



รายงาน

แบบบอท แนะนำวิธีการเลิกสุรา

เสนอ

อาจารย์ ผศ.ดร.อรรณณ อิมสมบัติ

จัดทำโดย

- | | |
|------------------------------|-----------------------|
| 1.นายเจตนิพัทธ์ แดงใส | รหัสสนิสิต 6530200061 |
| 2.นายธนธรณ์ จิตศาสน์สุวรรณค์ | รหัสสนิสิต 6530200207 |
| 3.นายรติพงษ์ วันวิเสส | รหัสสนิสิต 6530200398 |
| 4.นายจักรภพ ศิวะกุลรังสรรค์ | รหัสสนิสิต 6530200584 |
| 5.นางสาวสุภาวดี ม่วงประเสริฐ | รหัสสนิสิต 6530200860 |

รายงานนี้เป็นส่วนหนึ่งของวิชา Selected Topic in Computer Science 01418496-65

คณะวิทยาศาสตร์ ศรีราชา สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

ที่มาและความสำคัญ

เหล้า-เบียร์ หรือเครื่องดื่มแอลกอฮอล์ เป็นสิ่งที่ทุกคนทราบกันดีอยู่แล้วว่าเป็นเครื่องดื่มที่ไม่ก่อให้เกิดผลดีต่อสุขภาพ และยิ่งหากดื่มในปริมาณที่มากอย่างต่อเนื่องทุกวัน จนเกินเลยไปถึงขั้นกลายเป็น “ผู้ติดเหล้า” นอกจากจะเกิดปัญหาที่ส่งผลกระทบต่อสุขภาพร่างกายและจิตใจของบุคคลแล้ว ยังส่งผลกระทบต่อครอบครัวและสังคมอีกด้วย เพราะผู้ที่ติดเหล้ามักจะพบปัญหาทางสุขภาพที่ร้ายแรง และนำไปสู่การมีพฤติกรรมที่ไม่เหมาะสมตามมา

หลายคนร่างกายและสมองติดแอลกอฮอล์จนต้องหามาดื่มทุกวัน แต่อีกมุมหนึ่งก็อยากจะอยากเลิกแทบขาดใจ... แต่ก็ไม่สามารถเลิกได้อย่างถาวรสักที เพราะสารเคมีในสมองผิดเพี้ยนไปแล้ว บางคนตัดใจเลิกแบบหักดิบ กลายเป็นว่าต้องเผชิญกับความทรมานกับ “อาการลงแดง” จนแทบคลั่ง ทำให้ผู้ที่ต้องการหลุดพ้นจากวงจรความทุกข์นี้ไม่สามารถหลุดพ้นได้สักที ทั้งขาดกำลังใจ แรงจูงใจ และที่ปรึกษาที่พร้อมอยู่เคียงข้างให้สามารถเลิกเหล้าได้สำเร็จอย่างถูกวิธี

กลุ่มเราจึงอยากช่วยเหลือผู้ติดเหล้าให้ได้ในทุกมิติ โดยคิดนวัตกรรมแซทบอทช่วยเหลือผู้ต้องการเลิกเหล้า เหมาะสำหรับคนไม่กล้าโทร ไม่กล้าคุย แต่ชอบแซทผ่านมือถือในยุคสมัยนี้ ที่แค่ “แอตไลน์” เป็นเพื่อน

วัตถุประสงค์

1. ช่วยในการลดการบริโภคสุรา: แซทบอทเลิกเหล้าสามารถเป็นตัวช่วยในการส่งเสริมการลดการบริโภคเหล้าของบุคคล โดยให้ข้อมูลเกี่ยวกับผลกระทบของการดื่มเหล้าต่อสุขภาพ และแนะนำวิธีการเลิกเหล้าที่มีประสิทธิภาพ
2. ให้ข้อมูลเกี่ยวกับการเลิกเหล้า: แซทบอทเลิกเหล้าสามารถให้ข้อมูลเกี่ยวกับสาเหตุและผลกระทบทางสุขภาพของการดื่มเหล้ามากเกินไป และวิธีการเลิกเหล้าอย่างปลอดภัย ซึ่งอาจช่วยเพิ่มความเข้าใจและสามารถกระตุ้นให้บุคคลเริ่มพิจารณาการเลิกดื่มเหล้า
3. การสนับสนุนผู้ที่ต้องการเลิกเหล้า: แซทบอทเลิกเหล้าสามารถเป็นแหล่งสนับสนุนที่มีอยู่ตลอดเวลาสำหรับผู้ที่ต้องการเลิกดื่มเหล้า ผ่านการให้ข้อมูลที่แน่นนอน การสนับสนุนทางจิตใจ และการแนะนำสถานที่หรือแหล่งช่วยเหลือที่เหมาะสม

