การจำแนกเจตนาการสนทนาเพื่อการสร้างแชทบอท
Intent Classification for Building Chatbot

โดย

นายณัฐพล เดชประมวลพล รหัสนักศึกษา 6110210129
นางสาววริศรา พิสุทธิ์เธียร รหัสนักศึกษา 6110210373

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ วิทยาลัยสงขลานครินทร์
ปีการศึกษา 2564

กิตติกรรมประกาศ

การทำโครงงานทางวิทยาการคอมพิวเตอร์ “การจำแนกเจตนาการสนทนา เพื่อการสร้างแชทบอท” ในครั้งนี้สำเร็จลุล่วงได้ด้วยดี เพราะความอนุเคราะห์จากบุคคลหลายๆ ฝ่าย ดังนี้ ขอขอบคุณอาจารย์ ดร.นิวรรณ วัฒนกิจรุ่งโรจน์ อาจารย์ประจำสาขาวิชาวิทยาศาสตร์การคำนวณ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำปรึกษา แนะนำ และความช่วยเหลือในทุกๆ เรื่องสำหรับการทำงาน ให้ข้อคิด แรงกระตุ้น และกำลังใจในการพัฒนางาน นอกจากนี้ท่านยังสละเวลาในการติดตามผลการทำโครงงานครั้งนี้มาโดยตลอด ขอขอบคุณอาจารย์ ผศ.ดร.จารุณี ดวงสุวรรณ และอาจารย์ผศ.ดร.วิภาดา เวทย์ประสิทธิ์ คณะกรรมการในการสอบโครงงานที่กรุณาทำการสอบและให้คำชี้แนะที่ดีในการทำโครงงาน ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยายลัยสงขลานครินทร์ ที่เอื้อเผื่อสถานที่และอุปกรณ์ในการทำโครงงาน และการสอบโครงงาน ทางคณะผู้จัดทำรู้สึกซาบซึ้งและขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

คณะผู้จัดทำ

สารบัญ

สารบัญรูป

สารบัญตาราง

ตาราง **หน้า**

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

การตอบข้อความของลูกค้าหรือผู้ที่เข้ามาสอบถามเป็นส่วนหนึ่งของงานที่มีเจ้าหน้าที่คอยให้บริการอยู่เสมอ แต่ในการตอบกลับเหล่านั้นยังมีข้อจำกัดในหลายๆ ด้าน เช่น ต้องรอเวลาทำการเจ้าหน้าที่จึงสามารถตอบได้ หรือต้องรอให้เจ้าหน้าที่ตรวจสอบว่ามีข้อความเข้าจึงจะได้รับคำตอบ ซึ่งทำให้การบริการหรือคำถามไม่ได้คำตอบในทันทีทันใด และอาจทำให้ลูกค้าหรือผู้ใช้เกิดความไม่พึงพอใจขึ้นได้อีกด้วย ในปัจจุบันจึงมีตัวช่วยที่เรียกกันว่าแชทบอทเข้ามาเป็นส่วนหนึ่งในการให้บริการทางด้านนี้โดยเฉพาะ

ปัจจุบันแชทบอทหรือโปรแกรมคอมพิวเตอร์ชนิดหนึ่งที่ใช้ในการตอบกลับการสนทนากำลังเป็นที่แพร่หลายในองค์กรหรือบริษัทห้างร้าน เนื่องจากความสามารถในการช่วยอำนวยความสะดวกให้แก่ลูกค้าและเจ้าหน้าที่ เช่น ช่วยให้การสอบถามข้อมูลและบริการสะดวกรวดเร็วทั้งในและนอกเวลาทำการ ลดปัญหาเจ้าหน้าที่ไม่เพียงพอต่อความต้องการของลูกค้า เป็นต้น ความท้าทายอย่างหนึ่งในการสร้างแชทบอท คือ การระบุเจตนาจากการสนทนาหรือเรียกว่า intent ซึ่ง intent หนึ่งๆ มีข้อความได้หลากหลาย เราอาจใช้วิธีให้ผู้พัฒนาแชทบอทระบุข้อความที่เป็นไปได้ สำหรับแต่ละ intent แต่อาจไม่ครบถ้วน และต้องใช้เวลานาน

การเรียนรู้ของเครื่อง (Machine Learning) เป็นเทคนิคที่สามารถนำมาใช้ในการจำแนกข้อมูลได้ ผู้จัดทำโครงงานจึงสนใจศึกษาการนำการเรียนรู้ของเครื่องมาช่วยใน **การจำแนก intent** เพื่อการสร้าง **แชทบอท** โดยจะใช้กรณีศึกษาจากชุดข้อมูลมาตรฐาน และบทสนทนาถามตอบเกี่ยวกับการศึกษาของนักศึกษาคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.2 วัตถุประสงค์

1) เพื่อศึกษาการจำแนก intent โดยใช้การเรียนรู้ของเครื่องสำหรับสร้างแชทบอท
2) เพื่อสร้างแชทบอทตอบคำถามข้อมูลเบื้องต้นของสถาบันการศึกษา ซึ่งจะช่วยตอบคำถามให้แก่ผู้ใช้นอกช่วงเวลาทำการ

1.3 ขอบเขตของโครงงาน

ศึกษาการจำแนก intent โดยใช้การเรียนรู้ของเครื่อง อย่างน้อย 2 เทคนิค บนชุดข้อมูลมาตรฐาน และกรณีศึกษาแชทบอทตอบคำถามข้อมูลเบื้องต้นเกี่ยวกับการศึกษา สำหรับนักศึกษาคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ โดยใช้ภาษา python เป็นหลัก

1.4 ขั้นตอนและระยะเวลาในการดำเนินงาน

1) คิดหัวข้อโครงงาน
2) ศึกษาความเป็นไปได้ของโครงงาน
3) กำหนดขอบเขต
4) ศึกษาเครื่องมือที่เกี่ยวข้อง
5) ศึกษาเทคนิคที่เกี่ยวข้อง
6) วางแผนการทดสอบ
7) ทำการทดลอง
8) สรุปและวิเคราะห์ผลการทดลอง
9) นำ intent ไปสร้างแชทบอท
10) ทดสอบการใช้แชทบอท
11) ประเมินผลการใช้แชทบอท

1.5 ระยะเวลาดำเนินการ

ตารางที่ 1.1: ระยะเวลา

ขั้นตอนการดำเนินงาน	ภาคการเรียน 2/2563						ภาคการเรียน 1/2564					
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค
1) คิดหัวข้อโครงงาน												
2) ศึกษาความเป็นไปได้ของโครงงาน												
3) กำหนดขอบเขต												
4) ศึกษาเครื่องมือที่เกี่ยวข้อง												
5) ศึกษาเทคนิคที่เกี่ยวข้อง												
6) วางแผนการทดสอบ												
7) ทำการทดลอง												
8) สรุปและวิเคราะห์ผลการทดลอง												
9) นำ intent ไปสร้างแชทบอท												
10) ทดสอบการใช้แชทบอท												
11) ประเมินผลการใช้แชทบอท												

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้วิธีการจำแนก intent เพื่อการสร้างแชทบอทที่มีประสิทธิภาพ

2) ได้ต้นแบบแชทบอท เพื่อการสนทนาเกี่ยวกับการเรียน สำหรับนักศึกษาคณะวิทยาศาสตร์

1.7 สถานที่และเครื่องมือที่ใช้ทำโครงงาน

สถานที่ คือ อาคารภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

เครื่องมือที่ใช้

- ฮาร์ดแวร์
  - o Processor: 4.5 GHz Intel Core i5
  - o Ram: 16 GB

- ซอฟต์แวร์

    - Python 3.9.5
    - ระบบปฏิบัติการ windows 10

1.8 อาจารย์ที่ปรึกษาโครงงาน

อาจารย์ ดร.นิวรรณ วัฒนกิจรุ่งโรจน์

1.9 ผู้จัดทำโครงงาน

นายณัฐพล เดชประมวลพล รหัสนักศึกษา 6110210129
นางสาววริศรา พิสุทธิ์เธียร รหัสนักศึกษา 6110210373

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 องค์ประกอบแชทบอท

แชทบอท (Chatbot) คือ แอปพลิเคชันซอฟแวร์ที่ใช้ในการสนทนาออนไลน์ผ่านข้อความหรือเสียง แทนการติดต่อโดยตรงกับมนุษย์ ซึ่งถูกออกแบบมาเพื่อตอบคำถามอัตโนมัติให้กับผู้ใช้อย่างเป็นธรรมชาติ โดยแชทบอทถูกออกแบบให้ใช้เป็นระบบ โต้ตอบ เพื่อวัตถุประสงค์ต่างๆ เช่น การรับออเดอร์จากลูกค้า, การสอบถามข้อมูล และการขอรับบริการ เป็นต้น ซึ่งแชทบอทจะประกอบด้วยส่วนหลักทั้งหมด 3 ส่วนที่มีความสำคัญ และจะทำงานร่วมกันเพื่อให้แชทบอทสามารถใช้งานได้อย่างมีประสิทธิภาพ โดยส่วนประกอบทั้งหมดมีดังนี้

2.1.1 Intent

คือส่วนที่แชทบอทจะดึงเจตนารมณ์ที่ซ่อนอยู่ภายในข้อความที่ได้รับจากผู้ใช้ หรือความต้องการที่ผู้ใช้ต้องการกล่าวถึง ซึ่งอาจตรวจจับจาก keyword และช่วงคำหลัก โดยจะคำนึงถึงความจำเพาะที่บอทถูกสร้างขึ้นมาใช้งาน

2.1.2 Response

คือ การตอบกลับข้อความ ซึ่งจะเกิดขึ้นเมื่อมีการระบุ intent จากข้อความที่ผู้ใช้ส่งเข้ามา โดยแชทบอทจะตอบด้วยคำตอบที่เหมาะสมที่สุด

2.1.3 Story (flow)

คือ การวางแผนการตัดสินใจ หรือแผนผังตรรกะของการสนทนาเส้นตรง โดยคำนึงถึงการสนทนาจริง เพื่อเป็นการต่อประโยคการสนทนาอย่างเป็นธรรมชาติ และหากผู้ใช้ถามคำถามอื่นขึ้นมาโดยคำถามนั้นไม่มีความเกี่ยวข้องกับหัวข้อที่คุยกันอยู่ในช่วงก่อนหน้า แชทบอทสามารถเปลี่ยนการสนทนาให้เป็นไปตาม intent ที่ผู้ใช้ถามเข้ามาได้

2.2 ประเภทของแชทบอท

โดยทั่วไปแชทบอทถูกนำไปใช้เพื่อให้บริการในการตอบโต้บทสนทนาผ่านแอปพลิเคชัน หรือบนหน้าเว็บของแชทบอทเอง และวิธีการในการโต้ตอบกับแชทบอทแตกต่างกันไปตามประเภท ซึ่งสามารถแบ่งประเภทของแชทบอทได้ตามวิธีการทำงานของแชทบอท ดังนี้

2.2.1 Rule-base Chatbot

คือ แชทบอทแบบมีข้อกำหนด นิยมแบบมีปุ่มตัวเลือกให้ผู้ใช้เลือกถามคำถาม เพื่อเป็นการกำหนดให้ผู้ใช้ทำตามกฎที่เขียนไว้ล่วงหน้า ซึ่งจะทำให้ได้รับคำตอบที่ผู้สร้างได้กำหนดไว้ และผู้สร้างต้องมีการสร้างกฎขึ้นหลายข้อ เพื่อครอบคลุมหลายกรณีที่ต้องการให้แชทบอทสามารถโต้ต้อบได้ และหากผู้ใช้ต้องการถามคำถามนอกเหนือจากที่มี แชทบอทจะไม่สามารถตอบคำถามเหล่านั้นได้ โดยการทำงานเช่นนี้ของแชทบอทจะไม่มีการใช้งานปัญญาประดิษฐ์ จึงมีข้อจำกัดในการใช้งานอยู่มาก [1]

2.2.2 Artificial Intelligence (AI) Chatbot

คือ แชทบอทที่มีการใช้งานปัญญาประดิษฐ์ เพื่อช่วยในการตอบคำถาม แต่ในกรณีนี้ยังมีความไม่ชัดเจนเรื่องแนวคิดเบื้องหลังของระบบคอมพิวเตอร์ที่ทำให้เกิดการ ‘คิดเหมือนมนุษย์’ จึงทำให้มีการใช้ตรรกะทางความคิด การวางแผน และการเข้าใจภาษาเข้ามาเกี่ยวข้องด้วย ซึ่งการเข้าใจภาษามนุษย์จะทำให้แชทบอทมีการตอบโต้ที่ดูเป็นธรรมชาติ อันเกิดจากการใช้อัลกอริทึม หรือ Neural Networks เข้ามาประมวลผลภาษาธรรมชาติ จึงเป็นที่มาของ Natural Language Processing หรือ NLP ซึ่งทำให้มีความยากมากกว่าแชทบอทประเภทแรก [1]

2.3 การสร้างเวกเตอร์แทนข้อความ

การสร้างเวกเตอร์แทนข้อความ เป็นการสร้างเวกเตอร์เพื่อใช้แทนข้อความต่างๆ ซึ่งจะพิจารณาว่าคำดังกล่าวปรากฏอยู่ในข้อความหรือไม่ และไม่คำนึงถึงลำดับของคำ สามารถสร้างได้หลายวิธี ในที่นี้จะกล่าวถึงวิธีการสร้างเวกเตอร์แทนข้อความ ด้วยวิธี 4 วิธีดังนี้

2.3.1 Binary vector

ซึ่งจะประกอบด้วย 0 หรือ 1 เพื่อแสดงว่าไม่มีหรือมีคำนั้นอยู่ในข้อความตามลำดับ [2] ตัวอย่างมีคำดังนี้

```
Word dog cat tiger elephant the black white is are and
Id    0   1    2      3      4    5     6    7  8   9
```

(ที่มา สืบค้นจาก: https://ichi.pro/th/kar-thaen-kha-laea-khxkhwam-ni-kar-pramwl-phl-phas-a-thrrmchati-56300952731182)

รูปที่ 2.1: แสดงตัวอย่างลำดับของคำที่กำหนดให้

เมื่อมีประโยคมาให้ และเขียนเปรียบเทียบเป็นเลข 0 หรือ 1 จะได้ว่า

```
the dog and cat are black → 1100110010
```

```
1    1    0      0      1    1     0    0  1   0
dog cat tiger elephant the black white is are and
```

(ที่มา สืบค้นจาก: https://ichi.pro/th/kar-thaen-kha-laea-khxkhwam-ni-kar-pramwl-phl-phas-a-thrrmchati-56300952731182)

รูปที่ 2.2: แสดงตัวอย่างของการแทนข้อความด้วยเวกเตอร์จากการค้นหาคำในข้อความ

เมื่อเขียนให้อยู่ในรูปของเวกเตอร์ จะได้ว่า $[1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1]^T$

2.3.2 Term Frequency (TF)

Term Frequency (TF) เป็นการหาเวกเตอร์ค่าน้ำหนักของความถี่คำ ซึ่งไม่จำเป็นต้องเป็นเลข 0 หรือ 1 หากมีจำนวนคำมากกว่า 1 ขึ้นไป [2] พิจารณาได้ 2 ลักษณะ

- ค่าน้ำหนักความถี่โดยตรง แทนด้วย $tf^{(org)}$
- ค่าน้ำหนักความถี่ที่ได้รับการทำให้เป็นมาตราฐานขน แทนด้วย tf

โดยพิจารณาตัวอย่างข้อความที่มีบางคำซ้ำกันบ้าง ไม่ซ้ำกันบ้าง

The sky is blue. The sky is beautiful

(ที่มา TF-IDF คำไหนสำคัญนะ สืบค้นจาก: https://lukkiddd.com/tf-idf-คำไหนสำคัญนะ-dd1e1568312e)

รูปที่ 2.3: ตัวอย่างข้อความซึ่งอยู่ใน document 1

ให้ $\text{tf}_{t,d}^{(org)}$ แทนจำนวนครั้งของการปรากฏของคำ $t(term)$ ในข้อความ $d(document)$

f("The", document1) → 2
f("sky", document1) → 2
f("is", document1) → 2
f("blue", document1) → 1
f("beautiful", document1) → 1

(ที่มา TF-IDF คำไหนสำคัญนะ สืบค้นจาก: https://lukkiddd.com/tf-idf-คำไหนสำคัญนะ-dd1e1568312e)

รูปที่ 2.4: ตัวอย่างความถี่คำซึ่งอยู่ใน document 1

ทำค่าให้อยู่เป็นช่วงค่ามาตราฐาน (Normalization) โดยใช้สูตร $1 + log(tf^{org})$ แต่หากไม่มีคำนั้นปรากฏอยู่ให้ค่าเป็น 0 ทันที

logTF("The", document1) → 1 + log(2) ≈ 1.3
logTF("sky", document1) → 1 + log(2) ≈ 1.3
logTF("is", document1) → 1 + log(2) ≈ 1.3
logTF("blue", document1) → 1 + log(1) = 1
logTF("beautiful", document1) → 1 + log(1) = 1

(ที่มา TF-IDF คำไหนสำคัญนะ สืบค้นจาก: https://lukkiddd.com/tf-idf-คำไหนสำคัญนะ-dd1e1568312e)

รูปที่ 2.5: การทำค่า $tf_{t,d}$ จาก document 1

จากการทำในภาพที่ 2.5 จะสามารถเขียนออกมาในรูปเวกเตอร์ได้ว่า $[1.3\ 1.3\ 1.3\ 1\ 1]^T$

2.3.3 Inverse Document Frequency (IDF)

เนื่องจากคำบางคำมีปรากฏอยู่ในข้อความเกือบทุกข้อความที่สนใจ ซึ่งหมายความได้ว่าคำเหล่านั้นแทบไม่มีนัยสำคัญต่อการค้นหา ในทางกลับกันคำที่ปรากฏอยู่เพียงไม่กี่ข้อความจะถือว่ามีความ

สำคัญมาก เมื่อมีการสอบถามและปรากฏคำที่สำคัญ จึงต้องสนใจคำที่สำคัญนั้นมากกว่าคำที่พบในเกือบทุกข้อความ

ให้ df_t แทน จำนวนข้อความที่มีคำ t ปรากฏอยู่ ค่าผกผันของ df_t นิยามด้วย idf_t และ N คือจำนวนข้อความที่มีอยู่ จะใช้สูตร

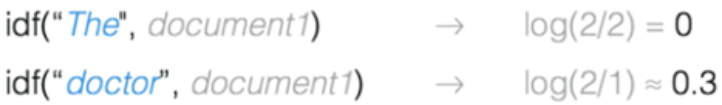
$$idf_t = log_{10} \frac{N}{df_t}$$

ในการหาค่า idft จากข้อความตัวอย่าง

(ที่มา TF-IDF คำไหนสำคัญนะ สืบค้นจาก: https://lukkiddd.com/tf-idf-คำไหนสำคัญนะ-dd1e1568312e)

รูปที่ 2.6: ตัวอย่างข้อความที่มีคำเหมือนและต่างกัน

หาค่า idf_t ได้จากสูตร โดยตัวอย่างจะสนใจคำว่า ‘the’ และคำว่า ‘doctor’ ที่มีปรากฏทั้งสองข้อความ และมีปรากฏใน document 1 เพียงอย่างเดียวตามลำดับ [2] ได้ว่า



