

Toxic comment classification Project Natural Language Processing

1. หลักการทำงาน

เป็น API สำหรับการทำนายผลด้านอารมณ์ (sentiment analysis) ของข้อความภาษาไทย โดยใช้โมเดลที่ผ่านการเทรนไว้และจัดเก็บในรูปแบบ ONNX (Open Neural Network Exchange). หลักการทำงานคือ:

1) รับข้อมูลผ่าน API

- ผู้ใช้งานส่งข้อความ JSON ผ่าน HTTP POST มาที่ API โดยใช้โครงสร้าง {"text": "ข้อความที่ต้องการวิเคราะห์"}

2) Preprocessing (การเตรียมข้อมูล)

- ข้อความถูกทำความสะอาดด้วยฟังก์ชัน `thai_clean_text` ซึ่งลบอักขระพิเศษ เช่น อีโมจิและช่องว่างซ้ำซ้อน และตัดคำภาษาไทยให้เหมาะสมสำหรับการวิเคราะห์
- ข้อความถูกแปลงจากรูปแบบตัวอักษรเป็น **Token IDs** โดยใช้ฟังก์ชัน `tokens_to_ids`

3) Inference (การพยากรณ์)

- ข้อมูล Token IDs ที่ได้ถูกส่งเข้าสู่โมเดล ONNX (`onnxruntime`) เพื่อคำนวณผลลัพธ์
- ผลลัพธ์จากโมเดลถูกแปลงจากค่าดัชนี (index) เป็นฉลาก (label) เช่น "positive", "negative"

4) การส่งผลลัพธ์กลับ

- API ส่งผลลัพธ์การวิเคราะห์กลับไปในรูปแบบ JSON เช่น {"prediction": "positive"}

2. วิธีการทาง NLP ที่ใช้

เทคนิค NLP มาใช้ในหลากหลายขั้นตอน ได้แก่:

(1) Tokenization (การตัดคำ)

- ใช้ `pythainlp.tokenize.word_tokenize` ในการตัดคำภาษาไทย เพราะภาษาไทยไม่มีการเว้นวรรคระหว่างคำ ทำให้ต้องใช้โมดูลที่เชี่ยวชาญในการระบุขอบเขตคำ

(2) Text Cleaning (การทำความสะอาดข้อความ)

- ฟังก์ชัน `thai_clean_text` ทำการลบข้อมูลที่ไม่จำเป็น เช่น:
 - อีโมจิ (`deEmojify`)
 - ช่องว่างซ้ำซ้อน
 - อักขระพิเศษ เช่น `\n`

(3) Text-to-Token Mapping (การแปลงข้อความเป็นตัวเลข)

- ข้อความที่ถูกตัดคำจะถูกแปลงเป็น Token IDs โดยอ้างอิงจากไฟล์ token2idx.json ซึ่งเก็บ mapping ของคำและหมายเลขไว้
- คำที่ไม่มีในพจนานุกรมจะถูกแทนด้วย unk_token_id (ID สำหรับ Unknown Words)

(4) Padding and Initialization

- ข้อความที่สั้นกว่าความยาวที่โมเดลต้องการจะถูกเติมค่าด้วย pad_token_id เพื่อความยาวที่เท่ากัน
- Token ID เริ่มต้นของทุกประโยคถูกตั้งด้วย init_token_id เพื่อบ่งชี้จุดเริ่มต้น

(5) Embedding and Feature Representation

- Token IDs ที่ได้จะถูกส่งต่อเข้าโมเดลที่มี embedding layer ภายใน ซึ่งแปลง Token IDs เป็นเวกเตอร์เชิงตัวเลขเพื่อวิเคราะห์ความหมาย

(6) Model Inference

- โมเดล ONNX ใช้ค่าดังกล่าวในการคำนวณความน่าจะเป็น (probability) ของแต่ละคลาส เช่น positive, negative, neutral
- ผลลัพธ์เป็น array ของความน่าจะเป็นที่เลือกค่ามากที่สุด (argmax) เพื่อระบุคลาส

(7) Mapping Output Index to Labels

- ผลลัพธ์จากโมเดลจะถูกแปลงกลับเป็นข้อความ (label) เช่น "positive", "negative" โดยอ้างอิงจากไฟล์ idx2lab.json

3. เทคนิคทาง NLP ที่ใช้

(a) Text Preprocessing

- ลบข้อมูลที่ไม่สำคัญ เช่น อีโมจิ ช่องว่างซ้ำซ้อน เพื่อให้ข้อความมีความสะอาดก่อนนำไปวิเคราะห์
- ใช้การตัดคำ (tokenization) เพื่อแยกคำแต่ละคำในภาษาไทยซึ่งไม่มีตัวเว้นวรรค

