การจำแนกรุ่นอายุผู้ใช้งานเฟซบุ๊กไทย โดยใช้การเรียนรู้เชิงลึกร่วมกับความน่าจะเป็นของคำ

นายศุภชัย ตั้งตรีรัตน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Classification of generation of Thai facebook users using deep learning with probability of words

Mr. Suppachai Tangtreerat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2019

หัวข้อวิทยานิพนธ์	การจำแนกรุ่นอายุผู้ใช้งานเฟซบุ๊กไทย โดยใช้การเรียนรู้เชิงลึกร่วมกับความน่าจะเป็นของคำ
โดย	นายศุภชัย ตั้งตรีรัตน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นัทที นิภานันท์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐพงศ์ ชินธเนศ)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.เด่นดวง ประดับสุวรรณ)

ศุภชัย ตั้งตรีรัตน์ : การจำแนกรุ่นอายุผู้ใช้งานเฟซบุ๊กไทย โดยใช้การเรียนรู้เชิงลึกร่วมกับความน่าจะเป็นของคำ. (Classification of generation of Thai facebook users using deep learning with probability of words) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุกรี สินธุภิญโญ

เฟซบุ๊กเป็นแพลตฟอร์มที่ได้รับความนิยมมากที่สุดในโลก นักการตลาดจึงต้องการใช้ข้อมูลเฟซบุ๊กจำนวนมากในการทำการตลาด ดังนั้นการวิเคราะห์รุ่นอายุของผู้ใช้งานเฟซบุ๊กจึงเป็นเรื่องสำคัญ เพื่อนำรุ่นอายุของผู้ใช้งานเฟซบุ๊กมาวิเคราะห์หากลุ่มเป้าหมายในการทำการตลาด ในงานวิจัยนี้ได้ทำการวิเคราะห์รุ่นอายุของผู้ใช้งานเฟซบุ๊กจากข้อมูลการโพสของผู้ใช้งาน โดยใช้การรวมกันระหว่างการเรียนรู้เชิงลึกกับข้อมูลความน่าจะเป็นของคำในแต่ละรุ่นอายุ ผลลัพธ์จากการทดลองได้ค่าความแม่นยำแบบต่อผู้ใช้เท่ากับ 82.90% และค่าความแม่นยำแบบต่อโพสต์เท่ากับ 52.48% ซึ่งได้ประสิทธิภาพดีกว่าการใช้แบบจำลองเพอร์เซ็ปตรอนหลายชั้น , นิวรอลเน็ตเวิร์กคอนโวลูชัน ,หน่วยความจำระยะสั้นแบบยาว เพียงอย่างเดียว จากผลการทดลองแสดงให้เห็นว่าการใช้ความน่าจะเป็นของคำในแต่ละรุ่นอายุเข้ามาช่วย ทำให้สามารถเพิ่มประสิทธิภาพของแบบจำลองได้ดียิ่งขึ้น

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2562 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070974421 : MAJOR COMPUTER SCIENCE

KEYWORD: Generation Classification Thai Facebook Deep Learning Convolutional Neural Network Long-Short Term Memory

Suppachai Tangtreerat : Classification of generation of Thai facebook users using deep learning with probability of words. Advisor: Asst. Prof. SUKREE SINTHUPINYO

Facebook is the most popular platform in the world. Marketers would like to use Facebook user data, which comprises large amounts of information which is useful for marketing. Therefore, analyzing the generation of Facebook users for marketing research is important to successfully capture the target market. In this research, posted data of Thai Facebook users will be analyzed using the combined methods of deep learning and probability of words data. The experiment result yields an accuracy of 82.90% per user and 52.48% per status, which is better than using other models alone such as Multi-Layers Perceptron (MLP), Convolution Neural Networks (CNN), or Long Short-Term Memory (LSTM). The experiment results show that using probability of words in each generation can help to increase the accuracy of the model.

Field of Study: Computer Science Student's Signature .....

Academic Year: 2019 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความกรุณาจากผู้ช่วยศาสตราจารย์ ดร. สุกรี สินธุภิญโญ อาจารย์ที่ปรึกษา ผู้ที่ให้คำปรึกษา แนวคิด คำแนะนำ ในการทำวิทยานิพนธ์ และให้ความรู้ในคิดและวิเคราะห์ในการทำวิทยานิพนธ์ รวมถึงเป็นผู้ตรวจทานแก้ไขวิทยานิพนธ์ฉบับนี้ให้ลุล่วง ขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ด้วย

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. นัทที นิภานันท์ ประธานกรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร. ณัฐพงศ์ ชินธเนศ กรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร. เด่นดวง ประดับสุวรรณ กรรมการภายนอก ที่ได้ให้คำแนะนำและชี้แนะแนวทางที่เป็นประโยชน์ต่อการทำวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่อบรม สั่งสอน ให้วิชาความรู้ในทุกๆด้าน จนกระทั่งมีทุกวันนี้

ขอขอบคุณเพื่อนๆพี่ๆน้องๆในภาควิชาวิศวกรรมคอมพิวเตอร์ภาคนอกเวลาทุกคน ที่คอยช่วยกันทำงานและแลกเปลี่ยนความรู้กัน รวมถึงกำลังใจในการเรียนและทำวิทยานิพนธ์

ขอขอบคุณ "พี่ปิ้น" นายปริญญา ตั้งกมลสถาพร ที่ให้คำแนะนำและให้ความรู้ในการทำวิทยานิพนธ์

ขอขอบคุณ "อ้อม" นางสาวสุทธนุช ไพฑูรย์ ที่คอยให้คำแนะนำ กำลังใจและคอยดูแล ในการทำวิทยานิพนธ์

สุดท้ายขอกราบขอบคุณพระคุณปาป๊า มาม๊า พี่ฝน ญาติพี่น้องทุกคนที่คอยดูแลและเป็นกำลังใจในการทำวิทยานิพนธ์

ศุภชัย ตั้งตรีรัตน์

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของปัญหา

ข้อมูลรุ่นอายุคือข้อมูลที่สำคัญในการทำการวิจัยด้านการตลาด เช่น การนำข้อมูลรุ่นอายุมาวิเคราะห์หากลุ่มผู้ที่ซื้อสินค้า หรือการนำข้อมูลรุ่นอายุมาวิเคราะห์หากลุ่มผู้ใช้บริการ เพื่อนำมาปรับปรุงหรือพัฒนาสินค้าและบริการ ในปัจจุบันคนไทยนิยมใช้เฟซบุ๊กในการซื้อสินค้าหรือโฆษณาบริการต่างๆเป็นจำนวนมาก การนำข้อมูลรุ่นอายุของผู้ใช้งานเฟซบุ๊กมาวิเคราะห์เพื่อทำการวิจัยด้านการตลาดจึงเป็นเรื่องที่สำคัญ แต่ยังมีผู้งานที่ไม่ได้กรอกข้อมูลอายุหรือกรอกข้อมูลอายุแต่ไม่เป็นไปตามจริง

การเรียนรู้ของเครื่อง (Machine Learning) เป็นวิธีมาตราฐานในการจำแนกข้อความ โดยนิยมใช้การแทนข้อความด้วยเวกเตอร์เช่น ถุงคำ (bag-of-word), ทีเอฟไอดีเอฟ (TFIDF) จากนั้นนำไปจำแนกด้วยวิธีการเรียนรู้ของเครื่อง เช่น ซัพพอร์ตเวกเตอร์แมชชีน (SVM), นาอีฟเบย์ (naive bayes) หรือต้นไม้ตัดสินใจ (decision tree) โดยมีการนำไปใช้กับงานวิจัยด้านการจำแนกข้อความในด้านต่างๆ เช่น การวิเคราะห์ความคิดเห็นของข้อความภาษาไทย [1]

การเรียนรู้เชิงลึก (Deep Learning) เป็นอีกวิธีที่ถูกนำมาใช้ในการจำแนกข้อความ โดยนิยมใช้การแทนข้อความด้วยกลุ่มของเวกเตอร์เช่น คำฝังตัว (Word Embedding) จากนั้นนำไปจำแนกด้วยวิธีการเรียนรู้เชิงลึก เช่น นิวรอลเน็ตเวิร์กคอนโวลูชัน (Convolutional Neural Network), หน่วยความระยะสั้นแบบยาว (Long-Short Term Memory หรือ LSTM) โดยมีการนำไปใช้กับงานวิจัยด้านการจำแนกข้อความในด้านต่างๆ เช่น การใช้นิวรอลเน็ตเวิร์กคอนโวลูชันในการจำแนกอายุและเพศของผู้ใช้งานทวิตเตอร์ [2], การเรียนรู้เชิงลึกกับถุงคำในการจำแนกความรู้สึกของข้อมูลเฟซบุ๊กภาษาไทย [3] ทั้งนี้ผลจากงานวิจัยของการใช้การเรียนรู้เชิงลึกสามารถให้ค่าความแม่นยำที่มากกว่าการใช้การเรียนรู้ของเครื่อง

วารสารภาษาและวัฒนธรรม ฉบับที่ 35 ฉบับพิเศษ ในหัวข้อเรื่องลักษณะเด่นของคำกริยาในภาษาวัยรุ่นไทย [4] ได้ทำแบบสอบถามการใช้คำกริยาในภาษาวัยรุ่น ระหว่างกลุ่มวัยรุ่นและกลุ่มผู้ใหญ่ เช่น คำว่า “แจ่ม” และ “แอ๊บ” กลุ่มผู้ใหญ่มีการใช้ร้อยละ 40 ส่วนกลุ่มวัยรุ่นมีการใช้ร้อยละ 100 หรือคำว่า “จัดหนัก” กลุ่มผู้ใหญ่มีการใช้ร้อยละ 30 ส่วนกลุ่มวัยรุ่นมีการใช้ร้อยละ 96.6

งานวิจัยนี้ขอเสนอการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กโดยใช้แบบจำลองการเรียนรู้เชิงลึกร่วมกับข้อมูลความน่าจะเป็นคำในแต่ละรุ่นอายุ เพื่อเพิ่มประสิทธิภาพในการจำแนกรุ่นอายุให้ดีขึ้น

1.2. วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอวิธีการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กไทย โดยใช้แบบจำลองการเรียนรู้เชิงลึกร่วมกับข้อมูลความน่าจะเป็นของการใช้คำในแต่ละรุ่นอายุ เพื่อเพิ่มประสิทธิภาพในการจำแนกรุ่นอายุให้ดีขึ้น

1.3. ขอบเขตการวิจัย

1.3.1. ข้อมูลที่นำมาใช้ในงานวิจัย คือข้อมูลที่ได้จากการโพสต์เฟซบุ๊กของผู้ใช้งานคนไทย จำนวน 300 คน คนละ 30 โพสต์ รวมทั้งสิ้น 9,000 โพสต์ โดยเริ่มรวบรวมข้อมูลการโพสต์ตั้งแต่วันที่ 01 พ.ย. 2561 ถึงวันที่ 28 ก.พ. 2562

1.3.2. การแบ่งกลุ่มผู้ใช้งานจะแบ่งตามรุ่นอายุของผู้ใช้งาน โดยแบ่งออกเป็น 3 กลุ่ม ดังนี้

- รุ่นอายุเอกซ์ (Generation X) คือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2508 – 2522
- รุ่นอายุวาย (Generation Y) คือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2523 – 2544
- รุ่นอายุแซด (Genertaion Z) คือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2545 – 2563

1.3.3. เปรียบเทียบประสิทธิภาพของแบบจำลองที่นำเสนอกับแบบจำลองการเรียนรู้ของเครื่องและแบบจำลองการเรียนรู้เชิงลึก

1.4. ประโยชน์ที่คำดาดว่าจะได้รับ

ได้วิธีการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊ก โดยการใช้การเรียนรู้เชิงลึกร่วมกับข้อมูลความน่าจะเป็นของการใช้คำในแต่ละรุ่นอายุ และได้ค่าความแม่นยำมากกว่าแบบจำลองของการใช้การเรียนรู้ของเครื่อง และได้คลังข้อมูลของคำฝังตัวแบบเวิร์ดทูเวกของภาษาไทยเพื่อที่จะสามารถนำไปใช้งานต่อไปได้

1.5. วิธีดำเนินการวิจัย

1.5.1. เตรียมข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กในประเทศไทย โดยทำการรวบรวมข้อมูลจากการโพสต์ของผู้ใช้งาน 300 คน จำนวนคนละ 30 โพสต์ รวมทั้งสิ้น 9,000 โพสต์ ในการทดลองนี้จะแบ่งผู้ใช้งานออกเป็น 3 กลุ่ม กลุ่มละ 100 คน โดยการแบ่งกลุ่มผู้ใช้งานจะแบ่งตามรุ่นอายุของผู้ใช้งาน ได้แก่ รุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด

1.5.2. เตรียมข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในประเทศไทย โดยทำการรวบรวมข้อมูลจากการทวีตของผู้ใช้งานจำนวน 130,000 ทวีต

1.5.3. ทำการประมวลผลก่อน (preprocessing) โดยการใช้โปรแกรมตัดคำภาษาไทยและลบคำภาษาไทยไม่สำคัญ ที่นิยม 2 โปรแกรม ได้แก่ pythainlp, deepcut

1.5.4. ทำการแทนข้อความ (Text Representation) ด้วยวิธีต่างๆเช่น คำฝังตัว (Word Embidding) โดยการประมวลด้วยเวิร์ดทูเวก (word2vec), ถุงคำ (Bag-of-words หรือ BoW), ทีเอฟไอดีเอฟ (Term Frequency-Inverse Document Frequency หรือ TF-IDF), ข้อมูลความน่าจะเป็นของคำ (Probability of words) ในแต่ละรุ่นอายุ

1.5.5. นำข้อมูลไปเรียนรู้ด้วยโมเดลเพอร์เซ็ปตรอนหลายชั้น (Multi-layer perceptron), นิวรอลเน็ตเวิร์กคอนโวลูชัน (Convolutional Neural Network), หน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory หรือ LSTM)

1..5.6. นำผลลัพธ์ที่ได้จากการหาค่าความแม่นยำสูงสุดจากการปรับค่าพารามิเตอร์ที่เกี่ยวข้อง (Hyperparameter Tuning) มาวัดประสิทธิภาพ ด้วยคอนฟิวชันเมทริกซ์ (Confusion Matrix) และตัววัดประสิทธิภาพจำแนกตามคลาส

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. ทฤษฎีที่เกี่ยวข้อง

2.1.1. การแทนข้อความ (Text Representation)

การแทนข้อความเป็นขั้นตอนหนึ่งที่ต้องทำ สำหรับการจำแนกประเภทของข้อความ วิธีการแทนข้อความนั้นมีขั้นตอนดังต่อไปนี้ ทั้งนี้ ในแต่ละ หัวข้อจะแสดงถึงการแทนข้อความ 2 ข้อความ ได้แก่ 1) “ฉันไปโรงเรียนโรงเรียนฉันสวย” และ 2) “โรงเรียนของฉันน่าอยู่”

2.1.1.1. ถุงคำ (Bag-of-words หรือ BoW)

