

การตอบกลับข้อความอัตโนมัติด้วยสติ๊กเกอร์

ที่มาและความสำคัญ

เนื่องจากปัจจุบันมีการใช้การสื่อสารผ่านโซเชียลมีเดีย(แชท) กันอย่างแพร่หลาย ซึ่งในปัจจุบันแทบทุกโซเชียลมีเดีย มีการใช้ฟังก์ชันสติ๊กเกอร์กันเกือบทุกโซเชียลมีเดีย การตอบด้วยสติ๊กเกอร์จึงเป็นที่นิยมของผู้ใช้งาน เนื่องจากง่าย เร็ว และบางครั้งสื่อความหมายได้ดีกว่าการพิมพ์ข้อความ

โดยโปรเจกต์นี้ได้ทำการนำข้อความจากการที่ผู้ใช้พิมพ์เข้ามาจากการแชท มาทำการแบ่งกลุ่มโดยใช้ machine learning แบบ unsupervised data เพื่อตอบคำถามให้แก่ผู้ใช้โดยอัตโนมัติ

วิธีการดำเนินงาน

- ข้อมูลที่ใช้: นำข้อมูลการแชทมาจากแอปพลิเคชัน Line และ Facebook (ประวัติการสนทนา) โดยใช้ฟังก์ชันดาวน์โหลดข้อความบนแอปพลิเคชันนั้น ๆ กรณีดาวน์โหลดจาก Line PC จะได้เป็นไฟล์ .txt และ กรณีโหลดจาก Facebook จะเป็นไฟล์ .json
- การดำเนินงานแบ่งเป็น 3 ส่วน
 1. การจัดการกับข้อมูล (ไฟล์ cleandata.py) เป็นการนำข้อมูลที่ได้จากแอปพลิเคชันมาสร้างเป็น sparse feature dict โดยใช้คำและ sentiment เป็น feature (ตอนแรกลองใช้ความยาวของข้อความและจำนวนคำรวมด้วย แต่พบว่ามีผลมากเกินไป จึงไม่นำมาพิจารณา) มีการใช้ package ดังนี้
 - re (regular expression) เพื่อขูดข้อมูลจาก Line
 - os เพื่อใช้ในการเปิดไฟล์ที่อยู่ต่าง folder กัน
 - pythainlp เพื่อใช้ในการตัดคำและ sentiment analysis จากข้อความ
 2. การสร้างโมเดลการ cluster ข้อมูลโดยใช้ KMeans จากแพ็คเกจของ sklearn (ไฟล์ make_model.py) โดยใช้ข้อมูลจากการจัดการในส่วนที่ 1 หลังจากนั้นใช้ pickle เพื่อ dump โมเดลไว้ใช้งาน
 3. การใช้งานโมเดลที่สร้างขึ้น (ไฟล์ display.py) นำโมเดลที่สร้างขึ้นมาใช้งาน โดยดูกลุ่มที่แบ่งได้ ทดสอบกับข้อความที่ลองพิมพ์ขึ้นว่าอยู่ในกลุ่มใด มีการหา top feature เพื่อนำมา label กลุ่มที่ได้ ว่าควร reply ด้วยสติ๊กเกอร์ใด

ผลการดำเนินงาน

โปรแกรมที่ได้สามารถแบ่งกลุ่มจากการพิมพ์ข้อความลงไป เพื่อการตอบกลับด้วยสติ๊กเกอร์ได้ แต่ค่อนข้างเป็นโมเดลที่มีประสิทธิภาพไม่เท่าไหร่ (ไม่ได้ทำการประเมินค่าประสิทธิภาพต่าง ๆ เนื่องจากการ reply ด้วยสติ๊กเกอร์อาจไม่ถูกไม่ผิด) บางครั้งโมเดลมักทายลงไปในกลุ่มเดียวกัน ทั้งที่ข้อความไม่ได้ออกไปทางเดียวกัน จากการทดลอง predict กลับด้วย training data พบว่ามีการจำแนกกลุ่มได้มากกว่า จึงคิดว่าโมเดลอาจมีการ fit กับข้อมูลมากเกินไป

```
[1097]: display.get_top_feature(0,20)
['-', 'C', '\t', '.', 'd', '>', '__sentiment', '=', ')', ':', ',', 'n', 'l', 'lu', ';', '2', 'the', 'ln', 'ã', ' ']

[1098]: display.get_top_feature(1,20)
[['', 'C', 'i', '.', 'j', ']', '-', ' ', ' ', ' ', 'int', 'O', ' ', ' ', '}', 'n', '{', ' ', ' ', '1', ');', 'k', ' ']

[1099]: display.get_top_feature(2,20)
['C', ' ', '=', '[', ']', ' ', '1', 'i', 'j', ' ', ' ', ' ', ' ', 'int', 'O', 'p', '}', ')', 'a', ' ', ' ', ']', 'k']

[1100]: display.get_top_feature(3,20)
['n', 'lu', 'q', 'æ', 'd', 'h', 'h', 'ã', 'uc', 'ñ', 'm', 'un', 'hu', 'zi', 'udr', 'desu', 'au', 'B', 'th', 'üer']

[1101]: display.get_top_feature(4,20)
['2', 'O', '7', '3', '6', '8', '4', '5', ' ', '1', '9', 'Q', 'u', 'K', 'O', ' ', '_', 'e', 'a', 'G', 'Z', 'P']

[1102]: display.get_top_feature(5,20)
['h', 'lu', 'q', 'æ', 'uc', 'äu', 'h', 'un', 'uc', 'udr', 'uo', ' ', 'm', 'B', 'ã', 'ä', 'h', 'de', 'ji', 'æ']

[1103]: display.get_top_feature(6,20)
['1', '[', '=', ' ', ' ', 'O', ' ', '/', 'C', '][', ' ', '2', 'i', 'j', 'int', ']', ' ', ' ', 'f', '}', 'a', ' ', ' ']

[1104]: display.get_top_feature(7,20)
['B', 'O', 'E', '8', '25', '9', 'A', '2', '3', ' ', '/', '1', 'F', '\n', '__sentiment', '=', '81', ':', '/', '4', '&']

[1105]: display.get_top_feature(8,20)
[' ', ' ', 'e', ' ', 'tk', '#', 'C', 'state', '=', ' ', ' ', ' ', 'r', "['", ']', '[', '[', 'print', 'O', 'parser', 'in', ')', '3']

[1106]: display.get_top_feature(9,20)
['.', 'C', '1', '=', ' ', ' ', ' ', ' ', ' ', ' ', 'int', 'j', 'C', '[', 'O', ' ', ' ', '+', ']', ' ', ' ', ' ', 'hamin', 'p']
```