Dataset

- Link Dataset

<https://docs.google.com/spreadsheets/d/1sPdWSGAgfqC80xUJGCEYbYIH0vgvcjBN6B6xdC3tc48/edit?usp=sharing>

- จำนวน class มี 6 คือ

0 : การหักท่าย	จำนวน 26
1 : วิธีการเลิกเหล้า	จำนวน 71
2 : วิธีการพบแพทย์สำหรับเลิกเหล้า	จำนวน 31
3 : ผลเสียของการดื่มเหล้า	จำนวน 30
4 : การรักษาบำบัด	จำนวน 30
5 : อาการลงแดงระหว่างเลิกเหล้า	จำนวน 30

Pre-processing

```
# mount drive
from google.colab import drive
drive.mount('/content/drive')
```

```
!pip install pythainlp
```

Requirement already satisfied: pythainlp in /usr/local/lib/python3.10/dist-packages (4.0.2)
Requirement already satisfied: requests>=2.22.0 in /usr/local/lib/python3.10/dist-packages (from pythainlp) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (2023.7.22)

```
[ ] import pandas as pd
import numpy as np
import tweepy
import re
import matplotlib.pyplot as plt
import itertools

from keras.preprocessing.text import Tokenizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.feature_extraction.text import CountVectorizer
from pythainlp.tokenize import word_tokenize
from pythainlp.corpus import thai_stopwords
from wordcloud import WordCloud
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Dense, Flatten
from tensorflow.keras.layers import Embedding, Bidirectional, LSTM, GlobalMaxPooling1D, Dense
from sklearn.metrics import confusion_matrix
```

```
# load data
data = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/data/dataset_Loek Lao1.xlsx')
data.head()
```

	หัวข้อคำถาม	คำถาม
0	0	สวัสดี
1	0	สวัสดีครับ
2	0	สวัสดีค่ะ
3	0	หวัดดี
4	0	ดีครับ

- คำสั่งการติดตั้ง package เพื่อเตรียมสำหรับเรียกใช้คำสั่งและจัดเตรียมไฟล์ dataset

- ฟังก์ชันการคลีนและการตัดคำภาษาไทย

```
# cleantext
def cleanText(text):
    text = str(text)
    stop_word = list(thai_stopwords())
    sentence = word_tokenize(text)
    result = [word for word in sentence if word not in stop_word and " " not in word]
    return " ".join(result)

def tokenize(d):
    result = d.split(" ")
    result = list(filter(None, result))
    return result

new_newmm = []
for txt in data['คำถาม']:
    new_newmm.append(cleanText(txt))

data['new_newmm'] = new_newmm
data
```

```
[ ] tokenizer = Tokenizer()
tokenizer.fit_on_texts(data['new_newmm'])
totalWords = len(tokenizer.word_index) + 1
print(f'Total unique words: {totalWords}')

Total unique words: 151
```

- คำสั่งคัด class ในแต่ละแถวของหัวข้อคำถาม

```
from tensorflow.keras.utils import to_categorical
print("Unique labels =", set(data['หัวข้อคำถาม']))
y = to_categorical([int(i) for i in data['หัวข้อคำถาม']])
print(y)
```

- คำสั่งเปลี่ยนค่าให้เป็นตัวเลขและนำเลขที่ได้มาเรียงให้เป็นประโยค

พร้อมเตรียม train_test_split

```
from keras.utils import pad_sequences
tts = tokenizer.texts_to_sequences(data['new_newmm'])
print("Raw Texts to Sequence:")
print(tts)
maxlen = max([len(s) for s in tts])
print("max sentence's length =", maxlen)
x = pad_sequences(tts, maxlen=maxlen, padding="post")
print("Sequence Padding at maxlen words, post padding:\n", x)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=64)
```

