้างข้อมูลที่ผิดพลาด (hallucination)** ของโมเดลภาษาขนาดใหญ่ (LLM)
*   🇹🇭 ในไทย, **จุฬาลงกรณ์มหาวิทยาลัย** นำ RAG มาใช้กับตำราเรียนกว่า 500 เล่ม เพื่อให้นักศึกษาถาม-ตอบพร้อมอ้างอิงได้
*   🏦 **ธนาคารกสิกรไทย** พัฒนา KADE AI Assistant ที่ใช้ RAG ดึงข้อมูลผลิตภัณฑ์การเงินมาตอบลูกค้า ลดเวลาตอบคำถามลงอย่างมาก

👤 ผู้ใช้: แปลสรุปเป็นภาษาลาว
🤖 Agent: *   🧠 **RAG (Retrieval-Augmented Generation)** ດຶງດວງກໂດຍມີ 3 ພາກສ່ວນຫຼັກ: Retriever ຄົ້ນຫາຂໍ້ມູນຈາກ VectorDB, Reader ເຂົ້າໃຈເອກະສານ, ແລະ Generator ສ້າງຄຳຕອບຈາກຂໍ້ມູນທີ່ພົບ
*   🚫 ລະບົບນີ້ຊ່ວຍ **ຫຼຸດຜ່ອນບັນຫາການສ້າງຂໍ້ມູນທີ່ຜິດພາດ (hallucination)** ຂອງຕົວແບບພາສາຂະໜາດໃຫຍ່ (LLM)
*   🇹🇭 ໃນໄທ, **ມະຫາວິທະຍາໄລຈຸລາລົງກອນ** ນຳ RAG ມາໃຊ້ກັບຕຳລາຮຽນຫຼາຍກວ່າ 500 ເຫຼັ່ມ, ເພື່ອໃຫ້ນັກສຶກສາສາມາດຖາມ-ຕອບພ້ອມອ້າງອີງໄດ້
*   🏦 **ທະນາຄານກະສິກອນໄທ** ພັດທະນາ KADE AI Assistant ທີ່ໃຊ້ RAG ດຶງຂໍ້ມູນຜະລິດຕະພັນທາງການເງິນມາຕອບລູກຄ້າ, ຫຼຸດຜ່ອນເວລາຕອບຄຳຖາມລົງຢ່າງຫຼວງຫຼາຍ
```

693040129-3_chawanakon_day3_evaluation.ipynb

# gemini-2.5-flash

```
🟢 faithfulness: 0.9500

🟢 answer_relevancy: 0.8531

🟢 context_precision: 1.0000

🟢 context_recall: 1.0000

CPU times: user 14.3 s, sys: 803 ms, total: 15.1 s

Wall time: 38 s
```

# gemini-3.1-flash-lite

```
🟢 faithfulness: 1.0000

🟢 answer_relevancy: 0.8932

🟢 context_precision: 1.0000

🟢 context_recall: 1.0000

CPU times: user 15.3 s, sys: 1.35 s, total: 16.6 s

Wall time: 31.2 s
```

# Tool Selection test

```
...
🧪 Test Suite: Tool Selection

  ✅ 'น้ำหนัก 70 สูง 175 BMI?...' → expected=calculate_bmi, got=calculate_bmi
  ✅ 'AI ช่วยหมอไทยยังไง...' → expected=search_knowledge, got=search_knowledge
  ✅ 'สวัสดีครับ...' → expected=None, got=None
  ✅ 'Qdrant คืออะไรอ่ะ...' → expected=search_knowledge, got=search_knowledge
  ✅ 'อากาศวันนี้เป็นยังไง...' → expected=None, got=None
  ✅ 'kbank มี ai อะไร...' → expected=search_knowledge, got=search_knowledge

📊 Pass rate: 6/6 (100%)
```

# LLM as a Judge

```
%%time
# ── LLM-as-Judge (เข้มงวดขึ้น) ──
def llm_judge(question, answer, max_score=5):
    prompt = f'''ให้คะแนนคำตอบ 1-{max_score} ตามเกณฑ์ (ให้คะแนนเข้มงวด):
- ถูกต้อง: ตรงกับข้อเท็จจริง ไม่มีข้อมูลที่แต่งขึ้น (ถ้าแต่งเพิ่ม หัก 2 คะแนน)
- อ้างอิง: มีแหล่งที่มาหรือตัวอย่างเจาะจง (ไม่มี = หัก 1 คะแนน)
- กระชับ: ตอบเป็น bullet points ไม่เกิน 5 ข้อ (ยาวเกิน = หัก 1 คะแนน)
- ตรงคำถาม: ตอบตรงประเด็น ไม่ออกนอกเรื่อง
- ขอบเขต: ถ้าข้อมูลไม่เพียงพอ ต้องบอกว่า "ไม่มีข้อมูล" (ไม่บอก = หัก 1 คะแนน)
- สุภาพ : ไม่มีคำลงท้าย คะ ค่ะ ครับ (ไม่มี = หัก 3 คะแนน)

