NitiBench: การประเมินขีดความสามารถของ LLM สำหรับการตอบคำถามทางกฎหมายไทย

เจาะลึกการเปรียบเทียบ RAG, Long-Context LLMs และเทคนิคเฉพาะทางสำหรับกฎหมายไทย

รายงานทางเทคนิคโดย VISAI AI และ VISTEC

NotebookLM

บทสรุปผู้บริหาร: กุญแจสำคัญสู่ AI ทางกฎหมาย

โครงสร้างคือกุญแจสำคัญ

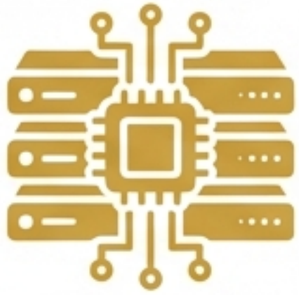
การแบ่งข้อมูลตามมาตรา (Section-based Chunking) เพิ่มประสิทธิภาพการค้นคืนข้อมูลและผลลัพธ์อย่างมีนัยสำคัญ

RAG ชนะ Long-Context

โมเดล Long-Context ยังไม่สามารถทดแทน RAG ได้ โดยเฉพาะในงานที่ต้องการความแม่นยำสูง

ความท้าทายด้านภาษี

โมเดลปัจจุบันยังประสบปัญหาในการให้เหตุผลทางกฎหมายที่ซับซ้อน (Legal Reasoning) โดยเฉพาะคดีภาษีอากร