เป็นวิธีการแทนข้อความให้อยู่ในรูปแบบของเวกเตอร์ที่มีขนาดเท่ากับจำนวนคำทั้งหมดในพจนานุกรมของชุดข้อมูล โดยใช้ความถี่ของคำที่ปรากฏแทนข้อความ จากข้อความตัวอย่าง สามารถสร้างพจนานุกรมของคำได้ดังนี้ [“ชั้น”,“ชอบ”,“หมา”,“น่ารัก”,“ของ”,“อ้วน”] โดยสามารถแทนข้อความตัวอย่างได้ดังนี้

1) “ชั้นชอบหมา หมาชั้นน่ารัก” แทนข้อความด้วย [2 1 2 1 0 0]
2) “หมาของชั้นอ้วน” แทนข้อความด้วย [1 0 1 0 1 1]

2.1.1.2. ทีเอฟไอดีเอฟ (Term Frequency-Inverse Document Frequency หรือ TF-IDF)

เป็นวิธีการแทนข้อความให้อยู่ในรูปแบบของเวกเตอร์คล้ายกับวิธีถุงคำ ต่างกันตรงที่การแทนคำของทีเอฟไอดีเอฟ ใช้ความถี่ของคำในข้อความคูณกับค่าผกผันของความถี่ของคำ โดยกำหนดให้ tf คือ ความถี่ของคำที่ปรากฏในข้อความ N คือจำนวนข้อความทั้งหมดในชุดข้อมูล $\boldsymbol{n_t}$ คือจำนวนข้อความที่มีคำนั้นๆ ค่า tfidf สามารถคำนวณได้ตามสมการที่ (1) ส่วนค่า idf สามารถคำนวณได้ตามสมการที่ (2)

$$\text{tfidf} = \text{tf x idf} \tag{1}$$

$$\text{idf} = \log\frac{N}{n_t} \tag{2}$$

จากข้อความตัวอย่าง สามารถคำนวณค่า idf ของแต่ละคำได้ดังนี้ [0 0.3 0 0.3 0.3 0.3] โดยสามารถแทนข้อความตัวอย่างได้ดังนี้

1) “ชั้นชอบหมา หมาชั้นน่ารัก” แทนข้อความด้วย [0 0.3 0 0.3 0 0]
2) “หมาของชั้นอ้วน” แทนข้อความด้วย [0 0 0 0 0.3 0.3]

2.1.1.3. เวกเตอร์วันฮอต (One-hot Vector)

เป็นวิธีการแทนข้อความด้วยกลุ่มของเวกเตอร์ที่เรียงลำดับคำตามข้อความ โดยกำหนดให้คำแต่ละคำแทนด้วยเวกเตอร์ที่มีขนาดเท่ากับจำนวนคำทั้งหมดในพจนานุกรมของคำ ค่าของคำที่ปรากฏมีค่าเท่ากับ 1 ส่วนคำที่ไม่ได้ปรากฏมีค่าเป็น 0 จากข้อความตัวอย่าง สามารถแทนคำด้วยเวกเตอร์วันฮอตได้ดังนี้

$$\text{“ชั้น”} = \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix}, \text{“ชอบ”} = \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}, \text{“หมา”} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}, \text{“น่ารัก”} = \begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix}, \text{“ของ”} = \begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix}, \text{“อ้วน”} = \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}$$

จากข้อความตัวอย่าง สามารถแทนข้อความตัวอย่างได้ดังนี้

1) “ชั้นชอบหมา หมาชั้นน่ารัก” แทนข้อความด้วย $\begin{bmatrix}1&0&0&0&1&0\\0&1&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&1\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}$

2) “หมาของชั้นอ้วน” แทนข้อความด้วย $\begin{bmatrix}0&0&1&0\\0&0&0&0\\1&0&0&0\\0&0&0&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$

2.1.1.4. คำฝังตัว (Word Embedding)

เป็นวิธีการแทนข้อความด้วยกลุ่มของเวกเตอร์ที่เรียงลำดับคำตามข้อความ โดยการนำข้อความทั้งหมดไปประมวลผลก่อน ขนาดของเวกเตอร์คำจะขึ้นอยู่กับการกำหนดขนาดมิติในตอนประมวลผล วิธีที่นิยมมี 2 แบบ ได้แก่ เวิร์ดทูเวก (word2vec) [5] และโกลฟ (GloVe) [6] ค่าของเวกเตอร์คำที่ได้จะแสดงถึงความใกล้เคียงกันของคำแต่ละคำ คำที่มีความหมายใกล้เคียงกัน จะมีค่าของระยะเวกเตอร์ใกล้เคียงกัน จากข้อความตัวอย่าง กำหนดให้เวกเตอร์คำมีขนาดมิติเท่ากับ 4 สามารถแทนคำด้วยเวกเตอร์คำได้ดังนี้

“ชั้น” = $\begin{bmatrix}0.45\\0.12\\0.31\\0.23\end{bmatrix}$, “ชอบ” = $\begin{bmatrix}0.68\\0.22\\0.35\\0.12\end{bmatrix}$, “หมา” = $\begin{bmatrix}0.17\\0.83\\0.16\\0.53\end{bmatrix}$, “น่ารัก” = $\begin{bmatrix}0.28\\0.08\\0.77\\0.62\end{bmatrix}$, “ของ” = $\begin{bmatrix}0.11\\0.16\\0.58\\0.64\end{bmatrix}$,

“อ้วน” = $\begin{bmatrix}0.52\\0.24\\0.82\\0.17\end{bmatrix}$

1) “ชั้นชอบหมา หมาชั้นน่ารัก” = $\begin{bmatrix}0.45&0.68&0.17&0.17&0.45&0.52\\0.12&0.22&0.83&0.83&0.12&0.24\\0.31&0.35&0.16&0.16&0.31&0.82\\0.23&0.12&0.53&0.53&0.23&0.17\end{bmatrix}$

2) “หมาของชั้นอ้วน” = $\begin{bmatrix}0.17&0.11&0.45&0.52\\0.83&0.16&0.12&0.24\\0.16&0.58&0.31&0.82\\0.53&0.64&0.23&0.17\end{bmatrix}$

2.1.2. นิวรอลเน็ตเวิร์ก (Neural Network)

นิวรอลเน็ตเวิร์กหรือโครงข่ายประสาทเทียมเป็นหนึ่งในเทคนิคของการทำเหมืองข้อมูล (Data mining) โดยมีแรงบันดาลใจมาจากสมองของมนุษย์ เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ด้วยการเรียนรู้จากข้อมูลที่มีอยู่ เพื่อใช้ในการทำนายข้อมูลที่ลักษณะที่คล้ายกัน

2.1.2.1. เพอร์เซ็ปตรอน (Perceptron)

เพอร์เซ็ปตรอนเป็นนิวรอลเน็ตเวิร์กแบบง่ายที่มีเพียงหน่วยเดียว จำลองลักษณะของเซลประสาทของมนุษย์ เพอร์เซ็ปตรอนถือเป็นหน่วยที่เล็กที่สุดของนิวรอลเน็ตเวิร์ก ทำหน้าที่รับข้อมูลรับเข้าเป็นเวกเตอร์เข้ามาแล้วคำนวณหาผลรวมเชิงเส้น (linear combination) แบบถ่วงน้ำหนักของข้อมูลรับเข้า และให้ข้อมูลส่งออกเป็นค่าคงที่ที่แตกต่างกันตามฟังก์ชันกระตุ้น (Activation Function) กำหนดให้ $\boldsymbol{x_i}$ คือข้อมูลรับเข้า, $\boldsymbol{w_i}$ คือน้ำหนัก, b คือค่าไบแอส (bias), z คือผลลัพธ์ของผลรวมเชิงเส้น, $\mathbf{g()}$ คือฟังก์ชันกระตุ้น และ o คือผลลัพธ์ที่ได้จากเพอร์เซ็ปตรอน การคำนวณเพอร์เซ็ปตรอนแสดงโดยสมการที่ (3) และ (4)

$$z = \sum_{i=0}^{n} w_i x_i + b \tag{3}$$

$$\mathrm{o} = \mathrm{g(z)} \tag{4}$$

สำหรับขั้นตอนการเรียนรู้เพอร์เซ็ปตรอน กำหนดให้ t คือข้อมูลส่งออกที่ถูกต้อง, o คือผลลัพธ์ที่ได้จากเพอร์เซ็ปตรอน, $\boldsymbol{\eta}$ คืออัตราการเรียนรู้ (learning rate) การเรียนรู้แสดงโดยสมการที่ (5) และ (6)

$$w_i = w_i + \Delta w_i \tag{5}$$

โดยที่

$$\Delta w_i = \eta(\mathrm{t} - \mathrm{o}) x_i \tag{6}$$

2.1.2.2. ฟังก์ชันกระตุ้น (Activation Function)

ฟังก์ชันกระตุ้นคือฟังก์ชันที่ช่วยให้ค่าส่งออกของเพอร์เซ็ปตรอนมีค่าที่แตกต่างกันไปตามฟังก์ชันกระตุ้น การเลือกใช้จะขึ้นอยู่กับค่าผลลัพธ์ที่ต้องการ

ตารางที่ 1 แสดงค่าฟังก์ชันกระตุ้น

ฟังก์ชัน	**ค่าผลลัพธ์**	**สมการฟังก์ชันกระตุ้น**
ฟังก์ชันซิกมอยด์ (Sigmoid Function)	0 ถึง 1	$\mathrm{g(z)} = \sigma(\mathrm{z}) = \frac{1}{1+e^{-z}}$ (7)
ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function)	0 ถึง 1	$\mathrm{g}(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{K} e^{z_i}}$ (8)
ฟังก์ชันเรคติไฟต์เชิงเส้น (Rectified Linear Unit Function หรือ ReLU)	0 หรือ จำนวนบวก	$\mathrm{g(z)} = \begin{cases} 0 \; if \; z < 0 \\ z \; if \; z \geq 0 \end{cases}$ (9)
ฟังก์ชันขีดแบ่ง (Threshold Function)	0 หรือ ค่า z	$\mathrm{g(z,t)} = \begin{cases} 0 \; if \; z < t \\ z \; if \; z \geq t \end{cases}$ (10)

2.1.2.3. นิวรอลเน็ตเวิร์กแบบป้อนไปข้างหน้า (Feedforward Neural Network)

เป็นนิวรอลเน็ตเวิร์กที่มีการคำนวณและส่งข้อมูลรับเข้าไปยังข้อมูลส่งออกในทิศทางเดียวกัน ข้อมูลจะถูกส่งไปเป็นลำดับชั้น โดยเพอร์เซ็ปตรอนในชั้นเดียวกันจะไม่เชื่อมต่อกัน แต่จะเชื่อมกับเพอร์เซ็ปตรอนในชั้นถัดไป และข้อมูลส่งออกของเพอร์เซ็ปตรอนชั้นที่แล้ว จะนำมาเป็นข้อมูลรับเข้าของเพอร์เซ็ปตรอนในชั้นถัดไป ขั้นตอนในการคำนวณข้อมูลรับเข้าของเพอร์เซ็ปตรอนในชั้นถัดไป กำหนดให้ $\boldsymbol{a_k^{l-1}}$ คือผลลัพธ์ที่ได้จากเพอร์เซ็ปตรอนตัวที่ k

ในลำดับชั้น l-1, $\boldsymbol{w_{jk}^{l}}$ คือค่าน้ำหนักของเพอร์เซ็ปตรอนตัวที่ j ในลำดับชั้นที่ l ที่มีเส้นเชื่อมมาจากเพอร์เซ็ปตรอนตัวที่ k, $\boldsymbol{b_{j}^{l}}$ คือค่าไบแอสของเพอร์เซ็ปตรอนตัวที่ j ในลำดับชั้นที่ l และ $\boldsymbol{g}$ คือฟังก์ชันกระตุ้น การคำนวณนิวรอลเน็ตเวิร์กแบบป้อนไปข้างหน้าแสดงโดยสมการที่ (11) และ (12)

$$z_j^l = \sum_{k=1}^{n} w_{jk}^l a_k^{l-1} + b_j^l \tag{11}$$

โดยที่

$$a_j^l = g(z_j^l) \tag{12}$$

2.1.2.4. การแพร่กระจายย้อนกลับและการเรียนรู้ (Back propagation and Training)

การทำการแพร่กระจายย้อนกลับถูกนำมาแก้ไขปัญหาของนิวรอลเน็ตเวิร์กแบบป้อนไปข้างหน้า เนื่องจากการทำนิวรอลเน็ตเวิร์กแบบป้อนไปข้างหน้าสามารถหาค่าความผิดพลาดได้แค่ในลำดับชั้นสุดท้าย แต่ไม่สามารถหาค่าความผิดพลาดในชั้นก่อนหน้าได้ จึงต้องใช้วิธีการแพร่กระจายย้อนกลับ กำหนดให้ $\boldsymbol{\delta_j^l}$ คือ ค่าความคลาดเคลื่อนของเพอร์เซ็ปตรอนตัวที่ j ในลำดับชั้น l, j คือฟังก์ชันกระตุ้น, z คือค่าที่คำนวณได้ก่อนจะผ่านฟังก์ชันกระตุ้น g แสดงโดยสมการที่ (13)

$$\delta_j^l = \frac{\partial_j}{\partial z_j^l} = \frac{\partial_j}{\partial a_j^l}\frac{\partial a_j^l}{\partial z_j^i} = \frac{\partial_j}{\partial a_j^l} g'(z_j^l) \tag{13}$$

ลำดับชั้นที่ l + 1 แสดงโดยสมการที่ (14)

$$\frac{\partial_j}{\partial a_j^l} = \sum_{k=1}^{m} \frac{\partial_j}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial a_j^l} = \sum_{k=1}^{m} \delta_k^{l+1} w_{kj}^{l+1} \tag{14}$$

เมื่อคำนวณค่าความผิดพลาดของแต่ละลำดับชั้นได้ ก็สามารถหาค่าความผิดพลาดเทียบกับค่าน้ำหนักและไบแอสได้ แสดงโดยสมการที่ (15) และ (16)

$$\frac{\partial_j}{\partial w_{jk}^l} = \frac{\partial_j}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l a_k^{l-1} \tag{15}$$

$$\frac{\partial_j}{\partial b_j^l} = \frac{\partial_j}{\partial z_j^l}\frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l \tag{16}$$

2.1.2.5. ฟังก์ชันต้นทุน (Cost function หรือ Loss function หรือ Objective function)

เป็นฟังก์ชันของนิวรอลเน็ตเวิร์กที่แสดงถึงต้นทุนของนิวรอลเน็ตเวิร์ก เป้าหมายของการเรียนรู้ของนิวรอลเน็ตเวิร์กคือการปรับค่าน้ำหนักเพื่อลดค่าฟังก์ชันต้นทุนให้เหลือน้อยที่สุด ฟังก์ชันต้นทุนที่นิยมมีรายละเอียดดังนี้ กำหนดให้ J คือฟังชันก์ต้นทุน, n คือจำนวนชุดข้อมูลที่ใช้ในการเรียนรู้, $\boldsymbol{y_i}$ คือข้อมูลส่งออกที่ถูกต้อง, $\boldsymbol{\hat{y}_i}$ คือผลลัพธ์ข้อมูลส่งออกที่ทำนายได้

