การจำแนกอารมณ์จากข้อความภาษาอังกฤษ
(Emotion in text classification)

ผู้จัดทำ

นางสาว ศศิวิมล วิทาทาน

รหัสประจำตัวนักศึกษา 610510707 ตอน 001

เสนอ

อาจารย์ ดร. ประภาพร เตชะอังกูร

เป็นส่วนหนึ่งของวิชา 204423 การทำเหมืองข้อมูล

ภาคเรียนที่ 2 ปีการศึกษา 2565

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

ชื่อเรื่อง การจำแนกอารมณ์จากข้อความภาษาอังกฤษ (Emotion in text classification)

อาจารย์ที่ปรึกษา อาจารย์ ดร. ประภาพร เตชะอังกูร

ชื่อผู้วิจัย นางสาว ศศิวิมล วิทาทาน รหัสประจำตัวนักศึกษา 610510707 ตอน 001

บทคัดย่อ

งานวิจัยนี้นำเสนอเกี่ยวกับระบบการจำแนกอารมณ์จากข้อความภาษาอังกฤษ โดยใช้ข้อมูลจากเว็บ Kaggle ที่มีจำนวนกว่า 20,000 ข้อความ ที่มีการระบุอารมณ์ของข้อความอยู่แล้วชัดเจน มาใช้เพื่อให้ได้ระบบที่สามารถจำแนก และทำนายผลอารมณ์จากข้อความภาษาอังกฤษได้ การทำระบบนี้จะใช้ภาษา Python ในการช่วยสร้างระบบ และใช้ Googe Colab เป็นซอฟต์แวร์ในการทำวิจัยครั้งนี้ ซึ่งหลักการที่นำมาใช้วิเคราะห์ข้อมูลคือโมเดลแบบ LSTM ที่เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลลำดับ เนื่องจากใน Python มีไลบารี่ที่ช่วยทำโมเดลแบบ LSTM อยู่แล้ว จากการทำโมเดลจะสามารถตรวจสอบประสิทธิภาพได้จากค่า Accuracy ซึ่งในงานวิจัยยนี้ได้ค่า Accuracy ประมาณ 85.43 % แสดงให้เห็นถึงความแม่นยำของโมเดลว่ามีโมเดลสารามถจำแนกข้อมูลได้อย่างมีความน่าเชื่อถือ และระบบที่ทำยังสามารถทำนายอารมณ์ของข้อความใหม่ที่ไม่เคยเห็นได้อีกด้วย

บทนำ

จากสถิติการใช้งานอินเทอร์เน็ตทั่วโลกพบว่าคนไทยซื้อของออนไลน์ผ่านมือถือมากเป็นอันดับ 8 ในโลกของปี 2021 และจากสถิติยังพบว่าคนไทย 52.5% ของคนที่ซื้อของออนไลน์หาข้อมูลสินค้าก่อนซื้อทางออนไลน์เป็นประจำ โดยสถิติดังกล่าวสามารถบอกได้ว่าคนไทยอ่านรีวิวก่อนซื้อสินค้าแต่ละชิ้น ซึ่งโดยส่วนมากแล้วรีวิวสินค้ามักเป็นข้อความของผู้ที่เคยซื้อมาก่อน โดยข้อความจะมาจากอารมณ์ของผู้ซื้อ ทำให้สามารถระบุได้ว่าสินค้านั้นเป็นที่พอใจต่อลูกค้าที่เคยซื้อหรือไม่ ยกตัวอย่างเช่น สินค้าสวยมาก สินค้านี้ไม่ตรงปกไม่ชอบ เป็นต้น ถ้าหากรีวิวที่ลูกค้าพอใจมีจำนวนมากก็จะมีส่วนในการตัดสินใจซื้อมากขึ้นไปด้วย แต่รีวิวที่มีจำนวนมากอาจทำให้ผู้ซื้ออ่านไม่ครบทำให้ไม่ได้ทราบถึงข้อมูลที่แท้จริงว่าเป็นสินค้าดีหรือไม่ ดังนี้นจึงกล่าวได้ว่าหากสามารถจำแนกอารมณ์จากข้อความจำนวนมากนี้ได้จะสามารถช่วยคัดกรองสินค้าที่ดีได้

จากการศึกษาพบว่าข้อมูลแบบข้อความ (text) นี้เป็นข้อมูลที่ต่อเนื่องกัน (Sequential data) ซึ่งสามารถใช้ LSTM (Long Short-Term Memory) ที่เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลลำดับ เพื่อทำการจำแนกอารมณ์จากข้อความได้ ดังนั้นในการทำการวิจัยนี้จึงใช้ LSTM เพื่อการจำแนกอารมณ์จากข้อความ

