CHATBOT

ROAR ALONE

OBJECTIVES

แชทบอทเพื่อการท่องเที่ยว

- เพื่อสร้างแอพพลิเคชั่นสำหรับการท่องเที่ยว
- เพื่อศึกษาเจตคติของผู้ใช้แอพพลิเคชั่น
- เพื่อส่งเสริมการท่องเที่ยวในประเทศไทย

FEATURES

Accommodation

Attraction

Restaurant

RecommendedRoute

FEATURES

Request notify

Record trip

CONTACT ROAR

APPLICATION LINE

หลักการทำงานเบื้องต้นของ CHATBOT คือ เมื่อเกิดคำถามจากฝั่งตรงข้าม ระบบจะทำการวิเคราะห์ด้วยการตรวจสอบคำที่มีความคล้ายคลึงกับข้อมูล ที่ตนเองถูกป้อนเข้าไป (คีย์เวิร์ด)
เมื่อรู้ว่าคำหลักคืออะไร ก็จะจัดการตอบกลับข้อมูลดังกล่าวด้วยเนื้อหาที่เหมาะสมและรวดเร็ว
ไม่ต้องเสียเวลาในการรอคนพิมพ์เองนาน ๆ

How are human languages processed by chatbots?

รู้จักกับ Dialogflow

Dialogflow คือ platform สำหรับสร้าง chatbot ของ Google ที่ใช้ machine learning ด้าน Natural Language Processing (NLP) มาช่วยในการทำความเข้าใจถึงความต้องการ (intent) และสิ่งที่ต้องการ (entity) ในประโยคสนทนาของผู้ใช้งาน และตอบคำถามตามความต้องการของผู้ใช้งานตามกฎ หรือ flow ที่ผู้พัฒนาวางเอาไว้ ซึ่ง Dialogflow จะช่วยเพิ่มความยืดหยุ่นของประโยคที่ chatbot รับมา ว่าไม่จำเป็นต้องตรงตามเงื่อนไข แบบ rule based เป๊ะๆ ก็สามารถเข้าใจถึงความต้องการของผู้ใช้งานได้

ขั้นตอนการทำ Dialogflow

สร้าง project Dialogflow ->

สร้าง project Line Chatbot ->

เชื่อมต่อ Line Chatbot กับ Dialogflow ->

สร้าง Application server ด้วย Flask บน Python

และทดสอบเชื่อมต่อ Flask กับ Dialogflow ->

สร้าง Sqlite database

Deploy application ด้วย Heroku ->

DIALOGFLOW

สร้าง PROJECT LINE CHATBOT

สร้าง PROJECT DIALOGFLOW

เชื่อมต่อ LINE CHATBOT กับ DIALOGFLOW

Chatbot.py

```python
#Chatbot Tutorial with Firebase
#Import Library
import json
import os
import requests
from flask import request
from flask import make_response
from function_chatbot import *
from flask import Flask,render_template
from requests.exceptions import HTTPError
#-----------------------------------

# Flask
app = Flask(__name__)
@app.route('/ACCOMMODATION/<place_id>')
def index(place_id):
    url = f"https://tatapi.tourismthailand.org/tatapi/v5/accommodation/{place_id}"

    try:
        Ac = requests.get(url,headers=my_headers).json()
    except HTTPError as http_err:
        print(f'HTTP error occurred: {http_err}')  # Python 3.6
    except Exception as err:
        print(f'Other error occurred: {err}')  # Python 3.6

    return render_template("GetAccommodationDetail.html",Ac=Ac['result'])
```

สร้าง Application server ด้วย Flask บน Python

สร้าง Sqlite database

DB Browser for SQLite - C:\Users\MoMo\Desktop\Vscode\Chatbot\database3.db

File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Database

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: record_model

Filter in any column

	User_ID	Trip_name	Participants_name	Accommodation_name	Restaurant_name	Expenses_total
	Filter	Filter	Filter	Filter	Filter	Filter
1	U8907e9f058b0cb82c0e0735e61d59d5f	เกาะล้าน	เจนนี่	โรงแรมวินเนอร์วิว	ร้านครัวเฉลียงลม	2,000บาท
2	U8907e9f058b0cb82c0e0735e61d59d5f	พักผ่อน	เจม,จัสมิน	โรงแรมคริสตัล	ร้านทะเลเดือด,ร้านบุฟเฟ่ซีฟู้ด,ร้านอาหารตามสั่ง	999บาท
3	U8907e9f058b0cb82c0e0735e61d59d5f	มันๆ	เลย์,อีม,อ้น	รีสอร์ทติดชายทะเล	ร้านอาหารทะเล,ร้านข้าวแกงอีสาน	1399บาท
4	U8907e9f058b0cb82c0e0735e61d59d5f	Hong kong	Ronaldo	โรงแรมพรูเดนเชียล	ร้านSharetea	5,000บาท
5	U8907e9f058b0cb82c0e0735e61d59d5f	ทริปDoi Inthanon	Messi,Neymar,Mbappe	Sibsan Resort & Spa,Tanita House	ร้านDongmadame,YANGZI JIANG	8855บาท