1) ค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean Squared Error หรือ MSE)

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (17)$$

2) ค่าเฉลี่ยครอสเอนโทรปีแบบทวิภาค (Binary Cross-entropy)

$$J = \frac{1}{n}\sum_{i=1}^{n}(y_i \log \hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \quad (18)$$

2.1.2.6. การหาค่าเหมาะที่สุด (Optimization)

เป็นการปรับน้ำหนักของเส้นเชื่อมในนิวรอลเน็ตเวิร์ก เพื่อลดค่าฟังก์ชันต้นทุนให้เหลือน้อยที่สุด การหาค่าเหมาะที่สุดที่นิยมมีรายละเอียดดังนี้

1) เกรเดียนต์แบบปรับตัวได้ (Adaptive Gradient Descent หรือ AdaGrad) กำหนดให้ g_t คือ เกรเดียนต์ที่เวลา t, $\frac{\partial_{Jt}}{\partial_w}$ คือเกรเดียนต์ของฟังก์ชันต้นทุนเทียบกับ w, w คือน้ำหนักที่เวลา t, η คืออัตราการเรียนรู้

$$g_t = \frac{\partial_{Jt}}{\partial_w} \quad (19)$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\sum_{k=1}^{t} g_i^2}} g_t \quad (20)$$

2) อาร์เอ็มเอสพรอพ (RMSProp) กำหนดให้ g_t คือเกรเดียนต์ที่เวลา t, $MeanSquare_t$ คือ ค่าเฉลี่ยของเกรเดียนต์, γ คืออัตราการใช้เกรเดียนต์ของอดีตในการเรียนรู้นิยมกำหนดให้เป็น 0.9, w คือน้ำหนักที่เวลา t, η คืออัตราการเรียนรู้

$$g_t = \frac{\partial_{Jt}}{\partial_w} \quad (21)$$

$$MeanSquare_t = \gamma MeanSquare_{t-1} - (1 - \gamma) g_t^2 \quad (22)$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{MeanSquare_t}} g_t$$

3) การประมาณโมเมนต์แบบปรับตัวได้ (Adaptive Moment Estimation หรือ Adam) กำหนดให้ g_t คือเกรเดียนต์ที่เวลา t, β_1, β_2 คืออัตราการใช้เกรเดียนต์ของอดีตในการเรียนรู้ นิยมกำหนดให้เป็น 0.9 และ 0.999 ตามลำดับ, w คือน้ำหนักที่เวลา t, η คืออัตราการเรียนรู้, ϵ คือค่าที่น้อยมากเพื่อป้องกันการหารด้วย 0

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (23)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (24)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (25)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (26)$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \quad (27)$$

2.1.3. นิวรอลเน็ตเวิร์กคอนโวลูชัน (Convolutional Neural Network)

นิวรอลเน็ตเวิร์กคอนโวลูชันเริ่มต้นถูกออกแบบมาเพื่อใช้กับงานวิจัยทางด้านการรู้จำภาพตัวอักษร โดยส่วนใหญ่จะใช้ข้อมูลรับเข้าเป็นเมทริกซ์ในรูปแบบต่างๆ ต่อมาได้ถูกนำมาประยุกต์ใช้งานวิจัยด้านต่างๆเช่น รูปภาพ, ข้อความ

2.1.3.1. ชั้นคอนโวลูชัน (Convolutional Layer)

เป็นชั้นที่ใช้สำหรับสร้างฟีเจอร์ใหม่ (Feature Map) จากข้อมูลรับเข้า โดยการใช้วิธีด็อตเมทริกซ์กับตัวกรอง (filter) กำหนดให้ I คือเมทริกซ์ข้อมูลรับเข้า, K คือเมทริกซ์ตัวกรองมีขนาด h x w ผลลัพธ์ของชั้นคอนโวลูชันสามารถคำนวณได้จากสมการ (28)

$$(\mathrm{I} * \mathrm{K}) = \sum_{i=1}^{h}\sum_{j=1}^{w} K_{ij} \, . \, I_{x+i-1,y+j-1} \tag{28}$$

2.1.3.2. ชั้นการรวม (Pooling Layer)

เป็นชั้นที่ใช้สำหรับลดขนาดข้อมูล ให้เหลือเฉพาะข้อมูลที่สำคัญ นิยมนำมาใช้ต่อจากชั้นคอนโวลูชัน วิธีของชั้นการรวมที่นิยมใช้มีดังนี้

1) การรวมชั้นโดยเลือกข้อมูลที่มีค่ามากที่สุด (Max Pooling)
2) การรวมชั้นโดยเฉลี่ยค่าของข้อมูล (Average Pooling)

2.1.3.3. ชั้นการเชื่อมโยงเต็มรูปแบบ (Fully Connected Layer)

ชั้นการเชื่อมโยงเต็มรูปแบบจะเป็นชั้นสุดท้ายของนิวรอลเน็ตเวิร์กคอนโวลูชัน ซึ่งหลังจากที่ผ่านชั้นคอนโวลูชันและผ่านชั้นการรวมมาแล้ว ในชั้นนี้จะประกอบด้วยชั้นย่อยๆที่มีเพอร์เซ็ปตรอนอยู่จำนวนหนึ่ง โดยในแต่ละชั้นจะมีเส้นเชื่อมระหว่างเพอร์เซ็ปตรอนในชั้นก่อนหน้าและชั้นถัดไป ทำให้สามารถคำนวณแบบนิวรอลเน็ตเวิร์กแบบป้อนไปข้างหน้าและการแพร่กระจายย้อนกลับได้

2.1.4. นิวรอลเน็ตเวิร์กแบบวกกลับ (Recurrent Neural Network)

นิวรอลเน็ตเวิร์กแบบวกกลับถูกออกแบบมาเพื่อใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) เช่น วีดิทัศน์ (video), ข้อความ โดยใช้การเรียนรู้ซึ่งพึ่งพาข้อมูลรับเข้าในอดีตในระยะยาว (Long-term Dependencies) ผลลัพธ์ที่ได้จากข้อมูลส่งออก ณ ลำดับที่ t ของชุดข้อมูลใดๆ สามารถคำนวณได้ดังนี้ กำหนดให้ $\boldsymbol{x_t}$ คือข้อมูลรับเข้า ณ ลำดับที่ t ของชุดข้อมูลใดๆ, U คือค่าน้ำหนักของข้อมูลรับเข้า, V คือค่าน้ำหนักวกกลับ (Recurrent Weight), W คือค่าน้ำหนักของข้อมูลส่งออก, $\boldsymbol{h_t}$ คือสถานะซ่อน (Hidden State) ณ ลำดับที่ t ของชุดข้อมูลใดๆ และ $\boldsymbol{o_t}$ คือข้อมูลส่งออก ณ ลำดับที่ t ของชุดข้อมูลใดๆ, $\boldsymbol{f_h}$ คือฟังก์ชันกระตุ้นในขั้นตอนการคำนวณสถานะซ่อน, $\boldsymbol{f_y}$ คือฟังก์ชันกระตุ้นในขั้นตอนการคำนวณข้อมูลส่งออก แสดงโดยสมการที่ (29) และ (30)

$$h_t = f_h(Ux_t + Vh_{t-1}) \tag{29}$$

$$o_t = f_y(Wh_t + b_y) \tag{30}$$

การฝึกสอนนิวรอลเน็ตเวิร์กแบบวกกลับ ใช้วิธีการแพร่กระจายย้อนกลับในการปรับน้ำหนัก ซึ่งอาจทำให้เกิดปัญหา หากข้อมูลรับมีขนาดยาวเกินไป เนื่องจากค่าเกรเดียนต์ของ V เกิดจากการคูณกันของลำดับก่อนหน้า หากค่าเกรเดียนต์มีค่าระหว่าง 0 ถึง 1 อาจส่งผลให้มีค่าเป็นศูนย์ (Vanishing Gradient) หรือถ้าหากมีค่ามากกว่า 1 อาจส่งผลให้มีค่าเพิ่มมากขึ้นเกินไป (Exploding Gradient)

2.1.5. หน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory หรือ LSTM)

นิวรอลเน็ตเวิร์กแบบหน่วยความจำระยะสั้นแบบยาวเป็นนิวรอลเน็ตเวิร์กแบบวกกลับแบบหนึ่ง ถูกออกแบบมาเพื่อแก้ปัญหาการลดหายหรือเพิ่มมากขึ้นของเกรเดียนต์ กำหนดให้ $\boldsymbol{f_t}$ คือประตูลืม (Forget Gate) ถ้าผลลัพธ์มีค่าเท่ากับ 0 จะลบค่าสถานะเซลล์ (Cell State) ก่อนหน้า แต่ถ้ามีค่าเท่ากับ 1 จะเก็บค่าสถานะเซลล์, $\boldsymbol{i_t}$ คือประตูรับเข้า (Input Gate) ทำหน้าที่ในการตัดสินใจว่าจะทำการปรับปรุงค่าหรือไม่, $\boldsymbol{C_t}$ คือค่าที่ใช้ในการปรับปรุงเมื่อประตูรับเข้าตัดสินใจ, $\boldsymbol{o_t}$ คือประตูข้อมูลออก (Output Gate) ทำหน้าที่ควบคุมปริมาณของข้อมูลที่ส่งต่อไปยังการทำงานในลำดับขั้นเวลาถัดไป

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{31}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{32}$$

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{33}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{34}$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{35}$$

$$h_t = o_t * \tanh(C_t) \tag{36}$$

2.1.6. การวัดประสิทธิภาพ (Performance Evaluation)

แบ่งออกเป็นทั้งหมด 2 แบบโดยมีรายละเอียดดังนี้

2.1.6.1. คอนฟิวชันเมทริกซ์ (Confusion Matrix)

คือเมทริกซ์ที่ใช้สำหรับการแจกแจงผลลัพธ์ของการจำแนกตามคลาส

ตารางที่ 2 แสดงคอนฟิวชันเมทริกซ์ของการจำแนกแบบ 3 คลาส

		คลาสที่ทำนาย		
		A	B	C
คลาสจริง	A	$M_{1,1}(TP_A)$	$M_{1,2}$	$M_{1,3}$
	B	$M_{2,1}$	$M_{2,2}(TP_B)$	$M_{2,3}$
	C	$M_{3,1}$	$M_{3,2}$	$M_{3,3}(TP_C)$

ค่าในแต่ละแถวแสดงจำนวนข้อมูลที่อยู่ในแต่ละคลาสจริงๆ ส่วนค่าในแต่ละสดมภ์แสดงจำข้อมูลที่ทำนายได้คลาสนั้นๆ กำหนดให้สำหรับคลาสใดๆ

1) TP คือจำนวนข้อมูลที่ทำนายคลาสได้ตรงกับคลาสจริง (True Positive)

ตัววัดประสิทธิภาพจำแนกตามคลาส มีดังนี้

1) ค่าความเที่ยง (Precision) คือค่าความแม่นยำของแบบจำลองโดยพิจารณาทีละคลาสในแนวสดมภ์ กำหนดให้ i คือลำดับของแถว และ j คือลำดับของสดมภ์

$$\text{Precision} = \frac{M_{i,i}}{M_{i,i} + \sum_{i=1}^{3} M_{i,j}(\text{ยกเว้น } i = j)} \tag{37}$$

2) ค่าความระลึก (Recall) คือค่าความถูกต้องของแบบจำลองโดยการพิจารณาทีละคลาสในแนวแถว (i) กำหนดให้ i คือลำดับของแถว และ j คือลำดับของสดมภ์

$$\text{Recall} = \frac{M_{i,i}}{M_{i,i} + \sum_{j=1}^{3} M_{i,j} \ (\text{ยกเว้น } i = j)} \quad (38)$$

3) ค่าเอฟวัน (F1) คือค่าเฉลี่ยระหว่างค่าความเที่ยงและค่าความระลึก

$$\text{F1} = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (39)$$

4) ค่าความแม่นยำ (Accuracy) คือค่าความแม่นยำของแบบจำลอง กำหนดให้ N คือจำนวนข้อมูลทั้งหมด

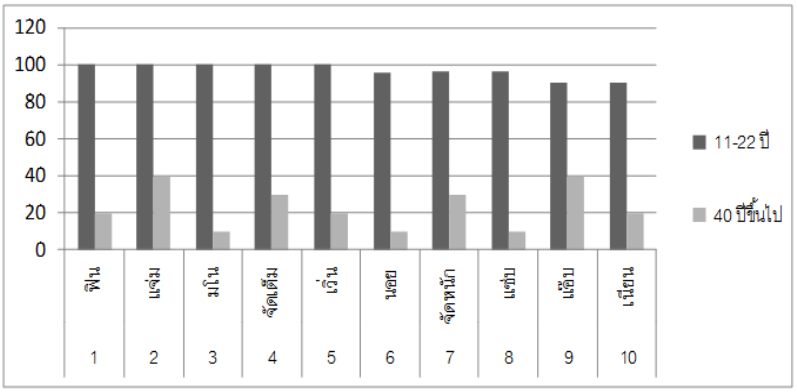
$$\text{Accuracy} = \frac{\sum_{i=1}^{3} TP_i}{N} \quad (40)$$

2.2. งานวิจัยที่เกี่ยวข้อง

ในการศึกษางานวิจัยที่เกี่ยวข้องของงานวิจัยนี้ จะแบ่งออกเป็น 3 โดยมีรายละเอียดดังนี้

2.2.1. งานวิจัยเกี่ยวกับการใช้คำในแต่ละช่วงอายุ

ในวารสารภาษาและวัฒนธรรม ฉบับที่ 35 ฉบับพิเศษ ในหัวข้อเรื่องลักษณะเด่นของคำกริยาในภาษาวัยรุ่นไทย ของ Pholnarat [4] (2016) ได้ทำแบบสอบถามการใช้คำกริยาในภาษาวัยรุ่น ระหว่างกลุ่มวัยรุ่นอายุ 11-22 ปี และกลุ่มวัยผู้ใหญ่อายุ 40 ปีขึ้น จำนวนกลุ่มละ 50 คน



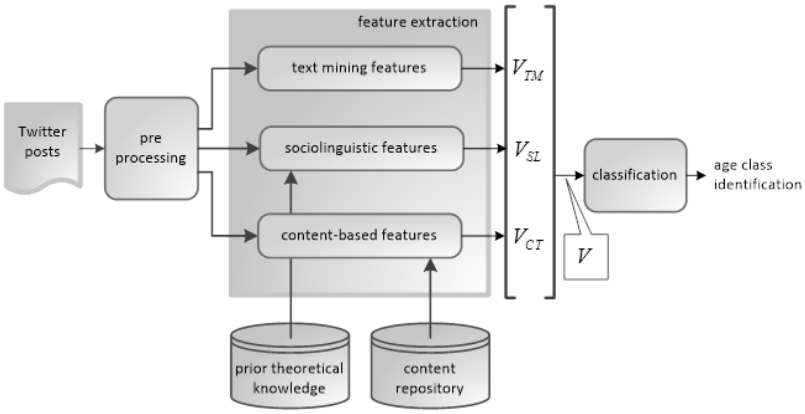
รูปที่ 1 แสดงการใช้คำกริยาจากแบบสอบถามของกลุ่มวัยรุ่นและกลุ่มวัยผู้ใหญ่