วัตถุประสงค์ของการวิจัย

1. เพื่อจำแนกอารมณ์จากข้อความภาษาอังกฤษ
2. เพื่อทำนายอารมณ์จากข้อความภาษาอังกฤษ

ประโยชน์ที่จะได้รับจากการศึกษา เชิงทฤษฎี หรือเชิงประยุกต์

1. ผลผลิต (Output)

   ระบบทำนายอารมณ์จากข้อความภาษาอังกฤษ
2. ผลลัพธ์ (Outcome) และผลกระทบ (Impact)

   สามารถนำระบบไปต่อยอดเพื่อใช้ในการวิเคราะห์อารมณ์จากข้อความในแอพซื้อขายต่างๆได้

ขอบเขตของการวิจัย

ประชากร	ข้อความภาษาอังกฤษที่ระบุอารมณ์ของข้อความจากเว็บไซต์ Kaggle จำนวน 21,405 ข้อความ
กลุ่มตัวอย่าง	ข้อความภาษาอังกฤษที่ระบุอารมณ์ของข้อความจากเว็บไซต์ Kaggle จำนวน 15,450 ข้อความ
ตัวแปรต้น	ข้อความภาษาอังกฤษ
ตัวแปรตาม	อารมณ์ของข้อความ

อุปกรณ์หรือเครื่องมือที่ใช้

1. ฮาร์ดแวร์ที่ใช้ในการพัฒนา
   - เครื่องคอมพิวเตอร์ส่วนบุคคล (ใช้สำหรับพัฒนาเกมบน Unity)

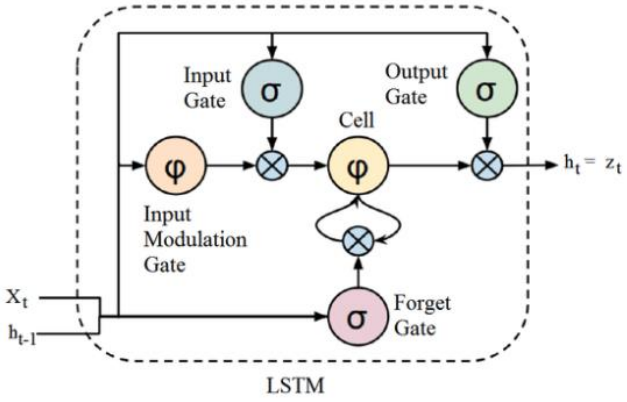
ชื่อรุ่น	Acer Predator Helios 300
หน่วยประมวลผล	อินเทล (Intel(R))
คอร์	Core i7-8750H
หน่วยความจำ	16.0 GB
2. ซอฟต์แวร์ใช้ในการพัฒนาระบบ
   - Google colab
3. ภาษาที่ใช้ในการพัฒนา
   - Python

เอกสารที่เกี่ยวข้อง

1. Long Short Term Memory: LSTM

Long Short Term Memory หรือ LSTM คือ โมเดล deep learning ที่สร้างขึ้นมาจำลองรูปแบบความจำของคน (memory) ที่มีความจุของความทรงจำอยู่จำกัด เมื่อมีเหตุการณ์ใหม่ๆ เข้ามาในความทรงจำ สมองจะเลือกที่จะรับ หรือไม่รับเหตุการณ์ใหม่เข้ามาในความทรงจำ ตามความสำคัญของเหตุการณ์ และเมื่อสมองเลือกที่จะรับเหตุการณ์ใหม่ๆ ที่มีความสำคัญเข้ามาเก็บไว้ในระบบความทรงจำแล้ว (memorize) ก็จำเป็นจะต้องมีเหตุการณ์บางอย่างในอดีตที่ถูกลืมไป

โครงสร้างพื้นฐานของ LSTM คือ มี forget gate มาจำลองเหตุการณ์ “ลืม” และ memory gate มาจำลองเหตุการณ์ “จำ” ดังรูปที่ 1