Deploy application ด้วย Heroku

แนะนำ Dialogflow
: Intent, Parameters และ Fulfillment

การทำงานของ intent

Intent แปลตรงตัวว่า เจตนา มันคือการนิยามคำสั่งของ chatbot ว่าจะให้ chatbot รับคำสั่งอะไรได้บ้าง โดย intent จะประกอบด้วย

- Training phrases: เป็นการใส่ตัวอย่าง input จาก user ซึ่ง Dialogflow จะใช้ตัวอย่างนี้ในการ train model
- Action and Parameters: คือการแกะค่าตัวแปรที่สำคัญออกมาจากประโยค เพื่อนำมาใช้งานต่อ
- โดยเราสามารถระบุตัวอย่างของ parameter ได้ใน training phrases
- Responses: ระบุข้อความที่ต้องการให้ chatbot ตอบกลับตาม intent นั้นๆ

Parameters

ในกรณีนี้ เป็นตัวอย่างของ intent ที่จะตอบข้อมูลพยากรณ์อากาศ (Forecast intent)
โดย parameter ที่ต้องใช้ในการพยากรณ์อากาศมี 2 ตัวคือ เวลา ($time) และ สถานที่ ($location)
ทางด้านซ้ายสุดคือ input จากผู้ใช้ ซึ่งมีความหลากหลาย และมี parameter ไม่เท่ากัน
ประโยคบนสุด ไม่บอกเวลาและสถานที่ ประโยคกลาง มีเวลา ส่วนประโยคล่างสุดมีครบทั้งเวลาและสถานที่
ในตัวอย่างนี้ Dialogflow จะ match input ทั้ง 3 แบบเข้ากับ Forecast intent (Dialogflow เข้าใจว่าผู้ใช้ต้องการข้อมูลพยากรณ์อากาศ)
และสกัด parameter ทั้ง $time และ $location ออกมาจากประโยคให้ด้วย ทำให้เราสามารถนำ parameter
ทั้ง 2 ตัวนี้ ไป query ระบบพยากรณ์อากาศ และนำคำตอบมาตอบกลับผู้ใช้งานได้ด้วย fulfillment

การทำงานของ fulfillment

โดยปกติ เราสามารถระบุ response ใน intent ได้เลยเป็น static response ข้อความที่ตอบกลับจะตายตัว (อย่างมากคือมีการนำ parameter มาใส่ใน response) แต่หากเราต้องการ response ที่ไม่ตายตัว เราสามารถใช้ fulfillment แทน response ได้

ใน fulfillment นั้น เราสามารถใช้ parameter ที่สกัดออกมา และส่งไปเรียก API ของระบบอื่นๆ เพื่อให้ได้คำตอบ และนำคำตอบนั้นกลับมาตอบกับผู้ใช้งานได้ ตามแผนภาพด้านล่าง

Entity

คือคุณลักษณะของตัวแปรแต่ละตัวที่ Dialogflow สามารถแกะออกมาได้จากประโยคที่ส่งเข้ามา

การนำ Developer Entities มาใช้ใน Intent

ยกตัวอย่าง Intent ของเราที่ชื่อ place_recommendation ให้เราลองเพิ่ม Training phrases ลงไปใหม่ คราวนี้จะเห็นว่ามันสามารถ detect ชื่อ @keyword_att ซึ่งเป็น Developer Entities ที่เราสร้างขึ้นมาได้แล้ว

รู้จักกับ ENTITIES และสร้าง DEVELOPER ENTITIES

หลักการทำงาน Chatbot

LINE
แนะนำสถานที่ท่องเที่ยว
DIALOGFLOW
อยากได้ที่แบบไหนคะ?
SERVER
ส่งข้อมูลแบบ json ที่ได้มาจากไลน์
LINE
ข้อมูลจากเซิร์ฟเวอร์ส่งกลับ
รับข้อมูลที่ได้ผ่านเซิร์ฟเวอร์
ส่งข้อมูลกลับให้บอท
หรือถ้ามีการเรียกใช้ข้อมูลส่งผ่าน
ส่งข้อมูลที่อ่านให้ไพธอน
PYTHON
ส่งข้อมูลไพธอนให้ SQLite
SQLite

BACK-END
ของ
SERVER
ส่วนที่ใช้กับ
FULLFILLMENT

BACK-END ของ SERVER

HEROKU

gunicorn

PORT
5000

BACK-END ของ SERVER

CSS

GitHub

Virtualenv

python Package

Flask

อุปกรณ์เสริม

HTML

CODE

LINE → DIALOGFLOW → PYTHON

การทำงานของ Code

การทำงานหลักของ Code

APP

File Edit View Navigate Code Help chatbot - app.py

app.py requirements.txt

```python
# Flask
app = Flask(__name__)
@app.route('/ACCOMMODATION/<place_id>')
def index(place_id):
    url = f"https://tatapi.tourismthailand.org/tatapi/v5/accommodation/{place_id}"

    try:
        Ac = requests.get(url,headers=my_headers).json()
    except HTTPError as http_err:
        print(f'HTTP error occurred: {http_err}')  # Python 3.6
    except Exception as err:
        print(f'Other error occurred: {err}')  # Python 3.6

    return render_template("GetAccommodationDetail.html",Ac=Ac['result'])


@app.route('/ATTRACTION/<place_id>')
def GetAttractionDetail(place_id):
    url = f"https://tatapi.tourismthailand.org/tatapi/v5/attraction/{place_id}"

    try:
        At = requests.get(url,headers=my_headers).json()
    except HTTPError as http_err:
        print(f'HTTP error occurred: {http_err}')  # Python 3.6
    except Exception as err:
        print(f'Other error occurred: {err}')  # Python 3.6
```