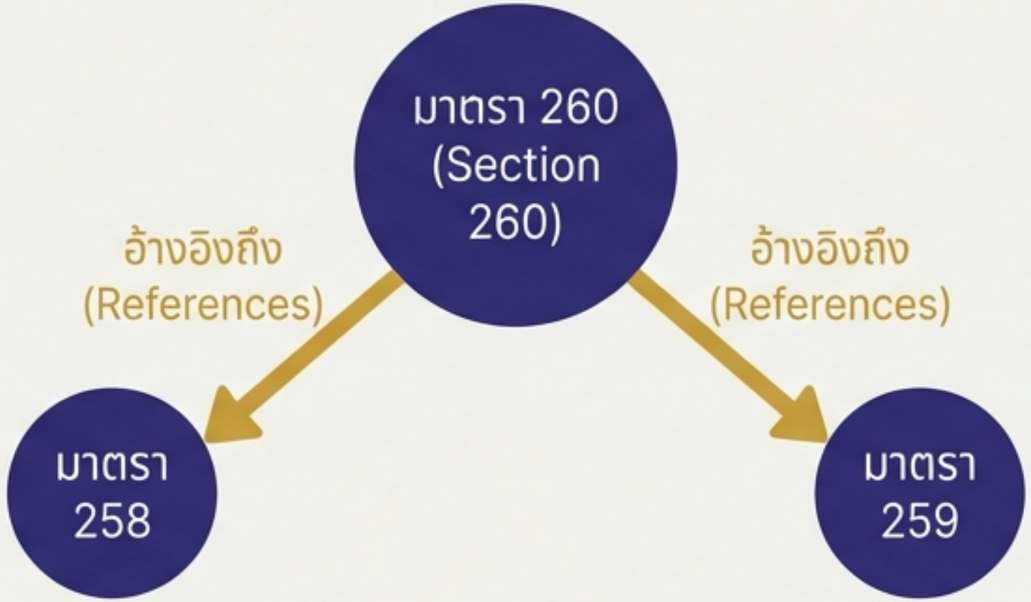


เครื่องมือที่ดีที่สุด

การใช้ BGE-M3 (Fine-tuned) ร่วมกับ Claude 3.5 Sonnet ให้ผลลัพธ์ที่ดีที่สุด

NotebookLM

ความท้าทายเฉพาะตัวของกฎหมายไทย



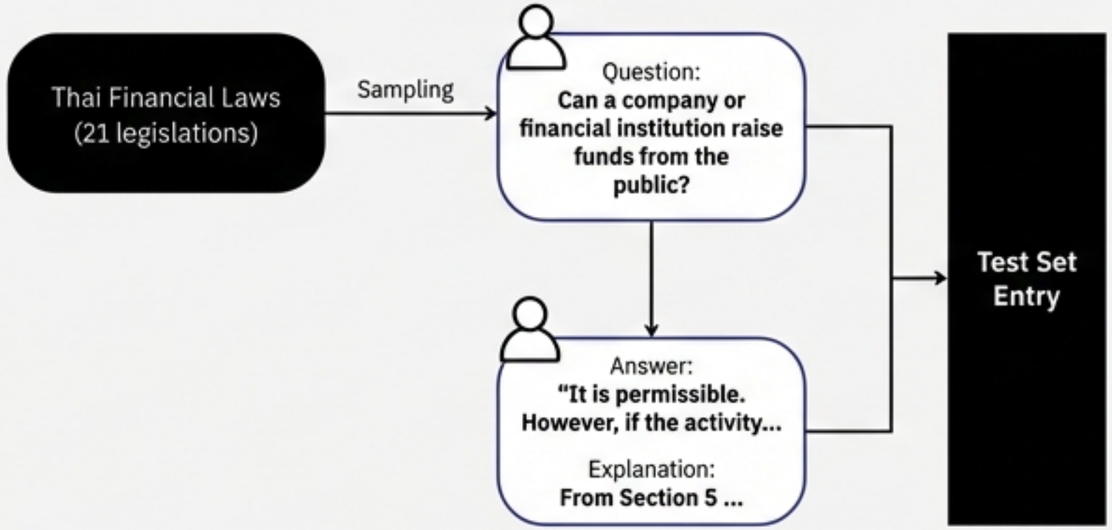
ปัญหา (The Problem): การแบ่งข้อมูลแบบดั้งเดิม (Generic Chunking) ทำลายความเชื่อมโยงเหล่านี้ ทำให้ AI สูญเสียบริบท

NotebookLM

NitiBench: เบนซ์มาร์กมาตรฐานใหม่

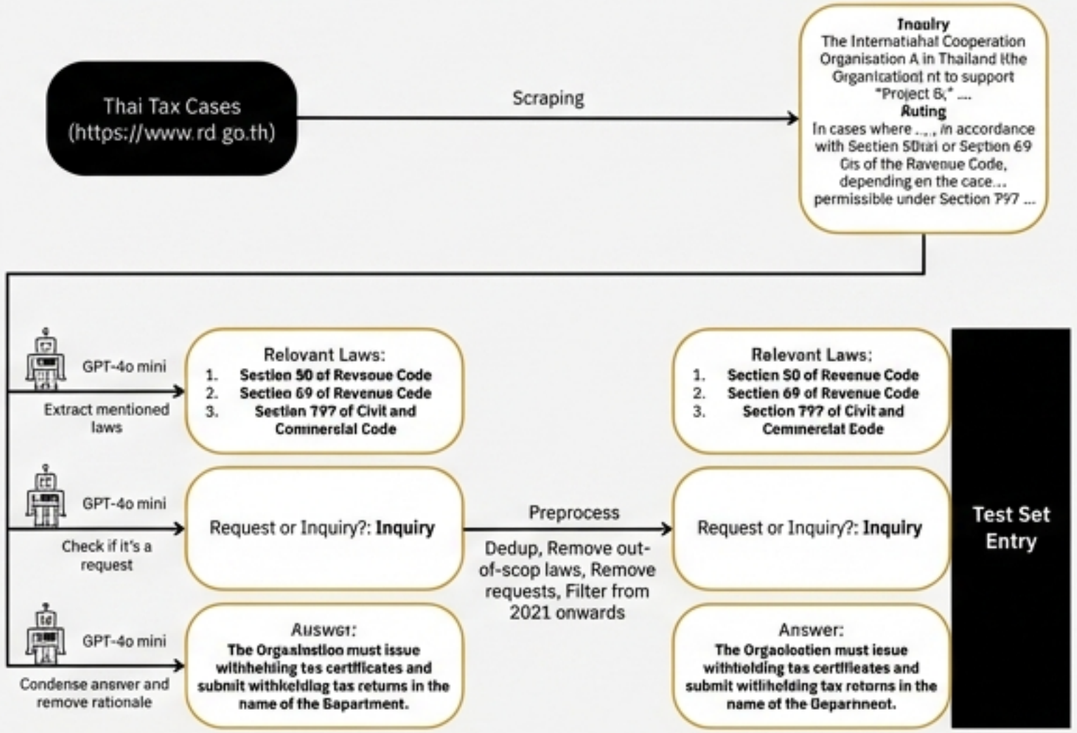
NitiBench-CCL (กฎหมายการเงินทั่วไป)