(ที่มา : อ้างอิงจากแผนภูมิที่ 1 ใน [4])

จากรูปแสดงให้เห็นว่าคำกริยาที่มีลักษณะเด่นในกลุ่มวัยรุ่น เช่น "มโน", "นอย", "แซ่บ" มีความน่าจะเป็นที่กลุ่มผู้ใหญ่จะนำไปใช้หรือเข้าใจความหมายค่อยข้างน้อย

2.2.2. งานวิจัยเกี่ยวกับการจำแนกอายุของผู้ใช้งานเครือข่ายสังคมออนไลน์

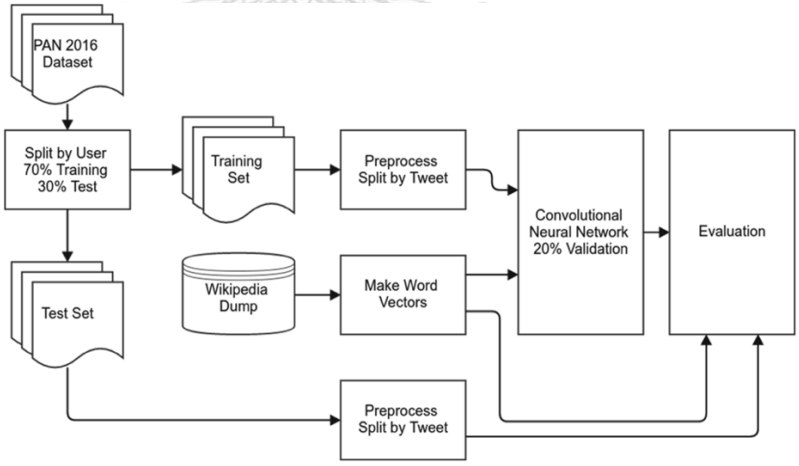
ในงานวิจัยของ Vasiliki Simaki [7] และคณะ (2016) ได้นำเสนอวิธีการเรียนรู้ของเครื่อง (machine learning) ในการจำแนกอายุของผู้ใช้งานทวิตเตอร์ (twitter)



รูปที่ 2 แสดงขั้นตอนการทำงานการจำแนกอายุของผู้ใช้งานทวิตเตอร์ (twitter)
(ที่มา : อ้างอิงจาก Fig. 1 ใน [7])

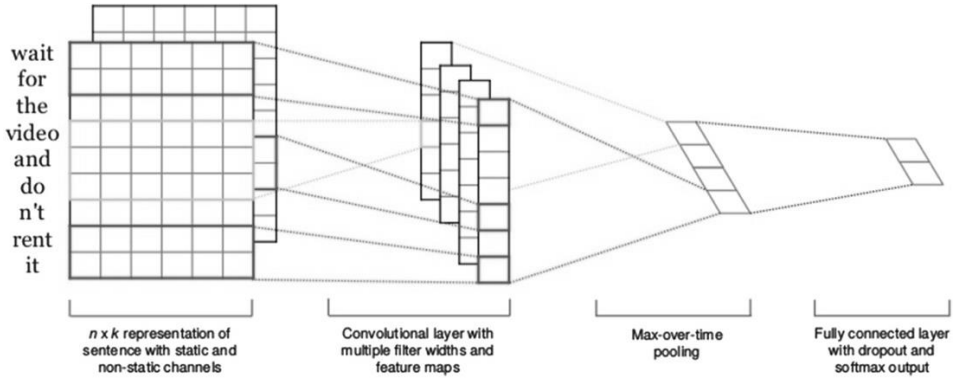
งานวิจัยนี้ใช้การแทนข้อความด้วยเวกเตอร์ที่มีฟีเจอร์ (feature) 3 แบบ แบ่งออกเป็น การวิเคราะห์ข้อความ(text mining) 40 ฟีเจอร์, พื้นฐานภาษาศาตร์ (sociolinguistic-based) 6 ฟีเจอร์ และเนื้อหาที่เกี่ยวข้อง (content-related) 3 ฟีเจอร์ จากนั่นนำไปจำแนกด้วยวิธีการเรียนรู้ของเครื่อง โดยกำหนดป้าย (label) ของการจำแนกอายุไว้ทั้งหมด 6 ช่วงอายุ ดังนี้ อายุ 14-19, 20-24, 25-34, 35-44, 45-59 และ มากกว่า 60 ขึ้นไป วิธีที่ได้ค่าความแม่นยำมากที่สุดคือ วิธีป่าสุ่ม (RandomForest) ซึ่งได้ค่าความแม่นยำเท่ากับ 61%

ในงานวิจัยของ Roy Khristopher Bayot [2] และคณะ (2018) ได้นำเสนอการใช้นิวรอลเน็ตเวิร์กคอนโวลูชันในการจำแนกอายุและเพศของผู้ใช้งานทวิตเตอร์ (twitter) โดยใช้ชุดข้อมูลจาก PAN 2016 Author Profiling [8] ซึ่งเป็นข้อมูลของผู้ใช้งานประเทศอังกฤษและประเทศสเปน



รูปที่ 3 แสดงขั้นตอนการทำงานการจำแนกอายุและเพศของผู้ใช้งานประเทศอังกฤษและสเปนจากข้อมูลทวิตเตอร์
(ที่มา : อ้างอิงจาก Fig. 1 ใน [2])

งานวิจัยนี้ใช้การแทนข้อความด้วยคำฝังตัว โดยใช้ข้อมูลจากวิกิพีเดีย (Wikipedia) ภาษาอังกฤษและสเปน นำไปประมวลผลด้วยเวิร์ดทูเวก โดยกำหนดขนาดมิติของเวกเตอร์คำเท่ากับ 100 และ 300 จากนั้นนำชุดข้อมูล มาแปลงเป็นเมทริกซ์ของเวกเตอร์คำตามลำดับคำของข้อมูล แล้วนำข้อมูลเมทริกซ์มาผ่านนิวรอลเน็ตเวิร์กคอนโวลูชันที่ถูกเสนอโดย Y. Kim [9]



รูปที่ 4 แสดงขั้นตอนการทำงานนิวรอลเน็ตเวิร์กคอนโวลูชันของ Y. Kim

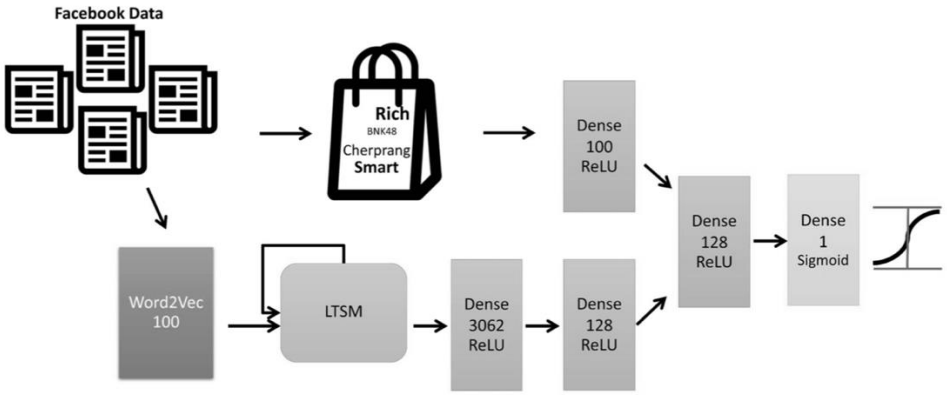
(ที่มา : อ้างอิงจาก Fig. 2 ใน [2])

โดยกำหนดป้าย (label) ของการจำแนกอายุไว้ทั้งหมด 5 ช่วงอายุ ดังนี้ อายุ 18-24, 25-34, 35-49, 50-64 และมากกว่า 65 ขึ้นไป และกำหนดป้าย (label) ของการจำแนกเพศเป็นเพศชายและเพศหญิง ผลจากงานวิจัยนี้ได้ค่าความแม่นยำมากกว่าแบบจำลองซัพพอร์ตเวกเตอร์แมชชีนโดยใช้วิธีการแทนคำด้วยทีเอฟไอดีเอฟ

2.2.3. งานวิจัยเกี่ยวกับการจำแนกข้อความภาษาไทย

ในงานวิจัยของ Thanabhat Koomsubha [10] (2016) ได้นำเสนอการใช้นิวรอลเน็ตเวิร์กคอนโวลูชันระดับตัวอักษรในการจำแนกประเภทข้อความภาษาไทย โดยใช้ข้อมูลข่าวจากหนังสือพิมพ์ของสำนักต่างๆของประเทศไทย งานวิจัยนี้ใช้การแทนข้อความด้วยเวกเตอร์วันฮอตแบบตัวอักษร โดยตัวอักษรที่ถูกนำมาแปลงมีทั้งหมด 151 ตัวอักษร แบ่งออกเป็น ตัวอักษรภาษาไทย 81 ตัวอักษร, ตัวอักษรภาษาอังกฤษและอักขระพิเศษ 70 ตัวอักษร จากนั้นนำข้อมูลมาผ่านนิวรอลเน็ตเวิร์กคอนโวลูชัน ผลจากงานวิจัยนี้ได้ค่าความแม่นยำในการจำแนกข้อความมากกว่านิวรอลเน็ตเวิร์กระดับตัวอักษรแบบดั้งเดิม นอกจากนี้ยังได้ผลลัพธ์ที่ดีกว่าวิธีแบบตัดคำ เช่น ซัพพอร์ตเวกเตอร์แมชชีน นาอีฟเบย์ เป็นต้น

ในงานวิจัยของ Phasit Charoenkwan [3] (2018) ได้นำเสนอวิธีการใช้การเรียนรู้เชิงลึกกับถุงคำในการจำแนกความรู้สึกของข้อมูลเฟซบุ๊กภาษาไทย โดยใช้ข้อมูลการโพสต์ของเฟซบุ๊กเพจ งานวิจัยนี้ได้เสนอการนำเน็ตเวิร์กของการเรียนรู้เชิงลึกแบบหน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory หรือ LSTM) โดยใช้การแทนข้อความด้วยคำฝังตัว โดยนำไปประมวลผลด้วยเวิร์ดทูเวก มารวมกับเน็ตเวิร์กของเพอร์เซ็ปตรอนหลายชั้นโดยใช้การแทนข้อความด้วยถุงคำ จากนั้นนำทั้ง2เน็ตเวิร์กมารวมกันด้วยชั้นการเชื่อมโยงเต็มรูปแบบ (Fully Connected Layer) ผลจากงานวิจัยนี้ได้ค่าความแม่นยำในการจำแนกมากกว่าการใช้การเรียนรู้เชิงลึกแบบหน่วยความจำระยะสั้นแบบยาวหรือการใช้เพอร์เซ็ปตรอนหลายชั้นเพียงอย่างเดียว



รูปที่ 5 ขั้นตอนในการจำแนกความรู้สึกของข้อมูลเฟซบุ๊กภาษาไทยโดยใช้การเรียนรู้เชิงลึกกับถุงคำ (ที่มา : อ้างอิงจาก Fig. 3 ใน [3])

บทที่ 3
แนวคิดและวิธีการวิจัย

จากงานวิจัยที่เกี่ยวข้องการจำแนกรุ่นอายุส่วนใหญ่มักจะใช้การเรียนรู้ด้วยเครื่องและการเรียนรู้เชิงลึกในการจำแนกช่วงอายุ ของผู้ใช้งาน และได้มีการนำถุงคำมาใช้ร่วมกับการเรียนรู้เชิงลึกเพื่อเพิ่มประสิทธิภาพของแบบจำลอง จากการทำแบบสอบถามการใช้คำเราสามารถใช้จำนวนการใช้คำมาช่วยในการจำแนกกลุ่มวัยรุ่นและกลุ่มวัยผู้ใหญ่ได้ ผู้วิจัยจึงมีแนวคิดที่จะใช้การเรียนรู้ด้วยเครื่องและการเรียนรู้เชิงลึกร่วมกับการใช้ฟีเจอร์ต่างๆเช่น ถุงคำ, ทีเอฟไอดีเอฟ, ความน่าจะเป็นของคำในแต่ละรุ่นอายุ เป็นต้น เพื่อเพิ่มประสิทธิภาพของแบบจำลอง

3.1. ขั้นตอนการวิจัย

3.1.1. การเตรียมข้อมูล

เก็บข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กในประเทศไทย โดยทำการรวบรวมข้อมูลจากการโพสต์ของผู้ใช้งาน 300 คน จำนวนคนละ 30 โพสต์ รวมทั้งสิ้น 9,000 โพสต์ การแบ่งกลุ่มผู้ใช้งานได้นำการแบ่งกลุ่มลักษณะเด่นของคำกริยาในภาษาวัยรุ่นไทยของ Pholnarat [4] (2016) มาประยุกต์ใช้ ในการทดลองนี้จะแบ่งผู้ใช้งานออกเป็น 3 กลุ่ม กลุ่มละ 100 คน โดยการแบ่งกลุ่มผู้ใช้งานจะแบ่งตามรุ่นอายุของผู้ใช้งาน ดังนี้ [11]

- รุ่นอายุเอกซ์คือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2508 – 2522
- รุ่นอายุวายคือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2523 – 2544
- รุ่นอายุแซดคือผู้ใช้งานที่เกิดระหว่าง พ.ศ. 2545 – 2563

3.1.2. การประมวลผลก่อน

การทำการประมวลผลก่อนจะเริ่มจากการตัดคำ โดยทั่วไปการตัดคำของภาษาต่างๆเช่น อังกฤษ เยอรมัน สเปน จะใช้การตัดคำจากช่องว่างของคำศัพท์ในแต่ละคำ แต่ในภาษาไทยการตัดคำคือส่วนที่ค่อนข้างยาก เนื่องจากภาษาไทยไม่เหมือนภาษาอื่นตรงที่ไม่มีการเว้นคำด้วยช่องว่าง (space) หรือใช้มหัพภาค (full stops) เพื่อจบประโยค ในงานวิจัยนี้ได้ทำการทดลองใช้โปรแกรมในการตัดคำไทย 2 โปรแกรม คือ Deepcut, PyThaiNLP ต่อมาหลักจากตัดคำแล้ว จะนำคำที่ตัดแล้วมาลบ เครื่องหมายวรรคตอน (punctuation) ออก และทำการลบคำหยุด (stopword) ออก โดยใช้คำที่ได้จากโปรแกรม Deepcut และ PyThaiNLP รวมกันเพื่อใช้ในการลบคำหยุดออก ในงานวิจัยนี้ไม่ได้ลบสัญรูปอารมณ์ (emotion) ออก เนื่องจากต้องการทดลองว่าสัญรูปอารมณ์มีผลต่อการจำแนกรุ่นอายุหรือไม่