การทำงานหลักของ Code

FUNCTION CHATBOT

```python
import json
import requests
from requests.exceptions import HTTPError
from datetime import date

my_headers = {'Authorization' : 'Bearer GUrPRVm8IIgIkclxZQ8wp9ptzoGsY()jYrNYYMtMi)xPo8fSSb0AxGxX4mT9)e0rDzgYG5s4TbfKkCqvuP)ALX0=====2',
              'Accept-Language' : 'th'}

def place_request(question_from_dailogflow_dict):
    #Print intent that recived from dialogflow.
    print(json.dumps(question_from_dailogflow_dict, indent=4 ,ensure_ascii=False))
    categorycodes = question_from_dailogflow_dict["queryResult"]["outputContexts"][0]["parameters"]["categorycodes"]
    province = question_from_dailogflow_dict["queryResult"]["outputContexts"][0]["parameters"]["province"]

    keyword_att = question_from_dailogflow_dict["queryResult"]["outputContexts"][0]["parameters"]["keyword_att"]
    if keyword_att != [] and "ATTRACTION" not in categorycodes:
        categorycodes.append("ATTRACTION")

    keyword_rest = question_from_dailogflow_dict["queryResult"]["outputContexts"][0]["parameters"]["keyword_rest"]
    if keyword_rest != [] and "RESTAURANT" not in categorycodes:
        categorycodes.append("RESTAURANT")

    keyword_acc = question_from_dailogflow_dict["queryResult"]["outputContexts"][0]["parameters"]["keyword_acc"]
    if keyword_acc != [] and "ACCOMMODATION" not in categorycodes:
        categorycodes.append("ACCOMMODATION")

```

API คืออะไร

API หรือ Application Programming Interface เป็นบริการช่องทางการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลจาก
ระบบหนึ่งไปสู่ระบบอื่น ๆ ที่มีความสะดวก รวดเร็ว ปลอดภัย
หน้าที่หลักของ API คือ คอยรับคำสั่งจากฝั่งลูกข่าย (Client) ซึ่งก็คือ Application ต่าง ๆ
เช่น Web Application, Mobile Application, Desktop Application
เป็นต้น

ตัวอย่าง
API
สำหรับการเรียกใช้งาน
CHATBOT
ท่องเที่ยว

ตัวอย่างการทำงานของแต่ละ intent

Accommodation

Accommodation

รายละเอียด

รายละเอียดสถานที่

รหัสสถานที่ : P02015612

ชื่อสถานที่ : เซนเตอร์พอยต์ เซอร์วิส อพาร์ตเมนต์ ทองหล่อ

ลองจิจูด : 100.5820661

ละติจูด : 13.73146105

รหัสแผนที่ :

SHA

sha_name : เซนเตอร์พอยต์ เซอร์วิส อพาร์ตเมนต์ ทองหล่อ

sha_type_code : 2

sha_type_description : โรงแรมและที่พัก สถานที่จัดการประชุม

sha_cate_id : 2

sha_cate_description : SHA Plus

ข้อมูลสถานที่

การแนะนำ :

รายละเอียด :

ประเภทร้านอาหาร

ใบอนุญาตที่ลงทะเบียน :

มาตรฐานดาว :

แสดงเวลาเช็คอิน :

แสดงเวลาชำระเงิน :

หมายเลขห้อง : None

ช่วงราคา :

เบอร์โทร

None

โทรศัพท์มือถือ

None

แฟกซ์

อีเมล

['saisunee.s@centrepoint.com']

URL

['www.centrepoint.com']

สิ่งอำนวยความสะดวก :

บริการ :

None

วิธีการชำระเงิน :

ปลายทาง

มาตรฐาน

TAGS

None

รางวัล

None

วันที่อัพเดท

2021-11-15T13:00:00

Attraction

DTAC
04:01
82%
ROAR ALONE
อ่านแล้ว
04:01
แนะนำที่ท่องเที่ยว
อยากได้ที่แบบไหนคะ?
04:01
อ่านแล้ว
04:01
วัด
แจ้งจังหวัดที่ท่านต้องการไปเลยค่ะ
04:01
อ่านแล้ว
04:01
กรุงเทพฯ
พระปรางค์วัดอรุณราชวราราม ราชวรมหาวิหาร
วัดอรุณราช วรมหาวิหาร