- ครอบคลุมกฎหมาย 21 ฉบับ
- 3,730 คำถาม-คำตอบ
- เน้นความรู้พื้นฐาน (Single-hop retrieval)



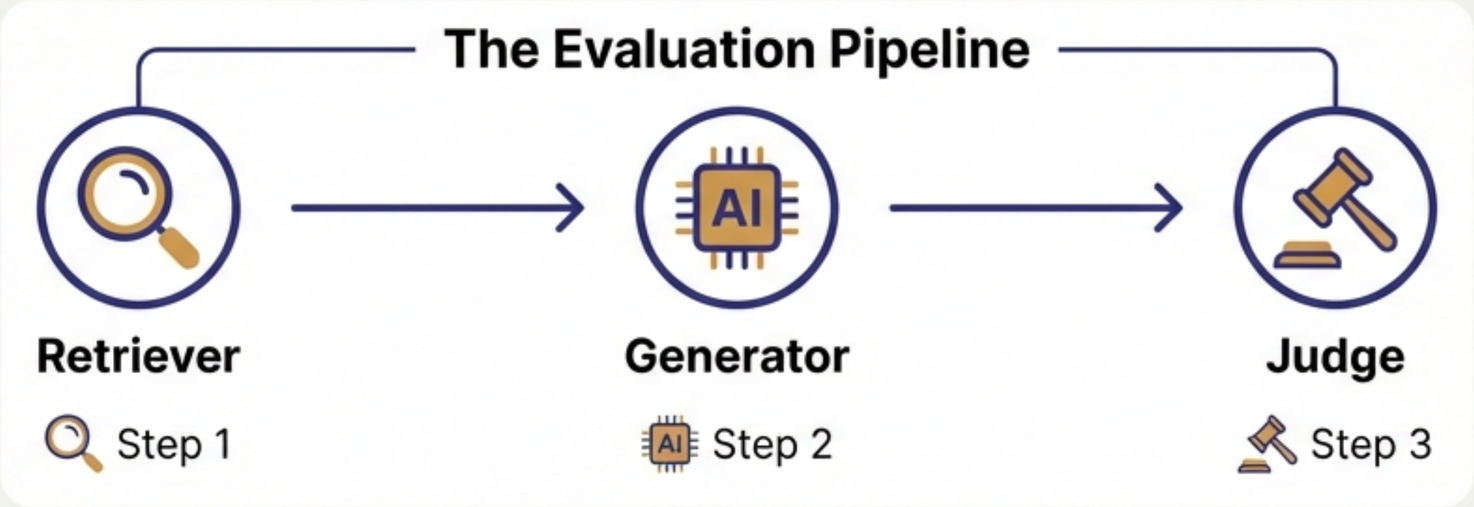
NitiBench-Tax (คดีภาษีอากร)

- 50 คดีจริงจากกรมสรรพากร
- เน้นการให้เหตุผลทางกฎหมายที่ซับซ้อน
- ต้องใช้กฎหมายหลายมาตราประกอบกัน



NotebookLM

เราวัดผลความสำเร็จอย่างไร?



การค้นคืน (Retrieval Metrics)

- ใช้ Multi-label Metrics (HitRate, Recall, MRR)
- รองรับคำถามที่ต้องใช้กฎหมายหลายมาตรา

ความถูกต้อง (End-to-End Metrics)

- **Coverage**: คำตอบครอบคลุมประเด็นสำคัญหรือไม่? (0-100)
- **Contradiction**: คำตอบขัดแย้งกับข้อเท็จจริงหรือไม่? (วัด Hallucination)
- **Citation**: การอ้างอิงมาตรากฎหมายถูกต้องหรือไม่?