(ที่มา TF-IDF คำไหนสำคัญนะ สืบค้นจาก: https://lukkiddd.com/tf-idf-คำไหนสำคัญนะ-dd1e1568312e)

รูปที่ 2.7: ตัวอย่างการหาค่า id$_f$ ของคำที่สนใจ

2.3.4 ค่าน้ำหนัก TF-IDF

เป็นปัจจัยที่ส่งผลต่อการคาดคะเนความเกี่ยวข้อง โดยพิจารณาจากค่า tf ซึ่งบ่งบอกว่าคำเหล่านั้น และข้อความมีความเกี่ยวข้องกันเพียงใด และ idf ซึ่งบ่งบอกว่าหากมีค่ามาก คำนั้นสำคัญต่อการค้นหาเป็นอย่างมากเช่นกัน โดยในการคำนวณ TF-IDF [2] จะหาจากสูตร

$$TF - IDF_{t,d} = tf \times idf_t$$

โดยนำตัวอย่างจากภาพที่ 2.6 หาค่า tf ของ the และ doctor ได้ว่า

ตารางที่ 2.1: ตารางค่า tf ของคำว่า the และ doctor ใน document 1 และ document 2

คำ	tf ใน document1	tf ใน document 2
the	1	1
doctor	1	0

ได้ค่า TF-IDF ของคำว่า the และ document จากสูตรดังนี้

ตารางที่ 2.2: ค่า TF-IDF ของคำว่า the และ doctor ใน document 1 และ document 2

คำ	TF-IDF ของ document 1	TF-IDF ของ document 2
the	0	0
doctor	0.3	0

2.4 ความคล้ายและความต่าง

เมื่อได้เวกเตอร์แทนข้อความแล้ว การตรวจสอบว่าข้อความที่สอบถาม กับข้อความที่มีอยู่เกี่ยวข้องกันมากน้อยเพียงใด พิจารณาได้ 2 แนวทาง ดังนี้

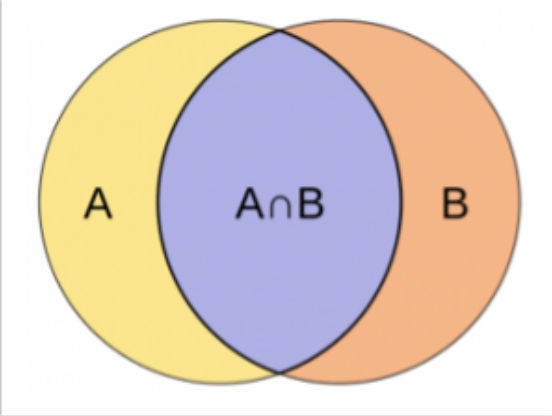
2.4.1 ความคล้ายกัน (similarity)

คือการหาความคล้ายกันของเวกเตอร์ ซึ่งพิจารณาได้จากทิศทางของเวกเตอร์ ซึ่งพุ่งออกจากจุดกำเนิด ไปยังพิกัดของเวกเตอร์นั้นๆ หากเวกเตอร์ใดมีทิศใกล้เคียงกัน หมายความได้ว่ามีคุณลักษณะในแต่ละมิติการกระจายตัวคล้ายกัน โดยขอกล่าวถึง 2 วิธี

1 Jaccard Similarity หรือ Intersect over union เป็นวิธีการหา similarity ที่ได้จากค่า A intersect B ซึ่งเป็นค่าที่มีความคล้ายคลึงกันมากที่สุด ซึ่งค่าสูงสุดของ Jaccard คือ 1 จะเกิดขึ้นเมื่อ A intersect B มีค่าเท่ากับ A union B [3] โดยคำนวณด้วยสูตร

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{A \cap B}{|A| + |B| - |A \cup B|}$$

เขียนเป็นรูปให้เข้าใจได้ง่ายว่า



(ที่มา Jaccard index สืบค้นจาก : https://en.wikipedia.org/wiki/Jaccard_index)

รูปที่ 2.8: ภาพวิธีการหาความคล้ายกันของ Jaccard similarity

2 Cosine Similarity คือการดูความคล้ายคลึงด้วยองศา ซึ่งคำนวณได้จากสูตร

$$cos\theta = \frac{\vec{q} \cdot \vec{d}}{\|\vec{q}\|\|\vec{d}\|} = SIM(\vec{q}, \vec{d})$$

เมื่อ $\vec{q}$ เป็นเวกเตอร์แทนข้อความการสอบถาม และ
$\vec{d}$ เป็นเวกเตอร์แทนข้อความใดๆ
θ เป็นมุมระหว่างเวกเตอร์ $\vec{q}$ และ $\vec{d}$
k เป็นจำนวนมิติเชิงเวกเตอร์ $\vec{q}$ และ $\vec{d}$

โดย $\|\vec{q}\| = \sqrt{\sum_{i=1}^{K} q_i^2}$

และ $\|\vec{d}\| = \sqrt{\sum_{i=1}^{K} d_i^2}$

เมื่อ $\vec{q}$ และ $\vec{d}$ มีความคล้ายกันมากที่สุดเมื่อ θ เข้าใกล้ 0 องศา หรือเข้าใกล้ 1 สามารถสมนัยสมการได้ว่า

$$SIM(\vec{q}, \vec{d}) = SIM(\vec{q}*, \vec{d}*) = \vec{q}* \cdot \vec{d}* = \sum_{i=1}^{K} q_i^* d_i^*$$

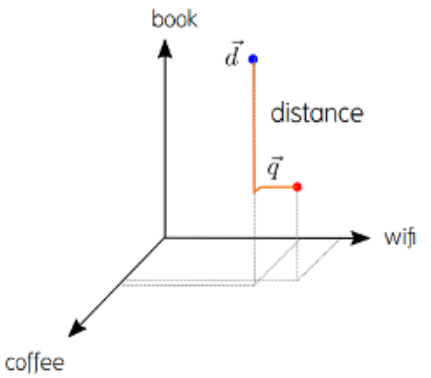
โดยที่ $\vec{q}^* = \frac{\vec{q}}{\|\vec{q}\|}$ และ $\vec{d}^* = \frac{\vec{d}}{\|\vec{d}\|}$ ซึ่ง $\vec{q}^*$ และ $\vec{d}^*$ คือเวกเตอร์มาตราฐานขนาด 1 หนึ่ง เมื่อค่าของ sim เข้าใกล้ 1 มากเท่าไรยิ่งหมายความว่ามีความคล้ายกันมากเท่านั้น [2]

2.4.2 ความต่างกัน (Distance)

เป็นการหาความต่างกันของเวกเตอร์ ซึ่งพิจารณาได้จากทิศทางของเวกเตอร์ ซึ่งพุ่งออกจากจุดกำเนิด ไปยังพิกัดของเวกเตอร์นั้นๆ หากเวกเตอร์ใดมีทิศต่างกัน หมายความได้ว่ามีคุณลักษณะในแต่ละมิติการกระจายตัวต่างกัน โดยขอกล่าวถึง 2 วิธี

1) Manhattan distance เป็นการหาระยะทางระหว่างเวกเตอร์หรือจุด ที่ได้จากผลบวกของระยะทางตาม แนวแกนในแต่ละมิติ ดังสมการ

$$dist_M(\vec{d},\vec{q}) = \sum_{i=1}^{K} |d_i - q_i|$$

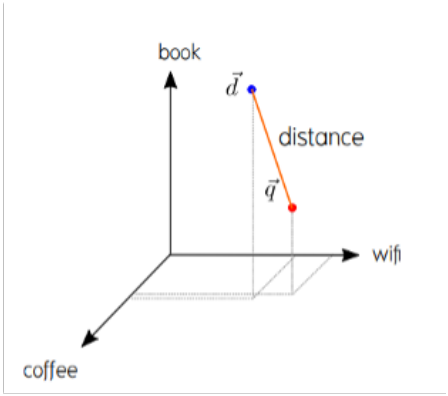


(ที่มา [2])

รูปที่ 2.9: การวัดระยะทางแบบ Manhattan distance

2) Euclidean distance เป็นการหาระยะทางระหว่างเวกเตอร์หรือจุด ซึ่งเป็นระยะแบบเส้นตรง (Straight-line) ดังสมการ

$$dist_E(\vec{d},\vec{q}) = \sqrt{\sum_{i=1}^{K}(d_i - q_i)^2}$$



(ที่มา [2])

รูปที่ 2.10: การวัดระยะทางแบบ Euclidean distance

โดยมีหลักการว่า หากค่าความต่างมีมากเท่าไร ข้อความสอบถามก็จะแตกต่างจากข้อความที่มีมากขึ้นเท่านั้น [2]

2.5 การใช้การเรียนรู้ของเครื่อง

ในการจัดกลุ่มว่าเราจะมี intent กี่กลุ่มนั้นสามารถใช้เทคนิคแบบไม่มีผู้สอน เพื่อจัดกลุ่ม intent เองได้ ซึ่งจะพูดถึงการเรียนรู้แบบไม่มีผู้สอน ได้แก่

2.5.1 การเรียนรู้แบบไม่มีผู้สอน

ในหัวข้อนี้จะกล่าวถึงการเรียนรู้แบบไม่มีผู้สอน 2 วิธี คือ K-Means และ DBSCAN

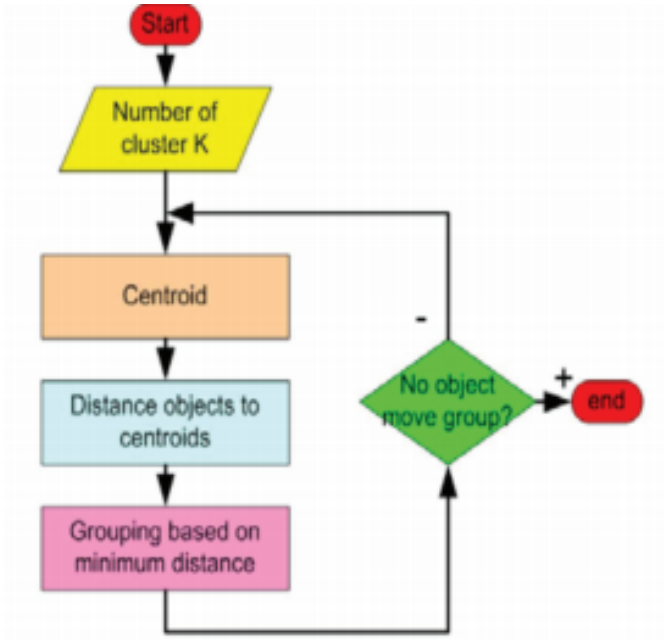
1) การแบ่งกลุ่มข้อมูลแบบเคมีน (**K-Means** clustering)
คือ วิธีการหนึ่งใน Data mining โดยหน้าที่หลักของ K-Means คือการแบ่งกลุ่มแบบ Cluster ซึ่งการแบ่งกลุ่มในลักษณะนี้จะต้องใช้ฟังก์ชั่นอย่างน้อย 2 ฟังก์ชั่นในการคำนวณระยะห่างระหว่างข้อมูลได้แก่ Euclidean distance metric และ Manhattan distance metric ซึ่งหน้าที่ของ Cluster คือการจับกลุ่มของข้อมูลที่มีลักษณะใกล้เคียงกันเป็นกลุ่มเดียวกัน ซึ่งในขั้นตอนการทำงานของ K-Means Clustering สามารถสรุปได้เป็น 4 ขั้นตอนดังนี้

(1) กำหนด Cluster (สามารถมีได้มากกว่า 1 ตัว)

(2) กำหนดหรือเคลื่อนที่ Centroid (ของพิกัดของ Cluster)

(3) คำนวณระยะห่างระว่างตำแหน่งของข้อมูลกับ Centroid

(4) จัดกลุ่มข้อมูลที่มีระยะห่างน้อยที่สุด

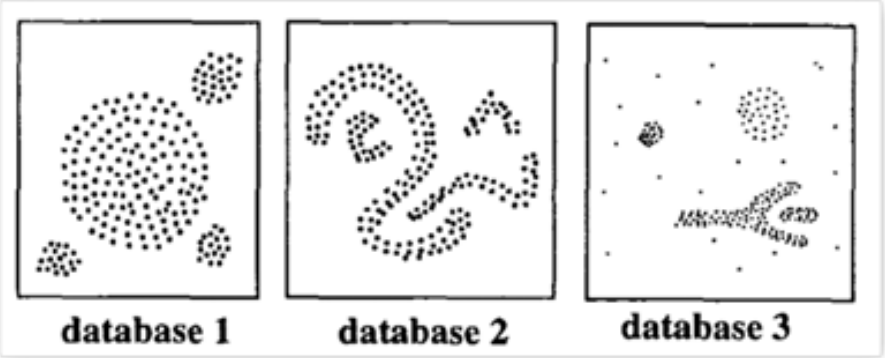


(ที่มา [4])

รูปที่ 2.11: ภาพอธิบายขั้นตอนของ K-Means Clustering

การจัดกลุ่มจะดูจากระยะห่างของข้อมูลกับ Centroid ว่าข้อมูลอยู่ใกล้ Centroid ใดมากกว่า ข้อมูลนั้นก็จะถูกจัดอยู่ใน Cluster นั้น ทำขั้นตอนที่ 2 – 4 วนซ้ำไปเรื่อยๆ จนกว่าค่า Centroid จะอยู่ในตำแหน่งจุดกึงกลางของข้อมูลใน Cluster [4]

2) การจัดกลุ่มของข้อมูล แบบ **DBSCAN** (Density-based spatial clustering of applications with noise) คือ การหาบริเวณข้อมูลที่อยู่รวมกันเป็นกลุ่มโดยจะหากลุ่มข้อมูลได้จากการคำนวณที่เกิดจาก Data point หรือจุดที่ข้อมูลแสดงอยู่



(ที่มา Ester et al. 1996 สืบค้นจาก : http://www.sthda.com/english/wiki/wiki.php?id_contents=7940)

รูปที่ 2.12: ภาพตัวอย่างชุดข้อมูลที่ใช้กับ DBSCAN

DBSCAN มักใช้กับชุดข้อมูลที่ไม่สามารถแบ่งแยกกลุ่มก้อนได้อย่างชัดเจน ไม่มี pattern ที่แน่นอน หรือมี outliner ในการทำงานของ DBSCAN นั้นจะใช้ 2 parameter เพื่อหากลุ่มข้อมูล [5] ได้แก่

(1) eps คือรัศมีจากจุดศูนย์กลางวงกลม

(2) MinPts คือจำนวน Data point ขั้นต่ำในการกำหนด center

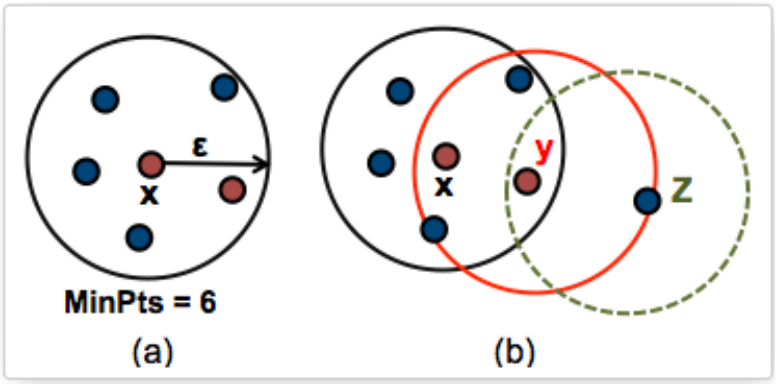
ซึ่งในขั้นตอนการทำงานของ DBSCAN สามารถสรุปเป็น 4 ขั้นตอน [5] ได้ดังนี้

(1) Data point ใดๆ ในชุดข้อมูลที่มีข้อมูลอื่นๆ อยู่รอบๆตัวมันในรัศมี eps มากกว่าหรือเท่ากับค่า MinPts จะถูกเรียกว่า core point

(2) Data point ใดๆ ในชุดข้อมูลที่มีข้อมูลอื่นๆ อยู่รอบๆ ตัวมันในรัศมี eps น้อยกว่าค่า MinPts แต่อย้ในรัศมี eps ของ core point จะเรียกว่า border point

(3) Data point ใดๆ ในชุดข้อมูลที่มีข้อมูลอื่นๆ อยู่รอบๆ ตัวมันในรัศมี eps น้อยกว่าค่า MinPts และไม่อยู่ในรัศมี eps ของ core point แต่อยู่ในรัศมี eps ของ border point จะเรียกว่า border point เช่นกัน

(4) Data point ใดๆ ที่มีอยู่ในขั้นตอนที่ 1-3 จะถูกเรียกว่า noise



(ที่มา DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning สืบค้นจาก : http://www.sthda.com/english/wiki/wiki.php?id_contents=7940)

รูปที่ 2.13: ตัวอย่าง DBSCAN

จากรูปที่ 2.13 สามารถบอกลักษณะของ data point ได้ว่า x คือ core point ส่วน y และ z คือ border point

2.5.2 การเรียนรู้แบบมีผู้สอน

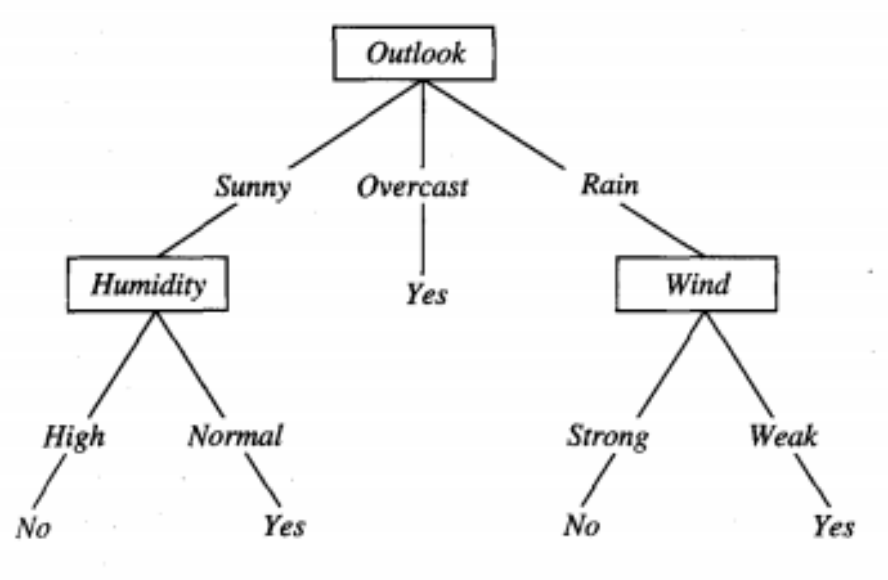
ในกรณีที่เราทราบแล้วว่าจะมี intent ใดบ้าง แล้วเราต้องการจำแนกข้อความที่เข้ามาว่าจัดอยู่ใน intent ใด สามารถทำได้ โดยใช้การเรียนรู้แบบมีผู้สอน ในที่นี้จะกล่าวถึงการเรียนรู้แบบมีผู้สอน 2 เทคนิค คือ Decision Tree และ Neural Network

1) ต้นไม้ตัดสินใจ (**Decision Tree**) ซึ่งเป็นวิธีการเรียนรู้ของเครื่องที่นิยมใช้เป็นอย่างมากรู้แบบหนึ่ง ใช้หรับหรือการจำแนก (Classification) ข้อมูลและคลาส (class) ต่างๆ โดยใช้คุณสมบัติ (attribute) ของข้อมูล ในการจำแนกข้อมูลจากคุณสมบัติของข้อมูลจะต้องดูว่า คุณสมบัติใดของข้อมูลที่ใช้ในการจำแนกคลาส และคุณสมบัตินั้นมีความสำคัญอย่างไร ซึ่งสามารถสรุปส่วนประกอบของ Decision tree ได้ 3 ส่วน [6] ดังนี้