Attraction

รายละเอียด

รายละเอียดสถานที่

รหัสสถานที่ : P03006845

ชื่อสถานที่ : พระปรางค์วัดอรุณราชวรารามราชวรมหาวิหาร

ลองจิจูด : 100.4889632

ละติจูด : 13.74379702

รหัสแผนที่ : TH-10DG421C.X5

ช่องทางการติดต่อ

เบอร์โทร

['0 2891 2185']

โทรศัพท์มือถือ

None

แฟกซ์

None

อีเมล

None

URL

None

บริการ : None

วิธีการชำระเงิน

วิธีการเดินทาง :

วันธรรมดา

DATE	วันที่ 1	วันที่ 2	วันที่ 3	วันที่ 4	วันที่ 5	วันที่ 6	วันที่ 7
DATE	จันทร์	อังคาร	พุธ	พฤหัสบดี	ศุกร์	เสาร์	อาทิตย์
เวลาเปิดต่อสัปดาห์							

ปลายทาง

กรุงเทพมหานคร

วันที่อัพเดท

2022-02-10T12:02:05.32

Restaurant

dtac
23:38
29%
ROAR ALONE
อ่านแล้ว 21:41
แนะนำที่กิน
อยากได้ที่แบบไหนคะ?
21:41
อ่านแล้ว 21:41
ก๋วยเตี๋ยว
จังหวัดอะไรคะ
21:41
อ่านแล้ว 21:41
ชลบุรี
NO IMAGE AVAILABLE
ก๋วยเตี๋ยวเป็ดป้าโชค ณ อ่างศิลา
RESTAURANT
NO AVA
ก๋วยเตี๋ยวไก่
RESTAU

Restaurant

รายละเอียด

รายละเอียดสถานที่

รหัสสถานที่ : P08018217

ชื่อสถานที่ : ก๋วยเตี๋ยวเป็ดป้าโชค ณ อ่างศิลา

ลองจิจูด : 100.9220178

ละติจูด : 13.32553865

รหัสแผนที่ :

SHA

sha_name : ก๋วยเตี๋ยวเป็ดป้าโชค ณ อ่างศิลา

sha_type_code : 1

sha_type_description : ภัตตาคาร/ร้านอาหาร

sha_cate_id : 1

sha_cate_description : SHA

ข้อมูลสถานที่

การแนะนำ :

รายละเอียด :

ประเภทร้านอาหาร

ประเภทที่ 1

ประเภทอาหาร : None

ที่ตั้ง

ที่อยู่ : 49 ม.5

สิ่งอำนวยความสะดวก : None

บริการ : None

วิธีการชำระเงิน :

CODE :

DESCRIPTION :

วิธีการเดินทาง :

เวลาทำการ

เปิดแล้ว :

ระยะเวลา :

วันธรรมดา

DATE	วันที่ 1	วันที่ 2	วันที่ 3	วันที่ 4	วันที่ 5	วันที่ 6	วันที่ 7
DATE	จันทร์	อังคาร	พุธ	พฤหัสบดี	ศุกร์	เสาร์	อาทิตย์
เวลาเปิดต่อสัปดาห์	07:00 - 17:00	07:00 - 17:00	07:00 - 17:00	07:00 - 17:00	07:00 - 17:00	07:00 - 17:00	07:00 - 17:00

ปลายทาง

มาตรฐาน

TAGS

None

รางวัล

None

พื้นที่

วันที่อัพเดท

2021-11-15T13:00:00

Request notify

Record trip

DTAC 23:13 VoLTE 71%
ROAR ALONE
วันนี้บันทึกข้อมูลหน่อย
อ่านแล้ว 23:12
1.) สวัสดีค่ะ ^_^ ทางเราสามารถให้ท่านบันทึกข้อมูลได้ดังต่อไปนี้นะค่ะ (ชื่อทริป,ชื่อผู้ร่วมทริป,สถานที่พัก,ชื่อร้านอาหาร,ยอดรวมค่าใช้จ่ายตลอดทั้งทริป)
2.) กรอกข้อมูลเข้ามาได้เลยค่ะ (เช่น ทริปเกาะล้านกับเจนนี่กับโรงแรมวินเนอร์วิวกับร้านครัวเฉลียงลมกับ2,000บาท)
23:12
ทริปเกาะล้านกับเจนนี่กับโรงแรมวินเนอร์วิวกับร้านครัวเฉลียงลมกับ2,000บาท
อ่านแล้ว 23:12
ชื่อทริปที่คุณต้องการบันทึก คือ เกาะล้าน และชื่อผู้ร่วมทริปคือ เจนนี่ และสถานที่พัก คือ โรงแรมวินเนอร์วิว และชื่อ