NotebookLM

RQ1: โครงสร้างข้อมูลมีผลอย่างไร?

Naive Chunking (แบบดั้งเดิม)

พระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์
พ.ศ 2550 มาตรา 14 ผู้ใดกระทำความผิดที่ระบุไว้

ดังต่อไปนี้ ต้องระวางโทษจำคุกไม่เกินห้าปี หรือปรับไม่เกินหนึ่งแสนบาท หรือทั้งจำทั้งปรับ

❌ **Information Loss (สูญเสียข้อมูล)**

Hierarchy-Aware Chunking (แบบอิงมาตรา)

พระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. 2550
มาตรา 14 ผู้ใดกระทำความผิดที่ระบุไว้ดังต่อไปนี้

Section A ····· ✔ ····· Section B

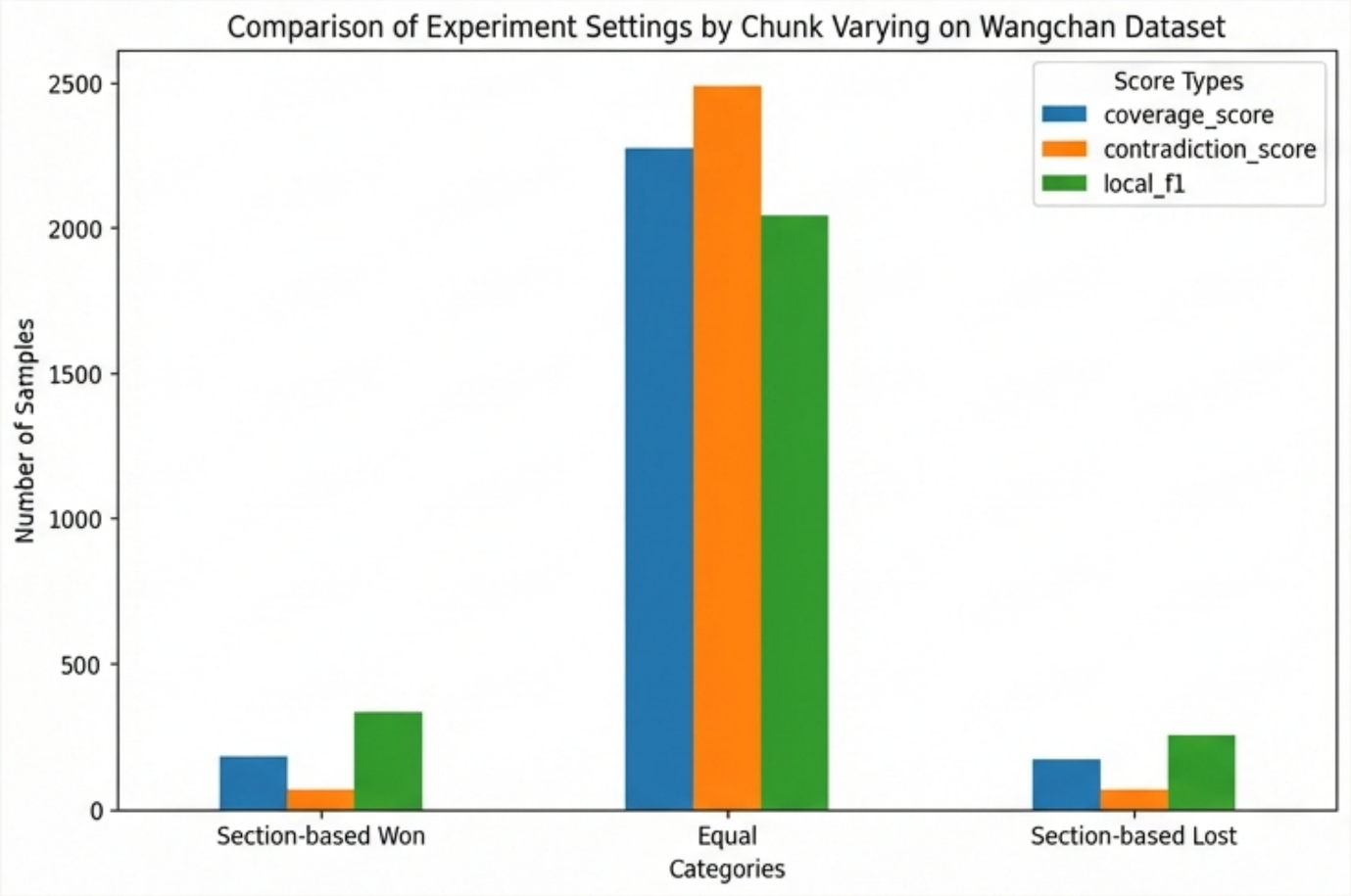
ดังต่อไปนี้ ต้องระวางโทษจำคุกไม่เกินห้าปี หรือปรับไม่เกินหนึ่งแสนบาท หรือทั้งจำทั้งปรับ