- Model deep learning , accuracy & confusion matrix graph

```

model_DL = Sequential()
model_DL.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=64, input_length=maxlen))
model_DL.add(Flatten())
model_DL.add(Dense(6, activation='sigmoid')) # Assuming binary classification
model_DL.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model_DL.summary()
# Train the model

model_DL.fit(x_train, y_train, epochs=40, validation_split=0.2)

#Predict with the model
y_predict = model_DL.predict(x)
y_t=[ np.argmax(i) for i in y]
y_p=[ np.argmax(i) for i in y_predict]
cm=confusion_matrix(y_t, y_p)
plot_confusion_matrix(cm=cm, classes=['0','1','2','3','4','5'], normalize=True)

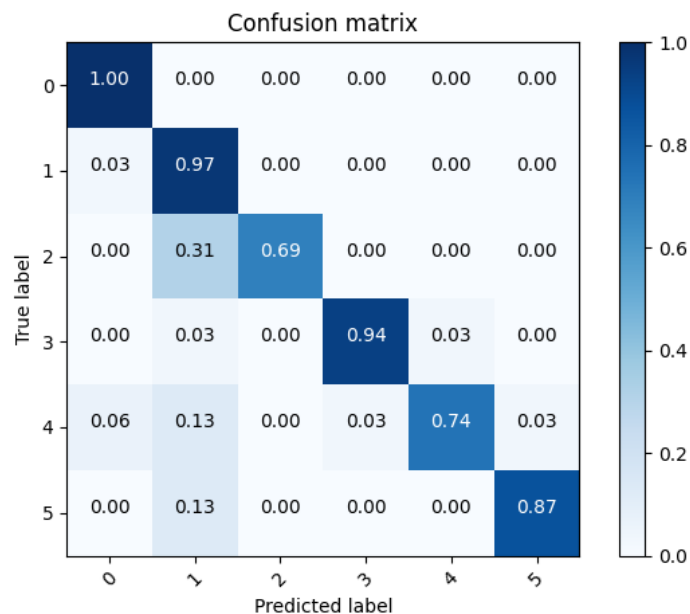
```

```

loss , accuracy , = model_DL.evaluate(x_test, y_test)
print(f"Test Loss : {loss*100:.2f}")
print(f"Test Accuracy : {accuracy*100:.2f}")

```

2/2 [=====] - 0s 8ms/step - loss: 0.2620 - accuracy: 0.7500
Test Loss : 26.20
Test Accuracy : 75.00



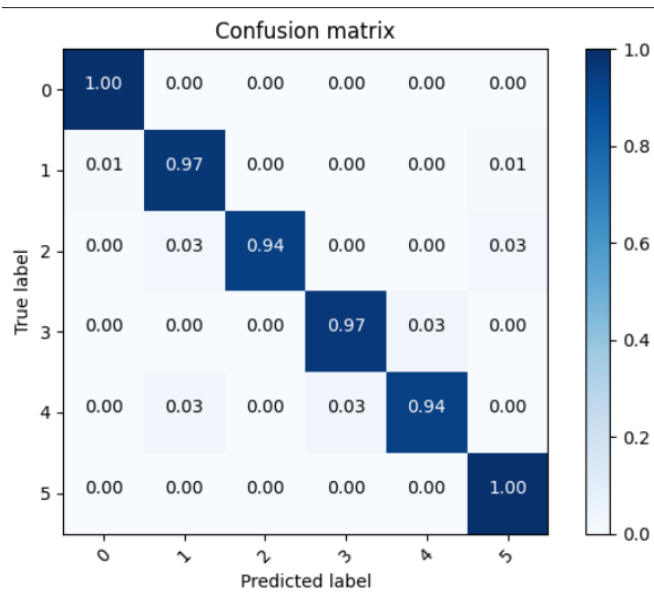
- Model LSTM , accuracy & confusion matrix graph



```
model = Sequential()
model.add(Embedding(len(tokenizer.word_index) + 1, 128, input_length=maxlen))
model.add(LSTM(128)) # LSTM layer
model.add(Dense(6, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
#Fit the model
model.fit(x_train, y_train, epochs=20)
#Predict with the model
y_predict = model.predict(x)
y_t=[ np.argmax(i) for i in y]
y_p=[ np.argmax(i) for i in y_predict]
cm=confusion_matrix(y_t, y_p)
plot_confusion_matrix(cm=cm, classes=['0','1','2','3','4','5'], normalize=True)
```

```
[ ] loss , accuracy , = model.evaluate(x_test, y_test)
print(f"Test Loss : {loss*100:.2f}")
print(f"Test Accuracy : {accuracy*100:.2f}")
```