   (1) โหนดภายใน (internal node) คือ คุณสมบัติต่างๆ ของข้อมูลที่ใช้ในการจำแนกว่าข้อมูลจะไปอยู่ในคลาสไหน โดยโหนดภายในที่เป็นโหนดเริ่มต้นเรียกว่า โหนดราก (root)

   (2) กิ่ง (branch, link) เป็นคุณสมบัติหรือเงื่อนไขของคุณสมบัติของโหนดที่ใช้ในการจำแนกข้อมูล ซึ่งโหนดภายในจะแตกกิ่งเท่ากับจำนวนคุณสมบัติของโหนดภายในนั้น

   (3) โหนดใบ (leaf node) คือคลาสต่างๆ ซึ่งเป็นผลลัพธ์ในการจำแนกข้อมูล



(ที่มา Machine Learning สืบค้นจาก : http://www.cs.ubbcluj.ro/ gabis/ml/ML-books/McGrawHill%20%20Machine%20Learning%20-Tom%20Mitchell.pdf)

รูปที่ 2.14: ตัวอย่าง Decision tree

โดยคุณสมบัติและลักษณะการเรียนรู้ของ Decision tree [6] สามารถจำแนกได้เป็น 5 ข้อ ดังนี้

(1) ผลของการเรียนรู้ของ Decision tree สามารถเข้าใจได้ง่ายเมื่อเทียบกับวิธีที่ใช้ในการจำแนกข้อมูลแบบอื่น

(2) เส้นทางทุกเส้นจากโหนดรากถึงโหนดใบ สามารถแสดงให้อยู่ในรูปของ IF-THEN ได้

(3) สามารถต้านทานข้อมูลรบกวน (noisy data) ได้

(4) มีความเร็วในการเรียนรู้สูง เมื่อเทียบกับวิธีการที่ใช้ในการจำแนกข้อมูลแบบอื่น

(5) เหมาะสำหรับการนำไปใช้ในการวิเคราะห์งานทางด้านธุรกิจ

2) โครงข่ายประสาท (**Neural Networks**) คือ แบบจำลองทางคณิตศาสตร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์ Neural Networks มีลักษณะของการส่งผ่านสัญญาณประสาทในสมองของมนุษย์ กล่าวคือ มีความสามารถในการรวบรวมความรู้ (knowledge) โดยผ่านการเรียนรู้ (learning process) และความรู้เหล่านั้นจะจัดเก็บอยู่ในโครงข่ายในรูปแบบค่าน้ำหนัก (weight) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้ใหม่ๆ เข้าไป

โหนด (node) เป็นการจำลองลักษณะการทำงานของเซลล์การส่งสัญญาณ (signal) ระหว่างโหนดที่เชื่อมต่อกัน (connection) จำลองมาจากการเชื่อมต่อกัน ภายในโหนดมีฟังก์ชันกำหนดสัญญาณส่งออกเรียกว่า ฟังก์ชันกระตุ้น (activation function) หรือฟังก์ชันการแปลง (transfer function) โดยในโครงสร้างของ Neural Networks ประกอบด้วย 5 องค์ประกอบ [7] ดังนี้

(1) ข้อมูลนำเข้า (input) คือ ข้อมูลที่เป็นตัวเลข หากเป็นข้อมูลเชิงคุณภาพ ต้องแปลงให้อยู่ในรูปเชิงปริมาณที่ Neural Network ยอมรับได้

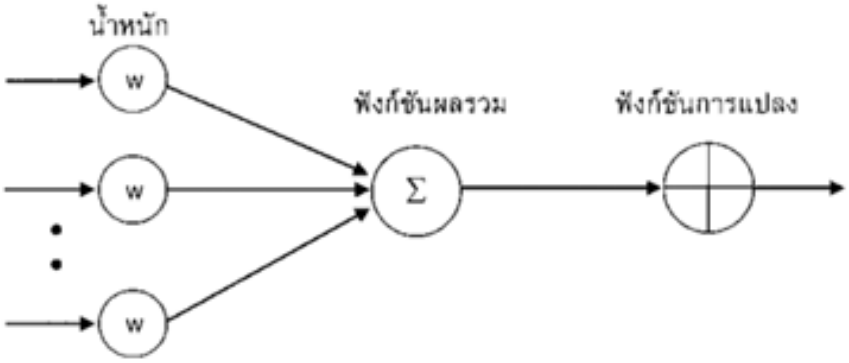
(2) ข้อมูลส่งออก (output) คือ ผลลัพธ์ที่เกิดขึ้นจริง (actual output) จากกระบวนการเรียนรู้ของ Neural Network

(3) ค่าน้ำหนัก (weight) คือ สิ่งที่ได้จากการเรียนรู้ของ Neural Network เรียกอีกอย่างว่า ค่าความรู้ (knowledge) ถูกเก็บเป็นทักษะที่ใช้ในการจดจำข้อมูลอื่นๆ ที่อยู่ในรูปแบบเดียวกัน

(4) ฟังก์ชันผลรวม (Summation function) เป็นผลรวมของข้อมูลป้อนเข้า (a_i) และค่าน้ำหนัก (w_i)

$$S = \sum_{i=1}^{n} a_i w_i$$

(5) ฟังก์ชันการแปลง (transfer function) เป็นการคำนวณการจำลองการทำงานของ Neural Network



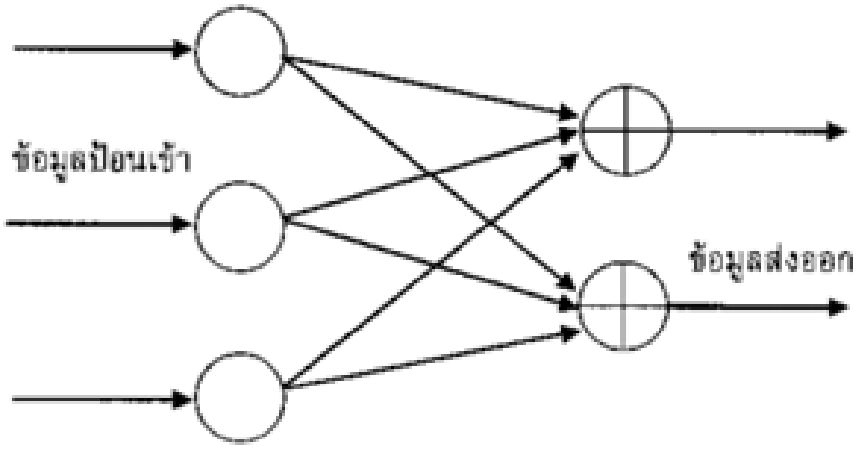
(ที่มา [7])

รูปที่ 2.15: ตัวอย่างของฟังก์ชันการแปลง

ลักษณะของ Neural Network จะประกอบไปด้วยโหนดจำนวนมากเชื่อมต่อกันกันกลุ่มย่อยเรียกว่า ชั้น (layer) ชั้นแรกเป็นชั้นข้อมูลเข้า เรียกว่า ขั้นรับข้อมูลป้อนเข้า (input layer) ส่วนชั้นสุดท้าย เรียกว่า ชั้นส่งข้อมูลส่งออก (output layer) และชั้นที่อยู่ระหว่างกลางของทั้ง 2 ชั้น เรียกว่า ชั้นแอบแฝง (hidden layer) ซึ่งโดยทั่วไปชั้นแอบแฝงอาจมีมากกว่า 1 ชั้นก็ได้ ทำให้สามารถจำแนกประเภทของ Neural Network ได้ 2 แบบ [7]

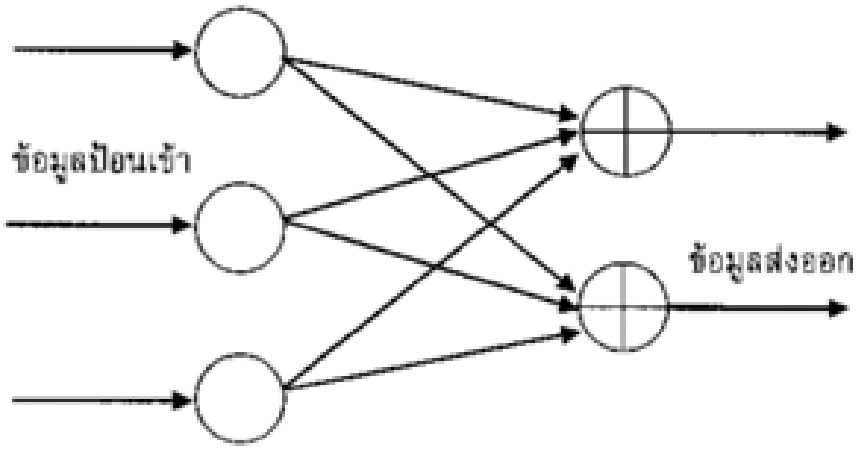
(1) Neural Network แบบชั้นเดียว มีเพียงชั้นรับข้อมูลป้อนเข้าและชั้นส่งข้อมูลออกเท่านั้น

(2) Neural Network แบบหลายชั้น ประกอบไปด้วยชั้นรับข้อมูลป้อนเข้า ชั้นส่งข้อมูลออกเท่านั้น และชั้นแอบแฝงตั้งแต่ 1 ชั้นขึ้นไป



(ที่มา [7])

รูปที่ 2.16: Neural Network แบบชั้นเดียว



(ที่มา [7])

รูปที่ 2.17: Neural Network แบบหลายชั้น

Neural Networks สามารถเรียนรู้ได้หลายประเภท ในที่นี้จะพูดถึงการเรียนรู้แบบมีผู้สอน (supervised learning) ข้อมูลสำหรับการสอนประกอบไปด้วยตัวอย่างข้อมูลที่ต้องการสอน และผลลัพธ์ที่ต้องการสอนให้ Network สร้างขึ้น เมื่อมีการนำข้อมูลในลักษณะเดียวกันมาเป็นข้อมูลป้อนเข้า Network จะกำหนดค่าผลลัพธ์ที่เป็นเป่าหมายให้กับข้อมูลป้อนเข้าแต่ละตัว Network จะนำค่าผิดพลาดระหว่างค่าเป้าหมายกับค่าผลลัพธ์ที่ได้ มาใช้ในการปรับค่าน้ำหนัก เพื่อให้ผลลัพธ์มีความใกล้เคียงกับเป้าหมายมากที่สุด [7]

2.6 เครื่องมือที่ใช้ในการสร้างแชทบอท

ในการสร้างแชทบอทโดยทั่วไปจะมี framework ให้เลือกใช้หลายตัว ขึ้นกับความสามารถของ framework แต่ละแบบว่าสามารถตอบโจทย์ความต้องการในการทำงานมากเพียงใด และบาง framework สามารถเชื่อมต่อไปยังแอปพลิเคชัน เพื่อเพิ่มความสะดวกให้แก่ผู้ใช้งาน แต่บาง framework ไม่สามารถทำได้ ซึ่งผู้จัดทำโครงงานได้สืบค้น framework ต่างๆ และขอยกตัวอย่างมาดังนี้

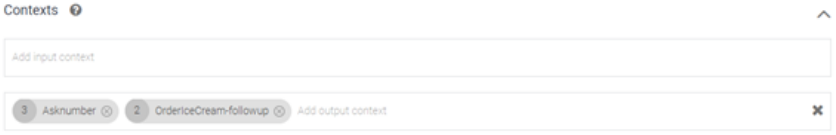
2.6.1 Dialogflow

Dialogflow คือ framework ที่สามารถสร้างแชทบอทได้หลายภาษารวมทั้งภาษาไทย โดยไม่ต้องมีการเขียนโค้ดใดๆ และมีฟังก์ชันการใช้งานที่หลากหลาย รวมทั้งสามารเชื่อมต่อไปยัง platform ได้หลากหลาย ซึ่งมีจุดเด่น คือการรองรับการทำ Natural Language Understanding โดยที่ไม่ต้องเขียนโปรแกรมอะไรเพิ่มเติม เพื่อช่วยในเรื่องการพิมพ์ผิดหรือเขียนมาไม่ตรงกับประโยคที่สอนไปเบื้องต้น แต่สามารถหาข้อมูลที่ถูกต้องได้ ซึ่งสามารถใช้งานผ่าน web browser ได้ โดยไม่ต้องติดตั้งโปรแกรมบนเครื่องคอมพิวเตอร์

การใช้งาน Dialogflow จะมีส่วนสำคัญคือการสอนคำถามให้กับแชทบอท ซึ่งจะมีประโยคอื่นๆ เพื่อสอนให้แชทบอททราบว่าคำเหล่านี้อยู่ใน intent ใด อันส่งผลให้แชทบอทเรียนรู้ และแยกประโยคเองได้

เมื่อมีการสอนประโยคให้สื่อถึง intent แล้ว ต่อมาจะต้องสอนให้แชทบอทรู้ว่าต้องตอบประโยคว่าอย่างไร โดยสามารถตอบกลับด้วยการเขียนข้อความหรือโค้ดก็ได้ และหากเชื่อมไปยังแอปพลิเคชันอื่น ผู้สร้างสามารถสร้างเป็นภาษาเฉพาะของแอปพลิเคชันได้ ซึ่งหากมีมากกว่า 1 คำตอบ ในส่วนเดียวกัน แชทบอทจะสุ่มเลือกขึ้นมาเอง และสามารถตั้งให้ถามคำถามเพื่อขอคำตอบเพิ่มเติมได้ โดยต้องมีการกำหนดว่าต้องการคำตอบชนิดใด

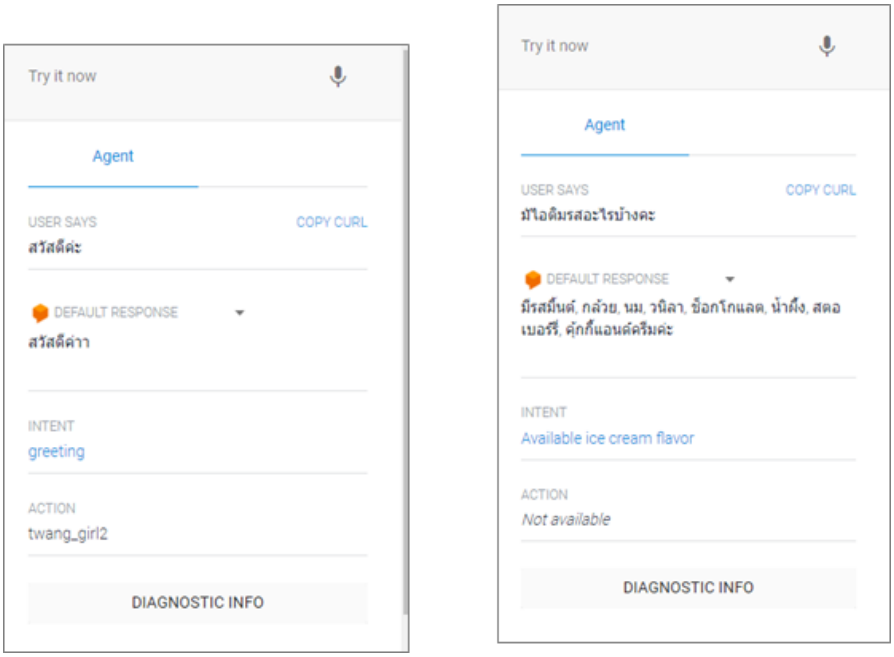
Dialogflow สามารถสร้าง flow ให้มีการพูดคุยในหัวข้อนั้นได้ ด้วยการส่งต่อหัวข้อไปยัง intent ที่เกี่ยวข้องต่อไป โดยจะมีตัวเลขเป็นการนับว่าจะยังคงอยู่ในหัวข้อดังกล่าวต่อไปอีกนานแค่ไหน



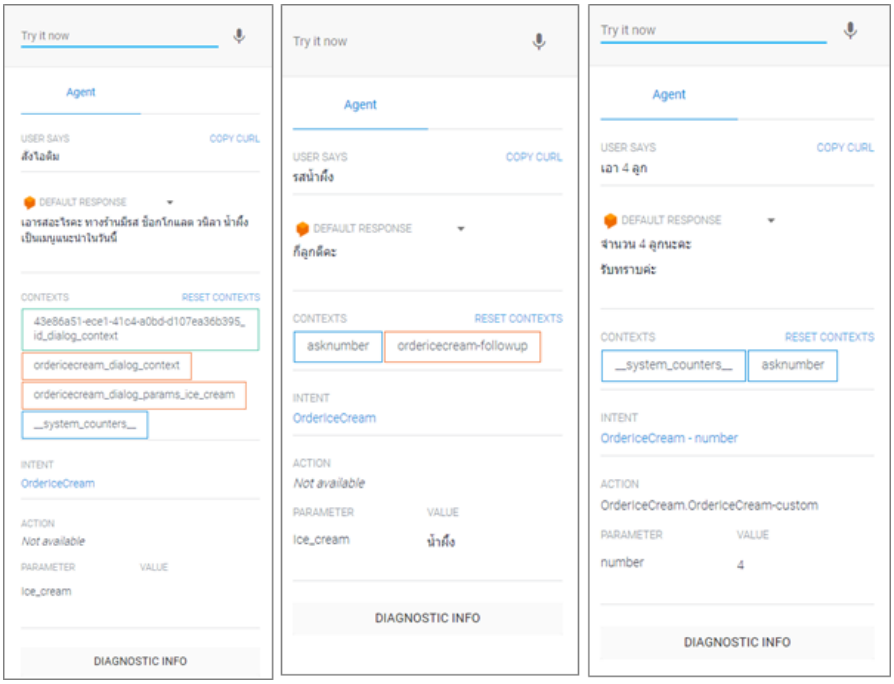
รูปที่ 2.18: หน้าจอ intent ส่วนของ contexts

ขั้นตอนในการทดสอบแชทบอท ก่อนนำไปใช้งานจริง ซึ่งในส่วนของ Dialogflow สามารถ

ทำการทดสอบได้ตลอดเวลาในฝั่งด้านขวาของหน้าเว็บ [8] จากการทดสอบจะได้ผลออกมาดังนี้



รูปที่ 2.19: หน้าจอแสดงผลการทดสอบแชทบอท Dialogflow (1)



รูปที่ 2.20: หน้าจอแสดงผลการทดสอบแชทบอท Dialogflow (2)

ข้อดีของ Dialogflow

(1) สามารถสร้างแชทบอท โดยไม่ต้องเขียนโค้ด

(2) ทำแชทบอทครั้งเดียว แต่สามารถใช้ได้กับหลาย platform เช่น Line, Facebook messenger และเว็บแชทของ Dialogflow เป็นต้น

(3) สามารถทำ pattern การสนทนาได้ คือ ต้องได้รับคำตอบครบทั้งหมดก่อน จึงสามารถประมวลผลได้ ยกตัวอย่างกรณีการคำนวณ BMI ที่ต้องได้รับค่าน้ำหนักและส่วนสูงครบก่อน จึงสามารถคำนวณค่า BMI ออกมาได้

(4) สามารถใช้งานได้ฟรี และไม่ต้องมี server เป็นของตัวเอง

(5) มีการรองรับ Natural Language understanding ทำให้ผู้ใช้เขียนข้อความนอกเหนือจากที่เคยสอนไปแชทบอทก็ยังสามารถเข้าใจได้