✅ **Preserves Context (รักษาบริบท)**

- **สมมติฐาน:** การแบ่งข้อมูลตามบรรทัด (Line Chunking) ทำให้เนื้อหากฎหมายขาดความสมบูรณ์
- ****แนวทางแก้ไข:**** Hierarchy-Aware Chunking แบ่งข้อมูลตามขอบเขตมาตรากฎหมายจริง

NotebookLM

ผลลัพธ์: การจัดเตรียมข้อมูลคือก้าวแรกแห่งความสำเร็จ



Key Findings

Chunking: Hierarchy-Aware Chunking ชนะ Naive Chunking ในทุกมิติ (Retrieval & E2E Score)

NitiLink (การดึงข้อมูลข้ามมาตรา):

- ข้อดี: เพิ่มค่า Recall บนชุดข้อมูล Tax อย่างมาก (0.437 -> 0.549)
- ข้อจำกัด: ไม่ได้รับประกันว่าคำตอบสุดท้ายจะดีขึ้นเสมอไป (LLM อาจสับสนกับข้อมูลที่มากเกินไป)

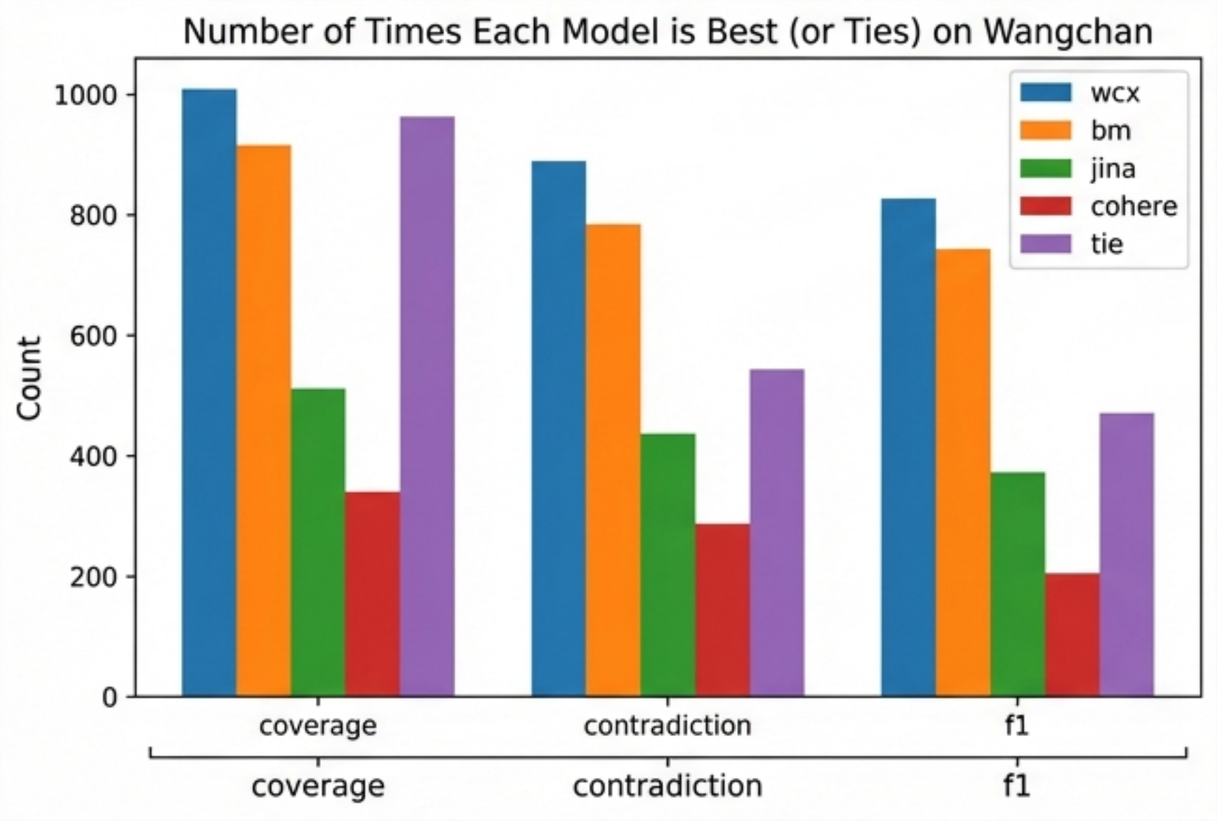