รูปที่ 1 โครงสร้างพื้นฐานของ LSTM

2. Classification

การจำแนกข้อมูล (Classification) ที่ เป็นปัญหาพื้นฐานของการเรียนรู้แบบมีผู้สอน โดยปัญหาคือการทำนายประเภทของวัตถุจากคุณสมบัติต่าง ๆ ของวัตถุ ซึ่งการเรียนรู้แบบมีผู้สอนจะสร้างฟังก์ชันเชื่อมโยง ระหว่างคุณสมบัติของวัตถุ กับประเภทของวัตถุจากตัวอย่างสอน แล้วจึงใช้ฟังก์ชันนี้ทำนายประเภทของวัตถุที่ไม่เคยพบ เครื่องมือหรือขั้นตอนวิธีที่ใช้สำหรับการแบ่งประเภทข้อมูลเช่น โครงข่ายประสาทเทียม ต้นไม้ตัดสินใจ

3. Accuracy

ความถูกต้อง หรือ ความแม่นยำ (accuracy) เป็นค่าที่บ่งบอกถึงความสามารถของเครื่องมือวัด (instrument) ในการอ่านค่าหรือแสดงค่าที่วัดได้เข้าใกล้ค่าจริง โดยค่า accuracy จะมีค่าอยู่ระหว่าง 0–1 ยิ่งเข้าใกล้ 1 แปลว่าโมเดลเราทำนายผลได้ดีมาก

$$\frac{TP + TN}{TP + TN + FP + FN}$$

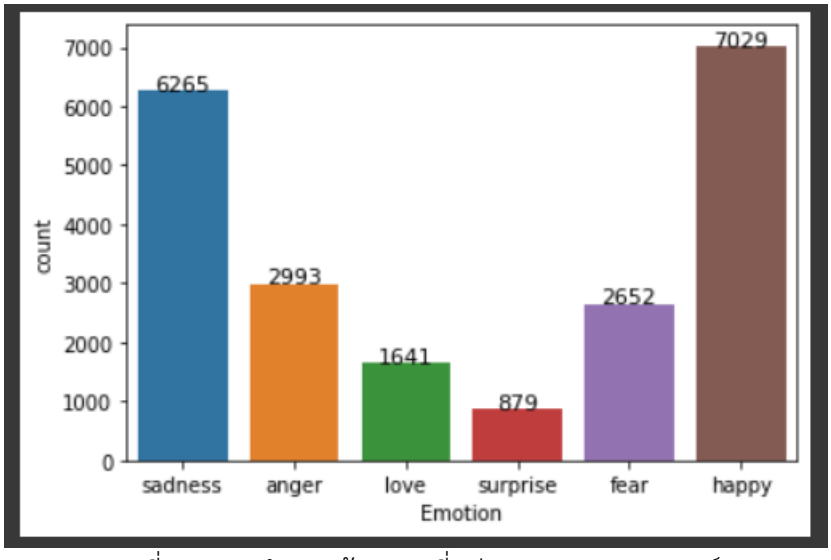
วิธีการ

1. การเก็บข้อมูล

จากข้อมูลที่ได้รับมาจากเว็บไซต์ Kaggle โดยข้อมูลที่ได้มาเป็นข้อความภาษาอังกฤษที่ระบุอารมณ์ของข้อความซึ่งมีจำนวน 21,405 ข้อความ โดยแบ่งข้อความตามอารมณ์ได้ดังนี้

- อารมณ์ “Sadness” จำนวน 6,265 ข้อความ
- อารมณ์ “Anger” จำนวน 2,993 ข้อความ
- อารมณ์ “Love” จำนวน 1,641 ข้อความ
- อารมณ์ “Surprise” จำนวน 879 ข้อความ
- อารมณ์ “Fear” จำนวน 2,652 ข้อความ
- อารมณ์ “Happy” จำนวน 7,029 ข้อความ

สามารถแสดงข้อมูลที่มีได้ดังรูปที่ 2



รูปที่ 2 กราฟจำนวนข้อความที่แบ่งตามประเภทอารมณ์

2. การวิเคราะห์ข้อมูล

จากข้อมูลที่ได้รับมาจากเว็บไซต์ Kaggle โดยข้อมูลที่ได้มาเป็นข้อความภาษาอังกฤษที่ระบุอารมณ์ของข้อความนั้นยังไม่ได้ทำการทำเตียมข้อมูลเพื่อนำไปสร้างโมเดลดังนั้น

- ลบข้อมูลที่เป็นว่างออก หรือข้อมูลที่ไม่ได้เติมข้อความหรืออารมณ์ออก
- เปลี่ยนข้อความโดยลบ 'stopwords' ในข้อความแต่ละข้อความออก เช่น “a”, “am”, “I” เป็นต้น และเก็บแต่ละข้อความรวมเป็น Array

```
[17] nltk.download('stopwords')

     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
     True
```