DTAC 23:13 VoLTE 71%
ROAR ALONE
ชื่อทริปที่คุณต้องการบันทึก คือ เกาะล้าน และชื่อผู้ร่วมทริปคือ เจนนี่ และสถานที่พัก คือ โรงแรมวินเนอร์วิว และชื่อร้านอาหารคือ ร้านครัวเฉลียงลม และยอดรวมค่าใช้จ่ายตลอดทั้งทริปคือ 2,000บาท ใช่มั้ยค่ะ
23:12
อ่านแล้ว 23:12
ใช่
บันทึกเรียบร้อยแล้วค่ะ
23:12
อ่านแล้ว 23:13
ขอดูข้อมูลหน่อย
ชื่อทริป : เกาะล้าน และชื่อผู้ร่วมทริป : เจนนี่ และสถานที่พัก : โรงแรมวินเนอร์วิว และชื่อร้านอาหาร : ร้านครัวเฉลียงลม และยอดรวมค่าใช้จ่ายทั้งหมด : 2,000บาท

เครื่องมือการสร้าง CHATBOT

LINE DEVELOP ผูกกับ DIALOGFLOW

GitHub

เพื่อ DEPLOY ขึ้น SERVER

จำเป็นที่จะต้องอัพไฟล์ลง GITHUB และเชื่อมต่อกับ HEROKU

FLASK

Flask คือ web framework ที่เขียนขึ้นมาสำหรับ Python เพื่อใช้ร่วมกัน webserver เช่น Apache และได้รับการยอมรับจาก community we pages ชั้นนำเช่น Pinterest, LinkedIn เป็นต้น โดย Flask ถูกเรียกว่า micro framework เพราะว่า มันไม่ต้องการเครื่องมือ หรือ library อะไรมาก อีกทั้ง ไม่จำเป็นต้องมี database ด้วย แต่อย่างไรก็ตาม Flask ก็ยังรองรับการเพิ่ม extensions พิเศษได้ ถ้ามันรองรับ Flask

FLASK

TF-IDF

TF-IDF

เป็น Algorithm สำหรับการตรวจสอบคำ (Text Mining) โดยนำประโยคสองประโยค
ขึ้นไปมา Dot กันเพื่อหา Weight ของ คำนั้นๆ

TF-IDF

$$TF = \frac{\text{จำนวนของคำนั้นๆใน doc}}{\text{จำนวนของคำทั้งหมดใน doc}}$$

$$IDF = \log\left(\frac{\text{จำนวน doc ทั้งหมดที่ใช้พิจารณา}}{\text{จำนวน doc ที่มีคำนั้นปรากฏอยู่}}\right)$$

Document A

ทำหน้าที่อะไร

Document B

หน้าที่ของคุณคืออะไร

Term	TF(A)	TF(B)	IDF	TF • IDF(A)	TF • IDF(B)
ทำ	$\frac{1}{3}$	0	$\log(\frac{2}{1})$	0.1	0
หน้าที่	$\frac{1}{3}$	$\frac{1}{5}$	$\log(\frac{2}{2})$	0	0
อะไร	$\frac{1}{3}$	$\frac{1}{5}$	$\log(\frac{2}{2})$	0	0
ของ	0	$\frac{1}{5}$	$\log(\frac{2}{1})$	0	0.06
คุณ	0	$\frac{1}{5}$	$\log(\frac{2}{1})$	0	0.06
คือ	0	$\frac{1}{5}$	$\log(\frac{2}{1})$	0	0.06

วิธีคำนวณหา intent

1.) รับ input ของข้อความ user

EX "หน้าที่ของเธอคืออะไรอะ"

2.) แบ่งคำจากประโยค

"หน้าที่","ของ","เธอ","คือ","อะไร","อะ"

3.) ตัดคำที่ไม่มีใน document

"หน้าที่","คือ","อะไร"