(b) Out-of-Vocabulary (OOV) Handling

- การจัดการคำที่ไม่มีในพจนานุกรม (unknown words) ด้วย unk_token_id เพื่อให้โมเดลยังสามารถทำงานได้แม้มีคำแปลกใหม่

(c) Sequence Padding

- ใช้ pad_token_id เพื่อเติมค่าให้ข้อความทุกข้อความมีความยาวเท่ากันในระหว่างการทำนาย

(d) Embedding Representation

- โมเดลภายใน ONNX มีการแปลง Token IDs เป็นเวกเตอร์เชิงตัวเลขใน embedding layer ซึ่งเป็นการลดมิติของข้อมูลและแปลงคำให้อยู่ในรูปแบบเชิงความหมาย

(e) ONNX Runtime Optimization

- โมเดลที่เทรนในรูปแบบ ONNX ถูกใช้เพื่อเพิ่มความเร็วในการพยากรณ์ (inference) เนื่องจาก ONNX รองรับการทำงานที่ปรับแต่งสำหรับฮาร์ดแวร์หลายแบบ

(f) Multi-Class Classification

- โมเดลทำงานในรูปแบบ multi-class classification โดยเลือกคลาสที่มีค่าความน่าจะเป็นสูงสุดจาก array ที่โมเดลคำนวณได้

สรุป

การประยุกต์ใช้ NLP อย่างครบถ้วนสำหรับงาน Sentiment Analysis ภาษาไทย ตั้งแต่การเตรียมข้อความ (preprocessing) การแปลงข้อความเป็นตัวเลข (tokenization และ embedding) การจัดการโมเดล (inference) และการจัดรูปผลลัพธ์ (output mapping) โดยใช้เทคนิค NLP ที่เหมาะสมกับลักษณะเฉพาะของภาษาไทยและเพิ่มประสิทธิภาพผ่าน ONNX Runtime