ข้อเสียของ Dialogflow

(1) อินเตอร์เฟสใช้งานยากสำหรับคนที่ไม่ใช่สายงาน developer

(2) ไม่มีการเก็บข้อมูลลงฐานข้อมูล

(3) ไม่สามารถดึงข้อมูลที่ผู้ใช้เขียนเข้ามาภายในแชทบอทออกมาได้

(4) เมื่อเขียนประโยคเพิ่มเข้าไปภายในแชทบอทผ่านหัวข้อ Training ด้านซ้าย ต้องจัดการแต่ละประโยคเองว่าเหมาะสมกับ intent ใด

(5) ในการเชื่อมต่อไปยังแอปพลิเคชันอื่นมีขั้นตอนเยอะเกินไปสำหรับคนใช้งานทั่วไป

แนวทางที่นำมาใช้กับงาน
ใช้งาน API ที่เชื่อมกับระบบอื่นๆ เช่น database เป็นต้น

2.6.2 chatterbot

chatterbot [9] คือ Library ภาษา Python ที่ใช้ทำแชทบอทแบบที่สามารถพิมพ์ผิดได้ เพราะ chatterbot มีการใช้ Machine Learning แบบ Supervised Learning ในการหาค่า Confidence เพื่อดูความคล้ายคลึงของประโยค และมี LogicAdapter คือวิธีที่จะใช้เลือกคำตอบที่อยู่ใน Data set นอกจากจะมีให้ใช้แล้ว ยังสามารถเขียนเพิ่มเข้าไปเองได้ ซึ่ง chatterbot นั้นต้องมีการเขียนข้อความ train ในรูปแบบ list

ข้อดีของ chatterbot

(1) เมื่อมีการพิมพ์ต่างจากที่สอนไป แชทบอทจะยังสามารถเข้าใจได้ด้วยการแยกคำ

(2) มีการจดจำบริบทก่อนหน้าไว้ในไฟล์ ทำให้มีความสนใจกับคำที่ถูกพูดถึงบ่อยๆ เป็นพิเศษ

ข้อเสียของ chatterbot

(1) มีคำตอบตามรูปแบบที่กำหนดเท่านั้น

(2) การสร้าง story ทำได้ยาก

```
trainer = ListTrainer(chatbot)

greeting = ["สวัสดีครับ", "สวัสดีค่ะ","หวัดดีครับ", "หวัดดีค่ะ", "ดีจ้า", "ทักครับ", "ทักค่ะ", "greeting"]

trainer.train(greeting)
```

รูปที่ 2.21: หน้าจอโปรแกรมเขียน train ให้กับ chatbot

```
(bot) C:\Chatterbot\bot>python chat.py
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Nut\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Nut\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Nut\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
You : สวัสดีครับ
Bot : สวัสดีค่ะ
You : ทักครับ
Bot : ทักค่ะ
You : ชื่ออะไรครับ
Bot : ไม่เข้าใจเลย
You :
```

รูปที่ 2.22: หน้าจอผลลัพธ์จากการรันโปรแกรมที่เขียนด้วย chatterbot

2.6.3 flow.ai

flow/ai คือ โปรแกรมสำหรับสร้างแชทบอท [10] ที่สามารถตอบกลับด้วยข้อความ เสียง วิดิโอและฯลฯ ซึ่งตัวโปรแกรมนี้ถูกออกแบบมาเพื่อการทำงานเป็นทีมแบบวันต่อวันผ่าน web browser และมีการรองรับภาษาที่หลากหลาย ซึ่งโปรแกรม flow.ai จะเน้นไปที่ flow ในการพูดคุยเป็นหลัก จึงทำให้หน้าต่างหลักจะแสดง flow ของข้อความเพื่อให้เข้าใจได้ง่าย และตรงจุดประสงค์หลักของการใช้งาน โดยในการสร้างแชทบอทจะให้เลือกว่าจะเชื่อมต่อไปยังแอปพลิเคชันใด และใช้งานในด้านใดเป็นหลัก ซึ่งจะส่งผลต่อข้อมูลต่อมาที่ระบบต้องการรับ และหน้าต่างแสดงผลเมื่อมีการรัน

ข้อดีของ flow.ai

(1) มีการเรียก api ไปบริการบนเว็บ

(2) สามารถสร้างแชทบอทโดยไม่ต้องเขียนโค้ด

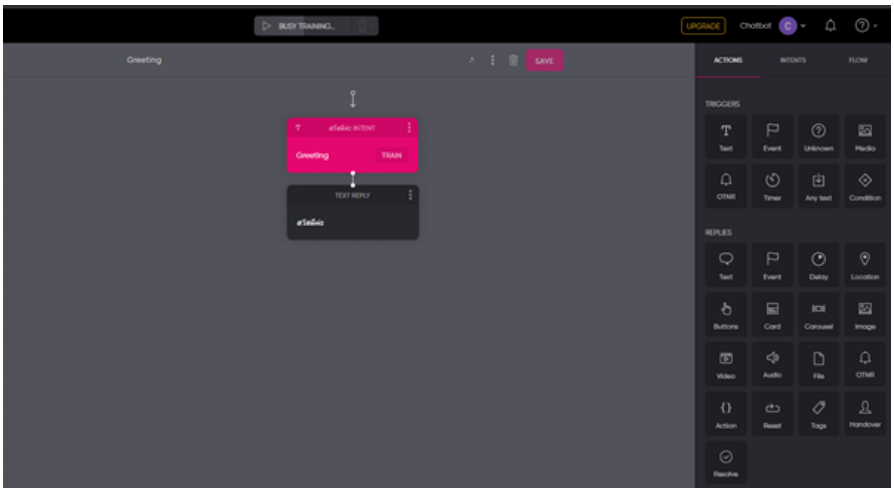
(3) สามารถใช้งานได้ฟรี และไม่ต้องมี server เป็นของตัวเอง

(4) มีการเข้าใจภาษาด้วย NLP ทำให้เมื่อเขียนข้อความที่ไม่ได้สอนก็ยังสามารถหาได้

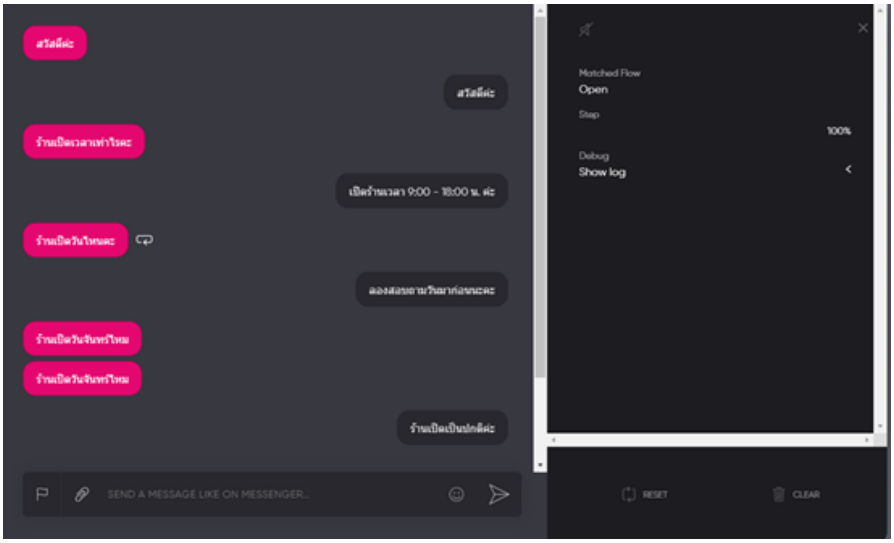
ข้อเสียของ flow.ai

(1) ไม่เหมาะกับบุคคลทั่วไปที่ไม่ใช่ developer

(2) ไม่มี entity เบื้องต้นมาให้ จึงต้องทำเองทั้งหมด



รูปที่ 2.23: หน้าจอหลักของ flow.ai



รูปที่ 2.24: หน้าจอแสดงผลการรันของ flow.ai

บทที่ 3

การวิเคราะห์และขั้นตอนวิธี

ในการสร้างแชทบอทนั้น เราจะต้องมีการเตรียมชุดข้อมูลที่ประกอบด้วยข้อความ (message) และการระบุเจตนา (intent มาจากคำว่า intention) โดยทั่วไปการระบุ intent ให้กับข้อความจำเป็นจะต้องใช้มนุษย์เป็นผู้พิจารณาว่าข้อความนั้นๆ ตรงกับ intent ใด ซึ่งหากมีข้อความจำนวนมาก จะทำให้ต้องใช้เวลานาน การจัดกลุ่มข้อความที่มีความคล้ายกันให้อยู่ในกลุ่มเดียวกันแล้วระบุ intent ในคราวเดียวกันจะทำให้การเตรียมข้อมูลมีความรวดเร็วมากขึ้น นอกจากนั้นแชทบอทที่มีการสร้างขึ้นเมื่อนำไปใช้งานสักระยะจะมีข้อความใหม่ที่ควรนำมาสอนแชทบอทเพิ่มเติม การระบุ intent ให้กับข้อความใหม่นี้หากมีจำนวนมาก และใช้มนุษย์ทำก็จะสิ้นเปลืองเวลาเช่นกัน จึงต้องมีโมเดลสำหรับจำแนกข้อความที่เกิดขึ้นใหม่เหล่านี้ว่าเป็น intent ใด

คณะผู้จัดทำโครงงานจึงได้วิเคราะห์และออกแบบขั้นตอนในการพัฒนางาน โดยประกอบไปด้วย 4 หัวข้อหลัก ดังนี้

- การจัดกลุ่มข้อความเพื่อกำหนด intent
- การสร้างโมเดล และทดสอบโมเดลในการจำแนก intent
- ชุดข้อมูลที่ใช้ทำการทดสอบ และตัวชี้วัดประสิทธิภาพ
- การประยุกต์ใช้ เพื่อพัฒนาต้นแบบแชทบอท และการอัพเดตแชทบอท

โดยรายละเอียดในการดำเนินการแต่ละหัวข้อ จะกล่าวดังต่อไปนี้

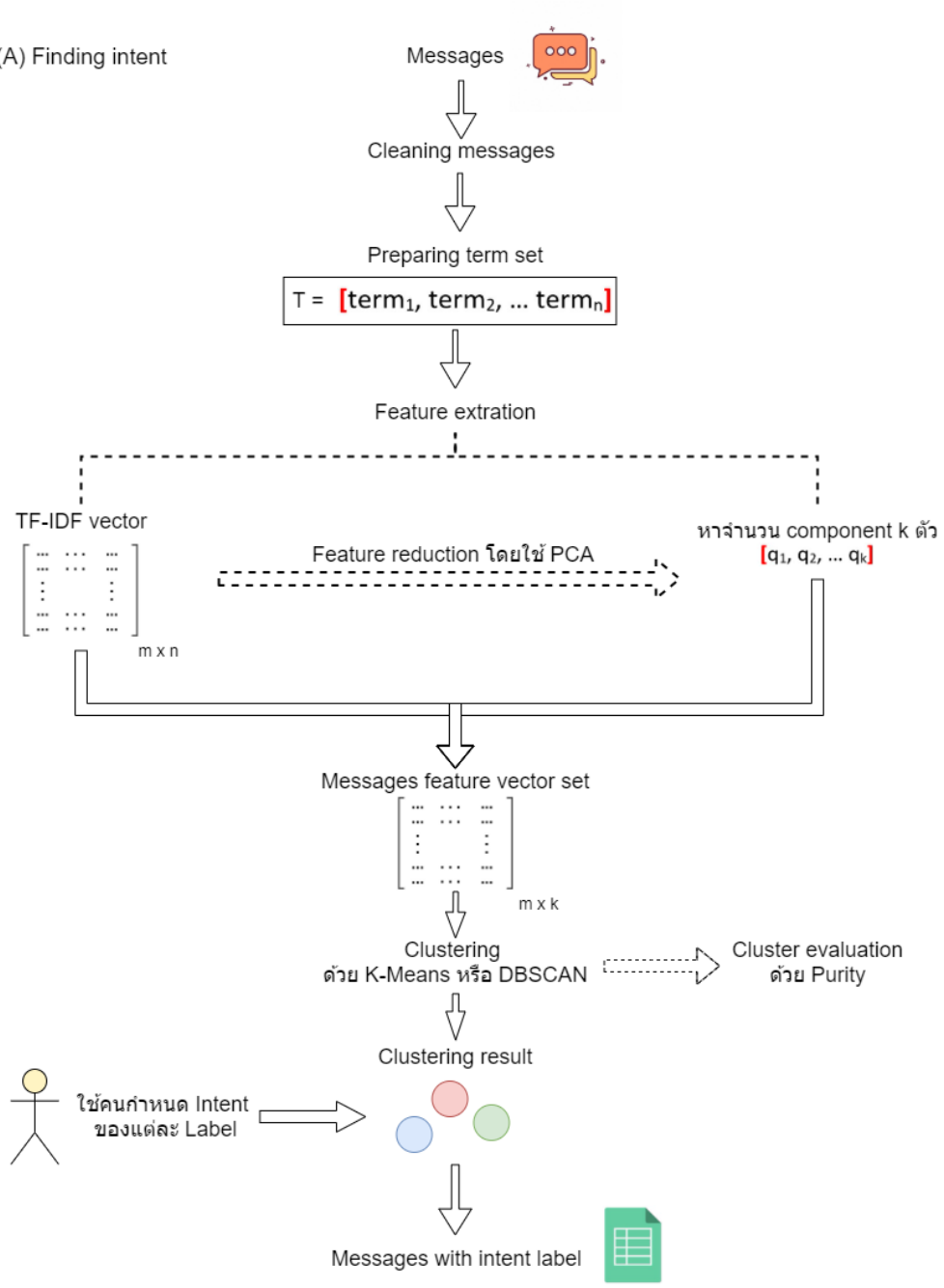
3.1 การจัดกลุ่มข้อความเพื่อกำหนด intent

ในขั้นตอนแรกจะเป็นการจัดกลุ่มของข้อความ โดยใช้การเรียนรู้ของเครื่อง แบบไม่มีผู้สอน เมื่อสามารถแบ่งกลุ่มเรียบร้อยแล้ว จะให้มนุษย์มาระบุ intent ให้กับข้อความแต่ละกลุ่ม ข้อความทั้งหมดที่อยู่ในกลุ่มนั้นๆ จะมี intent เดียวกัน โดยจะมีลำดับขั้นตอนดังภาพที่ 3.1 ประกอบด้วย

- การทำความสะอาดข้อความ
- การเตรียมชุดคำศัพท์

- การสกัดคุณลักษณะ
- การจัดกลุ่ม
- การกำหนด intent label

ดังรายละเอียดที่จะกล่าวต่อไปนี้



รูปที่ 3.1: ขั้นตอนการทำงาน A การจัดกลุ่มข้อความเพื่อกำหนด intent

3.1.1 การทำความสะอาดข้อความ (Cleaning messages)

ในขั้นตอนนี้จะเป็นการเตรียมข้อมูลให้เหมาะสมสำหรับการนำไปวิเคราะห์ [2] มีขั้นตอนดังนี้

1) ปรับอักขระให้อยู่ในรูปแบบเดียวกัน คือ พิมพ์เล็ก ดังตัวอย่างต่อไปนี้

Where is the Songkhla city? -> where is the songkhla city?

2) ลบเครื่องหมาย หรืออักขระพิเศษ และตัวเลข ด้วยการแทนค่าเหล่านั้นด้วยช่องว่าง ถ้าช่องว่างติดกันมากกว่า 1 ช่อง ให้ทำการลบออกจนเหลือเพียง 1 ช่อง ดังตัวอย่างต่อไปนี้

hi! how are you? -> hi how are you i am 22 year old -> i am 22 year old

3) ตัดแบ่งแต่ละคำด้วยช่องว่าง ดังตัวอย่างต่อไปนี้

hi how are you -> “hi”, “how”, “are”, “you”

4) ลบ stop word เนื่องจาก stop word เป็นคำที่ปรากฏในข้อความบ่อยครั้ง ซึ่งมีผลน้อยมากในการแบ่งกลุ่ม หรือจำแนกข้อความ เช่น a, an, do, to, or และ is ดังตัวอย่างต่อไปนี้

“hi”, “how”, “are”, “you” -> “hi”

5) เปลี่ยนรูปคำ/ลดรูปคำ (Stemming) ที่มีความหมายเหมือนกันให้อยู่ในรูปของรากศัพท์ ดังตัวอย่างต่อไปนี้

is, am, are -> be
loads, loaded, loading -> load

3.1.2 การสร้างชุดคำศัพท์ (Preparing term set)

ให้ m แทน จำนวนข้อความที่พิจารณา และ

$M_1, M_2, ..., M_m$ แทน เซตของคำศัพท์ ในข้อความที่ $1, 2, ..., m$ ที่ได้ผ่านการทำความสะอาดตามกระบวนการในข้อที่ 3.1.1 มาแล้ว

คำศัพท์ทั้งหมดจะถูกนำมายูเนียนกัน เพื่อสร้างเป็นชุดคำศัพท์ หรือดิกชันนารี (Dictionary) จะได้ว่า

$$T = M_1 \cup M_2 \cup ... \cup M_m = [term_1, term_2, ..., term_n]$$

เมื่อ n คือ จำนวนคำศัพท์ที่แตกต่างกัน ชุดคำศัพท์ที่ได้จะถูกนำไปใช้เป็นตัวกำหนดจำนวนมิติในการสกัดคุณลักษณะต่อไป

3.1.3 การสกัดคุณลักษณะของข้อมูล (Feature extraction)

ข้อความที่ทำความสะอาดแล้วตามหัวที่ 3.1.1 และสร้างชุดคำศัพท์ที่ได้ตามหัวข้อ 3.1.2 จะถูกนำมาใช้ในการสกัดคุณลักษณะข้อความให้ได้เป็นเวกเตอร์ โดยในโครงงานนี้ จะทำการแทนข้อความด้วยเวกเตอร์ด้วย TF-IDF ดังวิธีการที่ได้กล่าวไว้ในหัวข้อที่ 2.3 จากนั้นทำการลดจำนวนมิติของเวกเตอร์ลง (Feature reduction) โดยใช้หลักการ Principal Components Analysis (PCA) ซึ่งเราจะได้เวกเตอร์ component และค่า variance ออกมาทั้งหมดเท่ากับจำนวนมิติที่มี และ component ที่มีความสำคัญจะมีค่า variance สูง [11] ทำให้ในการลดจำนวนมิติ เราจึงเก็บ component ที่มีค่า variance สูง เพื่อไม่ให้ส่วนที่มีความสำคัญหายไปจากการลดมิติ และทำการลดมิติ component เมื่อค่า variance ต่ำ อย่างไรก็ตามการลดมิติมากไปก็ไม่ดี ในโครงงานนี้จึงใช้หลักการ ดังนี้

1) หาค่าเปอร์เซนต์ของ variance เมื่อเทียบกับผลรวมของ variance ทั้งหมด

$$Vp_i = \frac{V_i}{\sum_{j=1}^{n} V_j} * 100 i = 1, 2, ..., n$$

โดย Vp_i คือ ค่าความแปรปรวนของ component ที่ i โดยที่ $V_1 \geq V_2 \geq ... \geq V_n$ หน่วยเป็นเปอร์เซนต์

V_i คือ ค่าความแปรปรวนของ component ที่ i

$\sum_{j=1}^{n} V_j$ คือ ค่าความแปรปรวนรวมทั้งหมด ตั้งแต่ component ที่ 1 จนถึง n