4.) รวมคะแนนแต่ละคำในแต่ละ document

Tf-Idf (A) = 0+0+0 = 0

Tf-Idf (B) = 0+0+0.06 = 0.06

5.) นำ document ที่คะแนน Tf-Idf รวมมากที่สุด มาหา intent

เนื่องจาก document B อยู่ใน intent duty

ตอบ duty

```
{"intents": [
        {"tag": "greeting",
         "patterns": ["สวัสดี", "Hi", "หวัดดีค่ะ", "ว่าไง", "มีใครอยู่ป่าว","สวัสดีตอนเช้า",
    "สวัสดีตอนบ่าย","สวัสดีตอนค่ำจ้า","อยากคุยด้วยจัง","ทักทาย","สวัสดีปีใหม่ค่ะ","มีเรื่องจะคุยด้วยครับ",
    "ทำอะไรอยู่จ๊ะ","hello","ฮาโหล","ทัก","สวัสดีค่า","สายัณห์สวัสดิ์",
    "อรุณสวัสดิ์","สวัสดีน้า","สวีดัส","สวัสดีจ้ะ","ไม่เจอกันตั้งนาน","ขอทักทาย",
    "ทักครับ","หวัดดี","ไม่คุยกันนานมาก","ขอคุยด้วยหน่อยสิ","สวัสดีค่ะ","ดีครับ"
         ],
         "responses": ["สวัสดีค่ะ", "ดีจังที่ได้เจอคุณ", "สวัสดีค่ะ, มีอะไรให้ช่วยคะ"],
         "context_set": ""
        },
        {"tag": "duty",
         "patterns": ["ข้อมูลของคุณได้มากจากไหน","หน้าที่ของคุณคืออะไร","จุดประสงค์ของคุณคืออะไร","คุณต้องการสิ่งใด","โรอามีหน้าที่อะไรหรอ","อยากรู้หน้าที่ของคุณ",
    "เราเชื่อใจคุณได้มากขนาดไหน","เราต้องบอกคุณว่าอะไร","หน้าที่มีอะไรบ้าง","หน้าที่ของโรอาอ่ะ","ข้อมูลคุณได้มาจากไหน","หน้าที่",
    "คุณทำอะไร","ประโยชน์ของคุณ","คุณทำด้านใด","คุณต้องทำอะไรบ้าง","คุณทำอะไรให้เรา","จุดประสงค์",
    "โรอาคุณแนะนำอะไรได้บ้างอ่ะ","คุณต้องการสื่ออะไร","แชทบอทนี้ทำหน้าที่อะไร","คุณทำอะไรได้บ้าง","โรอาอโลนนีทำอะไรได้บ้าง","บทบาทของคุณคือ",
    "จุดประสงค์คืออะไร","อำนาจหน้าที่","คุณรับผิดชอบอะไรบ้าง","คุณทำอะไรอ่ะ","แชทบอทโรอาทำไร","ด้านท่องเที่ยวของคุณมีอะไรบ้าง"
         ],
         "responses": ["หน้าที่ของโรอาคือการแนะนำสถานที่ท่องเที่ยวให้คุณได้ทราบ ที่ตั้งจุดเด่นประวัติ การแบ่งเวลาของคุณว่าควรแบ่งทริปอย่างไร ให้ใช้เวลาคุ้มค่ามากที่สุด ในการไปเที่ยว ในกรณีที่มีธุระหลังจากไปเที่ยว หรือเวลาหลงทาง จากจุดที่เราจะไป
         "context_set": ""
        },
        {"tag": "story",
         "patterns": ["จุดเริ่มต้นของโรอามาจากไหนหรอ","โรอามาจากไหน","เล่าประวัติของคุณมาหน่อย","ทำไมถึงมาเป็นโรอา","คุณมาจากที่ใด","ทำไมถึงเป็นโรอา",
    "ทำไมถึงเป็นโรอาอโลน","ROAR ALONE คืออะไร","ทำไมต้อง ALONE","ROAR ALONE คืออะไรอ่ะ","ชื่อโรอามาจากไหน","คุณคืออะไร",
    "ทำไมคุณถึงมาเป็นแชทบอท","อะไรคือโรอา","โรอาเป็นชื่อของใคร","เล่าความเป็นมาของคุณที","ก่อนจะมาเป็นโรอาอโลน เที่ยวเหนื่อยไหม","โรอาคืออะไร",
    "โรอาคือสิ่งใด","เล่าจุดเริ่มต้นของคุณมาที","จุดเริ่มต้นชื่อของคุณมาจากที่ไหน","เล่าเรื่องโรอาให้ฟังหน่อย","ก่อนจะมาเป็นแชทบอทคุณเป็นอะไรมาก่อน","โรอาขอประวัติของเธอหน่อย",
    "เล่าประวัติของคุณมาให้เราทราบที","คุณผ่านอะไรมาบ้าง","ทำไมโรอาถึงรู้สถานที่ได้เยอะจัง","คุณไปเที่ยวด้วยตัวเองหรือเปล่า","เล่าการเป็นโรอาให้ฟังหน่อย","ROAR ALONE"
         ],
         "responses": ["โรอาเป็นชื่อของเจ้าหน้าที่คนหนึ่งที่ทำหน้าที่แนะนำสถานที่ท่องเที่ยวให้เด็กๆได้รับทราบ แต่ก่อนจะมาเป็นเจ้าหน้าที่นั้น โรอาไม่รู้สถานที่ท่องเที่ยวที่ไหนเลย เป็นแค่เจ้าหน้าที่หลงทาง แนะนำที่ตั้งที่พักจุดเด่นยังไม่ได้ มีผิดๆถูกๆ โรอาจึงจำ
         "context_set": ""
        },
```

```
    {"tag": "goodbye",
     "patterns": ["เจอกันนะ","ไว้เจอกันวานหลัง","บาย","เจอกาน","บ๊ายบาย","เจอกันนะ",
"ไปละ","ฉันไปดีกว่า","see ya","bye","โชคดีนะ ไปก่อนละกัน","ลาก่อน",
"ไปก่อนนะค่ะ","ขอตัวก่อนดีกว่าจ้า","ขอลาละ","ไว้เจอกันใหม่","ผมว่าผมไปก่อนดีกว่า","ไปละจ๊ะ",
"ไว้คุยกันใหม่","เอาไว้ค่อยคุย","โทษทีนะพอดีมีธุระ","ขอไปทำงานก่อนนะค่ะ","ขอตัวละ","เบื่อละไปก่อนนะ",
        "goodbye","เจอกันนะคะพอดีไม่งานต่อ","ขอให้เป็นวันที่ดี เจอกันค่ะ","ขอลานะคะ","ไปก่อนนะจ๊ะ","ลาละคับ"
     ],
     "responses": ["บายค่ะ", "ไว้เจอกันวานหลังค่ะ", "บาย!","ขอให้เป็นวันที่ดีนะค่ะ"],
     "context_set": ""
    },
    {"tag": "name",