ตัวอย่างหน้าจการทำงาน

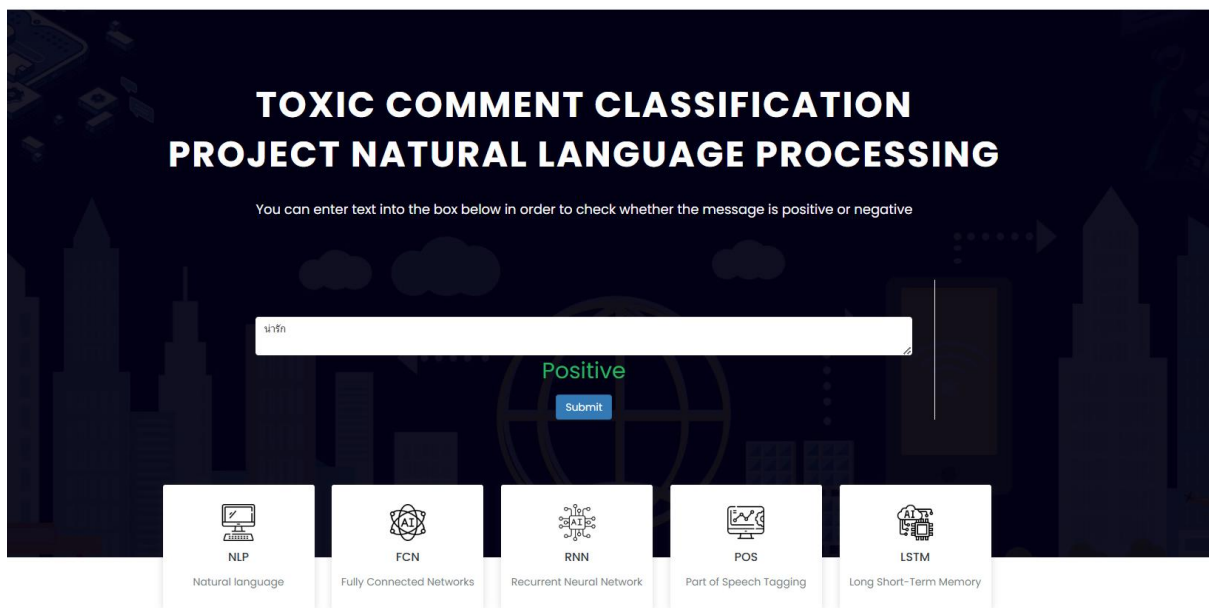
ขั้นตอนทำงาน

- 1) การรับข้อความ (Input Handling) : ผ่าน API /predict API ใช้ FastAPI และกำหนด endpoint /predict รองรับคำร้องแบบ HTTP POST ที่มีข้อมูล JSON รูปแบบ
- 2) การทำความสะอาดข้อความ (Text Cleaning) : ข้อความที่ได้รับจากผู้ใช้งานจะถูกส่งไปยังฟังก์ชัน `thai_clean_text` เพื่อเตรียมข้อความให้เหมาะสมกับการวิเคราะห์:
 - ลบอีโมจิ: ใช้ฟังก์ชัน `deEmojify` เพื่อลบอีโมจิและสัญลักษณ์พิเศษ
 - จัดรูปแบบข้อความ: ลบช่องว่างเกินจำเป็น และ ลบอักขระที่ไม่ต้องการ เช่น `\n`
 - ตัดคำ (Tokenization): ใช้ `pythainlp.tokenize.word_tokenize` ตัดข้อความออกเป็นคำตามภาษาไทย เช่น "สวัสดีครับ" → ["สวัสดี", "ครับ"]
- 3) การแปลงข้อความเป็น Token IDs (Text-to-Token Mapping) : ข้อความที่ตัดคำแล้ว (tokens) จะถูกแปลงเป็นตัวเลข (Token IDs) โดยฟังก์ชัน `tokens_to_ids`:
 - ใช้ `token_to_id` ซึ่งเป็นพจนานุกรมที่เชื่อมคำศัพท์กับตัวเลข
 - กรณีคำไม่อยู่ในพจนานุกรม (Out-of-Vocabulary): คำที่ไม่รู้จักจะถูกแทนด้วย `unk_token_id` (ค่า 2)
 - เพิ่ม Token เริ่มต้น: เพิ่ม `init_token_id` (ค่า 1) เป็น Token ตัวแรกของข้อความเตรียมข้อมูล: จัดรูปแบบข้อมูลในอาเรย์
- 4) การเตรียมข้อมูลสำหรับโมเดล (Data Preparation) ข้อมูล Token IDs ถูกจัดรูปแบบให้อยู่ในรูปของอาเรย์ NumPy (`np.array`) เพื่อส่งเข้าสู่โมเดล

- 5) การทำนายผลด้วยโมเดล (Model Inference) ใช้โมเดล ONNX (onnxruntime) สำหรับการพยากรณ์ผล:
 - โหลดโมเดลจากไฟล์ sentiment_model.onnx
 - กำหนดชื่ออินพุตและเอาต์พุตของโมเดล (input_name, output_name)
- 6) การแปลงผลลัพธ์โมเดล (Result Mapping) คำนวณคลาสที่มีความน่าจะเป็นสูงสุดด้วย np.argmax และแปลงดัชนี (index) ของคลาสเป็นฉลาก (label) เช่น "positive" หรือ "negative" โดยใช้ ids_to_labels
- 7) การส่งผลลัพธ์กลับ (Output Response) ส่งฉลากผลลัพธ์กลับไปยังผู้ใช้งานในรูปแบบ JSON
- 8) การจัดการคำร้องเพิ่มเติม (Optional) ระบบใช้ CORS Middleware เพื่ออนุญาตคำร้องจากทุกต้นทาง (origins=["*"]) ทำให้ API สามารถใช้งานได้จากทุกที่ รองรับการใช้คลาวด์เซอร์เวอร์โดยใช้ uvicorn เพื่อพัฒนาและปรับปรุงระบบอย่างต่อเนื่อง

ตัวอย่างหน้าจอระบบ

- 1) ถ้าเราป้อนข้อมูลเข้าไปในช่องแล้วก็ยืนยันระบบจะประมวลผลออกมาให้ว่าคำที่ป้อนไปเป็นข้อความเชิงบวกหรือเชิงลบ เช่น ป้อนคำว่า น่ารัก ระบบจะประมวลผลออกมาเป็น Positive



TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

บางชิ้นแสบปี้โธมสเลย

Positive

Submit

TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

ตัวเองน่ารักจังเลย

Positive

Submit

TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

หวานไปหน่อยสิ

Positive

Submit

- 2) ถ้าหากป้อนคำหยาบเพียง 1-2 คำ ระบบจะประมวลผลเป็นคำ Negative


TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING


You can enter text into the box below in order to check whether the message is positive or negative


ไม่รักกันกับออกมาอีตีด้อย


Negative


Submit


NLP


FCN


RNN


POS


LSTM

TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

ไม่ คัดถึง อี ตีด ซี สะ แม่ งง งง ง

Negative

Submit

TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

หน้าเมืองจันทระ

Negative

Submit

TOXIC COMMENT CLASSIFICATION PROJECT NATURAL LANGUAGE PROCESSING

You can enter text into the box below in order to check whether the message is positive or negative

เมืองหน้าเหิยาะ

Negative

Submit