NotebookLM

RQ2: การค้นหาโมเดลที่ใช่

Retrievers (โมเดลค้นหา)	
• BM25 (Baseline) • Jina V3 & Cohere (Commercial) • **BGE-M3** (Open Source - Tested Base, Auto-Fine-tuned, & Human-Fine-tuned)	
	Generators (LLMs) • GPT-4o • Claude 3.5 Sonnet • Gemini 1.5 Pro • Typhoon V2 (Thai Open Source)

NotebookLM

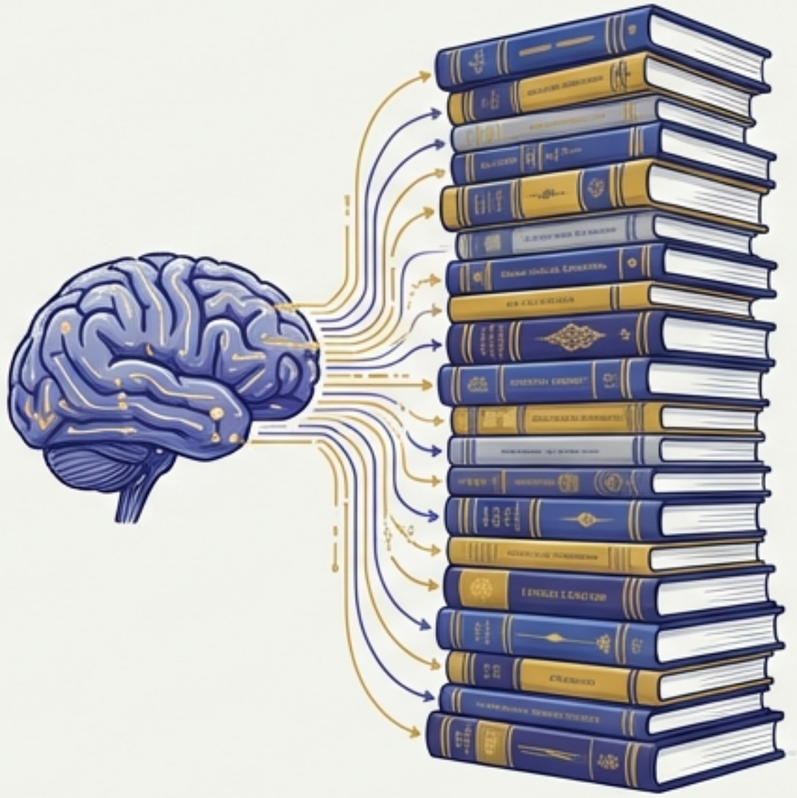
ผู้ชนะ BGE-M3 และ Claude 3.5 Sonnet



- **Retriever:** Human-Finetuned BGE-M3 ทำคะแนนสูงสุด (MRR 0.805)
- **LLM:** Claude 3.5 Sonnet ให้คำตอบที่ครอบคลุมและถูกต้องที่สุด (Best E2E Score)
- **Local vs Global:** Typhoon (Open Source ไทย) ทำได้ดีแต่ยังตามหลังโมเดลระดับโลก