2) นำค่า Vp_iของ i ตั้งแต่ 1 จนถึง n ที่ได้มาคำนวณหาค่า component ที่ดีที่สุด ด้วยสมการดังต่อไปนี้

$$sumV_{pi} = \sum_{i=1}^{k} V_{pi}$$

โดยที่จำนวน component ที่ดีที่สุดจะได้จากจำนวนผลรวม k ตัวแรก ของ V_{pi} ที่ทำให้ $sumV_{pi} \geq 80\%$ ซึ่งแทนว่ามีจำนวน component ที่ดีที่สุด k ตัว เขียนได้ว่า $[q_1, q_2, ..., q_k]$ ซึ่งจะใช้ในการลดมิติของข้อมูล ผลลัพธ์สุดท้ายจะได้ออกมาเป็นเวกเตอร์ข้อความตามขนาดของมิติ คือ ขนาด k

3.1.4 การแบ่งกลุ่ม (Clustering)

ข้อความที่จะนำมาสร้างแชทบอทจะถูกนำมาจัดกลุ่ม เพื่อให้การระบุ intent label เป็นไปอย่างรวดเร็ว และมีประสิทธิภาพ ในโครงงานนี้ทำการทดลองจัดกลุ่มข้อความด้วย 2 วิธี ได้แก่

1) K-Means เป็นการจัดกลุ่มข้อมูลแบบ center-based ดังหลักการที่ได้กล่าวไว้ในหัวข้อ 2.5.1 หัวข้อที่ 1) โดยจะต้องมีการกำหนดพารามิเตอร์ k ซึ่งแสดงถึงจำนวนกลุ่มที่ต้องการจัดข้อมูล ในการทำโครงงานนี้ทางคณะผู้จัดทำจะหาค่า k ที่ดีที่สุด [12] เพื่อนำมาแบ่งกลุ่มข้อมูล โดยไม่ต้องใช้มนุษย์เข้ามาช่วย ด้วยหลักการ The Silhouette Method

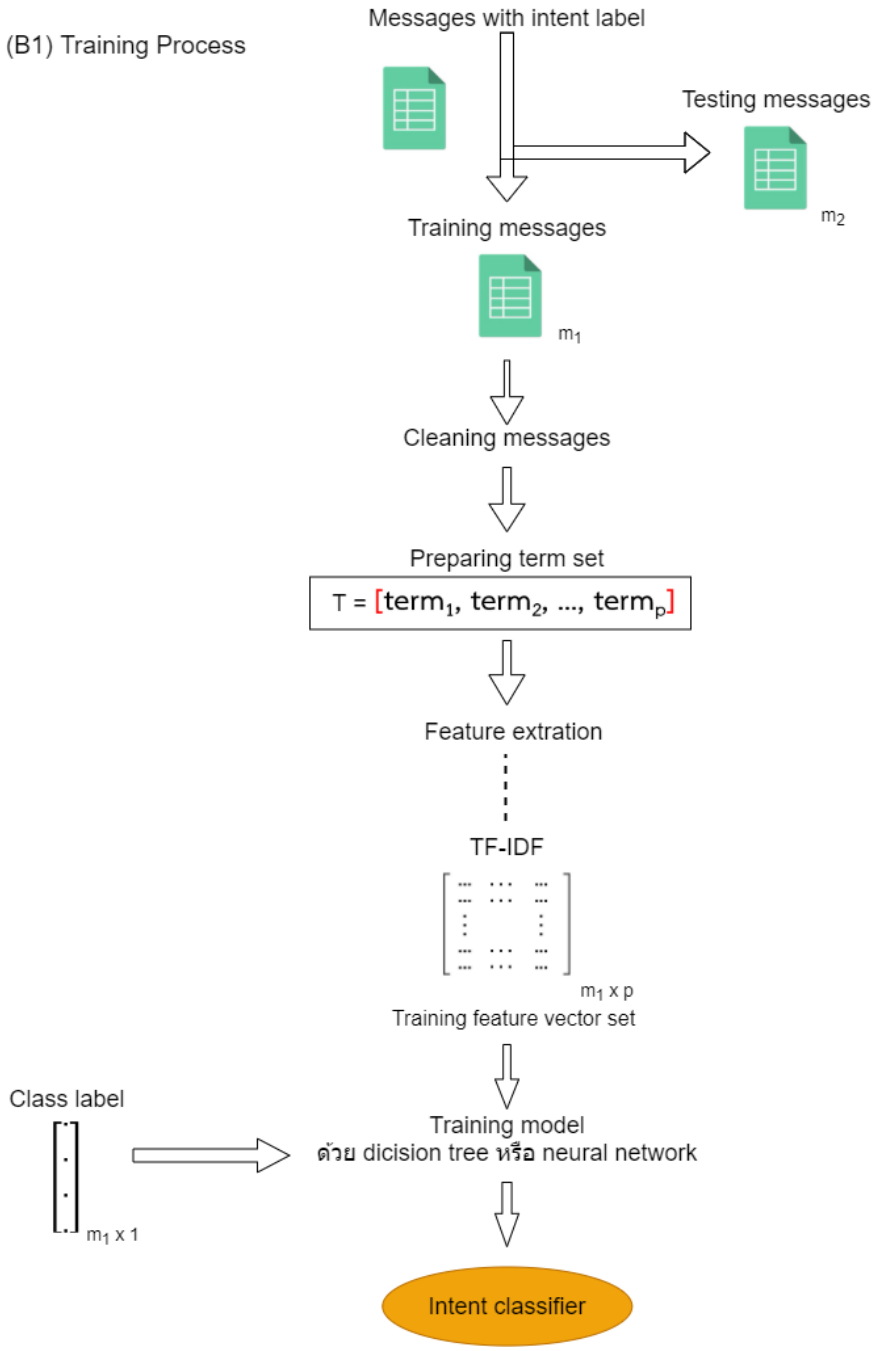
2) DBSCAN เป็นการจัดกลุ่มแบบ density-based ดังที่ได้กล่าวไว้ในหัวข้อที่ 2.5.1 หัวข้อที่ 2) พารามิเตอร์ที่ต้องใช้ในการจัดกลุ่ม คือ Eps ซึ่งแสดงถึงรัศมีของบริเวณพื้นที่ และ MinPts ซึ่งแสดงถึงจำนวนจุดข้อมูลที่อยู่ในบริเวณที่พิจารณา

ผลจากการจัดกลุ่มทั้ง 2 วิธี จะถูกนำมาประเมินผลเปรียบเทียบประสิทธิภาพต่อไปในหัวข้อที่ 3.9 เพื่อเลือกเทคนิคการจัดกลุ่มที่เหมาะสมไปใช้จัดกลุ่มข้อความ ก่อนที่จะนำไปพิจารณาระบุ intent ของแต่ละกลุ่ม ซึ่งข้อความที่อยู่ในกลุ่มเดียวกันจะได้รับการระบุว่าเป็น intent เดียวกันทั้งกลุ่ม จากนั้นข้อความที่มีการระบุ intent แล้วจะถูกนำไปใช้ในขั้นตอนต่อไป

3.2 การสร้างโมเดลและทดสอบโมเดลในการจำแนก intent

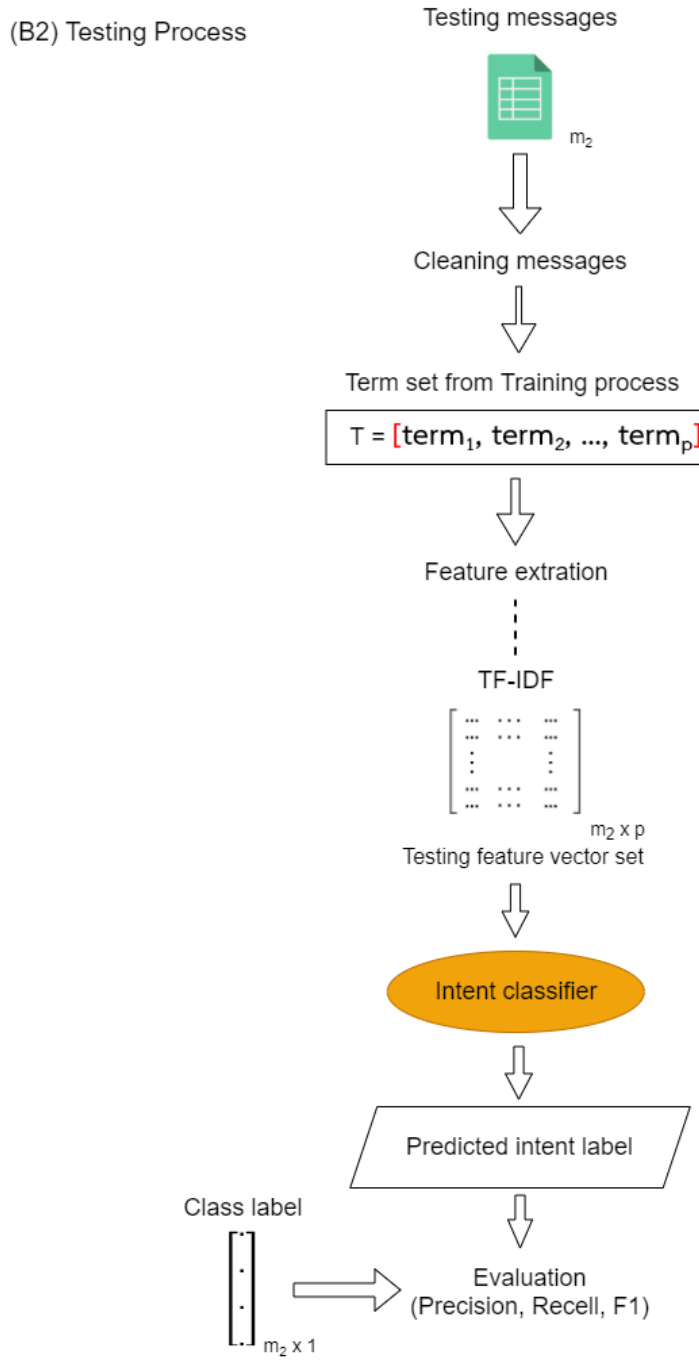
ในขั้นตอนนี้เป็นการนำชุดข้อความที่ผ่านการแบ่งกลุ่ม และระบุ intent เรียบร้อยแล้ว มาสร้างโมเดลสำหรับทำนาย (Predict) intent ให้กับข้อความ โดยใช้การเรียนรู้ของเครื่องแบบมีผู้สอน ข้อมูลทั้งหมดจะถูกแบ่งเป็น 2 ส่วน คือ ชุดข้อมูลสอน (Training Data) และชุดข้อมูลทดสอบ (Testing Data) โดยจะทำการทดลองด้วยวิธีการแบ่งข้อมูลเป็นส่วน ข้อความ และ intent ของแต่ละข้อความ โดยจะนำข้อมูลสอนเข้าสู่กระบวนการเรียนรู้ (Training Process) จนได้เป็นโมเดล โดยจะมีลำดับขั้นตอนดังภาพที่ 3.2 ดังรายละเอียดที่จะกล่าวต่อไปนี้

เริ่มต้นจะต้องมีการนำข้อความในชุดข้อมูลสอน มาทำความสะอาดตามหลักการที่ได้กล่าวไว้ก่อนหน้านี้ในหัวข้อที่ 3.1.1 แล้วทำการสร้างชุดคำศัพท์ตามวิธีในหัวข้อที่ 3.1.2 ข้อความที่ผ่านการทำความสะอาด และ intent label จะใช้เป็นข้อมูลนำเข้าสู่กระบวนการเรียนรู้ เพื่อสร้างโมเดล ในโครงงานนี้จะทำการทดลองสร้างโมเดล โดยใช้ 2 เทคนิค คือ Decision Tree และ Neural Networks ทำให้โมเดลที่ได้จะถูกนำไปใช้ในกระบวนการทดสอบต่อไป สำหรับกระบวนการทดสอบโมเดล จะนำข้อความจากชุดทดสอบที่ถูกแบ่งไว้ก่อนหน้านี้ มาทำนาย intent โดยใช้โมเดลที่สร้างขึ้น โดยจะมีลำดับขั้นตอนดังภาพที่ 3.3



รูปที่ 3.2: ขั้นตอนการทำงาน B1 การสร้างโมเดล

เริ่มต้นจากการทำความสะอาดข้อความ แล้วทำชุดคำศัพท์ที่ได้จากกระบวนการสอนมาใช้ในการสกัดคุณลักษณะให้เป็นเวกเตอร์ TF-IDF เมื่อนำเข้าสู่โมเดลทำนายว่าเป็น intent ใด จากนั้นเราจะนำผลการทำนายไปเปรียบเทียบประสิทธิภาพ โดยใช้ตัวชี้วัดที่กล่าวไว้ในหัวข้อ 3.3.3 ต่อไป



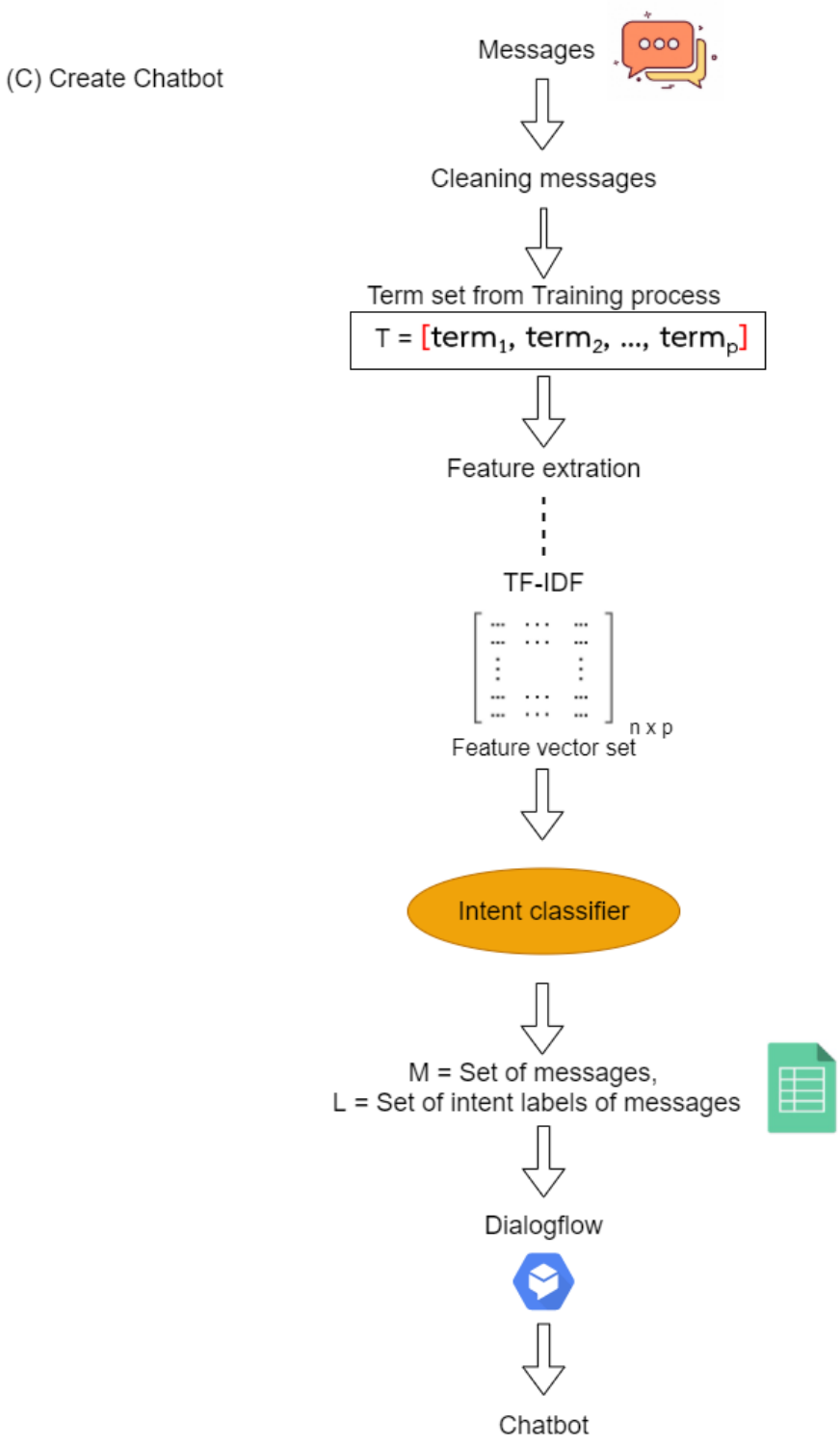
รูปที่ 3.3: ขั้นตอนการทำงาน B2 ทดสอบโมเดลในการจำแนก intent

3.3 การประยุกต์ใช้เพื่อพัฒนาต้นแบบแชทบอทและการอัพเดตแชทบอท

ในหัวข้อนี้จะกล่าวถึงการประยุกต์ใช้ 2 ส่วน คือ การสร้างแชทบอท และการอัพเดต แชทบอท ดังรายละเอียดที่จะกล่าวต่อไปนี้

3.3.1 การสร้างแชทบอท

ในขั้นตอนนี้เป็นการนำชุดข้อความที่ยังไม่มีการระบุ intent ที่เตรียมไว้สำหรับสร้าง แชทบอท มาระบุ intent ของข้อความโดยใช้โมเดลที่สร้างขึ้นจากขั้นตอนก่อนหน้า โดยมีการทำความสะอาดข้อความ และการสกัดคุณลักษณะ ซึ่งเป็นหลักการเดียวกันกับตอนสร้างโมเดล หลังจากการนำข้อความเข้าสู่โมเดล ค่าผลลัพธ์ที่ได้จะปรากฏในรูป เซตข้อความ แทนด้วย M และเซต intent ของข้อความ แทนด้วย L ซึ่งจะนำเข้าสู่ Dialogflow โดยจะมีการกำหนดคำตอบที่เหมาะสมตาม intent เพื่อเป็นการสอนแชทบอท โดยจะมีลำดับขั้นตอนดังภาพที่ 3.4