2/2 [=====] - 1s 10ms/step - loss: 0.6931 - accuracy: 0.8929
Test Loss : 69.31
Test Accuracy : 89.29



- Model BI-LSTM , accuracy & confusion matrix graph

```

model = Sequential()
model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=100, input_length=maxlen))
model.add(Bidirectional(LSTM(64, return_sequences=True)))
model.add(GlobalMaxPooling1D())
model.add(Dense(6, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(x_train, y_train, epochs=20)

y_predict = model.predict(x)
y_t = [np.argmax(i) for i in y]
y_p = [np.argmax(i) for i in y_predict]
cm = confusion_matrix(y_t, y_p)
plot_confusion_matrix(cm=cm, classes=['0','1','2','3','4','5'], normalize=True)

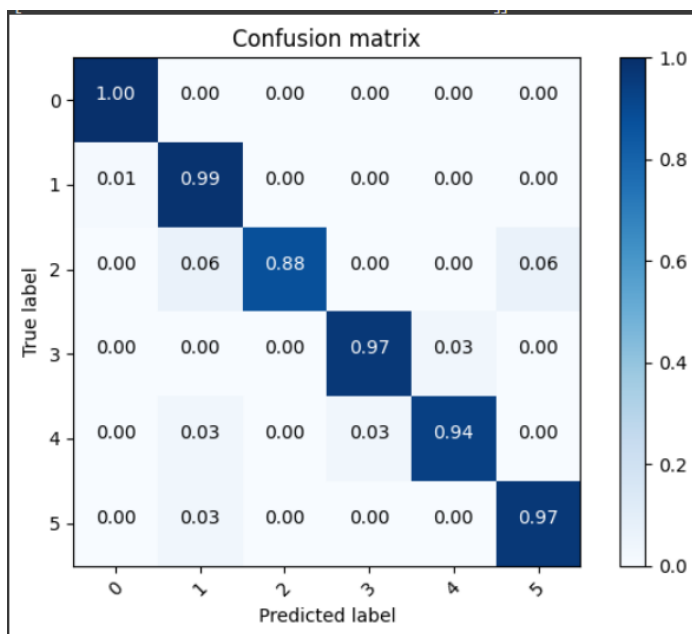
```

```

loss , accuracy , = model.evaluate(x_test, y_test)
print(f"Test Loss : {loss*100:.2f}")
print(f"Test Accuracy : {accuracy*100:.2f}")

```

2/2 [=====] - 1s 10ms/step - loss: 0.6313 - accuracy: 0.8750
 Test Loss : 63.13
 Test Accuracy : 87.50



- คำสั่งที่ใช้ทดสอบ predict

```
text_to_predict = ["อาการของคนติดสุรา"]
tokenized_text = [word_tokenize(text, keep_whitespace=False) for text in text_to_predict]
text_sequences = tokenizer.texts_to_sequences(tokenized_text)
max_sequence_length = maxlen
padded_sequence = pad_sequences(text_sequences, maxlen=max_sequence_length, padding="post")
predicted_class = model.predict(padded_sequence)
predicted_class_index = np.argmax(predicted_class)
class_labels = ['0', '1', '2', '3', '4', '5']
predicted_label = class_labels[predicted_class_index]
for text, prediction in zip(text_to_predict, predicted_label):
    print(f"Text: {text} => Predicted Class: {prediction}")
```