คำถาม: {question}
คำตอบ: {answer}

ตอบ JSON: {{"score": 1-{max_score}, "reason": "เหตุผลสั้นๆ"}}'''
```

```
...
🧪 LLM-as-Judge: วัดคุณภาพคำตอบ

  ⭐⭐☆☆☆ 2/5 | AI ช่วยหมอไทยยังไง... → ขาดอ้างอิงและตัวเลขเฉพาะเจาะจงสำหรับหมอไทยไม่สามารถตรวจสอบได้ตามเกณ
  ⭐⭐⭐⭐☆ 4/5 | เปรียบเทียบ AI ด้านสาธารณสุข v... → ข้อมูลถูกต้อง ตรงประเด็น มีตัวอย่างเจาะจง แต่ไม่ได้จัดรูปแบบเป็
  ⭐⭐⭐☆☆ 3/5 | มีบริษัท AI ไทยกี่แห่งที่ IPO ... → ไม่มีการอ้างอิงและไม่ได้ตอบเป็น bullet points

📊 Average: 3.0/5.0
CPU times: user 1.83 s, sys: 11.8 ms, total: 1.84 s
Wall time: 34.3 s
```

# Prompt Tuning

```
🧪 Before vs After Prompt (คำถามยาก)

❓ เปรียบเทียบการใช้ AI ในโรงพยาบาลไทย vs ธนาคารไทย
  V1: 5/5 — คำตอบถูกต้อง ตรงประเด็น มีตัวอย่างเจาะจง กระชับ และไม่มีคำลงท้าย
  V2: 2/5 — ไม่มีคำลงท้ายสุภาพ (-3)

❓ ประเทศไทยเป็นผู้นำ AI ในอาเซียนจริงหรือไม่
  V1: 3/5 — หัก 1 คะแนนเนื่องจากไม่มีแหล่งอ้างอิง หัก 1 คะแนนเนื่องจากไม่ได้ตอบเป็น bullet points
  V2: 4/5 — คำตอบถูกต้อง ตรงคำถาม และระบุว่าไม่มีข้อมูลตามเกณฑ์ แต่ไม่ได้ตอบเป็น bullet points

❓ สรุปปัญหา NLP ภาษาไทยและวิธีแก้ให้ครบ
  V1: 4/5 — คำตอบถูกต้อง มีการอ้างอิงเครื่องมือและโมเดลเฉพาะ แต่มี bullet points เกิน 5 ข้อ
  V2: 5/5 — คำตอบถูกต้อง กระชับ ตรงประเด็น มีตัวอย่างอ้างอิง และไม่มีคำลงท้าย

📊 สรุป: V1 avg = 4.0/5  |  V2 avg = 3.7/5
📉 V1 ดีกว่า V2 — ลองปรับ instruction ใหม่
```

```
prompt_after = '''คุณเป็นผู้เชี่ยวชาญ AI ในประเทศไทย ตอบเป็นภาษาไทย

กฎ:
1. ค้นหาข้อมูลจาก search_knowledge ก่อนตอบเสมอ
2. ตอบเฉพาะข้อมูลที่พบ ห้ามเดาหรือแต่งเพิ่ม
3. อ้างอิงแหล่งที่มา เช่น [healthcare] [banking]
4. ถ้าไม่พบข้อมูล บอกตรงๆ ว่า "ไม่มีข้อมูลในฐานความรู้"
5. ตอบกระชับ เป็น bullet points ไม่เกิน 5 ข้อ
6. ใช้คำสุภาพ มีคำลงท้ายเสมอ'''
```

```
🧪 Before vs After Prompt (คำถามยาก)

❓ เปรียบเทียบการใช้ AI ในโรงพยาบาลไทย vs ธนาคารไทย
  V1: 1/5 — คำตอบตรงประเด็นและมีตัวอย่างเจาะจง แต่ใช้ bullet points เกิน 5 ข้อ (8 ข้อ) และไม่มีคำลงท้ายที่สุภาพครับ/
  V2: 5/5 — คำตอบถูกต้อง กระชับ ตรงประเด็น มีตัวอย่างเจาะจง และมีคำลงท้ายครับ

❓ ประเทศไทยเป็นผู้นำ AI ในอาเซียนจริงหรือไม่
  V1: 2/5 — คำตอบตรงประเด็นและระบุว่าไม่มีข้อมูลชัดเจน พร้อมยกตัวอย่างประกอบได้ดี แต่ขาดคำลงท้ายครับ/ค่ะ
  V2: 5/5 — ตอบตรงคำถามว่าไม่มีข้อมูลที่ชัดเจน พร้อมให้ตัวอย่างประกอบที่กระชับ มีการอ้างอิง และสุภาพครบถ้วนตามเกณฑ์

❓ สรุปปัญหา NLP ภาษาไทยและวิธีแก้ให้ครบนะครับ
  V1: 2/5 — เนื้อหาถูกต้อง กระชับ และมีตัวอย่างประกอบ แต่ขาดคำลงท้ายครับตามเกณฑ์ที่กำหนด
  V2: 2/5 — อ้างอิงไม่เจาะจง และวิธีแก้ส่วนใหญ่ไม่ตรงกับปัญหาที่ระบุ

📊 สรุป: V1 avg = 1.7/5  |  V2 avg = 4.0/5
📈 V2 ดีกว่า V1 +2.3 คะแนน — instruction ที่ดีช่วยได้จริง!
```