3.1.3. การใช้คำฝังตัว

สำหรับการแปลงข้อความให้เป็นข้อมูลเมทริกซ์ในการงานวิจัยนี้ใช้คำฝังตัว โดยการประมวลด้วยเวิร์ดทูเวก โดยรวบรวมข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในประเทศไทย จำนวน 130,000 ทวีต จากนั้นใช้โปรแกรม Deepcut, PyThaiNLP ในการตัดคำ ต่อมานำคำที่ตัดแล้วมาลบคำไม่สำคัญออก โดยใช้คำจากโปรแกรม Deepcut และ PyThaiNLP แล้วลบเครื่องหมายวรรคตอนออก แล้วนำมาประมวลผลด้วยเวิร์ดทูเวก เพื่อสร้างคลังข้อมูลคำไทย จากการรวบรวมข้อมูลได้คำมาทั้งหมด 147,611 คำ

3.1.4. ความน่าจะเป็นของคำในแต่ละรุ่นอายุ

ในงานวิจัยนี้ได้นำความน่าจะเป็นของคำในแต่ละรุ่นอายุมาช่วย เพื่อเพิ่มประสิทธิภาพในการจำแนกรุ่นอายุ การคำนวณความน่าจะเป็นสามารถคำนวณได้ดังนี้ กำหนดให้ WC1 คือจำนวนของคำหนึ่งคำที่พบในรุ่นอายุเอกซ์, WC2 คือจำนวนของคำหนึ่งคำที่พบในรุ่นวาย, WC3 คือจำนวนของคำหนึ่งคำที่พบในรุ่นแซด, AW คือผลรวมของจำนวนของคำหนึ่งคำ

$$\mathrm{AW} = \mathrm{WC1} + \mathrm{WC2} + \mathrm{WC3} \quad (41)$$

$$\text{เวกเตอร์[คำ]} = [\frac{WC1}{AW}, \frac{WC2}{AW}, \frac{WC3}{AW}] \quad (42)$$

ตัวอย่าง เช่น คำว่า "สอบ" พบในรุ่นอายุเอกซ์ ทั้งหมด 2 ครั้ง, พบในรุ่นอายุวาย ทั้งหมด 8 ครั้ง, พบในรุ่นอายุแซด ทั้งหมด 10 ครั้ง สามารถคำนวณความน่าจะเป็นได้ดังนี้

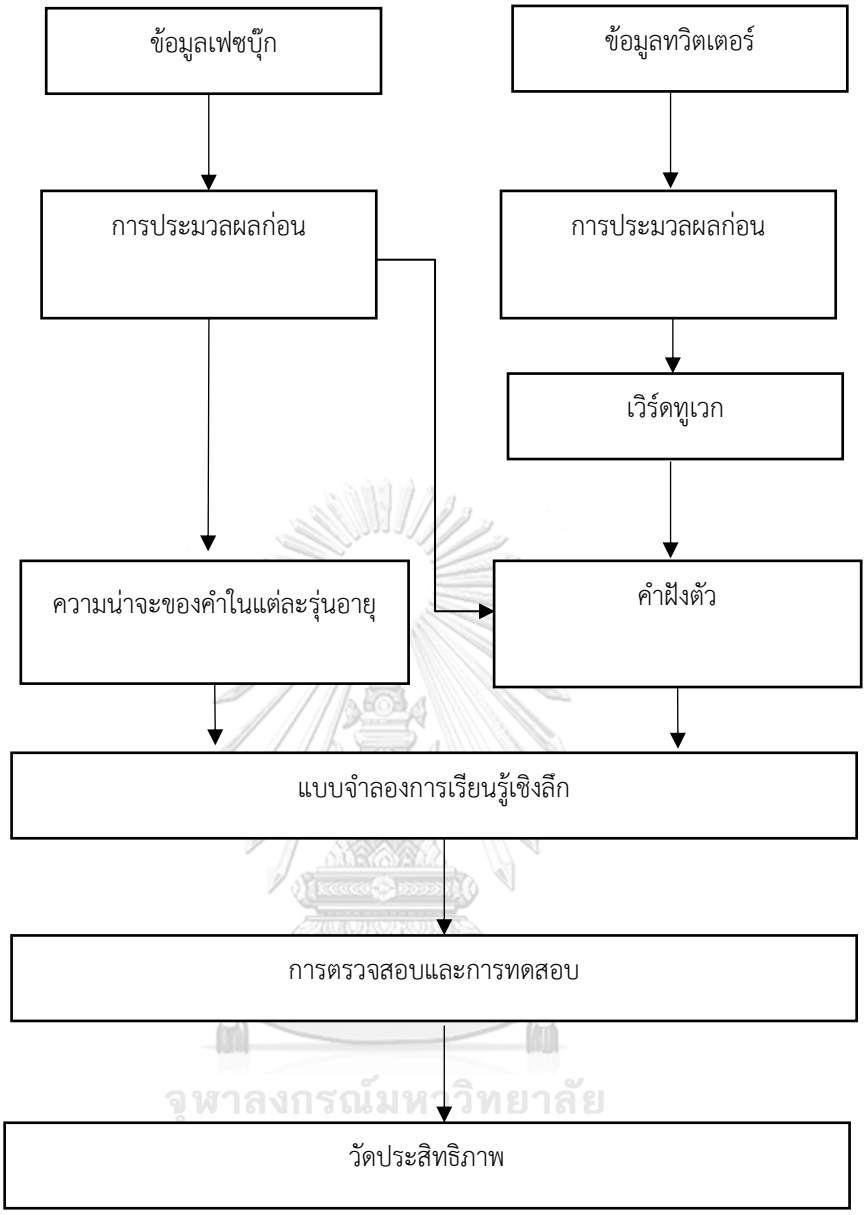
WC1 = 2, WC2 = 8, WC3 = 10

AW = 2+8+10 = 20

เวกเตอร์["สอบ"] = $[\frac{2}{20}, \frac{8}{20}, \frac{10}{20}]$ = [0.1, 0.4, 0.5]

3.1.5. แบบจำลอง

ในงานวิจัยนี้ได้เสนอวิธีการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กไทย โดยกำหนดให้ข้อมูลมี 3 ป้าย ได้แก่ รุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด โดยใช้แบบจำลองการเรียนรู้เชิงลึกร่วมกับข้อมูลฟีเจอร์ต่างๆ ได้แก่ ถุงคำ, ทีเอฟไอดีเอฟ, ความน่าจะเป็นของคำในแต่ละรุ่นอายุ



รูปที่ 6 แสดงแบบจำลองการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กไทย

จากรูปที่ 6 แสดงแบบจำลองการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กไทย โดยเริ่มจากการรวบรวมข้อมูลทวิตเตอร์จำนวน 130,000 ทวีต มาทำการประมวลผลก่อน เพื่อลบคำหยุดและเครื่องหมายวรรคตอน จากนั้นนำมาทำคำฝังตัวด้วยการประมวลผลด้วยเวิร์ดทูเวก เพื่อสร้างเป็นคลังข้อมูลคำไทย ต่อมารวบรวมข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กในประเทศไทยจำนวน 300 คน คนละ 30 โพสต์ จากนั้นทำการประมวลผลก่อน โดยการลบคำหยุดและเครื่องหมายวรรคตอนออก จากนั้นนำข้อมูลเฟซบุ๊กที่ผ่านการทำการประมวลผลก่อนแล้ว มาแปลงข้อความเป็นเมทริกซ์ โดยวิธีการแปลงมี 2 วิธี ได้แก่

1) การแปลงข้อความด้วยการใช้คำฝังตัว โดยการประมวลผลแบบเวิร์ดทูเวก จากคลังข้อมูลคำไทย

2) การแปลงข้อความด้วยการใช้ความน่าจะเป็นของคำในแต่ละรุ่นอายุ

จากนั้นนำข้อมูลทั้ง 2 อย่างมาเป็นข้อมูลรับเข้าของแบบจำลองการเรียนรู้เชิงลึก การทำการตรวจสอบ (Validation) งานวิจัยนี้เลือกใช้วิธี 10 fold cross validation โดยมีการแบ่งข้อมูลเฟซบุ๊กออกเป็น 2 ส่วน ดังนี้

ส่วนแรก 90% จากนั้นนำมาแบ่งออกเป็น 80% ของข้อมูลส่วนแรก จำนวน 6,480 โพสต์ เพื่อใช้เป็นชุดข้อมูลเรียนรู้ (Training set) และ 20% ของข้อมูลส่วนแรก จำนวน 1,620 โพสต์ ใช้เป็นชุดข้อมูลตรวจสอบ (Validation set) ส่วนที่สอง 10% จำนวน 900 โพสต์ ใช้เป็นชุดข้อมูลทดสอบ (Test set)

3.1.6. การประเมินผล

ในการประเมินผลของแบบจำลองจะแบ่งออกเป็น 2 วิธี ดังนี้ วิธีที่หนึ่งประเมินผลแบบต่อโพสต์ (Per Status) โดยนำผลลัพธ์ที่ได้จากแบบจำลองที่อยู่ในรูปแบบของ [%GenX ,% GenY ,% GenZ] โดยเราจะเลือกรุ่นอายุที่ได้ค่าเปอร์เซ็นต์มากที่สุดมาเป็นคำตอบ ส่วนวิธีที่สองประเมินผลแบบต่อผู้ใช้ (Per User) วิธีนี้จะนำผลลัพธ์ที่ได้จากแบบจำลองมารวมค่ากันโดยจัดกลุ่มตามผู้ใช้ เพื่อหาผลรวมเปอร์เซ็นต์ของรุ่นอายุที่ได้ค่ามากที่สุดมาเป็นคำตอบของแต่ละผู้ใช้

ในการวัดประสิทธิภาพของแบบจำลอง เราจะนำผลลัพธ์ที่ได้จากการหาค่าความแม่นยำสูงสุดจากการปรับค่าพารามิเตอร์ที่เกี่ยวข้อง มาวัดประสิทธิภาพ โดยแบ่งออกเป็น 2 วิธีดังนี้

1) คอนฟิวชันเมทริกซ์
2) ตัววัดประสิทธิภาพจำแนกตามคลาส ได้แก่ ค่าความเที่ยง ,ค่าความระลึก ,ค่าเอฟวัน ,ค่าความแม่นยำ

บทที่ 4
การประเมินผลการวิจัย

ในงานวิจัยนี้ได้ทำการทดลองโมเดลทั้งหมด 3 แบบ ได้แก่โมเดลเพอร์เซ็ปตรอนหลายชั้น, นิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว เพื่อทำการเปรียบเทียบผลลัพธ์ของประสิทธิภาพของแต่ละโมเดล

4.1. ระบบที่ใช้ในการทดลอง

ระบบที่ใช้ในการทดลองมีดังนี้

4.1.1. คอมพิวเตอร์ที่ใช้ทำการทดลอง

การทดลองของงานวิจัยนี้ ทำงานบนระบบ google colab มีหน่วยประมวลผลกลาง Intel Xeon ความเร็ว 2.2 Ghz มีหน่วยความจำขนาด 13 GB มีหน่วยประมวลกราฟฟิกคือ Tesla K80 และมีหน่วยความจำกราฟฟิกขนาด 12 GB

4.1.2. การเขียนโปรแกรม

การเขียนโปรแกรมของงานวิจัยนี้ทำโดยใช้ภาษา Python โดยโปรแกรมของโมเดลเพอร์เซ็ปตรอนหลายชั้น ใช้ไลบรารี scikit learn ส่วนโปรแกรมของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาว ใช้ไลบรารี Keras

4.2. ข้อมูลที่ใช้ในการทดลอง

4.2.1. ข้อมูลสำหรับสร้างแบบจำลอง

ข้อมูลที่ใช้สำหรับสร้างแบบจำลอง เป็นข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กในประเทศไทย โดยทำการรวบรวมข้อมูลจากการโพสต์ของผู้ใช้งาน 300 คน จำนวนคนละ 30 โพสต์ รวมทั้งสิ้น 9,000 โพสต์ โดยแบ่งกลุ่มผู้ใช้งานออกเป็น 3 กลุ่มรุ่นอายุ

ตารางที่ 3 แสดงตัวอย่างข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กในประเทศไทย แยกตามกลุ่มรุ่นอายุ

รุ่นอายุ	ตัวอย่างข้อมูลการโพสต์
รุ่นอายุเอกซ์	ยังพักผ่อนต่ออีกวัน 🎉
รุ่นอายุเอกซ์	งานน้อยคงต้องหาคอนแทคเพิ่มขึ้นปีนี้เพราะอุปกรณ์ที่มีอยู่คือต้นทุนต้องเอาทุนคืนบ้างละ
รุ่นอายุวาย	คนท้องทำไมยังสวย 😂
รุ่นอายุวาย	อู ดาฟ จะแต่งแล้วจ้าาา ตื่นเต้นๆๆ 👰👨❤️
รุ่นอายุแซด	ทีมอยากมีที่เรียนดีดีกับเขาสักที
รุ่นอายุแซด	วันเด็กของเราและพี่ชาย

4.2.2. ข้อมูลสำหรับสร้างคลังข้อมูลคำไทย

ข้อมูลที่ใช้สำหรับสร้างคลังข้อมูลคำไทย เป็นข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในประเทศไทย จำนวน 130,000 ทวีต เพื่อนำมาทำคำฝังตัวโดยการประมวลผลด้วยเวิร์ดทูเวก เพื่อนำมาสร้างเป็นคลังข้อมูลคำไทย

ตารางที่ 4 แสดงตัวอย่างข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในประเทศไทย

ตัวอย่างข้อมูลการทวีต
ความเท่ของจอ มีไฟด้วย ตอน notification มา #oneplus7series
คนไทยแห่เที่ยวนอก เหตุเงินบาทแข็ง ค่าทัวร์ถูก - หนีหมอกควัน #ข่าวช่อง3 #Ch3ThailandNews
เรารู้ว่าแบกอยู่ แต่ยินดีที่ได้แบก ยินดีที่ได้ประสบการณ์
ผมไม่หล่อและก็ไม่เคยคิดว่าตัวเองหล่อ แต่ผมแค่อยากทำดีให้คนอื่นบ้างเท่านั้นเองอะครับ... #อย่าเหยียดกันเลย 

4.2.3. ตัวอย่างการตัดคำและการลบคำหยุด

ในงานวิจัยนี้ได้ทำการทดลองการตัดคำด้วยโปรแกรม Deepcut และ PyThaiNLP หลังจากการตัดคำแล้วจะทำการลบคำหยุดออก โดยใช้คำหยุดจากโปรแกรม Deepcut และ PyThaiNLP ตารางที่ 5 แสดงข้อมูลตัวอย่างคำหยุด โดยฝั่งซ้ายคือคำหยุดของโปรแกรม Deepcut และฝั่งขวาคือคำหยุดของโปรแกรม PyThaiNLP