     "patterns": ["ขอถามชื่อหน่อย","เธอชื่ออะไร","คุณคือใคร","ฉันควรเรียกเธอว่าอะไร","นามของคุณคืออะไรอะ","ไม่ทราบว่ามีชื่อหรือเปล่า",
"อยากรู้ชื่ออะ","อยากให้เรียกคุณว่าอะไร","ชื่ออ่านยังไง","ชื่อเสียงเรียงนามเจ้าคืออะไร","ชื่ออะไรจ๊ะ","ชื่อจริงว่าอะไร",
"มีชื่อป่ะ","อยากรู้จังครับ ว่าคุณชื่ออะไร","เรียกเธอว่าอะไรดีน้า","มีชื่อป่ะเนี่ย","ขอชื่อเล่นหน่อย","คนสร้างตั้งชื่ออะไรให้",
"ใครอะ","น้องชื่ออะไรครับ","พี่ชื่อไรเหรอครับ","ขอสอบถามชื่อหน่อยได้มั้ยค่ะ","ชื่ออะไรเหรอ","ชื่อ?",
"ขอทราบชื่อด้วยคับ","อยากรู้ชื่อจัง","ป้าชื่อไร","น้าชื่ออะไรจ๊ะ","อยากให้เรียกว่าอะไร","ชื่ออะไรน้า"
     ],
     "responses": ["เรียกฉันโรอาได้เลยค่ะ", "ฉันชื่อโรอาค่ะ", "โรอา อโลนค่ะ" , "ฉันชื่อ โรอา, คนช่วยแนะนำที่เที่ยวให้คุณค่ะ"],
     "context_set": ""
    },
    {"tag": "hungry",
     "patterns": ["หิวมากเลย","ไม่ได้กินอะไรตั้งแต่เที่ยวแล้วอะ กินอะไรดี","เลือกไม่ถูกว่ากินอะไรดีแนะนำหน่อย","เลือกเมนูอาหารมาหน่อยสิ","หิวไม่ไวแล้ว","แนะนำอาหารหน่อย",
"กินอะไรดีนะ","อยากกินข้าว เอาอะไรดีจ๊ะ","แดกอะไรดี","รับประทานอาหารอะไรดีครับ","โครตหิวเลย","ไม่มีอาหารตกถึงท้องเลย",
        "hungry","ไม่รู้จะทานอะไร","หิวโว้ย","อยู่ร้านข้าว เอาอะไรดี","อาหารมีอะไรบ้าง","มีเมนูอะไรน่าสนใจ",
"เมนูอาหารยอดฮิตคืออะไร","คนไทยชอบกินอะไร","โรอาแนะนำเมนูอาหารหน่อย","โรอาชอบกินอะไร","คิดเมนูให้เราหน่อย","อาหารที่น่ากินมีไรบ้าง",
"หิวไม่ไหวแล้ว","ทานไรดีครับเนี่ย","หาไรกินดีกว่า","เมนูอาหารที่ต้องกิน","โครตหิวบอกตรงๆ","กินเมนูอะไรดีค่ะ"
     ],
     "responses": ["กะเพราไก่ไข่ดาวค่ะ", "ลองไก่กรอบ 3 รสค่ะ"],
     "context_set": ""
    },
```

```
        {"tag": "homework",
         "patterns": ["การบ้านการงานเสร็จยังอะ","การบ้านภาษาไทยเสร็จยังอะ","การบ้านอัลกอสร็จยังอะ","การบ้านดิสครีตเสร็จยังอะ","การบ้านแอปแสต็กเสร็จยังอะ","การบ้านโปรแกรมสกิลเสร็จยังอะ",
        "การบ้านไพธอนเสร็จยังอะ","การบ้านจาวาเสร็จยังอะ","การบ้านจาวายากจัง","การบ้านไพธอนยากจัง","การบ้านที่ต้องส่งพรุ่งนี้มีอะไรบ้าง","การบ้านเยอะจัง",
        "อยากทำการบ้านแล้วอะ","การบ้านสังคมเสร็จยังอะ","การบ้านฟิสิกเสร็จยังอะ","การบ้านคณิตยากจัง","การบ้านฟิสิกยากจัง","การบ้านสังคมยากจัง",
        "การบ้านอิเล็กยากจัง","การบ้านอัลกอยากจัง","การบ้านการงานเสร็จยังอะ","การบ้านภาษาไทยเสร็จยังอะ","การบ้านอัลกอสร็จยังอะ","การบ้านดิสครีตเสร็จยังอะ",
        "การบ้านแอปแสต็กเสร็จยังอะ","การบ้านโปรแกรมสกิลเสร็จยังอะ","การบ้านไพธอนเสร็จยังอะ","การบ้านจาวาเสร็จยังอะ","การบ้านจาวายากจัง","การบ้านไพธอนยากจัง"
         ],
         "responses": ["ได้เลย", "ดูวันเวลาส่งใน team ได้เลย","เสร็จแล้วววว","ยังเลยย"],
         "context_set": ""
        },
        {"tag": "sleep",
         "patterns": ["เมื่อคืนนอนกี่โมงอะ","นอนวันละกี่ ชม. อ่อ","ง่วงจัง อยากนอนแล้ว","ถ้าสอบเสร็จจะนอนสัก 10 ชม. เลยคอยดู","นอนไม่หลับทำไงดี","ทำไมตอนนอนชอบหิวอะ",
        "นอนเล่นไปมา เผลอหลับเลย","เมื่อคืนฉันนอนหลับเร็วหละ","นอนให้เพียงพอดีต่อสุขภาพยังไงอะ","ทำไมผู้ใหญ่ถึงแนะนำให้นอนหลับให้เพียงพอ","เมื่อคืนนอนน้อยอะ","เมื่อคืนนอนเต็มอิ่มเลย",
        "อยากนอนสัก 10 ชม.","ที่ไหนมีแข่งนอนไหม อยากแข่งจัง","เรียนมหาลัยเห็นคุณค่าของการนอนเลยอะ","ทำไมฉันนอนดึกอย่างนี้","อยากปรับเปลี่ยนเวลานอนต้องทำไงบ้างอะ","ฉันตื่นสายเพราะเมื่อคืนฉันนอนดึก",
        "ขอโทษทีที่มาสาย เมื่อคืนนอนดึกไปหน่อย","เดี๋ยวนอนฉันนอนดึก เพราะติดเกม","เดี๋ยวนอนฉันนอนดึก เพราะติดซีรี่","เดี๋ยวนอนฉันนอนดึก เพราะติดหนัง","เดี๋ยวนอนฉันนอนดึก เพราะต้องเคลียการบ้าน","เดี๋ยวนอนฉันนอนดึก เพราะติดเที่ยว",
        "อยากนอนไวๆจังงง","นอนไปก็เท่านั้น","ทำไมคนเราต้องนอนด้วย","วันนี้เราคงไม่ได้นอนแล้วแหละ การบ้านเยอะจัด","ไปนอนกันดีกว่า","ไม่ไหวละ ขอนอนก่อนเด้อ"
         ],
         "responses": ["8 ชม. ต่อวัน", "ก็นอนสิ"],
         "context_set": ""
        }
    ]
}
```