รูปที่ 3.4: ขั้นตอนการทำงาน C พัฒนาต้นแบบแชทบอท

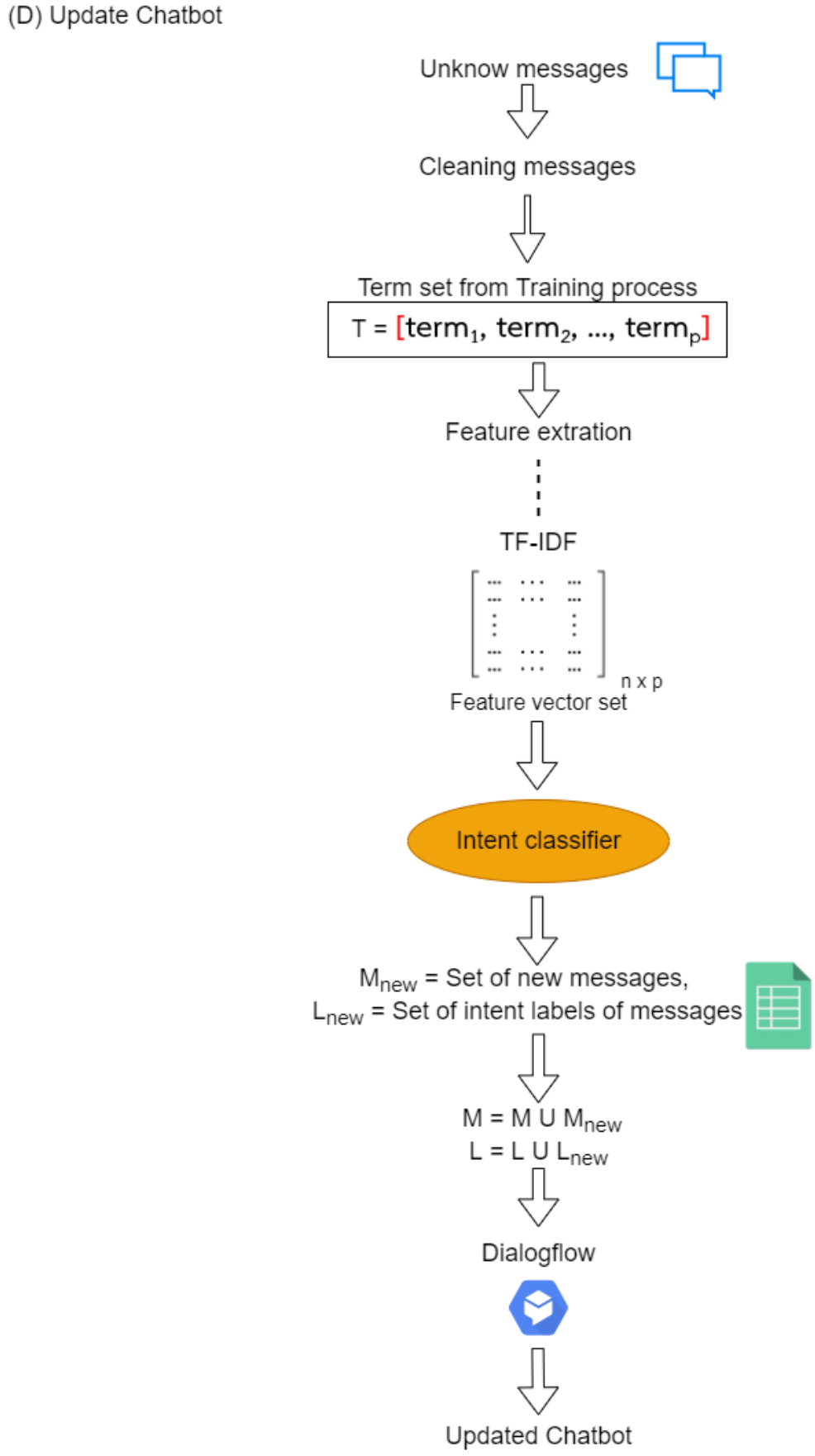
3.3.2 การอัพเดตแชทบอท

เมื่อมีคำถาม ซึ่งไม่มีอยู่ภายในแชทบอท เช่น คำถามจากแชทบอทที่แชทบอทไม่สามารถตอบได้ หรือข้อมูลเพิ่มเติมที่หาได้ ข้อความเหล่านี้จะถูกนำเข้ามาในกระบวนการ เพื่ออัพเดตแชทบอทอีกครั้ง ดังขั้นตอนในภาพที่ 3.5 จะต้องนำมาระบุ intent ด้วยการนำเข้าสู่โมเดล และนำผลลัพธ์ที่เรียกว่า เซตของข้อความใหม่ แทนด้วย M_{new} และเซตของ intent ของข้อความใหม่ แทนด้วย L_{new} และนำมารวมกับข้อมูลเดิม จะเขียนออกมาได้ว่า

$$M = M \cup M_{new}$$

$$L = L \cup L_{new}$$

ก่อนนำเข้า Dialogflow เพื่ออัปเดตแชทบอท ให้มีข้อมูลเพิ่มขึ้น และเป็นการอัพเดตแชทบอท ให้มีความสามารถมากขึ้น



รูปที่ 3.5: ขั้นตอนการทำงาน D การอัพเดตแชทบอท

บทที่ 4

การทดลองและผลการทดลอง

4.1 ชุดข้อมูลที่ใช้

4.1.1 ชุดข้อมูลมาตรฐาน

ในโครงงานฉบับนี้ได้ใช้ชุดข้อมูลมาตรฐานสำหรับทดสอบข้อมูลจำนวน 3 กลุ่ม ได้แก่

1) ชุดข้อมูล ATIS หรือ Airline Travel Information System จาก https://www.kaggle.com/hassanamin/atis-airlinetravelinformationsystem?select=atis_intents.csv เป็นชุดข้อมูลมาตรฐานของที่ใช้กันอย่างแพร่หลายสำหรับสร้าง chatbot โดยประกอบด้วยข้อมูลจำนวน 4978 ตัว

2) ชุดข้อมูล Corona Dataset จาก https://github.com/botxo/corona_dataset ชุดข้อมูลนี้สามารถนำไปใช้ในการสอนแชทบอทให้สามารถเข้าใจคำถามเกี่ยว corona virus ได้ โดยประกอบด้วยข้อมูลจำนวน 1053 ตัว

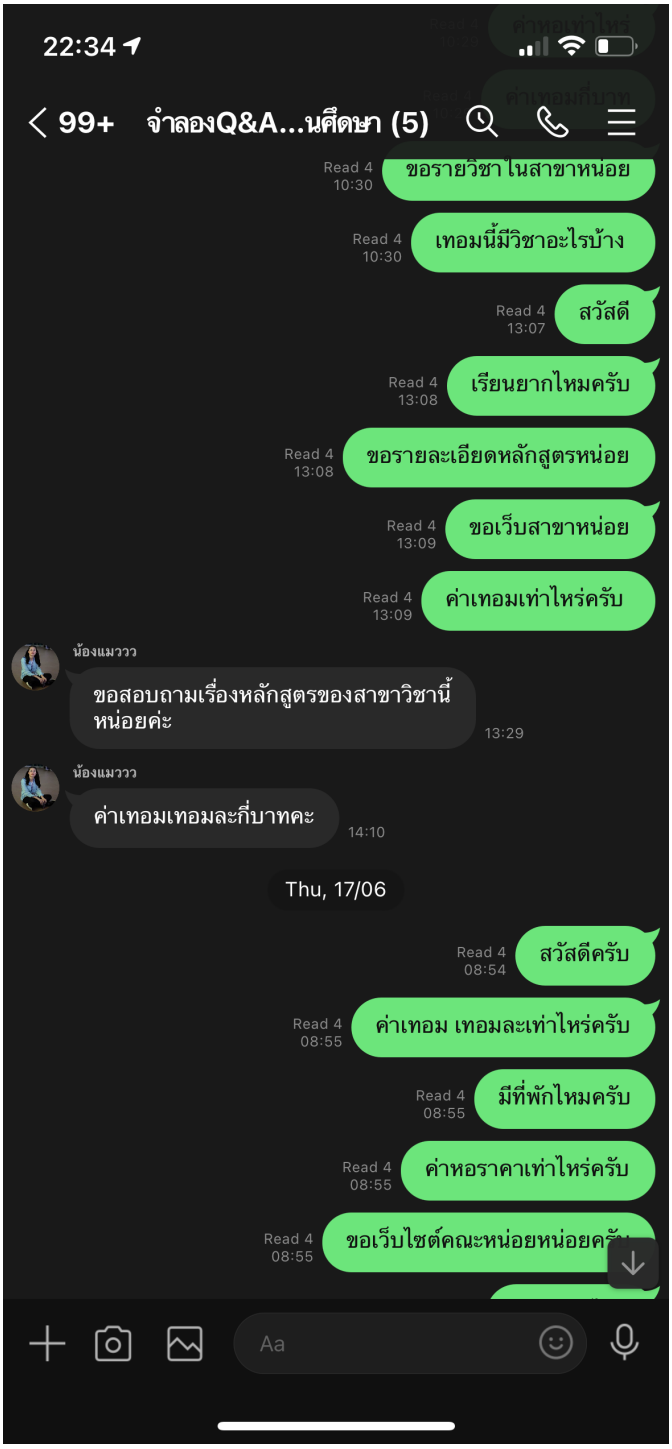
3) ชุดข้อมูล case routing intent จาก language - Einstein intent - training dataset error - Salesforce Stack Exchange เป็นชุดข้อมูลคำถามเกี่ยวกับการซื้อของ โดยประกอบด้วยข้อมูลจำนวน 150 ตัว

4.1.2 ชุดข้อมูลที่สร้างขึ้นเอง

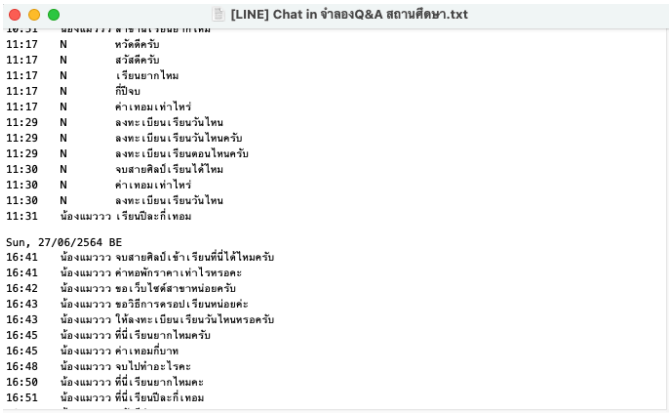
คือ ชุดข้อมูลจำลองเพื่อใช้สำหรับสร้างแชทบอทโดยอาศัยวิธีการจัดกลุ่มและจำแนกที่นำเสนอในโครงงานนี้ ถูกจัดทำขึ้นภายใต้หัวข้อ "แชทบอทสำหรับสถาบันการศึกษา" โดยจะมีข้อความไม่ต่ำกว่า 500 ข้อความ และมีขั้นตอนในการจัดทำดังต่อไปนี้

- สร้างกลุ่มไลน์สำหรับรวบรวมคำถามเกี่ยวกับหัวข้อที่กำหนด (รูปที่ 4.1)
- เก็บข้อมูลการสนทนาภายในกลุ่มไลน์ โดยการใช้คุณสมบัติ Export chat history ของไลน์ จากโทรศัพท์มือถือ จะได้เป็นไฟล์ .txt (รูปที่ 4.2)
- เขียนโปรแกรมสำหรับสกัดเอาเฉพาะข้อความแชทจากไฟล์ที่ได้จากไลน์กลุ่ม

- สุดท้ายจะได้เป็นไฟล์ .csv สำหรับไปทำต่อในขั้นตอนต่อไป (รูปที่ 4.4)



รูปที่ 4.1: ตัวอย่างข้อความในกลุ่มไลน์



รูปที่ 4.2: ไฟล์บันทึก Chat history จากไลน์กลุ่ม

```python
import json, csv,codecs
from pythainlp import sent_tokenize, word_tokenize


with open('comsci_res.txt',"r") as reader:
    texts = reader.readlines()
    set_texts = []
    check = True;
    for text in texts:
        if len(text) <= 1:
            check = False
        if (text.find("joined the group.") != -1):
            check = False
        if (text.find("[Notes]") != -1):
            check = False
        # add text
        if (check):
            set_texts.append(text.split("\t"))
        check = True

    last_texts = ["text"]

    for text in set_texts:
        message = ""
        for index in range(2,len(text)):
            message += text[index]
        if message != "" and message !="BE":
            last_texts.append(message)


    with codecs.open("comsci_data.csv","w", "utf-8") as write:
        text_write = csv.writer(write)
        for text in last_texts:
            text_write.writerow([text])

```

รูปที่ 4.3: source code สำหรับแปลงไฟล์จากไลน์

	message
1	message
2	สวัสดีค่ะ
3	ขอข้อมูลหลักสูตรหน่อยค่ะ
4	เรียนกี่ปีจบ
5	ค่าเทอมกี่บาท
6	เรียนแมทกี่ตัว
7	มีที่พักไหมคะ
8	เรียนจบไปทำงานอะไร
9	มีหอพักสำหรับนักศึกษามั้ยคะ
10	ค่าหอเท่าไหร่คะ
11	เรียนยากมั้ยคะ
12	ขอระเบียบการหอพัก
13	เปิดเทอมวันไหน
14	Hi
15	จบอะไรถึงเรียนได้คะ
16	สายศิลป์เรียนได้ไหม
17	สวัสดี
18	สวัสดีครับ ตึกสาขาตั้งอยู่ตรงไหนครับ
19	เปิดเรียนปีการศึกษาละกี่เทอมคะ
20	ค่าเทอมลดกี่เปอเซ็นคะช่วงโควิด
21	อรุณสวัสดิ์ค่ะ :)
22	สนใจเรียนสมัครอย่างไรคะ
23	ขอเว็ปมหาลัยหน่อยจ้า

รูปที่ 4.4: ตัวอย่างข้อมูลที่ได้จากการสกัดข้อความจากไฟล์ Chat history

4.2 ตัวชี้วัดประสิทธิภาพ

ในหัวข้อนี้จะกล่าวถึงตัวชี้วัดประสิทธิภาพในการจัดกลุ่ม และตัวชี้วัดประสิทธิภาพในการจำแนก ซึ่งมีรายละเอียดดังต่อไปนี้

4.2.1 การประเมินผลการแบ่งกลุ่ม (Clustering evaluation)

ตัวชี้วัดประสิทธิภาพ ที่ใช้ในโครงงานนี้ ได้แก่ Putiry และ Sum square of error (SSE)

1) **Purity** เป็นตัวชี้วัดประสิทธิภาพการจัดกลุ่มที่พิจารณาผลการจัดกลุ่มกับคลาสจริงของข้อมูล ข้อมูลที่ถูกจัดให้อยู่ในกลุ่มเดียวกันควรมาจากคลาสเดียวกัน โดยมีสูตร [13] ดังนี้

$$P_{ij} = \frac{m_{ij}}{m_i}$$

เมื่อ $i = 1, 2, ..., k$ กลุ่ม และ $j = 1, 2, ..., l$ คลาส

โดย m_i คือ จำนวนข้อมูลที่อยู่ภายในกลุ่ม i ทั้งหมด

m_{ij} คือ จำนวนข้อมูลในคลาส j ที่อยู่ในกลุ่ม i ทั้งหมด

หลังจากได้ค่า P_{ij} จะนำมาคำนวณหาค่า Purity ของกลุ่มที่ i เขียนแทนด้วย P_i มีสูตรดังสมการ

$$P_i = max_{j=1}^{l} P_{ij}$$

จากนั้นนำค่า P_i ที่ได้ไปหาค่า Purity ของการแบ่งกลุ่มด้วยสมการ

$$purity = \sum_{i=1}^{k} \frac{m_i}{m} P_i$$

โดย m คือ จำนวนข้อมูลทั้งหมดที่นำมาแบ่งกลุ่ม

2) **Sum Square of Error (SSE)** เป็นการพิจารณาประสิทธิภาพของการจัดกลุ่ม โดยการหาผลรวมค่าระยะห่างของข้อมูลแต่ละตัวกับจุดกึ่งกลางของกลุ่มข้อมูลนั้นๆ [13] หากค่าที่หาได้เข้าใกล้ 0 หมายความว่าข้อมูลภายในกลุ่มมีการกระจายตัวน้อย ซึ่งถือว่ามีประสิทธิภาพดี การหาค่า SSE มีสูตรการคำนวณ ดังนี้

$$SSE = \sum_{i=i}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

โดยที K คือ จำนวนของกลุ่มทั้งหมด
C_i คือ เซตของข้อมูลในกลุ่มที่ i
c_i คือ จุดศูนย์กลางของกลุ่มที่ i
x คือ จุดข้อมูล
$i =$ $1, 2, ..., K$

4.2.2 การประเมินผลการจำแนก (Classification evaluation)

เป็นการตรวจสอบประสิทธิภาพในการจำแนกข้อมูล โดยใช้เกณฑ์ในการประเมินได้แก่ Precision, Recall และ F1-Score [14] ซึ่งก่อนที่จะทำความเข้าใจ จำเป็นจะต้องรู้คำศัพท์ที่ต้องใช้เสียก่อน

TP (True Positive) คือ จำนวนข้อมูลทดสอบที่จำแนกว่าอยู่ในคลาสที่สนใจ และถูกต้องตามเฉลย

TN (True Negative) คือ จำนวนข้อมูลทดสอบที่จำแนกว่าไม่อยู่ในคลาสที่สนใจ และถูกต้องตามเฉลย

FN (False Negative) คือ จำนวนข้อมูลทดสอบที่จำแนกว่าอยู่ในคลาสที่สนใจ แต่ไม่ถูกต้องตามเฉลย

FP (False Positive) คือ จำนวนข้อมูลทดสอบที่จำแนกว่าไม่อยู่ในคลาสที่สนใจ แต่ไม่ถูกต้องตามเฉลย

จากนั้นนำไปใช้กับวิธีการในการประเมินประสิทธิภาพ ดังนี้

1) **Precision** คือ สัดส่วนความถูกต้องของผลลัพธ์ที่ได้จากการจำแนกคลาสที่สนใจ ต่อจำนวนข้อมูลที่อยู่ในคลาสที่สนใจทั้งหมดตามผลเฉลย เป็นการตรวจสอบความแม่นยำในการจำแนกของโมเดล

$$Precision = \frac{TP}{TP + FP}$$

2) **Recall** คือ สัดส่วนความถูกต้องของผลลัพธ์ที่ได้จากการจำแนกคลาสที่สนใจ ต่อจำนวนข้อมูลที่ได้จากการจำแนกจากคลาสที่สนใจทั้งหมด เป็นการตรวจความครบถ้วนในการจำแนกของโมเดล

$$Recall = \frac{TP}{TP + FN}$$

3) **F1-Score** คือ การหาค่าเฉลี่ยแบบ harmonic mean ของ Precision และ Recall เพื่อใช้ในการอธิบายประสิทธิภาพของโมเดล

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.3 การจัดกลุ่มข้อความ

ในการทำแชทบอท เมื่อเราได้ทำการข้อความการสนทนา แล้วจะนำข้อความเหล่านั้นมาจัดกลุ่มก่อนที่จะมีการระบุ intent ดังในหัวข้อนี้จะทำการทดลองเพื่อหาวิธีการจัดกลุ่มที่เหมาะสม

ในโครงงานนี้ ใช้ TF-IDF เป็นคุณลักษณะแทนข้อความ ตามที่ได้กล่าวไว้ในบทที่ 2 โดยมีกำจัดคำที่พบบ่อย หรือเรียกว่า คำหยุด (stop word) ได้แก่ คำต่อไปนี้

ค่ะ ครับ หน่อย คะ จ้า

เมื่อได้คุณลักษณะที่ต้องการ เพื่อนำไปใช้งานการจัดกลุ่มต่อไป

นอกจากนี้ ได้ทำการลดจำนวนมิติของเวกเตอร์ TF-IDF เพื่อให้มีขนาดของข้อมูลเล็กลง และทำให้จัดกลุ่มข้อมูลได้รวดเร็วขึ้น ทางคณะผู้จัดทำจึงได้นำการลดจำนวนมิติ โดยใช้ PCA แบบหาจำนวนมิติที่เหมาะสมแบบอัตโนมัติ ดังแสดงในขั้นตอนในรูปที่ 4.5 มาใช้ในการทดลอง ได้ผลดังตาราง 4.1

```python
import numpy as np
from sklearn.decomposition import PCA

def bestPCA (feature, n_component):
    pca = PCA(n_components=n_component)
    pca.fit(feature)
    sum_variance_ratio = np.sum(pca.explained_variance_ratio_)
    number_component = 0

    for i in pca.explained_variance_ratio_:
        number_component += 1
        if np.sum(pca.explained_variance_ratio_[0:number_component]) >= 0.8:
            break

    pca = PCA(n_components=number_component).fit(feature)

    return pca.transform(feature)
```