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
corpus = []
for i in range(0, len(messages)):
    review = re.sub('[^a-zA-Z]', ' ', messages['Text'][i])
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

- ทำการเข้ารหัสเพื่อเปลี่ยนข้อความเป็นตัวเลข

```
onehot_repr=[one_hot(words,voc_size)for words in corpus]
```

- ทำให้ข้อความทุกข้อความมีขนาดเท่ากัน

```
sent_length=35
embedded_docs=pad_sequences(onehot_repr,padding='pre',maxlen=sent_length)
print(embedded_docs)
```

```
[[   0    0    0 ... 2375 6686  469]
 [   0    0    0 ... 8488 5737 5180]
 [   0    0    0 ... 6686 6762 4389]
 ...
 [   0    0    0 ... 8734 7939 7483]
 [   0    0    0 ... 8215 7483 1340]
 [   0    0    0 ...    0 7483 2015]]
```

3. การออกแบบการทดลอง

โครงสร้างพื้นฐานของ LSTM คือ มี forget gate มาจำลองเหตุการณ์ “ลืม” และ memory gate มาจำลองเหตุการณ์ “จำ” โดยจะใช้โมเดล LSTM จากไลบารี่ keras ซึ่งเป็น library สำหรับทำ deep learning ใช้งานง่าย โดย class ที่ใช้สำหรับการสร้าง deep learning โมเดลใน keras เรียกว่า Sequential ซึ่งเป็นเสมือนโครงสร้างเปล่าๆ ที่เราสามารถเพิ่ม layer ต่างๆ เข้าไปได้ โดยโครงสร้างประกอบไปด้วย layer แรก คือ LSTM ซึ่งเราสามารถกำหนดขนาดของ hidden layer ได้ และ กำหนด shape ของข้อมูลที่จะ input เข้ามา

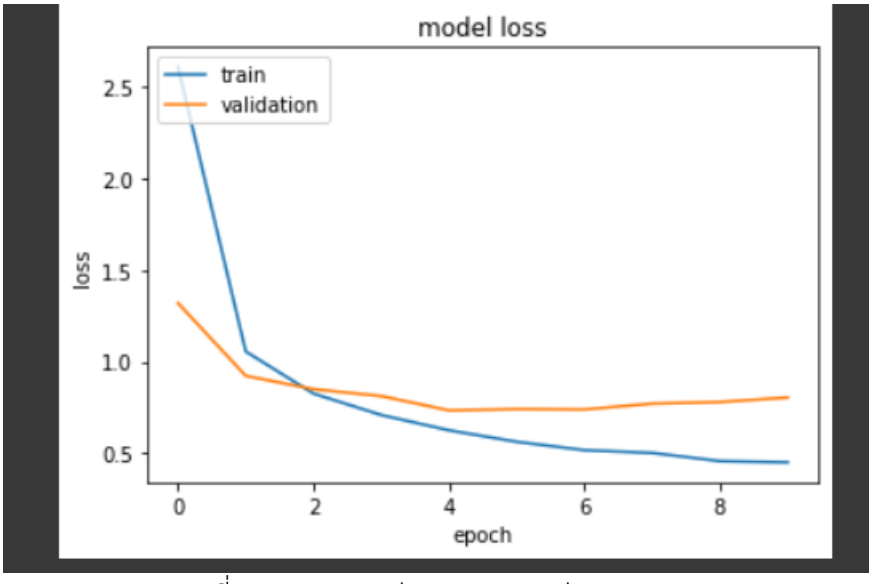
4. การทดลอง

สร้างโมเดล LSTM จากไลบารี่ keras

```
# Creating model
embedding_vector_features=100
model=Sequential()
model.add(Embedding(voc_size,embedding_vector_features,input_length=sent_length))
model.add(Dropout(0.3))
model.add(Bidirectional(LSTM(64)))
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(Dropout(0.3))
model.add(Dense(6,activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy',optimizer= tf.keras.optimizers.Adam(learning_rate=0.01),
              metrics=['accuracy'])
model.summary()
```

```
[28] model_save = ModelCheckpoint('weights.h5', save_best_only = True, save_weights_only = True, monitor = 'val_loss',
                                  mode = 'min', verbose = 1)
     history = model.fit(X_train,y_train,validation_data=(X_val,y_val),epochs=10,batch_size=256,callbacks = [model_save])
```