ตารางที่ 5 แสดงข้อมูลตัวอย่างคำหยุดของโปรแกรม Deepcut และ PyThaiNLP

Deepcut	PyThaiNLP
ไว้, ไม่, ไป, ได้, ให้, ใน,โดย, แห่ง, แล้ว, และ, แรก, แบบ,แต่, เอง, เห็น, เลย, เริ่ม, เรา, เมื่อ, เพื่อ, เพราะ, เป็นการ, เป็น, เปิดเผย, เปิด, เนื่องจาก, เดียวกัน, เดียว ,เช่น, เฉพาะ, เคย, เข้า, เขา, อีก, อาจ, อะไร, ออก, อย่าง, อยู่, อยาก, หาก, หลาย	ณ, ๆ, ๆ, กู, ก็, ขอ, คง, คะ, คำ, จง, จด, จน, จะ, จำ, ซะ, ตน, ที, นะ, นำ, บน, ผล, พบ, พอ, พา, มา, มิ, มี, ยก, รึ, ลง, ละ, หน, ฯล, แก, โต, ใน, ไง, ไป, ไร, ๆ, กัน, กับ, การ, ขณะ, ของ, ขาด, ข้า, ครบ, ครา, ควร, คิด, คือ, คุณ, ค่ะ, ครับ, จรด, จวน, จวบ, จัง, จัด, จับ

ส่วนตารางที่ 6 แสดงข้อมูลตัวอย่างการตัดคำและลบคำหยุดด้วยโปรแกรม Deepcut และ PyThaiNLP โดยฝั่งซ้ายมือคือการตัดคำและลบคำหยุดด้วยโปรแกรม Deepcut ฝั่งขวาคือการตัดคำและลบคำหยุดด้วยโปรแกรม PyThaiNLP

ตารางที่ 6 แสดงตัวอย่างการตัวคำและลบคำหยุดด้วยโปรแกรม Deepcut และ PyThaiNLP

ประโยคตัวอย่าง	Deepcut	PyThaiNLP
ชาวขอนแก่นระแวกใกล้เคียง สนใจสมัครได้ครับ	ขอนแก่น|ระแวก|ใกล้เคียง|สนใจ|สมัคร	ชาว|ขอนแก่นระแวก|ใกล้เคียง|สนใจ|สมัคร
วันนี้ทำงาน พรุ่งนี้ได้หยุดแว้ววววว 	ทำ|งาน|พรุ่ง|หยุด|แว้ววววว| 	นี้|ทำ|งาน|พรุ่ง|นี้|หยุด|แว้ววววว| 
ออกไปทำงานนอกสถานที่สองชั่วโมง ภูมิแพ้มาเลย แสรดดดดด !!!	ทำ|งาน|นอก|สถานที่|สอง|ชั่วโมง|ภูมิแพ้|แสร|ดดดดด	ทำ|งาน|สถานที่|สอง|ชั่วโมง|ภูมิแพ้|แสร|ดดดดด
มันต้องแบบนี้ทีมชาติไทย มันเป็นของคนไทย 	มัน|ทีม|ชาติ|ไทย|มัน|คน|ไทย| 	นี้|ทีม|ชาติ|ไทย|คน|ไทย| 
คนท้องทำไมยังสวย??? 	คน|ท้อง|ทำไม|สวย| 	คน|ท้อง|สวย| 

จากตารางที่ 6 แสดงให้เห็นว่าการตัดคำและการลบคำหยุดของโปรแกรม Deepcut และ PyThaiNLP ให้ผลลัพธ์ที่แตกต่างกัน ซึ่งมีผลจากการตัดคำและการลบคำหยุดที่ไม่เหมือนกัน เช่นประโยค “ชาวขอนแก่นระแวก

ใกล้เคียง” โปรแกรม Deepcut ได้ผลลัพธ์เป็น “ขอนแก่น|ระแวก|ใกล้เคียง|” ส่วนโปรแกรม PyThaiNLP ได้ผลลัพธ์เป็น “ชาว|ขอนแก่นระแวก|ใกล้เคียง|” เป็นต้น

4.2.4. การแทนข้อความ

ในงานวิจัยนี้มีได้ทดลองการแทนข้อความทั้งหมด 4 แบบ ดังนี้

4.2.4.1. การแทนข้อความด้วยถุงคำ

การแทนข้อความแบบถุงคำจะแทนข้อความให้อยู่ในรูปแบบของเวกเตอร์ที่มีขนาดเท่ากับจำนวนคำทั้งหมดในพจนานุกรมของชุดข้อมูล โดยในงานวิจัยนี้มีคำที่ได้จากการโพสต์ของผู้ใช้งานเฟซบุ๊กทั้งหมด 19,893 คำ ทำให้เวกเตอร์ที่ใช้ในการแทนข้อความของถุงคำมีขนาดเท่ากับ 19,893

จากตัวอย่างประโยค “รอ||น้อง||ชาย||  ||ท่า||น้ำ||นะ||  ” สามารถแทนข้อความได้ดังนี้

จากพจนานุกรมของชุดข้อมูล [“รอ”, “น้อง”, “ชาย”, “  ”, “ท่า”, “น้ำ”, “นะ”, “  ”, “คน”, ..., “ไก่”]

เราแทนข้อความด้วย [1, 1, 1, 1, 1, 1, 1, 1, 0, ..., 0]

4.2.4.2. การแทนข้อความด้วยทีเอฟไอดีเอฟ

การแทนข้อความแบบทีเอฟไอดีเอฟ จะเป็นวิธีการแทนข้อความคล้ายกับวิธีถุงคำโดยมีขนาดของเวกเตอร์ของทีเอฟไอดีเอฟเท่ากับ 19,893 แต่จะต่างกันที่ทีเอฟไอดีเอฟใช้ค่าความถี่ของคำในข้อความคูณกับค่าผกผันของความถี่ของคำ แทนความถี่ของคำที่ปรากฏในข้อความ

จากตัวอย่างประโยค “รอ||น้อง||ชาย||  ||ท่า||น้ำ||นะ||  ” สามารถแทนข้อความได้ดังนี้

จากพจนานุกรมของชุดข้อมูล [“รอ”, “น้อง”, “ชาย”, “  ”, “ท่า”, “น้ำ”, “นะ”, “  ”, “คน”, ..., “ไก่”]

เราแทนข้อความด้วย [0.0543, 0.3286, 0.4142, 0.1759, 0.2685, 0.2881, 0.3078, 0.4896, 0.4435, 0, ..., 0]

4.2.4.3. การแทนข้อความด้วยเวิร์ดทูเวก

ในงานวิจัยนี้ใช้โปรแกรม gensim ในการประมวลผลเวิร์ดทูเวกและกำหนดขนาดของเวกเตอร์เวิร์ดทูเวกเท่ากับ 100 โดยใช้ข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ จำนวน 130,000 ทวีตมาประมวลผล การนำเวิร์ดทูเวกมาแทนข้อความ เราจะนำเวกเตอร์ของคำในข้อความนั้นมาต่อกันเป็นเมทริกซ์ โดยมีขนาดเท่ากับจำนวนคำที่มากที่สุดของโพสต์ในเฟซบุ๊ก ซึ่งมีค่าเท่ากับ 208 ทำให้ขนาดของเมทริกซ์ของเวิร์ดทูเวกที่ใช้แทน 1 ข้อความ มีขนาดเท่ากับ 208 x 100

จากตัวอย่างประโยค “รอ||น้อง||ชาย||  ||ท่า||น้ำ||นะ||  ” สามารถแทนข้อความได้ดังนี้

คำว่า “รอ” แทนด้วย [-0.4854 0.4341 ... -1.8216] เป็นเวกเตอร์ขนาด 100

คำว่า “น้อง” แทนด้วย [-0.6648 -0.0758 ... 0. 3338] เป็นเวกเตอร์ขนาด 100 นำมาต่อกันเป็นเมทริกซ์ จะได้ [[-0.4854 0.4341 ... -1.8216], [-0.6648 -0.0758 ... 0. 3338], ..., [0.00 0.00 ... 0.00], [0.00 0.00 ... 0.00]] เมทริกซ์ขนาด 208 x 100

4.2.4.4. การแทนข้อความด้วยความน่าจะเป็นของคำ

การแทนข้อความแบบความน่าจะเป็นของคำจะมีขนาดเวกเตอร์ของคำแต่ละคำเท่ากับ 3 โดยแทนค่าด้วยความน่าจะเป็นของคำในแต่ละรุ่นอายุ การนำเวกเตอร์ของคำมาแทนข้อความ เราจะนำเวกเตอร์ของคำในข้อความนั้นมาต่อกันเป็นเมทริกซ์เหมือนกับการแทนข้อความด้วยเวิร์ดทูเวก ทำให้ขนาดของเมทริกซ์ความน่าจะเป็นของคำที่ใช้แทน 1 ข้อความ มีขนาดเท่ากับ 208 x 3

จากตัวอย่างประโยค “รอ||น้อง||ชาย||||ท่า||น้ำ||นะ||  ” สามารถแทนข้อความได้ดังนี้

คำว่า “รอ” แทนด้วย [0.4642 0.3571 0.1785] เป็นเวกเตอร์ขนาดเท่ากับ 3

คำว่า “น้อง” แทนด้วย [0.4404 0.3035 0.2559] เป็นเวกเตอร์ขนาดเท่ากับ 3 นำมาต่อกันเป็นเมทริกซ์ จะได้ [[0.4642 0.3571 0.1785], [0.4404 0.3035 0.2559], ..., [0.000 0.000 0.000]] เมทริกซ์ขนาด 208 x 3

4.3. ผลการทดลอง

4.3.1. การทดลองเพื่อหาวิธีการแทนข้อความและโมเดลที่ดีที่สุด

4.3.1.1. การแทนข้อความด้วยถุงคำและทีเอฟไอดีเอฟด้วยโมเดลเพอร์เซ็ปตรอนหลายชั้น

ในการทดลองของแบบจำลองโมเดลเพอร์เซ็ปตรอนหลายชั้น เรากำหนดป้ายเป็น รุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด ในแบบจำลองนี้ใช้การแทนข้อความ 2 แบบ ได้แก่ ถุงคำและทีเอฟไอดีเอฟ และใช้ Grid Search เพื่อหาค่าพารามิเตอร์ที่ดีที่สุด โดยกำหนดค่าพารามิเตอร์ดังนี้ max_iter = [200], hidden_layer_sizes = [50,100,[150,50]], learning_rate_init = [0.001,0.0001]

ตารางที่ 7 ผลลัพธ์ค่าความแม่นยำของโมเดลเพอร์เซ็ปตรอนหลายชั้น

MODEL	โปรแกรมตัดคำ	ACCURACY (%)	
		USER	STATUS
MLP (BOW)	Deepcut	77.30	50.90
	PyThaiNLP	74.60	50.10
MLP (TF-IDF)	Deepcut	80.00	51.25
	PyThaiNLP	77.60	50.80

ตารางที่ 8 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวันของโมเดลเพอร์เซ็ปตรอนหลายชั้นแบบต่อผู้ใช้

MODEL	โปรแกรมตัดคำ	LABEL	USER		
			PRECISION	RECALL	F1
MLP (BOW)	Deepcut	GEN X	76.86	72.00	74.03
		GEN Y	72.83	63.00	66.59
		GEN Z	83.64	97.00	89.40
MLP (BOW)	PyThaiNLP	GEN X	77.06	69.00	72.80
		GEN Y	69.64	59.00	63.87
		GEN Z	78.34	96.00	86.27
MLP (TF-IDF)	Deepcut	GEN X	77.87	71.00	73.32
		GEN Y	71.64	74.00	72.36
		GEN Z	92.84	95.00	93.59
MLP (TF-IDF)	PyThaiNLP	GEN X	74.63	71.00	72.36
		GEN Y	70.22	66.00	67.16
		GEN Z	89.03	96.00	91.99

ตารางที่ 9 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวันของโมเดลเพอร์เซ็ปตรอนหลายชั้นแบบต่อโพสต์

MODEL	โปรแกรมตัดคำ	LABEL	USER		
			PRECISION	RECALL	F1
MLP (BOW)	Deepcut	GEN X	53.25	44.13	48.23
		GEN Y	45.22	38.43	41.53
		GEN Z	53.20	70.13	60.47
MLP (BOW)	PyThaiNLP	GEN X	52.61	44.43	48.10
		GEN Y	44.55	35.26	39.33
		GEN Z	52.10	70.80	59.97
MLP (TF-IDF)	Deepcut	GEN X	51.78	46.33	48.86
		GEN Y	44.13	42.53	43.24
		GEN Z	57.07	64.90	60.62
MLP (TF-IDF)	PyThaiNLP	GEN X	51.89	45.96	48.70
		GEN Y	43.94	40.80	42.27
		GEN Z	55.59	65.63	60.13

จากตารางที่ 7 - 9 แสดงผลสรุปค่าความแม่นยำของโมเดลเพอร์เซ็ปตรอนหลายชั้น โดยโมเดลเพอร์เซ็ปตรอนหลายชั้น ที่ใช้การแทนข้อความแบบทีเอฟไอดีเอฟ และใช้โปรแกรม Deepcut ในการตัดคำ และมีค่าพารามิเตอร์ hidden_layer_sizes = 50, learning_rate_init = 0.0001 ได้ค่าความแม่นยำกับชุดข้อมูลทดสอบ (Test Set) มากที่สุด โดยมีค่าความแม่นยำแบบต่อผู้ใช้งาน (Accuracy Per User) เท่ากับ 80.00% และค่าความแม่นยำแบบต่อโพสต์ (Accuracy Per Status) เท่ากับ 51.25% และมีค่าเอฟวันแบบต่อผู้ใช้งาน (F1 Per User) ของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด เท่ากับ 73.32, 72.36, 93.59 ตามลำดับ และค่าเอฟวันแบบต่อโพสต์ (F1 Per Status) ของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซดเท่ากับ 48.86, 43.24, 60.62 ตามลำดับ

4.3.1.2. การแทนข้อความด้วยเวิร์ดทูเวกและการใช้เวิร์ดทูเวกร่วมกับการแทนข้อความในรูปแบบต่างๆ ด้วยโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาว

ในการทดลองแบบจำลองโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาว เรากำหนดป้ายเป็นรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด ในแบบจำลองนี้ใช้การแทนข้อความทั้งหมด 4 แบบ ได้แก่ เวิร์ดทูเวก, เวิร์ดทูเวกร่วมกับถุงคำ, เวิร์ดทูเวกร่วมกับทีเอฟไอดีเอฟ และ เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ และมีการกำหนดค่าพารามิเตอร์ดังนี้ optimizer algorithm = Adaptive Moment Estimation (Adam), loss function = Binary Cross-entropy, Batch size = 50

ตารางที่ 10 ผลลัพธ์ค่าความแม่นยำของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว

MODEL	โปรแกรมตัดคำ	ACCURACY (%)	
		USER	STATUS
CNN (word2vec)	Deepcut	74.33	48.36
CNN (word2vec)	PyThaiNLP	70.99	47.98