ทดสอบความแม่นยำ

เราได้ทำการ สร้างตารางขึ้นมา แล้วจากนั้นก็ได้ ทำการทดสอบกลุ่มตัวอย่าง ของแต่ละ Intent ที่เรามี จากนั้น เราจะดูว่า จาก กลุ่มตัวอย่างทั้งหมดที่ได้มา มีผ่านเท่าไหร่บ้าง จากนั้น เอาสัดส่วนที่ผ่านมา หารกับจำนวนกลุ่มตัวอย่าง ทั้งหมด แล้วได้มาเป็นผลของความแม่นยำ

โดยแบ่งข้อมูลแต่ละ intent มา 20% ใช้ทดสอบ

```
[ ] import random
    from sklearn.feature_extraction.text import  TfidfTransformer , CountVectorizer
    from sklearn.naive_bayes import MultinomialNB
    import json
    import numpy as np
    from attacut import tokenize
```

```
[ ] count_vect = CountVectorizer(tokenizer=tokenize)
```

```
[ ] with open("/content/drive/MyDrive/roar/intents.json") as file:
        data = json.load(file)
    Doc = []
    intents_doc = []
    Xtest = []
    intents_Xtest = []
```

```
[ ] for intent in data["intents"]:
      for i in range(0,30):
        if i > 23:
          Xtest.append(intent['patterns'][i])
          intents_Xtest.append(intent["tag"])
        else:
          Doc.append(intent['patterns'][i])
          intents_doc.append(intent["tag"])
```

```
[ ] doc_count =  count_vect.fit_transform(Doc).toarray()
    tfidf_transformer = TfidfTransformer()
    tfidf_transformer.fit(doc_count)
    training_vector = tfidf_transformer.transform(doc_count).toarray()
```

```
training_vector[30]
```

```
array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.4713608 , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.22090358, 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.39538088, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.4713608 , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.37362533, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.31940097,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.33000106, 0.        ])
```

```
[ ] clf = MultinomialNB()
```

```
[ ] clf.fit(training_vector, intents_doc)
```

```
MultinomialNB()
```

```
[ ] def match_intent(text):
      text_count = count_vect.transform(text)
      text_tfidf = tfidf_transformer.transform(text_count)
      return clf.predict(text_tfidf)
```

```
[ ] correct = 0
    Xtest_match = match_intent(Xtest)
    for i in range(0,len(Xtest)):
      if Xtest_match[i] == intents_Xtest[i]:
        correct += 1
```

ACCURACY TEST

83.3%

```
[ ] print('Accuracy percentage is {percent:.2f}%'.format(percent= (correct/len(Xtest))*100 ))

    Accuracy percentage is 83.33%
```