ใบงานที่ 8 การสร้างระบบวิเคราะห์อารมณ์ความรู้สึก (Sentiment Analysis) ด้วย NLP และ LSTM

วัตถุประสงค์การเรียนรู้

1. ทำความสะอาดข้อความและแปลงคำศัพท์ภาษาไทยเป็นตัวเลข (Word Embedding) ด้วยโมเดล Word2Vec ได้
2. ปรับความยาวประโยคและสร้างโครงข่าย LSTM สำหรับวิเคราะห์ข้อความได้
3. อธิบายสถิติของข้อมูลและนำโมเดลไปใช้ทำนายอารมณ์ความรู้สึกของข้อความได้

เครื่องมือและอุปกรณ์

1. เครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรม PyCharm IDE
2. ชุดข้อมูลดิบ dataset.txt
3. ไฟล์โค้ดตั้งต้น ได้แก่ cleaned_dataset.py และ count_dataset.py
4. เอกสารใบความรู้หน่วยที่ 8
5. แบบฟอร์มบันทึกผลการปฏิบัติงาน (ท้ายใบงาน)

ลำดับขั้นตอนการปฏิบัติงาน

คำชี้แจง: ให้ผู้เรียนเขียนและรันโค้ดลงใน PyCharm ตามขั้นตอนต่อไปนี้

ขั้นตอนที่ 1: การทำความสะอาดและสำรวจข้อมูล (Data Cleaning & Exploration)

1. นำไฟล์ dataset.txt, cleaned_dataset.py และ count_dataset.py ใส่ลงในโปรเจกต์
2. รันไฟล์ cleaned_dataset.py เพื่อกำจัดข้อมูลขยะ บรรทัดว่าง หรือข้อมูลที่ไม่มี Label ออกจากระบบ
3. รันไฟล์ count_dataset.py เพื่อตรวจสอบสัดส่วนของข้อมูลแต่ละประเภท (Positive, Negative, Neutral) และบันทึกผลลงในแบบฟอร์ม

ขั้นตอนที่ 2: การตัดคำภาษาไทย (Tokenization)

1. สร้างไฟล์ format dataset file.py
2. เขียนคำสั่ง Import ไลบรารี deepcut และสร้างฟังก์ชันสำหรับตัดคำข้อความจากไฟล์ที่ทำความสะอาดแล้ว
3. สั่งรันโปรแกรมเพื่อแยกผลลัพธ์ออกเป็น 2 ไฟล์ คือ word.txt (สำหรับเทรน Word2Vec) และ dataset_format.txt (สำหรับเทรน LSTM)

ขั้นตอนที่ 3: การแปลงคำศัพท์เป็นตัวเลข (Word Embedding)

1. สร้างไฟล์ model Word2Vec.py
2. เขียนคำสั่งใช้งาน Word2Vec จากไลบรารี gensim โดยกำหนดพารามิเตอร์ vector_size=100 และ window=5 เพื่อเรียนรู้บริบทของคำจากไฟล์ word.txt
3. บันทึกโมเดลที่ได้ลงในโฟลเดอร์ model

ขั้นตอนที่ 4: การสร้างและเทรนโมเดล LSTM

1. สร้างไฟล์ train.py
2. เขียนคำสั่งโหลดพจนานุกรมจาก Word2Vec และแจกหมายเลขบัตรคิว (Word Index) ให้กับคำศัพท์
3. เขียนคำสั่งใช้งาน pad_sequences เพื่อปรับความยาวประโยค (maxlen) ให้เท่ากับ 50 คำ
4. ประกอบเลเยอร์ของโมเดลด้วย Embedding, Bidirectional(LSTM) และ Dense (Softmax)
5. สั่งเทรนโมเดลและบันทึกผลลัพธ์เป็นไฟล์ final_lstm_model.h5

ขั้นตอนที่ 5: การนำโมเดลไปใช้งาน (Deployment)

1. สร้างไฟล์ run_model.py โหลดโมเดล Word2Vec และ LSTM ที่เทรนเสร็จแล้ว
2. สร้างหน้าต่างแชทใน Terminal เพื่อรับข้อความจากผู้ใช้งาน
3. ทดสอบพิมพ์ข้อความต่อไปนี้ และบันทึกผลลัพธ์ลงในแบบฟอร์ม
   - “ของคุณภาพดี สะอาด”
   - “ของพังง่ายไม่ชอบเลย”
   - “อากาศแจ่มใส”