ระหว่าง train จะเก็บค่า loss แล้วนำมาแสดงผลได้ดังรูปที่ 3



รูปที่ 3 กราฟแสดงค่า Loss ของแต่ละ Epoch

5. การหาประสิทธิภาพ

การหาประสิทธิภาพจะใช้ Confusion Matrix ในการแสดงประสิทธิภาพ และแสดงรายละเอียดของค่า Accuracy, Precision, Recall, F1-Score เพื่อดูความน่าเชื่อถือของโมเดลที่ทำว่ามีความแม่นยำหรือไม่ โดยการหา Confusion Matrix สามารถใช้ไลบารี่ใน Python ช่วยหาได้

```
print(sns.heatmap(confusion_matrix(y_test, y_pred),annot=True,fmt="d"))
```

ทำการแสดงรายละเอียดของค่า Accuracy, Precision, Recall, F1-Score ผ่านทางการแสดงผลของ Classification_report ของข้อมูลที่ใช้ทดสอบและทำการทำนายอารมณ์จากข้อความใหม่เพื่อแสดงประสิทธิภาพของโมเดล

```
print(classification_report(y_test, y_pred, digits=5))
```

ผลการทดลองและการอภิปราย

จากการทดลองได้โมเดลที่ทำการ Classification โดยจะแสดงผลการดำเนินการเป็น Confusion Matrix ซึ่งเมทริกซ์ตัวนี้สามารถอธิบายค่า TP, TN, FN และ FP ทำให้ทราบว่ามีข้อมูลที่นำมาทดสอบจำนวน 4258 ข้อความ นี้ที่ทำนายผิดหรือถูกอย่างไรบาง โดยจากการทดลองได้ผลดังนี้

TP ของอารมณ์ “Sadness” อยู่ที่ 543 ข้อความ

TP ของอารมณ์ “Anger” อยู่ที่ 460 ข้อความ

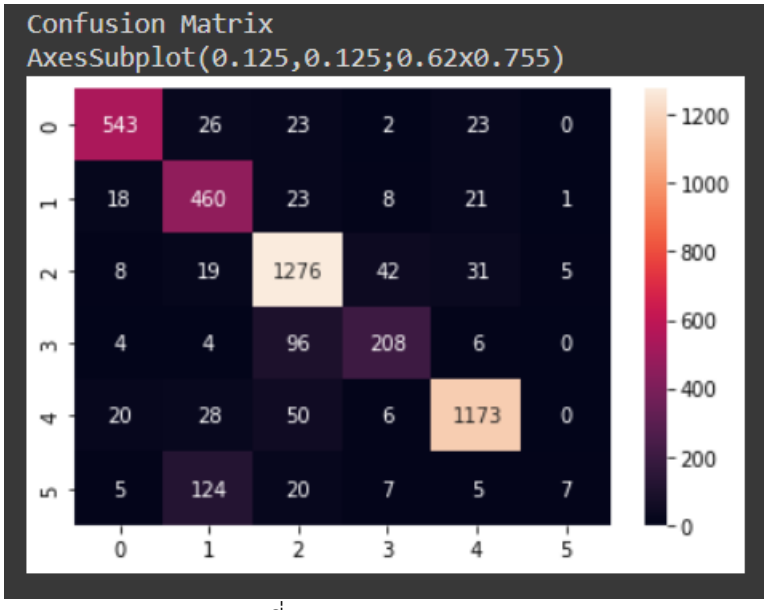
TP ของอารมณ์ “Love” อยู่ที่ 1276 ข้อความ

