**WEEK03 WEEK04 WEEK05 Report**

**By Nonpawit Silabumrungrad, Natchanan Lordee**

**Key Point #**

1. เรียกใช้งาน API โมเดล OCR จาก Google Vision
2. เรียกใช้งาน API โมเดล OCR จาก PyTesseract
3. พัฒนาสคริปต์วัดประสิทธิภาพโมเดล OCR จากทั้งสอง API ด้วยข้อมูลที่มีลักษณะเป็นฉลากยา
4. พัฒนาแอปพลิเคชันตามหน้าออกแบบใน Figma
5. เชื่อมต่อ Back-End และแอปพลิเคชัน
6. เลือกโมเดล LLM ที่จะทำ Fine-Tuning
7. เตรียมข้อมูลสำหรับการทำ Fine-Tuning โมเดล LLM
8. เรียกใช้งาน API โมเดล OCR จาก Google Vision
9. เรียกใช้งาน API โมเดล OCR จาก PyTesseract
10. พัฒนาสคริปต์วัดประสิทธิภาพโมเดล OCR จากทั้งสอง API ด้วยข้อมูลที่มีลักษณะเป็นฉลากยา

Metric ที่ใช้ในการวัดผลประสิทธิภาพของโมเดล OCR จากทั้งสอง API Platform มีดังนี้

1. **Character Error Rate (CER)**

เมตริกนี้พิจารณาในระดับ Character-Level เป็นการเปรียบเทียบว่าผลทำนายจากโมเดล (Predicted Text) นั้น ต้องแก้ไขข้อความมากขนาดไหนถึงจะตรงกับข้อความเฉลย (Ground Truth Text)

**วิธีการคำนวณ**

เป็นการคำนวณจาก Levenshtein Distance ซึ่งเป็นจำนวนการแก้ไขที่ต้องทำ มี 3 รูปแบบ ดังนี้ 1. Insertion (เพิ่มตัวอักษร) 2. Deletion (ลบตัวอักษร) 3. Substitution (แทนที่ตัวอักษร)

CER = จำนวนการแก้ไขทั้งหมด / จำนวนตัวอักษรทั้งหมด

**ตัวอย่างเช่น**

“ยาแก้ไอ” เป็น Ground Truth Text และ “ยาแก้ไข” เป็น Predicted Text

จำนวนตัวอักษรมีทั้งหมด 5 ตัว มีจุดที่ต้องแก้ไข 1 จุด คำตอบ 1/5 = 0.2 หรือ 20%

หมายความว่าผิดทั้งหมด 20% จากข้อความทั้งหมด (ยิ่งเยอะยิ่งไม่ดี)

1. **Word Error Rate (WER)**
2. พัฒนาแอปพลิเคชันตามหน้าออกแบบใน Figma
3. เชื่อมต่อ Back-End และแอปพลิเคชัน
4. เลือกโมเดล LLM ที่จะทำ Fine-Tuning
5. เตรียมข้อมูลสำหรับการทำ Fine-Tuning โมเดล LLM