

## DADS5001's dash application with LLM

### วิธีการรวบรวมข้อมูล

เริ่มต้นจากการรับข้อความ input ตัวอักษร จากลูกค้า ผ่านเว็บไซต์ของเรา จากนั้นนำ input ไปผ่าน language detection library ที่มีชื่อว่า “Lingua” โดยขอบเขตของโมเดลเราจะใช้ 4 ภาษา ดังนี้ ภาษา อังกฤษ ภาษาฝรั่งเศส ภาษาสเปน และภาษาเยอรมัน

หาก input เป็นภาษาอื่นที่ไม่ใช่ภาษาอังกฤษ จะถูกแปลงเป็นภาษาอังกฤษก่อน ด้วย translator api ที่มีชื่อว่า “Helsinki-NLP/opus-mt”

ข้อความภาษาอังกฤษที่ผ่านการแปลมาเรียบร้อยแล้ว จะถูกนำไปประมวลผลด้วย zero-shot text classification api ที่มีชื่อว่า “facebook/bart-large-mnli” ซึ่งเป็น api ที่จะช่วยให้เราสามารถหาค่า likelihood ระหว่าง input กับ class names ซึ่งทางเราได้ทำการปรับใช้ให้ input คือ ข้อความตัวอักษรจากลูกค้า และ class names จะเป็นชื่อของแผนกต่าง

โดย class names มีดังนี้

1. Waterworks (ฝ่ายช่างประปา)
2. Electrical (ฝ่ายช่างไฟฟ้า)
3. Structural (ฝ่ายโครงสร้าง หรือโยธา)
4. Arborist (ฝ่ายดูแลสวน หรือรุกขกร)
5. Pesticide (ฝ่ายจัดการสัตว์ หรือแมลงรบกวน)

### วิธีการเก็บข้อมูล

หลังจากที่ลูกค้าได้ป้อนข้อมูลลงในเว็บไซต์ของเรา ก็จะถูกนำไปจัดเก็บใน database ในรูปแบบ SQL บน PostgreSQL โดย เราจะมี 2 ตารางของข้อมูล มีรายละเอียดดังนี้

1. Accounts เป็นตารางสำหรับเก็บข้อมูลการเข้าใช้งานบนเว็บไซต์ ของพนักงาน และลูกค้า ประกอบด้วยหลักดังนี้
  - a. id: เลขแสดงลำดับที่ผู้ใช้งานสมัครเข้ามาในระบบ
  - b. username: ชื่อผู้ใช้งานในระบบ
  - c. password: รหัสเข้าระบบของผู้ใช้งาน
  - d. name: ชื่อจริงและนามสกุลของผู้ใช้งาน
  - e. email: อีเมลของผู้ใช้งาน
  - f. address: ที่อยู่ของผู้ใช้งาน
  - g. user\_role: สถานะในระบบของผู้ใช้งาน
2. Orders เป็นตารางสำหรับเก็บข้อมูลคำสั่งของลูกค้าที่ได้ทำการป้อนเข้ามาในระบบ ผ่านกล่องข้อความบนเว็บไซต์ ประกอบด้วยหลักดังนี้
  - a. id: เลขแสดงลำดับคิวคำสั่งของลูกค้า
  - b. user\_id: เลขแสดงลำดับที่ผู้ใช้งานสมัครเข้ามาในระบบ
  - c. timestamp: เวลา วัน เดือน ปี ชั่วโมง นาที ที่คำสั่งถูกส่งเข้ามาในระบบ
  - d. order\_details: รายละเอียดของคำสั่งที่ผู้ใช้งานแจ้งเข้ามาในระบบ
  - e. status: สถานะของคำสั่งว่าถูกดำเนินการไปหรือยัง
  - f. type: คำสั่งนี้เป็นงานของแผนกใด
  - g. percent: ตัวเลขค่า likelihood ของคำสั่งต่อแผนกในหลัก “type”

### ผลการวิเคราะห์ข้อมูล และการนำเสนอข้อมูล

Dash ของทีมเรา จะมีการแสดงผลเป็น 4 หน้าจอ โดยผู้ที่สามารถเห็นหน้า dash นี้ได้ ต้องเป็นพนักงานของบริษัทเท่านั้น โดยจะแสดงผลแตกต่างกันตาม role และแผนกของผู้ใช้งาน

1. Geography map ที่ใช้แสดงรัฐที่ลูกค้าต้องการให้เข้าดำเนินงาน
2. Data logs ที่ใช้แสดง request queue ของลูกค้า ตามรัฐที่ผู้ใช้งานเลือก
3. Pie chart ที่ใช้แสดงแผนกของทีมช่างที่จะถูกส่งไปหน่วยงานของรายการงานนั้น
4. Histograms แสดงความถี่ของงานในฝ่ายของผู้ใช้งาน ที่ลูกค้าส่งมาในแต่ละรัฐ

โดยแผนกที่ถูกเลือก จะถูกกำหนด จากการคำนวณด้วย bart-large-mnli โดยจะเลือกแผนกที่มีค่า likelihood ออกมาสูงที่สุด เพราะมีความสอดคล้องกับเนื้อหาที่ต้องทำมากที่สุด

### Idea ของการนำ LLM มาใช้

เนื่องจากทีมของเรา ได้มีความสนใจในการแก้ไข pain point ที่เกิดขึ้นใน แผนก customer service ที่ต้องมีความรู้ที่เกี่ยวข้องกับเนื้อหาอื่นๆ และต้องมีความเข้าใจในภาษาอื่นๆ ซึ่งทำให้

1. แผนกนี้จำเป็นต้องมีการ training ทักษะและองค์ความรู้ต่างๆ ให้สามารถเข้าใจปัญหาของลูกค้า แล้วสามารถประสานงานต่อไปยังแผนกอื่นๆ ได้ ซึ่งการ training นั้น เสียเวลา และค่าใช้จ่าย ทีมของเราคิดว่าด้วย api นี้ จะสามารถแก้ไข pain point นี้ได้
2. การตีความของพนักงานที่ขาดความชำนาญในภาษาอื่นๆ หรือมีความรู้ทางวิศวกรรมไม่เพียงพอ อาจสร้างความผิดพลาด แล้วเกิดความเสียหายให้กับลูกค้า และบริษัทได้

ทางทีมของเราจึงเลือกใช้ NLP models เป็นจำนวน 3 รูปแบบ มาช่วยแก้ไขปัญหาดังกล่าว

1. Language detection model เป็นโมเดลที่ช่วยในการตรวจว่า ข้อความที่ผู้ใช้งานป้อนเข้าไป เป็นภาษาอะไร โดยเราเลือกใช้
  - a. Lingua
2. Translation models เป็นโมเดลที่ช่วยในการแปลภาษาจากข้อความที่ผู้ใช้งานป้อนเข้าไป โดยเราเลือกใช้
  - a. Helsinki-NLP/opus-mt-es-en
  - b. Helsinki-NLP/opus-mt-fr-en
  - c. Helsinki-NLP/opus-mt-de-en
3. Zero-shot classification model เป็นโมเดลที่ช่วยในการหาความสอดคล้องระหว่าง keywords ที่ผู้ใช้งานกำหนด และประโยคข้อความที่ผู้ใช้งานป้อนเข้าไป โดยเราเลือกใช้
  - a. facebook/bart-large-mnli

1.นาย จักรพันธ์ นิ่มทอง 6610412008  
2.นาย ภัทรดนัย เลิศกิตติสกุล 6610412011  
3.นาย รนกร บุญสวนเกริกชัย 6610412003