MODEL	โปรแกรมตัดคำ	ACCURACY (%)	
		USER	STATUS
CNN (word2vec + BOW)	Deepcut	78.00	51.04
CNN (word2vec + BOW)	PyThaiNLP	77.99	50.41
CNN (word2vec + TF-IDF)	Deepcut	81.33	50.75
CNN (word2vec + TF-IDF)	PyThaiNLP	77.99	50.96
CNN (word2vec + Prob of words)	Deepcut	82.90	52.48
CNN (word2vec + Prob of words)	PyThaiNLP	81.60	51.30
LSTM (word2vec)	Deepcut	69.33	47.74
LSTM (word2vec)	PyThaiNLP	69.30	47.68
LSTM (word2vec + BOW)	Deepcut	78.00	50.80
LSTM (word2vec + BOW)	PyThaiNLP	77.00	50.77
LSTM (word2vec + TF-IDF)	Deepcut	78.66	51.24
LSTM (word2vec + TF-IDF)	PyThaiNLP	78.50	51.17
LSTM (word2vec + Prob of words)	Deepcut	81.66	51.84
LSTM (word2vec + Prob of words)	PyThaiNLP	81.00	51.15

ตารางที่ 11 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวัน ของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาวที่มีค่าความแม่นยำมากที่สุดแบบต่อผู้ใช้

MODEL	LABEL	USER		
		PRECISION	RECALL	F1
CNN (word2vec + Prob of words)	GEN X	79.43	82.00	79.89
	GEN Y	80.09	72.00	74.40
	GEN Z	92.75	95.00	93.59
LSTM (word2vec + Prob of words)	GEN X	77.68	77.00	76.86
	GEN Y	74.30	72.00	72.49
	GEN Z	93.66	96.00	94.65

ตารางที่ 12 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวัน ของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาวที่มีค่าความแม่นยำมากที่สุดแบบต่อผู้โพสต์

MODEL	LABEL	STATUS		
		PRECISION	RECALL	F1
CNN (word2vec + Prob of words)	GEN X	53.65	49.76	51.49
	GEN Y	45.58	42.50	43.80
	GEN Z	57.66	65.20	61.07
LSTM (word2vec + Prob of words)	GEN X	53.29	47.76	50.26

	GEN Y	46.20	39.40	42.34
	GEN Z	54.87	68.36	60.79

จากตารางที่ 10 - 12 แสดงผลสรุปค่าความแม่นยำของโมเดลโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว โดยโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน ที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำและใช้โปรแกรม Deepcut ในการตัดคำ ได้ค่าความแม่นยำกับชุดข้อมูลทดสอบมากที่สุด โดยมีค่าความแม่นยำแบบต่อผู้ใช้งานเท่ากับ 82.90% และค่าความแม่นยำแบบต่อโพสต์เท่ากับ 52.48% และมีค่าเอฟวันแบบต่อผู้ใช้งานของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด เท่ากับ 79.89, 74.40, 93.59 ตามลำดับ และค่าเอฟวันแบบต่อโพสต์ของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด เท่ากับ 51.49, 43.80, 61.07

จากผลการทดลองการหาวิธีการแทนข้อความและโมเดลที่ดีที่สุด เมื่อนำผลการทดลองทั้งหมดมาเปรียบเทียบค่าความแม่นยำกัน โมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน ที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำและใช้โปรแกรม Deepcut ในการตัดคำ ได้ค่าความแม่นยำมากที่สุด ทำให้สามารถสรุปได้ว่าการนำความน่าจะเป็นของคำ เข้ามาช่วยในการแทนข้อความนั้นสามารถช่วยเพิ่มประสิทธิภาพให้กับโมเดลได้ นอกจากนี้การทดลองของการใช้โปรแกรมตัดคำ Deepcut และ PyThaiNLP เมื่อนำโมเดลเดียวกันมาเปรียบเทียบกันนั้น จะเห็นได้ว่าโปรแกรมตัดคำนั้นมีผลต่อประสิทธิภาพของโมเดล ซึ่งในการทดลองนี้โปรแกรม Deepcut ได้ค่าความแม่นยำที่ดีกว่า PyThaiNLP

ตารางที่ 13 ผลรวม 10 fold cross validation ของตารางคอนฟิวชันเมทริกซ์แบบต่อผู้ใช้

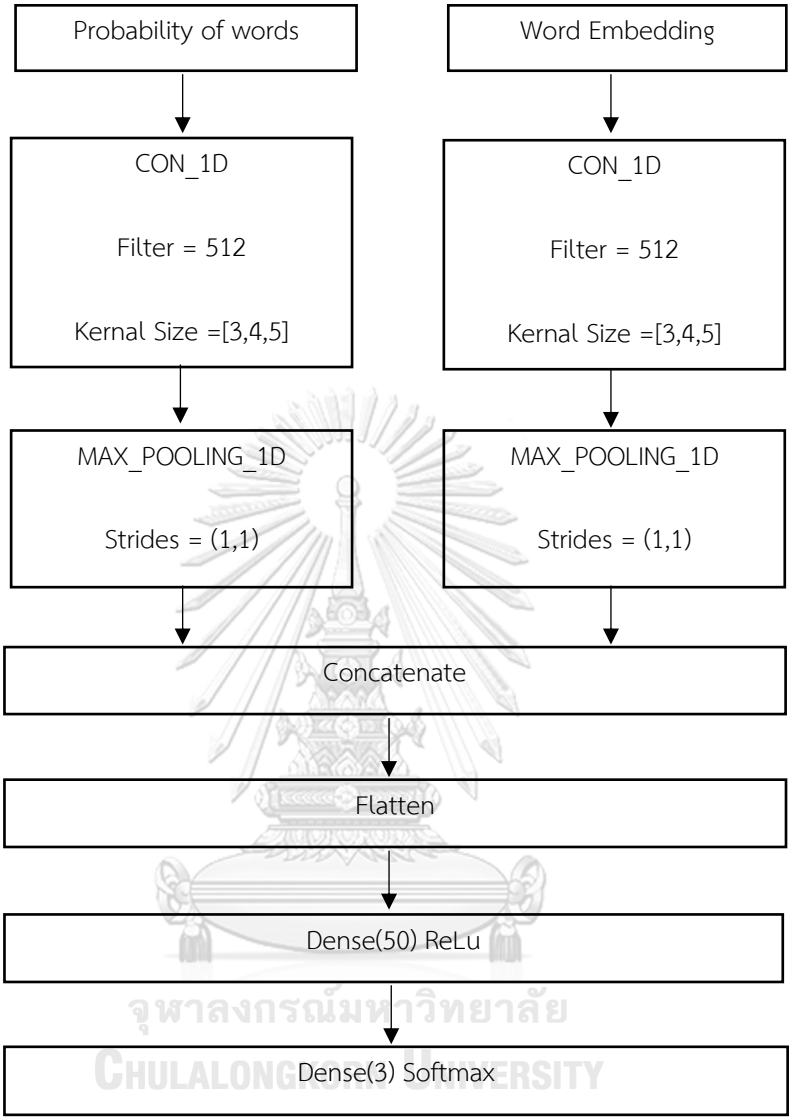
		Predicted Class		
		X	Y	Z
Actual Class	X	82	18	0
	Y	20	72	8
	Z	3	2	95

ตารางที่ 14 ผลรวม 10 fold cross validation ของตารางคอนฟิวชันเมทริกซ์แบบต่อผู้โพสต์

		Predicted Class		
		X	Y	Z
Actual Class	X	1,493	910	597
	Y	872	1,275	853
	Z	428	616	1,956

จากตารางที่ 13 และ 14 แสดงผลรวม 10-fold cross validation ของตารางคอนฟิวชันเมทริกซ์ของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน ที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำของชุดข้อมูลทดสอบ ตารางที่ 13 แสดงผลลัพธ์แบบต่อผู้ใช้ มีผู้ใช้งานทั้งหมด 300 คน แบ่งเป็นรุ่นอายุละ 100 คน โดยได้ค่า TP ของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด เท่ากับ 82, 72, 95 ตามลำดับ ส่วนตารางที่ 14 แสดงผลลัพธ์แบบต่อโพสต์ มีโพสต์ทั้งหมด 9,000 โพสต์ แบ่งเป็นรุ่นอายุละ 3,000 โพสต์ ได้ค่า TP ของรุ่นอายุเอกซ์, รุ่นอายุวาย, รุ่นอายุแซด เท่ากับ 1,493, 1,275, 1,956 ตามลำดับ จากตาราง 10-fold cross validation แสดงให้เห็นว่าในการจำแนกรุ่นอายุนั้นสามารถจำแนกรุ่นอายุแซดได้ดีที่สุด ตามด้วยรุ่นอายุเอกซ์และรุ่นอายุวายตามลำดับ

4.3.1.3. โครงสร้างโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน (เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ)

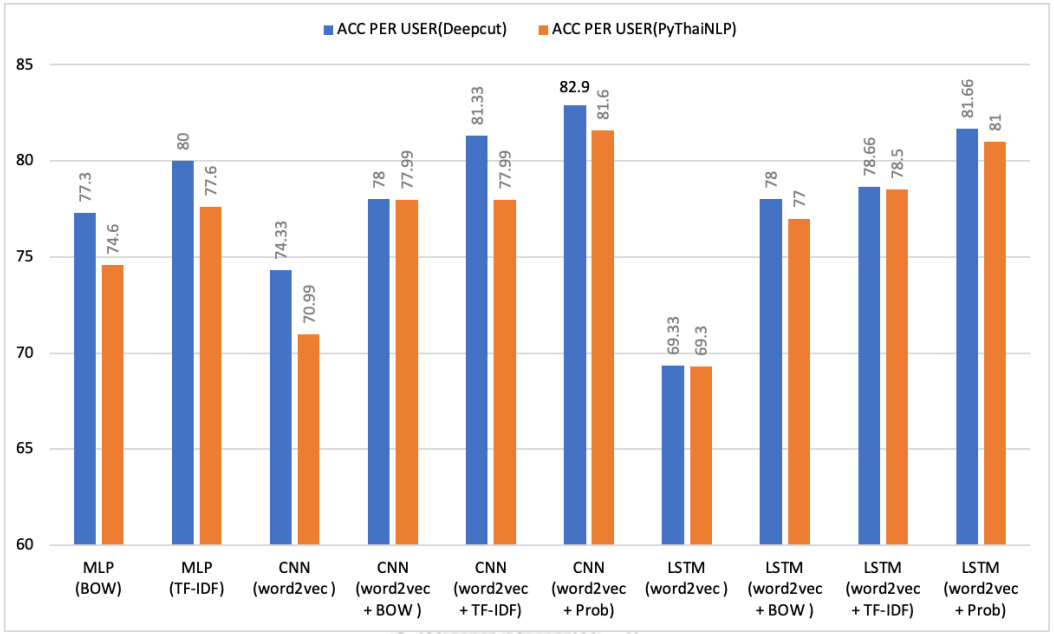


รูปที่ 7 โครงสร้างโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน (เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ)

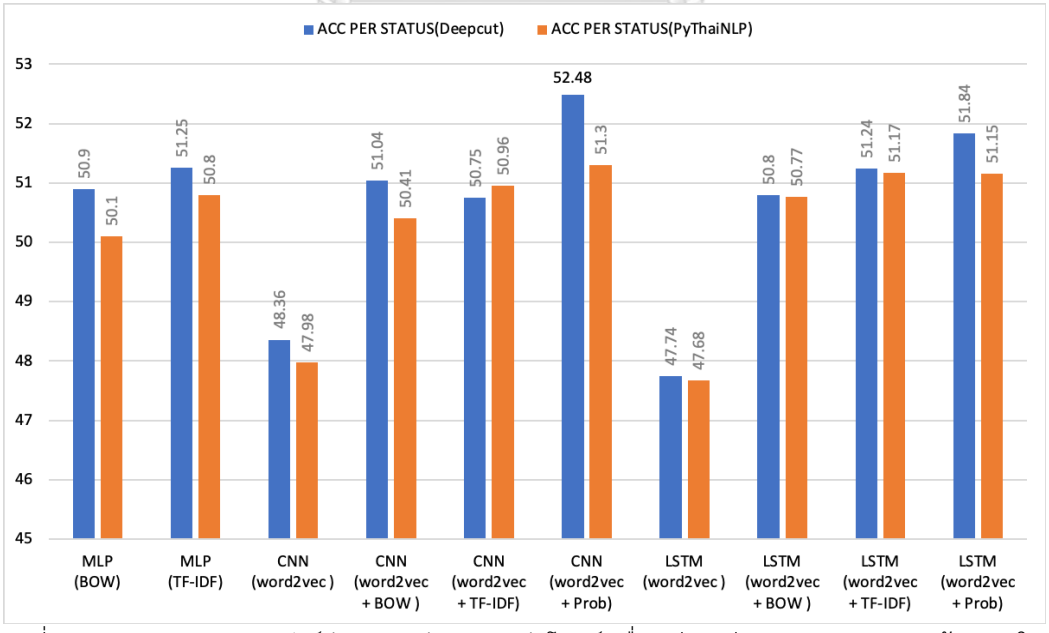
จากรูปที่ 7 แสดงโครงสร้างของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน ที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ โดยหลังจากทำการแทนข้อความเรียบร้อยแล้ว จะนำข้อมูลเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ ที่ได้นำข้อมูลมาผ่าน 1D Convolution layer มีการกำหนดค่าพารามิเตอร์ดังนี้ Filter = 512, Kernal Size ทั้งหมด 3 แบบคือ 3, 4, 5 และมี Activation = ReLu ต่อมานำข้อมูลไปผ่าน 1D Max pooling layer กำหนดให้ Strides =(1,1) จากนั้นนำผลลัพธ์ที่ได้จาก Max pooling layer ของเวิร์ดทูเวกและความน่าจะเป็นของคำ มารวมกัน โดยใช้ Concatenate และใช้ Flatten เพื่อแปลงข้อมูลไปยัง Fully connected layer ในชั้นแรกใช้ Dense = 50, Activation = ReLu และชั้นสุดท้ายใช้ Dense = 3, Activation = Softmax

4.3.1.4. ความน่าจะเป็นของคำ

จากการทดลองที่ผ่านมาจะเห็นได้ว่า การแทนข้อความในรูปแบบต่างๆนั้นมีผลต่อค่าความแม่นยำของการจำแนกรุ่นอายุ ซึ่งการนำการแทนข้อความในรูปแบบต่างๆมาใช้ร่วมกัน ส่งผลให้ประสิทธิภาพของโมเดลนั้นดีขึ้น การใช้เวิร์ดทูเวกร่วมกับถุงคำ หรือ ทีเอฟไอดีเอฟ หรือ ความน่าจะเป็นของคำ ทำให้ค่าความแม่นยำสูงกว่าการใช้เวิร์ดทูเวกเพียงอย่างเดียว ซึ่งการแทนข้อความด้วยเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ นั้นมีค่าความแม่นยำมากที่สุด



รูปที่ 8 แผนภูมิภาพแสดงผลลัพธ์ค่าความแม่นยำแบบต่อผู้ใช้ เพื่อเปรียบเทียบประสิทธิการแทนข้อความในแบบต่างๆ



รูปที่ 9 แผนภูมิภาพแสดงผลลัพธ์ค่าความแม่นยำแบบต่อโพสต์ เพื่อเปรียบเทียบประสิทธิการแทนข้อความในแบบต่างๆ