ปัญหาอาจเกิดจากตัวสมการของโมเดลที่ใช้ไม่เหมาะกับประเภทของข้อมูลหรือข้อมูลที่นำไปเทรนไม่ได้มีการจัดการที่ดีพอ สังเกตได้จากเมื่อแสดงผล top feature มักสลับลักษณะแปลก ๆ มากกว่าค่า หรืออาจจากการที่ดาวนโหลดมาจากเซตส่วนตัว บางครั้งมีการคัดลอกข้อความหรือ code โปรแกรมมาแปะ เพื่อส่งกันในแชท ทำให้ไม่ใช่ข้อความที่มนุษย์พิมพ์กันในชีวิตประจำวันจริง ๆ หรือ fit จากการคูณงานมากเกินไป




อุปสรรค

- การ clean data:

06:59 Anan Sojipan รูปนี้ที่ Tokyo ครับ

07:04 Anan Sojipan Lake okutama

07:04 'หนึ่ง.. 🙏🙏🙏 ปีนี้หรือปีก่อนครับ สวยจัง

07:05 'หนึ่ง..    เห็นผ่านๆว่าปีนี้บางที่เริ่มเหลืองแล้วด้วย

จากภาพจะเป็นข้อมูลที่ดาวน์โหลดมาจากไลน์ จะยากต่อการชุดข้อมูลออกมา จะสังเกตได้ว่า บางคนมีการตั้งชื่อไลน์ที่เว้นวรรคด้วย space จะไม่สามารถชุดข้อความออกมาโดยไม่มีข้อติดมาได้

- การดาวน์โหลดข้อมูลจากLine: การกวดาวน์โหลดจาก Line PC นั้นแต่ละครั้งจะโหลดเฉพาะข้อมูลที่แสดงในหน้าที่เราเปิดอยู่เท่านั้น และความยาวจำกัด (โดยเฉลี่ยอยู่ที่ 5 วันต่อครั้งการดาวน์โหลด) ทำให้ต้องดาวน์โหลดหลายครั้งและเสี่ยงต่อการซ้อนทับของข้อมูล
- การดาวน์โหลดข้อมูลจากFacebook: ผู้จัดทำมาทราบช่วงหลัง ๆ ของการทำโปรเจกต์ว่าสามารถดาวน์โหลดแชทออกมาได้เป็น .json แต่การดาวน์โหลดจาก Facebook นั้นจะต้องดาวน์โหลดมาทั้งรูปภาพ วิดีโอ และไฟล์ โดยเมื่อกด request การดาวน์โหลดแล้วทาง Facebook จะใช้เวลาประมวลผลประมาณ 1-2 วันถึงจะสามารถดาวน์โหลดได้
- การตัดคำภาษาไทยและการ sentiment analysis ใช้เวลาในการรันโปรแกรมนาน ทำให้ไม่สามารถสร้างข้อมูลจากที่ดาวน์โหลดลงมาทั้งหมดได้ จึงเลือกใช้ 100,000 ข้อความ
- การไม่ทราบเครื่องมือที่เหมาะสมกับการใช้งานจริง ๆ อาจทำให้เลือกเครื่องมือได้ไม่ดีพอ ทำให้โมเดลไม่ตีเท่าที่ควร และผู้จัดทำมีประสบการณ์กับการทำ machine learning น้อย ทำให้อาจไม่ทราบการปรับค่าบางอย่างที่ช่วยให้ผลดีขึ้น

ร้านนาน

ปัญหาจากข้อมูล

ข้อมูลจากไลน์ จะมาในรูปแบบดังภาพ

สรุปผล

มองว่าตัวโปรเจกต์ที่ได้ผลถ้ามองในเรื่องการใช้งานยังไม่ได้เท่าที่ควร แต่ได้ประโยชน์จากการเรียนรู้การใช้เครื่องมือต่าง ๆ เช่น การตัดคำภาษาไทย การ cluster จาก unsupervised data เป็นต้น ได้นำความรู้และทักษะการเขียนโปรแกรมภาษาไพธอน และการทำ NLP เบื้องต้นมาวิเคราะห์ออกแบบการทำงานว่าควรจัดการอย่างไรให้สามารถวิเคราะห์ข้อมูลที่เป็น string ออกมาได้ มองเห็นแนวทางในการประมวลผลข้อความมากยิ่งขึ้น