แบบบันทึกผลการปฏิบัติงาน

ส่วนที่ 1: สถิติของชุดข้อมูล (Dataset Statistics)

1. หลังจากรันไฟล์ count_dataset.py ชุดข้อมูลที่ทำความสะอาดแล้วมีจำนวนแต่ละหมวด
   - ข้อความเชิงบวก (Positive): ..... ประโยค
   - ข้อความเชิงลบ (Negative): ..... ประโยค
   - ข้อความทั่วไป (Neutral): ..... ประโยค

ส่วนที่ 2: การตั้งค่าโมเดล (Model Configurations)

1. ในขั้นตอนการทำ Word Embedding มีการกำหนดให้ 1 คำศัพท์ ถูกแปลงเป็นชุดตัวเลข (Vector Size) จำนวนกี่ตัวเลข?
   - ตอบ: ..... ตัวเลข
2. ในขั้นตอนการเตรียมข้อมูลก่อนเข้า LSTM มีการกำหนดความยาวประโยคสูงสุด (Maxlen) ไว้ที่กี่คำ?
   - ตอบ: ..... ตัวเลข

ส่วนที่ 3: ผลการทดสอบโมเดลวิเคราะห์อารมณ์ความรู้สึก (Inference Results)

จงบันทึกผลลัพธ์ (หมวดหมู่) และค่าความมั่นใจ (เปอร์เซ็นต์/ความน่าจะเป็น) ที่ AI ทำนายได้จากข้อความทดสอบ

1. ข้อความ: "ของคุณภาพดี สะอาด"
   - ผลการทำนาย (Label): .....
   - ค่าความมั่นใจ (Confidence): ..... %
2. ข้อความ: "ของพังง่ายไม่ชอบเลย"
   - ผลการทำนาย (Label): .....
   - ค่าความมั่นใจ (Confidence): ..... %
3. ข้อความ: "อากาศแจ่มใส"
   - ผลการทำนาย (Label): .....
   - ค่าความมั่นใจ (Confidence): ..... %

ส่วนที่ 4: การทดลองและประยุกต์ใช้ (Experiment & Application)

คำชี้แจง: ให้ผู้เรียนทำการปรับแก้โค้ดตามคำสั่งด้านล่าง และสังเกตผลลัพธ์ที่เปลี่ยนแปลงไป

การทดลองที่ 4.1: การปรับลดขนาดพื้นที่ความหมาย (Vector Size Tuning)

ให้ผู้เรียนกลับไปที่ไฟล์ model Word2Vec.py และทำการแก้โค้ดบรรทัด vector_size=100 ให้ลดลงเหลือเพียง vector_size=10 จากนั้นให้ รันโปรแกรมใหม่ตั้งแต่ Step 3 ถึง Step 5 แล้วนำประโยค "ของคุณภาพดีสะอาด" ไปทดสอบอีกครั้ง

- ผลการทำนายที่ได้ (Label): .....
- ค่าความมั่นใจ (Confidence): ..... %
- คำถามวิเคราะห์: ค่าความมั่นใจที่ได้ แตกต่างจากการทดสอบในส่วนที่ 3 หรือไม่? ผู้เรียนคิดว่าการลดค่า vector_size ส่งผลอย่างไรต่อการเรียนรู้ความหมายของคำศัพท์ของ AI?
- ตอบ:..... .....
  ....

การทดลองที่ 4.2: การทดสอบข้อจำกัดของโมเดล (Testing Limitations)

ให้ผู้เรียนแต่งประโยคภาษาไทย 1 ประโยค ที่มีความหมาย "ประชดประชัน" (เช่น พิมพ์เหมือนจะชมแต่ความหมายจริง ๆ คือด่า) แล้วนำไปทดสอบใน run_model.py

- ประโยคที่ใช้ทดสอบคือ: "....."
- ผลการทำนายของ AI (Label): .....
- คำถามวิเคราะห์: AI ทำนายอารมณ์ของประโยคนี้ "ถูก" หรือ "ผิด" ตามบริบทของมนุษย์? หากทำนายผิด ผู้เรียนคิดว่าเป็นเพราะข้อจำกัดใดของชุดข้อมูล (Dataset) หรือกระบวนการเรียนรู้ของโมเดล?
- ตอบ:..... .....
  ....