จากรูปที่ 8 เมื่อเทียบค่าความแม่นยำแบบต่อผู้ใช้ ระหว่างโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันการแทนข้อความด้วยเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ กับการแทนข้อความด้วยเวิร์ดทูเวกเพียงอย่างเดียว ที่ใช้โปรแกรม Deepcut นั้น สามารถเพิ่มค่าความแม่นยำ จาก 74.33% เป็น 82.90% และจากรูปที่ 9 เมื่อค่าความแม่นยำแบบต่อโพสต์ สามารถเพิ่มค่าความแม่นยำ จาก 48.36% เป็น 52.48%

ตารางที่ 15 ค่าความน่าจะเป็นของคำของแต่ละรุ่นอายุ

คำ	GEN X		GEN Y		GEN Z	
	PROB.	No.	PROB.	No.	PROB.	No.
สถานที่	0.857	12	0.142	2	0	0
ครัว	0.83	20	0.166	4	0	0
😝	0.77	57	0.108	8	0.121	9
🌵	0	0	0.857	12	0.142	2
🍭	0.166	4	0.791	19	0.041	1
ปลา	0.178	5	0.785	22	0.035	1
🔥	0	0	0.138	5	0.861	31
ป่ะ	0	0	0.157	3	0.842	16
ค่าย	0.172	5	0.034	1	0.793	23

จากตาราง 15 แสดงตัวอย่างค่าความน่าจะเป็นของคำที่พบในแต่ละรุ่นอายุ จากข้อมูลนี้เราสามารถนำคำเหล่านี้มาช่วยในการจำแนกรุ่นอายุได้ เช่น คำว่า "สถานที่" มีการใช้ในรุ่นอายุเอกซ์ทั้งหมด 12 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.857 และมีการใช้ในรุ่นอายุวายทั้งหมด 2 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.142 ส่วนในรุ่นอายุแซดนั้นไม่มีการใช้เลย หรือมีความน่าจะเป็นของคำเท่ากับ 0 หรือ สัญรูปอารมณ์ "🔥" มีการใช้ในรุ่นอายุแซดทั้งหมด 31 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.861 และมีการใช้ในรุ่นอายุวายทั้งหมด 5 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.138 ส่วนในรุ่นอายุเอกซ์ไม่มีการใช้งานเลย หรือมีความน่าจะเป็นของคำเท่ากับ 0 หรือคำว่า "ค่าย" มีการใช้ในรุ่นอายุแซดทั้งหมด 23 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.793 และมีการใช้งานในรุ่นอายุเอกซ์ทั้งหมด 5 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.172 และมีการใช้งานในรุ่นอายุวายทั้งหมด 1 ครั้ง หรือมีความน่าจะเป็นของคำเท่ากับ 0.034 จากตัวอย่างข้างต้น การนำความน่าจะเป็นของคำมาช่วยในการจำแนกรุ่นอายุนั้น ทำให้เราสามารถสร้างกฎที่มนุษย์สามารถคาดเดาได้อยู่แล้วให้กับโมเดลของเราได้ เช่น ผู้ใช้งานที่โพสต์คำว่า "ค่าย" มีความน่าจะเป็นที่ค่อนข้างสูงที่จะอยู่ในวัยเรียน ซึ่งก็คือรุ่นอายุแซดหรือผู้ใช้งานที่โพสต์คำว่า "ครัว" มีความน่าจะเป็นที่ผู้ใช้งานจะอยู่ในช่วงวัยทำงาน ซึ่งมีโอกาสเป็นรุ่นอายุเอกซ์และรุ่นอายุวายได้

4.3.2. การทดลองเปรียบเทียบการจำแนกรุ่นอายุแบบ 2 ป้าย ด้วยโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาว

ในการทดลองนี้เราได้ทดลองการจำแนกรุ่นอายุแบบ 2 ป้าย โดยการนำข้อมูลรุ่นอายุมาจำแนกทีละ 2 รุ่น เพื่อตรวจสอบว่าคู่รุ่นอายุไหนที่มีการจำแนกได้ดีที่สุด โดยมีการกำหนดป้ายเป็น 3 แบบ ได้แก่ 1.รุ่นอายุเอกซ์, รุ่นอายุวาย 2.รุ่นอายุเอกซ์, รุ่นอายุแซด 3.รุ่นอายุวาย, รุ่นอายุแซด ในแบบจำลองนี้ใช้การแทนข้อความทั้งหมด 4 แบบ

ได้แก่ เวิร์ดทูเวก, เวิร์ดทูเวกร่วมกับถุงคำ, เวิร์ดทูเวกร่วมกับทีเอฟไอดีเอฟ และเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ และใช้โมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาวในการทดลอง

ตารางที่ 16 ผลลัพธ์ค่าความแม่นยำของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว แบบต่อผู้ใช้ (2 ป้าย)

MODEL	ACCURACY PER USER (%)		
	X,Y	X,Z	Y,Z
CNN (word2vec)	73.50	95.00	82.50
CNN (word2vec + BOW)	76.50	97.00	90.00
CNN (word2vec + TF-IDF)	75.50	97.00	92.50
CNN (word2vec + Prob of words)	80.00	98.00	93.00
LSTM (word2vec)	74.00	95.00	85.50
LSTM (word2vec + BOW)	79.49	96.49	90.50
LSTM (word2vec + TF-IDF)	77.00	97.00	92.50
LSTM (word2vec + Prob of words)	79.00	95.50	94.00

ตารางที่ 17 ผลลัพธ์ค่าความแม่นยำของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว แบบต่อโพสต์ (2 ป้าย)

MODEL	ACCURACY PER STATUS (%)		
	X,Y	X,Z	Y,Z
CNN (word2vec)	58.58	71.01	63.56
CNN (word2vec + BOW)	58.26	73.14	66.45
CNN (word2vec + TF-IDF)	58.33	73.75	67.00
CNN (word2vec + Prob of words)	58.35	73.86	67.01
LSTM (word2vec)	58.20	71.16	63.85
LSTM (word2vec + BOW)	58.28	73.16	65.99
LSTM (word2vec + TF-IDF)	58.70	73.10	66.28
LSTM (word2vec + Prob of words)	58.00	72.04	67.73

ตารางที่ 18 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวันของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน (เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ) แบบต่อผู้ใช้ (2 ป้าย)

MODEL	LABEL	USER		
		PRECISION	RECALL	F1
CNN (word2vec+Prop of words)	GEN X	83.35	76.00	78.79
	GEN Y	78.70	84.00	80.85
CNN (word2vec+Prop of words)	GEN X	96.51	100.00	98.13
	GEN Z	100.00	96.00	97.83
CNN (word2vec+Prop of words)	GEN Y	90.10	97.00	93.26
	GEN Z	97.07	89.00	92.66

ตารางที่ 19 ผลลัพธ์ค่าความเที่ยง, ค่าความระลึก, ค่าเอฟวันของโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน (เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ) แบบต่อโพสต์ (2 ป้าย)

MODEL	LABEL	STATUS		
		PRECISION	RECALL	F1
CNN (word2vec+Prop of words)	GEN X	58.55	57.10	57.80
	GEN Y	58.17	59.60	58.86
CNN (word2vec+Prop of words)	GEN X	74.29	73.33	73.70
	GEN Z	73.69	74.40	73.95
CNN (word2vec+Prop of words)	GEN Y	66.79	67.90	67.28
	GEN Z	67.38	66.13	66.68

จากตารางที่ 16 - 19 แสดงผลสรุปความแม่นยำของโมเดลโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน, หน่วยความจำระยะสั้นแบบยาว แบบ 2 ป้าย จากการผลการทดลองจะเห็นได้ว่าโมเดลนิวรอลเน็ตเวิร์กคอนโวลูชันที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ ได้ค่าความแม่นยำที่มากที่สุด โดยสามารถจำแนกรุ่นอายุเอกซ์และรุ่นอายุแซดได้ดีที่สุด โดยมีค่าความแม่นยำแบบต่อผู้ใช้งาน เท่ากับ 98.00% และค่าความแม่นยำแบบต่อโพสต์ เท่ากับ 73.86% อันดับที่สองเป็นรุ่นอายุวายและรุ่นอายุแซด มีค่าความแม่นยำแบบต่อผู้ใช้งานเท่ากับ 93.00% และค่าความแม่นยำแบบต่อโพสต์ เท่ากับ 67.01% และอันดับที่สามเป็นรุ่นอายุเอกซ์และรุ่นอายุวาย มีความแม่นยำแบบต่อผู้ใช้งาน เท่ากับ 80.00% และค่าความแม่นยำแบบต่อโพสต์เท่ากับ 58.35%

บทที่ 5
บทสรุป

5.1. สรุปงานวิจัย

วิทยานิพนธ์นี้ได้เสนอวิธีการจำแนกรุ่นอายุของผู้ใช้งานเฟซบุ๊กไทยโดยใช้การเรียนรู้เชิงลึกร่วมกับความน่าจะเป็นของคำ โดยนำข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในประเทศไทยมาทำคำฝังตัวด้วยการประมวลผลแบบเวิร์ดทูเวก เพื่อสร้างคลังข้อมูลคำไทย ต่อมานำข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กไทยมาแปลงข้อมูลด้วยวิธีคำฝังตัวโดยการประมวลแบบเวิร์ดทูเวกจากคลังข้อมูลคำไทย และแปลงข้อมูลด้วยการใช้ความน่าจะเป็นของคำ จากนั้นนำข้อมูลที่ได้มาประมวลผลด้วยนิวรอลเน็ตเวิร์กคอนโวลูชันเพื่อจำแนกรุ่นอายุของผู้ใช้งาน

จากการทดสอบการใช้เวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำได้ค่าความแม่นยำแบบต่อผู้ใช้งาน เท่ากับ 82.90% และได้ค่าความแม่นยำแบบต่อโพสต์ เท่ากับ 52.48% ซึ่งได้ผลลัพธ์ที่ดีกว่าเมื่อนำมาเปรียบเทียบกับการใช้เวิร์ดทูเวกเพียงอย่างเดียว หรือการใช้เวิร์ดทูเวกร่วมกับถุงคำ หรือการใช้เวิร์ดทูเวกร่วมกับทีเอฟไอดีเอฟ นอกจากนี้การนำค่าความน่าจะเป็นของคำมาช่วยนั้น ยังสามารถทำให้เราสามารถสร้างกฎที่มนุษย์สามารถคาดเดาได้ให้กับแบบโมเดลของเรา เช่น การโพสต์คำว่า "ครัว" มีความน่าจะเป็นที่จะเป็นโพสต์ของผู้ใช้งานที่อยู่ในวัยทำงานซึ่งก็คือรุ่นอายุเอกซ์, รุ่นอายุวาย เป็นต้น

การใช้คำนวณค่าความแม่นยำแบบต่อผู้ใช้งาน ทำให้ประสิทธิภาพในการจำแนกรุ่นอายุมีความแม่นยำมากขึ้น เนื่องจากบางโพสต์ของผู้ใช้งาน อาจจะไม่ได้บ่งบอกถึงรุ่นอายุของผู้ใช้งาน แต่เมื่อนำผลลัพธ์จากหลายๆโพสต์ของผู้ใช้มารวมกัน แล้วนำรุ่นอายุที่ได้ค่ามากที่สุดมาเป็นคำตอบ จะช่วยให้โมเดลของเราเพิ่มประสิทธิภาพมากขึ้นได้

เมื่อเปรียบเทียบที่โมเดลเดียวกันการตัดคำและลบคำหยุดด้วยการใช้โปรแกรม Deepcut และ PyThaiNLP นั้นมีผลต่อประสิทธิภาพของโมเดล เช่น โมเดลนิวรอลเน็ตเวิร์กคอนโวลูชัน ที่ใช้การแทนข้อความแบบเวิร์ดทูเวกร่วมกับความน่าจะเป็นของคำ ค่าความแม่นยำแบบต่อผู้ใช้งานของโปรแกรม Deepcut เท่ากับ 82.90% ส่วนค่าความแม่นยำแบบต่อผู้ใช้งานของโปรแกรม PyThaiNLP เท่ากับ 81.60% ซึ่งแสดงให้เห็นว่าโปรแกรม Deepcut มีประสิทธิภาพมากกว่าโปรแกรม PyThaiNLP

5.2. แนวทางการวิจัยในขั้นถัดไป

5.2.1. เพิ่มการเก็บข้อมูลการทวีตของผู้ใช้งานทวิตเตอร์ในไทยให้มากขึ้น เพื่อสร้างคลังข้อมูลคำที่มากขึ้น

5.2.2. เพิ่มการเก็บข้อมูลการโพสต์ของผู้ใช้งานเฟซบุ๊กไทยให้มากขึ้น เพื่อสร้างความน่าจะเป็นของคำที่มีประสิทธิภาพมากขึ้น และเพื่อเพิ่มประสิทธิภาพของแบบจำลองให้แม่นยำมากขึ้น

5.2.3. เปลี่ยนการออกแบบโมเดลในแต่ละชั้น layer และปรับเปลี่ยนพารามิเตอร์ต่างๆของโมเดล

บรรณานุกรม

1. Sarakit, P., et al. *Classifying emotion in Thai youtube comments*. in *2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. 2015.
2. Bayot, R.K. and T. Gonçalves. *Age and Gender Classification of Tweets Using Convolutional Neural Networks*. 2018. Cham: Springer International Publishing.
3. Charoenkwan, P., *ThaiFBDeep: A Sentimental Analysis Using Deep Learning Combined with Bag-of-Words Features on Thai Facebook Data*, in *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*. 2018. p. 565-569.
4. Pholnarat, A., *Distinctive verbs of Thai teenagers' speech.* Journal of language and culture, 2016. **35**: p. 231-235.
5. T. Mikolov, I.S., K. Chen, G. S. Corrado, and J. Dean,, *Distributed Representations of Words and Phrases and their Compositionality.* p. 3111-3119.
6. Jeffrey Pennington, R.S., Christopher D. Manning, *GloVe: Global Vectors for Word Representation.* p. 1532-1543.
7. Simaki, V., I. Mporas, and V. Megalooikonomou, *Age Identification of Twitter Users: Classification Methods and Sociolinguistic Analysis*. 2016.
8. Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Pottast, M., Stein, B., *Overview of the 4th author profiling task at PAN 2016.* 2016: p. 750– 784.
9. Kim, Y., *Convolutional Neural Networks for Sentence Classification.* CoRR, 2014. **abs/1408.5882**.
10. Koomsubha, T., *Text Categorization for Thai Corpus using Character-Level Convolutional Neural Network.* 2016.
11. Berkup, S.B., *Working With Generations X And Y In Generation Z Period: Management Of Different Generations In Business Life*. 2014. Vol. 5. 2014.

ประวัติผู้เขียน

ชื่อ-สกุล	ศุภชัย ตั้งตรีรัตน์
วัน เดือน ปี เกิด	21/11/2528
สถานที่เกิด	กทม.
วุฒิการศึกษา	ปริญญาตรี
ที่อยู่ปัจจุบัน	7 วชิรธรรมสาธิต 64 สุขุมวิท101/1 บางจาก พระโขนง กทม. 10260