ซึ่งเราเลือกใช้ model BI-LSTM ที่มีค่า accuracy น้อยกว่า LSTM เนื่องจากเราได้มีการทดลอง predict ด้วยโมเดลดังกล่าวแล้ว แต่ผลลัพธ์ไม่ค่อยออกมาอย่างที่ต้องการเท่า BI-LSTM

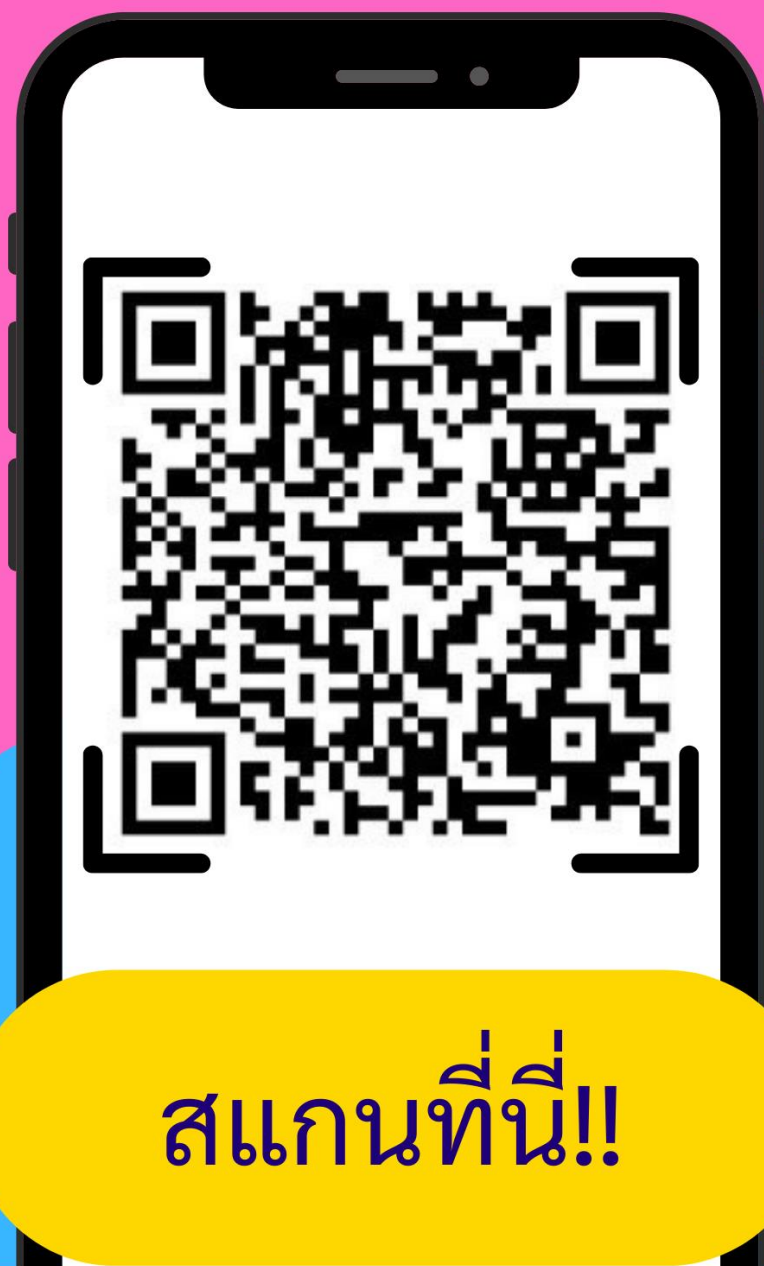
Link dataset

https://docs.google.com/spreadsheets/d/1sPdWSGAgfqC80xUJGCEYbYIH_OvgvcjBN6B6xdC3tc48/edit?usp=sharing

link colab

https://colab.research.google.com/drive/1l8ZlirO90qKd9cZ_AAwcX-BOy1_1CNiE?usp=sharing

อยากเลิกเล่นให้น้องถูกขโมย
ช่วยนะครับ



สแกนที่นี่!!