รูปที่ 4.5: source code pca

ตารางที่ 4.1: ผลการทำ PCA

ชุดข้อมูล	เวลาที่ใช้ (วินาที)	จำนวนมิติของข้อมูล	จำนวนมิติหลังทำ PCA ที่ได้
case_routing_intent	0.022895098	197	64
corona	0.264032841	609	172
atis_intents	0.433625937	498	111

4.3.1 การจัดกลุ่มโดยใช้ PCA และไม่ใช้ PCA บนชุดข้อมูลมาตรฐาน

จากการลดจำนวนมิติใน ตาราง 4.1 จะเห็นว่า จำนวนมิติในแต่ละชุดข้อมูลลดลงมากกว่า 50% แต่เราไม่ทราบว่าการลดจำนวนมิติของข้อมูลจะส่งผลดีต่อการจัดกลุ่มจริงหรือไม่ ทำให้เกิดการทดลองเพื่อทดสอบประสิทธิภาพของการลดจำนวนมิติ โดยใช้การจัดกลุ่มทั้งสองวิธีที่ได้กล่าวไว้ในบทที่ 2 ได้-แก่

- K-means

  ในการทดลองวิธีการจัดกลุ่มแบบ K-means จำเป็นจะต้องกำหนดจำนวน centroid โดยทางคณะผู้จัดทำได้ใช้วิธีการ Silhouette Method ในการหาจำนวน centroid ที่เหมาะสมที่สุด ดังแสดงในรูปที่ 4.6

- DBSCAN

  ในการทดลองวิธีการจัดกลุ่มแบบ DBSCAN จำเป็นจะต้องกำหนดพารามิเตอร์ 2 ตัวได้แก่ MinPts และ eps โดยในการทดลองได้กำหนด ให้ MinPts = 3 และ MinPts = 5 ส่วนค่า eps จะกำหนดโดยวัดระยะทางของข้อมูล 2 ตัวที่อยู่ใกล้กันที่สุดจนครบทุกตัวในชุดข้อมูล นำมาเรียงจากน้อยไปมาก และหาจุดที่ระยะทางที่เรียงติดกันมีค่าต่างกันมากที่สุด โดยในการทดลองจะใช้

ฟังก์ชัน NearestNeighbors ใน package scikit-learn มาประยุกต์ในการกำหนดค่าที่เหมาะสม ดังแสดงในรูปที่ 4.7

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

def best_k (feature, max_):
    num_sil = []
    k = 2
    maximum = -1

    for i in range(2, max_):
        model = KMeans(n_clusters=i).fit(feature)
        label_ = model.labels_
        centroids = model.cluster_centers_
        num_sil.append(silhouette_score(feature,label_,metric='euclidean'))
        if num_sil[-1] > maximum:
            maximum = num_sil[-1]
            k = i

    return k
```

รูปที่ 4.6: source code สำหรับหาค่าจำนวน centroid ด้วย Silhouette Method

```python
import numpy as np
from sklearn.neighbors import NearestNeighbors
from matplotlib import pyplot as plt

def bestEps (feature):
    neigh = NearestNeighbors(n_neighbors=2)
    nn = neigh.fit(feature)
    distances, indices = nn.kneighbors(feature)

    distances = distances[:,1]
    distances = list(set(distances))
    distances = np.sort(distances, axis=0)

    eps_ = 0.1
    for i in range(len(distances)-1):

        if distances[i+1] - distances[i] > eps_:
            eps_ = distances[i+1]

    eps_ = round(eps_, 1)

    return eps_
```

รูปที่ 4.7: source code สำหรับหาค่า eps ด้วย NearestNeighbors

ประสิทธิภาพของการจัดกลุ่มข้อความด้วยคุณลักษณะ TF-IDF ที่ใช้ PCA และไม่ใช้ PCA บนชุดข้อมูลมาตรฐาน จะพิจารณาจากคลาสของข้อมูลผ่านตัวชี้วัดประสิทธิภาพ คือ purity ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2: ผลการเปรียบเทียบการใช้ PCA และไม่ใช้ PCA บนชุดข้อมูลมาตรฐาน

ข้อมูล	วิธีการ	ใช้ PCA	S	D	purity	เวลาจัดกลุ่ม
case_routing_intent	K-means	no	NaN	k=14	**0.6667**	0.1180
case_routing_intent	K-means	yes	NaN	k=14	0.6333	**0.1090**
case_routing_intent	DBSCAN	no	minPts=3	eps=0.6	0.3133	0.0618
case_routing_intent	DBSCAN	yes	minPts=3	eps=0.1	0.2933	0.0608
case_routing_intent	DBSCAN	no	minPts=5	eps=0.6	0.2933	0.0609
case_routing_intent	DBSCAN	yes	minPts=5	eps=0.1	0.2933	0.0590
corona	K-means	no	NaN	k=99	0.5745	1.8632
corona	K-means	yes	NaN	k=102	**0.6135**	**0.8775**
corona	DBSCAN	no	minPts=3	eps=0.3	0.0997	0.3211
corona	DBSCAN	yes	minPts=3	eps=0.1	0.0969	0.3012
corona	DBSCAN	no	minPts=5	eps=0.3	0.0551	0.2940
corona	DBSCAN	yes	minPts=5	eps=0.1	0.0551	0.2579
atis_intents	K-means	no	NaN	k=489	0.6135	28.8498
atis_intents	K-means	yes	NaN	k=485	**0.8763**	**10.0008**
atis_intents	DBSCAN	no	minPts=3	eps=0.1	0.7704	1.7692
atis_intents	DBSCAN	yes	minPts=3	eps=0.1	0.7808	1.6541
atis_intents	DBSCAN	no	minPts=5	eps=0.1	0.7521	1.5810
atis_intents	DBSCAN	yes	minPts=5	eps=0.1	0.7618	1.4044

จากตาราง 4.2 จะเห็นว่า เมื่อพิจารณาจากค่า purity ในกรณีชุดข้อมูล case_routing_intent เมื่อไม่มีการลดจำนวนมิติด้วย PCA ค่า purity ที่ได้ออกมาจะมีค่ามากกว่า แต่ในกรณีชุดข้อมูล corona และ atis_intents เมื่อมีการลดจำนวนมิติด้วย PCA ค่า purity ที่ได้ออกมาจะมีค่ามากกว่า และในทุกชุดข้อมูลในกรณีที่มีการลดจำนวนมิติด้วย PCA จะใช้เวลาน้อยกว่ากรณีที่ไม่มีการใช้การลดจำนวนมิติ ซึ่งในกรณีชุดข้อมูล case_routing_intent จะเป็นชุดข้อมูลที่มีจำนวนข้อความน้อยกว่าอีกสองชุดข้อมูล จึงสรุปได้ว่าการลดจำนวนมิติด้วยวิธีการ PCA จะเหมาะกับข้อมูลที่มีขนาดใหญ่ ทางคณะผู้จัดทำจึงนำการลดจำนวนมิติมาใช้ในการทดลอง เนื่องจากคำนึงถึงขนาดของชุดข้อมูลจริงที่จะใช้ในอนาคต ซึ่งจะมีขนาดใหญ่

ดังนั้น ในการจัดกลุ่มข้อมูลที่จะทำแชทบอท เมื่อนำข้อความมาสกัดคุณลักษณะได้เป็น เวกเตอร์ TF-IDF แล้ว จึงนำมาลดมิติด้วย PCA จากนั้น นำไปจัดกลุ่มเพื่อให้สะดวกในการนำไประบุ intent สำหรับการสร้างแชทบอท

4.3.2 การจัดกลุ่มโดยใช้ K-means และ DBSCAN เมื่อใช้ PCA บนชุดข้อมูลมาตรฐาน และชุดข้อมูลที่สร้างขึ้นเอง

หลังจากได้คุณลักษณะของชุดข้อมูล ขั้นตอนต่อมาคือการจัดกลุ่มของข้อมูล โดยในการวัดประสิทธิภาพในการจัดกลุ่ม ทางคณะผู้จัดทำได้เพิ่มตัววัดประสิทธิภาพอีกหนึ่งตัวคือ SSE เพื่อใช้สำหรับดูความใกล้ชิดของข้อมูลภายในกลุ่ม ได้ผลการทดลองดังนี้

ตารางที่ 4.3: ผลการเปรียบเทียบประสิทธิภาพการจัดกลุ่มระหว่าง K-means และ DBSCAN

ข้อมูล	วิธีการ	sse	purity
case_routing_intent	K-mean	**85.5811**	**0.6333**
case_routing_intent	DBSCAN(minPts=3)	116.5259	0.2933
case_routing_intent	DBSCAN(minPts=5)	116.5259	0.2933
corona	K-mean	**356.5779**	**0.6135**
corona	DBSCAN(minPts=3)	715.9563	0.0969
corona	DBSCAN(minPts=5)	763.7182	0.0551
atis_intents	K-mean	**1146.6829**	**0.8763**
atis_intents	DBSCAN(minPts=3)	3207.7457	0.7808
atis_intents	DBSCAN(minPts=5)	3509.3300	0.7618
ข้อมูลที่สร้างขึ้นเอง	K-mean	**118.8681**	NaN
ข้อมูลที่สร้างขึ้นเอง	dbscan(minPts = 3)	171.8013	NaN
ข้อมูลที่สร้างขึ้นเอง	dbscan(minPts = 5)	243.9920	NaN

จากตาราง 4.3 พบว่าในทุกๆ ชุดข้อมูลที่ใช้ในการทดสอบ K-means มีค่า SSE ที่น้อยกว่า DBSCAN นั้นหมายความว่าระยะห่างของข้อมูลภายในกลุ่มมีความใกล้ชิดกันมากกว่า ดังนั้นจึงสามารถสรุปได้ว่าประสิทธิภาพของ K-means มีมากกว่า DBSCAN แต่ถึงอย่างนั้น ประสิทธิภาพของ K-means ก็ยังไม่เป็นที่น่าพอใจสำหรับนำไปใช้งานจริง ทางคณะผู้จัดทำจึงได้พัฒนาวิธีการในการเพิ่มประสิทธิภาพให้กับ K-means มาเป็นวิธีการจัดกลุ่มแบบ deep K-means

4.4 การจัดกลุ่มโดยใช้ deep K-means

ในโครงงานนี้ เสนอวิธีการจัดกลุ่มชื่อว่า deep K-means ซึ่งเป็นวิธีการในการแบ่งกลุ่มของข้อมูลลึกลงไป 3 ชั้น โดยแนวคิดดังนี้

1) *level1*: จัดกลุ่มข้อมูลโดยใช้วิธี K-means แบบ optimal

2) สำหรับแต่ละกลุ่ม ที่ j คำนวณหาระยะทางเฉลี่ยจากข้อมูลในกลุ่มไปยังจุดศูนย์กลางของกลุ่มแทนด้วย d_j

$$d_j = \frac{\sum_{i=1}^{n_j} |centroid_j - x_i|}{n_j}$$

3) หาค่าเฉลี่ยของค่าที่ได้ในข้อที่ 2)

$$D_{avg} = \frac{\sum_{j=1}^{N} d_j}{N}$$

4) *level2*: สำหรับแต่ละกลุ่ม ที่ j ถ้า $d_j > D_{avg}$ จะนำข้อมูลที่อยู่ในกลุ่มที่ j ไปจัดกลุ่มอีกครั้ง

5) พิจารณาผลการจัดกลุ่มใหม่ที่ได้ในข้อ 4) ตามหลักการในข้อ 2) - 3)

6) *level3*: สำหรับแต่ละกลุ่ม ที่ j ถ้า $d_j > D_{avg}$ จะนำข้อมูลที่อยู่ในกลุ่มที่ j ไปจัดกลุ่มอีกครั้ง

โดยที่ n คือ จำนวนข้อมูลภายในกลุ่มที่ j

i คือ ตำแหน่งของข้อมูลในกลุ่มที่ j
x_i คือ ตำแหน่งข้อมูลตัวที่ i ของกลุ่ม
$centroid$ คือ จุดศูนย์กลางของกลุ่ม
N คือ จำนวนกลุ่มทั้งหมดภายในการจัดกลุ่ม
j คือ ตำแหน่งกลุ่มภายในการจัดกลุ่ม

ผลการทดลองเปรียบเทียบประสิทธิภาพการจัดกลุ่มแบบ deep K-means การจัดกลุ่มแบบ K-means และการจัดกลุ่มแบบ DBSCAN แสดงดังตารางที่ 4.4

ตารางที่ 4.4: ผลการเปรียบเทียบประสิทธิภาพการจัดกลุ่มระหว่าง K-mean, DBSCAN และ deep K-mean

ข้อมูล	วิธีการ	sse	purity
case_routing_intent	K-mean	85.5811	0.6333
case_routing_intent	deep K-mean	**32.3764**	**0.86**
case_routing_intent	DBSCAN(minPts=3)	116.5259	0.2933
case_routing_intent	DBSCAN(minPts=5)	116.5259	0.2933
corona	K-mean	356.5779	0.6135
corona	deep K-mean	**140.9437**	**0.8072**
corona	DBSCAN(minPts=3)	715.9563	0.0969
corona	DBSCAN(minPts=5)	763.7182	0.0551
atis_intents	K-mean	1146.6829	0.8763
atis_intents	deep K-mean	**493.2257**	**0.9249**
atis_intents	DBSCAN(minPts=3)	3207.7457	0.7802
atis_intents	DBSCAN(minPts=5)	3509.3300	0.7616
ข้อมูลที่สร้างขึ้นเอง	K-mean	118.8681	NaN
ข้อมูลที่สร้างขึ้นเอง	deep K-mean	**34.4778**	NaN
ข้อมูลที่สร้างขึ้นเอง	dbscan(minPts = 3)	171.8013	NaN
ข้อมูลที่สร้างขึ้นเอง	dbscan(minPts = 5)	243.9920	NaN

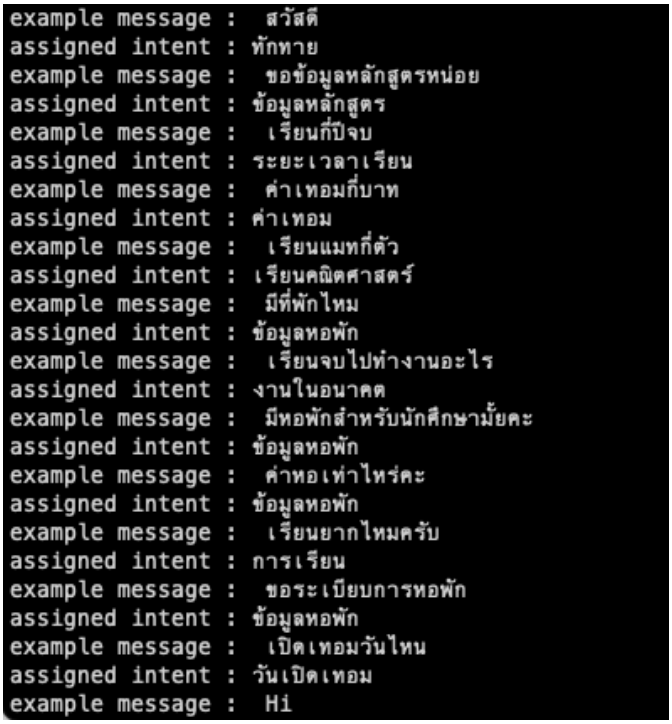
จากตาราง 4.4 จะเห็นได้ว่า ค่า SSE ของ deep K-means มีค่าน้อยที่สุด และ ค่า purity มีค่ามากที่สุด และมีค่ามากกว่า 80% ในทุกชุดข้อมูล ดังนั้น ทางคณะผู้จัดทำจึงได้เลือกวิธีการ deep K-means ในการจัดกลุ่มข้อความ

4.5 การระบุ intent

หลังจากทำการแบ่งกลุ่มข้อความด้วยวิธีการ deep K-means จำเป็นจะต้องกำหนดชื่อ intent ของแต่ละกลุ่ม เพื่อให้ง่ายต่อความเข้าใจ และสามารถระบุคำตอบได้ตรงตามเป้าหมายของแต่ละกลุ่มข้อมูล ทางคณะผู้จัดทำจึงได้เขียนโปรแกรมสำหรับระบุ intent โดยใช้วิธีการแสดงข้อความที่มีระยะทางอยู่ใกล้กับ centroid ของกลุ่มมากที่สุดเป็นตัวแทนของข้อความทั้งหมดภายในกลุ่ม จากนั้นรับข้อความที่เป็นชื่อ intent จากผู้ใช้ เพื่อเป็นการระบุชื่อให้กับกลุ่มข้อมูล

```python
import pandas as pd
import numpy as np

if __name__ == "__main__":
    deep_Kmean = "kmean_three_level.csv"
    df = pd.read_csv("comsci_result/"+deep_Kmean) # get data from deep Kmean

    label_group = df.centroids_id.unique() # get label number
    df["intent"] = np.NaN
    df['target'] = np.NaN

    count_target = -1
    for i in label_group: # insert label name by user
        insert_label = df[df.centroids_id == i]
        ex_text = insert_label[insert_label.dist_score == insert_label.dist_score.min()]\
                        .text.sample(1).to_string(index=False)
        print("example message : ", ex_text)
        label_name = input("assigned intent : ")
        intent = df.intent.tolist()
        target = -1
        if label_name not in intent:
            count_target += 1
            target = count_target
        else:
            target = df[df.intent == label_name].target.tolist()[0]
        df.at[df.centroids_id == i, "intent"] = label_name
        df.at[df.centroids_id == i, "target"] = target

    label_name = df.intent.unique().tolist()

    data_ = pd.DataFrame({'target':df.target, 'intent':df.intent, 'text':df.text })
    save_at = "data_training/intent_group.csv"
    data_.to_csv(save_at,encoding='utf-8-sig',index=False)
```

รูปที่ 4.8: source code โปรแกรมสำหรับระบุ intent



รูปที่ 4.9: ตัวอย่างโปรแกรมระบุ intent

จากรูปที่ 4.9 จะเห็นว่ามีข้อมูลบางกลุ่มที่มีความหมายเหมือนกัน แต่ถูกจัดให้อยู่คนละกลุ่ม ในกรณีนี้จะระบุให้ชื่อ intent เหมือนกัน หลังจากนั้นจะได้กลุ่มของข้อความใหม่ ที่มีการระบุ intent เรียบร้อยแล้ว

4.6 การสร้างแชทบอท เวอร์ชั่น 1

เมื่อทำการระบุ intent ครบทุกกลุ่ม จำเป็นต้องแยกข้อมูลเป็นกลุ่มตาม intent เพื่อกำหนด response ให้กับกลุ่มนั้นๆ ก่อนจะนำข้อมูลที่ได้ไปจัดระเบียบเพื่อเตรียมสอนให้กับ dialogflow แต่เนื่องจากการสอน dialogflow จำเป็นจะต้องมีรูปแบบการเขียนข้อมูลนำเข้าตามที่ถูกกำหนดไว้ ทางกลุ่มจึงเขียนอัลกอริทึมเพื่อจัดข้อมูลให้อยู่ในรูปแบบฟอร์มเฉพาะของ dialogflow