TP ของอารมณ์ “Surprise” อยู่ที่ 208 ข้อความ

TP ของอารมณ์ “Fear” อยู่ที่ 1173 ข้อความ

TP ของอารมณ์ “Happy” อยู่ที่ 7 ข้อความ

จาก Confusion Matrix ยังแสดงค่าอื่นอีกด้วยดังรูปที่ 4



รูปที่ 4 Confusion Matrix

ซึ่งจาก Confusion Matrix ที่ได้มาสามารถนำมาหาค่า Accuracy, Precision, Recall, F1-Score เพื่อแสดงถึงความถูกต้องของโมเดลได้ โดยค่า Accuracy เท่ากับ 0.85438 หรือก็คือ 85.438 % ซึ่งมีค่าเข้าใกล้ 1 แสดงให้เห็นว่ามีความแม่นยำที่สูงเป็นโมเดลที่มีความหน้าเชื่อถือได้ ค่า Precision ของแต่ละอารมณ์นั้นมีค่าเข้าใกล้ 1 หมดเลยแสดงถึงความแม่นยำของโมเดล ค่า Recall ของแต่ละอารมณ์นั้นมีค่าเข้าใกล้ 1 หมดเลยแสดงถึง8ความถฮกต้องของโมเดล และ ค่า f1_score ของแต่ละโมเดลนั้นเป็นค่าเฉลี่ยของ ค่า Precision และค่า Recall ซึ่งค่าที่ได้มีค่า f1_score ค่าเข้าใกล้ 1 ที่แสดงให้ว่าข้อมูลที่นำมาทสอบ และโมเดลนี้มีความน่าเชื่อถือ โดยจะแสดงรายละเอียดค่าต่างๆ ดังรูปที่ 5

	precision	recall	f1-score	support
0	0.90803	0.88006	0.89383	617
1	0.69592	0.86629	0.77181	531
2	0.85753	0.92397	0.88951	1381
3	0.76190	0.65409	0.70389	318
4	0.93169	0.91856	0.92508	1277
5	0.53846	0.04167	0.07735	168
accuracy			0.85438	4292
macro avg	0.78225	0.71411	0.71024	4292
weighted avg	0.84728	0.85438	0.84061	4292

รูปที่ 5 ค่า Accuracy, Precision, Recall, F1-Score

จากโมเดลที่ได้สามารถนำมาทำนายข้อความใหม่เพื่อหาอารมณ์ได้อีกด้วย และจากการทดลองพบว่าสามารถทำนายออกมาได้ค่อนข้างถูกต้อง โดยได้แสดงรายละเอียดการทำนายดังรูปที่ 6

```
[47] predict_emotion('I am very happy and joyful today')

     'happy'

[48] predict_emotion('He is an arrogant and rude person')

     'anger'

[49] predict_emotion('The teacher is intimidating and scary')

     'fear'

[51] predict_emotion('I am pretty bad')

     'sadness'
```

รูปที่ 6 ตัวอย่างการทำนายผลจากข้อความภาษาอังกฤษ

สรุป

จากผลการดำเนินการสามารถสรุปได้ว่าจากข้อมูลที่ได้รับมาจากเว็บไซต์ Kaggle โดยข้อมูลที่ได้มาเป็นข้อความภาษาอังกฤษที่ระบุอารมณ์ของข้อมูลซึ่งมีจำนวน 21,405 ข้อความ เมื่อทำการทำความสะอาดข้อมูลแล้วเลือกข้อมูลแบบสุ่มจำนวน 15,450 ข้อความ เพื่อใช้เป็น Training Set สำหรับสร้างโมเดลเพื่อจัดกลุ่ม และทำนายผลอารมณ์จากข้อความ เมื่อสร้างโมเดลโดยใช้หลักการ LSTM แล้วได้ค่า Accuracy ประมาณ 85.43 % แสดงให้เห็นถึงความแม่นยำของโมเดลว่ามีโมเดลสารามถจำแนกข้อมูลได้อย่างมีความน่าเชื่อถือ และระบบที่ทำยังสามารถทำนายอารมณ์ของข้อความใหม่ที่ไม่เคยเห็นได้อีกด้วย

เอกสารอ้างอิง

[1] Nattapon Muangtum. (2565). *เปิดสถิติ พฤติกรรมผู้ใช้อินเทอร์เน็ต-โซเชียลมีเดียยอดนิยม ตลอดปี 2021*. https://www.prachachat.net/ict/news-855712

[2] Sirinart Tangruamsub. (2560). *Long Short-Term Memory (LSTM)*. https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6

[3] Sanparith Marukatat. (2560). *LSTM เท่าที่เข้าใจ.* https://sanparithmarukatat.medium.com/lstm-เท่าที่เข้าใจ-75027db3167f

[4] Aman kharwal. (2564). *Text Emotions Detection with Machine Learning.* https://thecleverprogrammer.com/2021/02/19/text-emotions-detection-with-machine-learning/

[5] Porntiva Visitsora-at. (2562). *Machine Learning.* https://medium.com/@615162020027/metrics-พื้นฐานสำหรับวัดประสิทธิภาพของโมเดล-machine-learning-c00fcc32fa30

[6] วิกิพีเดีย. (2563). *classification.* https://th.wikipedia.org/wiki/การแบ่งประเภทข้อมูล

[7] พิมพ์เพ็ญ พรเฉลิมพงศ์. (2555). *classification.* http://www.foodnetworksolution.com/wiki/word/4289/accuracy-ความถูกต้อง-ความแม่นยำ