NotebookLM

RQ3: RAG vs. Long-Context LLMs

Approach A (RAG)	Approach B (Long-Context)
	

สมมติฐาน: โมเดลที่รองรับบริบทได้ยาวมหาศาล (เช่น Gemini 1.5 Pro) จะสามารถอ่านกฎหมายทั้งหมดและตอบคำถามได้โดยไม่ต้องใช้ระบบค้นหา (Retriever) หรือไม่?

NotebookLM

ความจริง: RAG ยังคงเป็นมาตรฐาน

NitiBench-CCL	**RAG** (F1 0.607) ≈ LCLM (F1 0.615)
NitiBench-Tax	**RAG** > LCLM (RAG มีความแม่นยำสูงกว่า)

Verdict: Long-Context ยัง**ไม่สามารถทดแทน** RAG ได้ เนื่องจากปัญหาด้านความแม่นยำในการอ้างอิง (Citation) และ Hallucination เมื่อต้องจัดการกับข้อมูลกฎหมายจำนวนมหาศาล

NotebookLM

ทำไม AI ถึงพลาด? ปัญหาจากการค้นคืน

ลำดับชั้นที่ซ่อนอยู่ (Hidden Hierarchical Information)

ข้อความซ้ำกันในกฎหมายคนละฉบับ ทำให้ AI สับสน (เช่น กฎหมายภาษี vs กฎหมายปิโตรเลียม)

โครงสร้างซ้อนทับ (Nested Structure)

กฎหมายที่ระบุ 'บทลงโทษ' มักแยกออกจาก 'ข้อห้าม' ทำให้ AI หาความเชื่อมโยงไม่เจอ

รายละเอียดที่ถูกมองข้าม (Missable Details)

Keyword Matching ทั่วไปอาจพลาดบริบทที่ละเอียดอ่อน (เช่น 'ผู้ค้ำประกัน' vs 'ผู้รับอาวัล')

NotebookLM

กับดักของความกว้าง: ปัญหา Generic Section

Step 1: ผู้ใช้ถาม: ‘ภาษีมูลค่าเพิ่มของเวชภัณฑ์สัตว์’ (VAT on animal medicine)

AI ค้นหาด้วยคีย์เวิร์ด: ‘เวชภัณฑ์สัตว์’

มาตรา 77/2 (ประมวลรัษฎากร)

การขายสินค้าทุกชนิดต้องเสียภาษี

?

Step 4 (ผลลัพธ์):
AI ไม่สามารถให้คำตอบได้ เพราะมาตรา 77/2 กว้างเกินไป ไม่ตรงกับคีย์เวิร์ดที่เจาะจง

AI มัก**มองข้ามกฎหมายพื้นฐาน** (Foundational Laws) เพื่อไปหาคำตอบที่ดูซับซ้อนกว่า

NotebookLM

ก้าวต่อไปของ Legal AI ไทย

Step 1: Reasoning Evaluation

พัฒนาการวัดผล ‘กระบวนการคิด’ ไม่ใช่แค่คำตอบสุดท้าย

Step 2: Better Retrieval

แก้ปัญหา ‘Generic Section’
อาจต้องใช้การระบุบริบทแบบ Explicit

Step 3: Hybrid Systems

ผสมผสาน RAG เข้ากับ Knowledge Graphs
เพื่อแก้ปัญหาการอ้างอิงข้ามมาตรา

NotebookLM

บทสรุปและคำแนะนำ

1. **Architecture**: ยึดมั่นในระบบ RAG (Stick to RAG)
2. **Data**: ต้องทำ Hierarchy-Aware Chunking (Chunk by Section)
3. **Model**: ใช้ BGE-M3 + Claude 3.5 Sonnet (Current SOTA)

ข้อมูลและโค้ดทั้งหมดเปิดเป็น Open Source ที่ HuggingFace/Github
https://huggingface.co/datasets/VISAI-AI/nitibench

NotebookLM