```
1  import re, csv, json
2  from copy import deepcopy
3  from pprint import pprint
4
5  def csvToDialogflow(file_name):
6      test_ = open('function/test.json',)
7      question=json.loads(test_.read())
8      test_ = open('function/test_usersays_en.json')
9      userSays=json.loads(test_.read())
10
11     file_ = open(file_name, en  delimiter: Any
12     reader = csv.reader(file_, delimiter=",")
13
14     render_line = 0
15     intent = None
16     message = []
17     respons = []
18
19     for row in reader:
20         if render_line != 0:
21             if intent == None:
22                 intent = row[0]
23             if row[1] != '':
24                 message.append(row[1])
25             if row[2] != '':
26                 respons.append(row[2])
27         render_line += 1
28
29     question_ = deepcopy(question)
30     userSays_ = deepcopy(userSays)
31     text_format = userSays_[0]
32     userSays_ = []
```

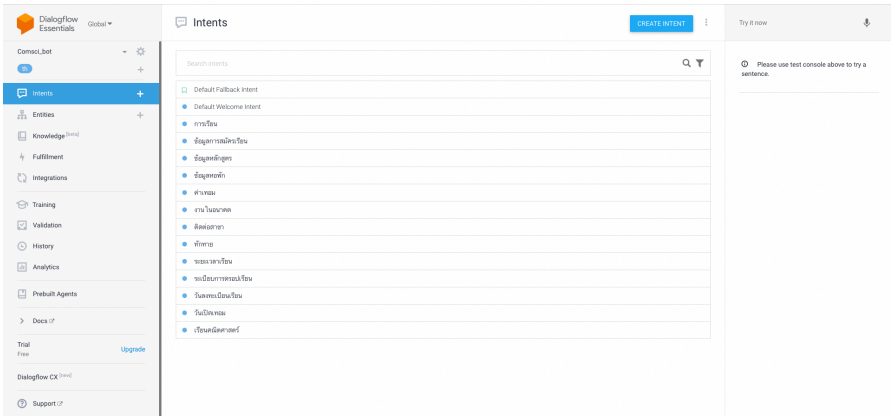
รูปที่ 4.10: ตัวอย่าง source code แปลงไฟล์

```
33
34      question_['name'] = intent
35      question_['responses'][0]['messages'][0]['speech'] = respons
36
37      # save file
38      name_ = intent+".json"
39      name_user_ = intent+"_usersays_th.json"
40
41      with open('dialogflow_file/'+name_,'w') as f:
42          f.write(json.dumps(question_, ensure_ascii=False))
43
44      id = 0
45      for i in range(len(message)):
46          tmp = deepcopy(text_format)
47          tmp['id'] = str(id)
48          tmp['data'][0]['text'] = message[i]
49          id += 1
50          userSays_.append(tmp)
51          # userSays_[0]['data'][0]['text'] = message[i]
52          # userSays_[0]['id'] = id
53          # id += 1
54
55      with open('dialogflow_file/'+name_user_,'w') as f:
56          f.write(json.dumps(userSays_, ensure_ascii=False))
57
```

รูปที่ 4.11: ตัวอย่าง source code แปลงไฟล์

เมื่อเสร็จสิ้นการทำงานจะได้ไฟล์รูปแบบ json ที่มีฟอร์มเฉพาะของ dialogflow โดยในขั้นตอนต่อไปจะเป็นการสร้าง dialogflow จากไฟล์ json ซึ่งมีขั้นตอนดังนี้

1) สร้าง agent ของ dialogflow ใหม่
2) export agent ที่สร้างขึ้นใหม่ และ แตก zip ไฟล์
3) นำไฟล์ json ที่สร้างขึ้นใหม่ ใส่ในโฟลเดอร์ intents
4) zip ไฟล์ agent แล้ว import เข้า dialogflow อีกครั้ง

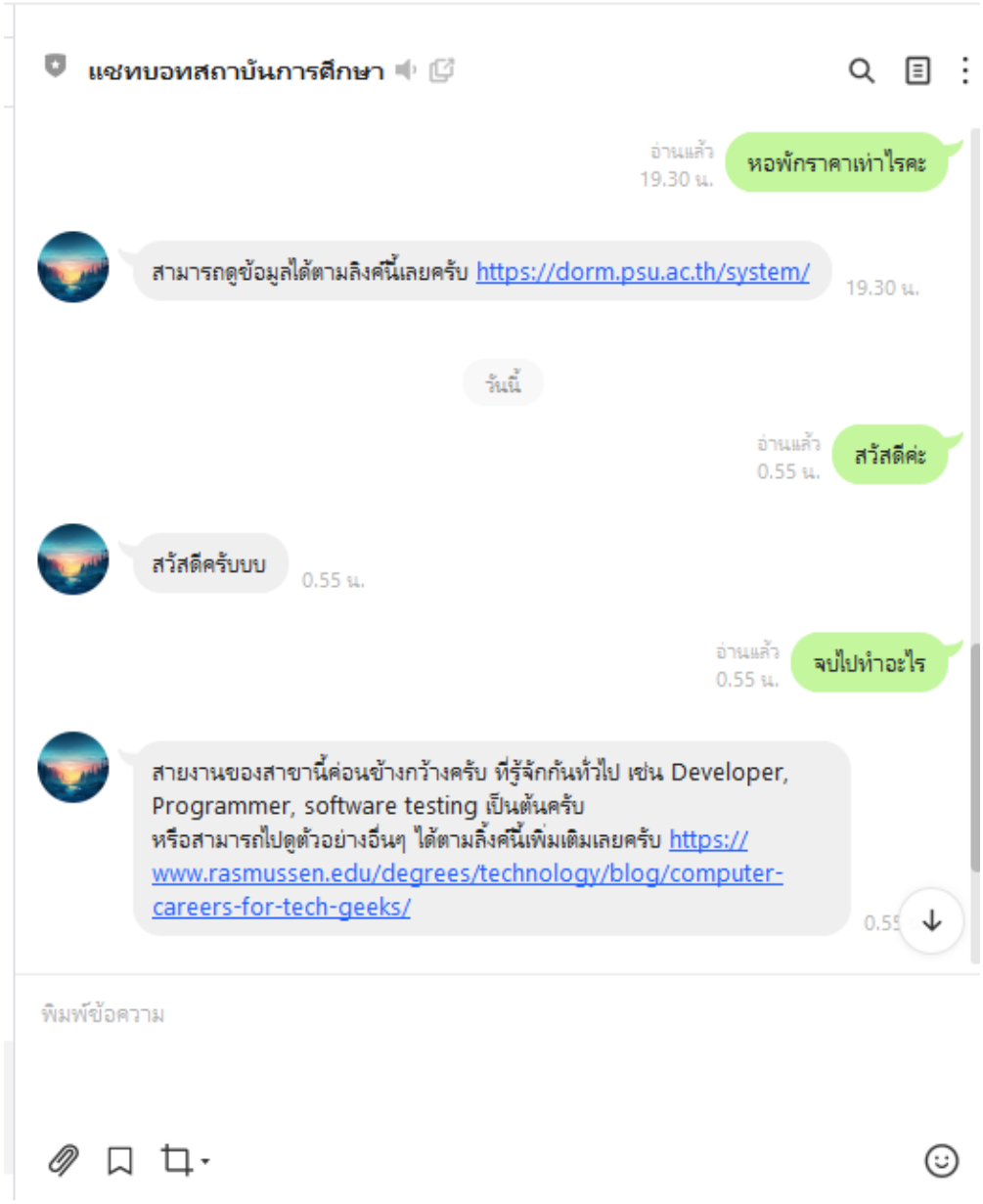


รูปที่ 4.12: ตัวอย่าง dialogflow หลังจากสร้างแชทบอทเรียบร้อยแล้ว

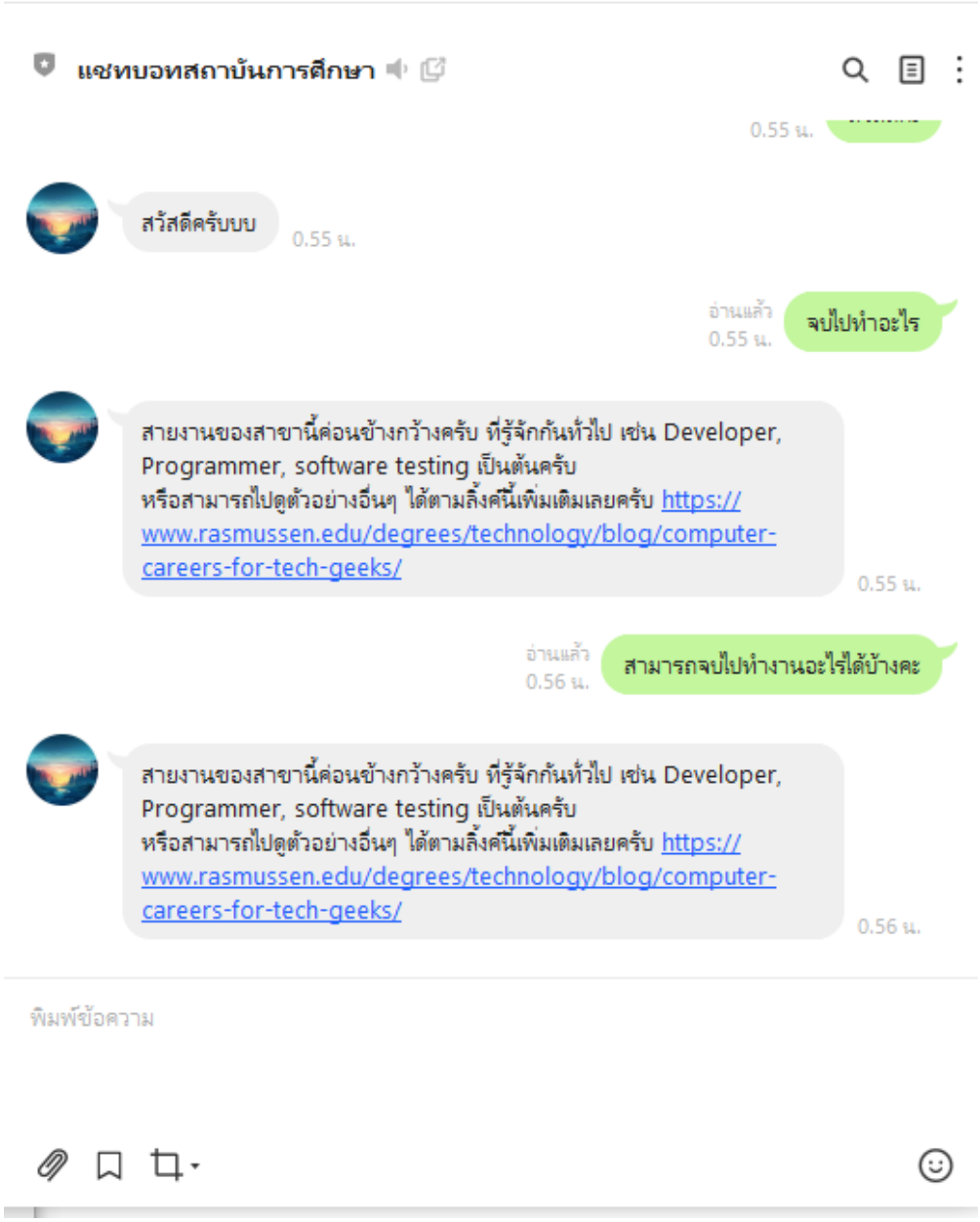
หลังจากนั้นจะทำการเชื่อมต่อไปยัง Line โดยขั้นตอนดังต่อไปนี้

1) ทำการสร้าง account Line developer หรือ Log in account Line ที่ใช้อยู่ก่อนแล้ว
2) สร้าง providers และทำการกรอกข้อมูลเบื้องต้นจนครบ
3) เมื่อสร้าง providers เป็นที่เรียบร้อย ให้กลับที่ dialogflow ไปที่หน้า Integrations เลื่อนหา Line ในหัวข้อ Text based
4) ดูข้อมูลที่ dialogflow ต้องการ แล้วกลับไปยังหน้า Line developer เพื่อนำข้อมูลกลับมาใส่ใน dialogflow จนครบ
5) นำ webhook url จาก dialogflow ไปใส่ใน Line developer และเปิดใช้งาน webhook เป็นอันเสร็จสิ้น

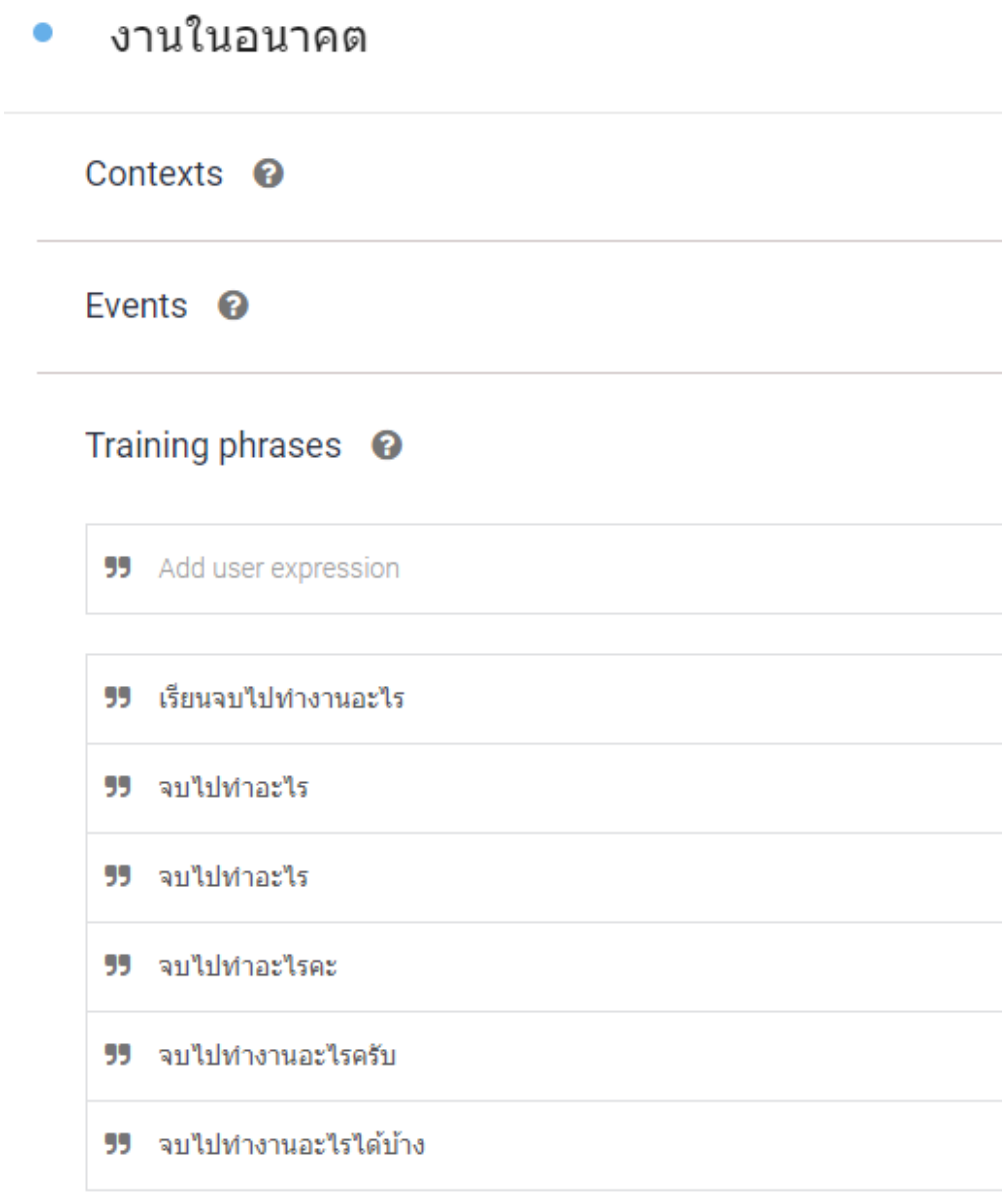
หลังจากนั้นทำการทดสอบระบบของ dialogflow ผ่าน Line



รูปที่ 4.13: ตัวอย่างผลลัพธ์การทดสอบแชทบอท จากคำที่เคยสอนแชทบอทแล้ว



รูปที่ 4.14: ตัวอย่างผลลัพธ์การทดสอบแชทบอท จากคำที่ไม่เคยสอนแชทบอท



รูปที่ 4.15: ข้อความที่อยู่ภายใน intent เพื่อทดสอบความสามารถของแชทบอท

บรรณานุกรม

[1] Andrew Mironov และคณะ. Different types of chatbots: Rule-based vs. nlp. https: //flow.ai/blog/kb-different-kinds-of-chatbots, 2016. (Accessed on 01/08/2021).

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008. (Accessed on 01/10/2021).

[3] Bhavika Kanani. Jaccard similarity - text similarity metric in NLP. https: //studymachinelearning.com/jaccard-similarity-text-similarity-metric-in-nlp/, 2020. (Accessed on 01/10/2021).

[4] Shruti Kapil, Meenu Chawla, and Mohd Dilshad Ansari. On k-means data clustering algorithm with genetic algorithm. In *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 202--206, 2016. (Accessed on 01/09/2021).

[5] Abhishek Sharma. How does dbscan clustering work? | dbscan clustering for ml. https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/, September 2020. (Accessed on 01/10/2021).

[6] Prince Yadav. Decision tree in machine learning. https://towardsdatascience.com/ decision-tree-in-machine-learning-e380942a4c96, November 2018. (Accessed on 01/10/2021).

[7] ธนาวุฒิ ประกอบผล. โครงข่ายประสาทเทียม artificial neural. *วารสาร มฉก.วิชาการ*, 12(24):73--87, 2552. (Accessed on 01/10/2021).

[8] Petch Kruapanich. ลองทำแชทบอทง่ายๆด้วย dialogflow กันเถอะ. https://medium.com/ readmoreth/ลองทำแชทบอทลงทะเบียนง่ายๆด้วย-dialogflow-กันเถอะ-4bd3a8c550de, April 2018. (Accessed on 01/03/2021).

[9] Gunther Cox. About chatterbot. https://chatterbot.readthedocs.io/en/stable, 2019. (Accessed on 01/04/2021).

[10] Andrew Mironov และ คณะ. What can flow.ai do? https://flow.ai/docs/what-can-flow-do. (Accessed on 01/09/2021).

[11] Pop Phiphat. Principal components analysis (PCA) ต่างจาก factor analysis (FA) ยังไง ? (ตอนที่ 1). https://medium.com/ingenio/principal-components-analysis-pca-ต่างจาก-factor-analysis-fa-ยังไง-ตอนที-1-c395e55bdc3, June 2018. (Accessed on 03/01/2021).

[12] Khyati Mahendru. How to determine the optimal k for k-means? https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb, June 2018. (Accessed on 03/03/2021).

[13] Tan และคณะ. *Introduction to Data Mining (2nd Edition)*. Pearson, 2nd edition, 2004. (Accessed on 03/08/2021).

[14] Jérémie du Boisberranger และคณะ. 3.3. metrics and scoring: quantifying the quality of predictions. https://scikit-learn.org/stable/modules/model_evaluation.html?fbclid=IwAR2JOz6tVB2PNu0oQUbssP48jiXQbhZvS1lbQXrXMlCTQjws2OCsGajAdi8#the-scoring-parameter-defining-model-evaluation-rules, 2007. (Accessed on 03/10/2021).