เทคนิคการเรียนรู้เชิงลึกเพื่อวิเคราะห์ความรู้สึกจากผู้ใช้ผลิตภัณฑ์

นายภูมิรพี ภูมิค้า

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

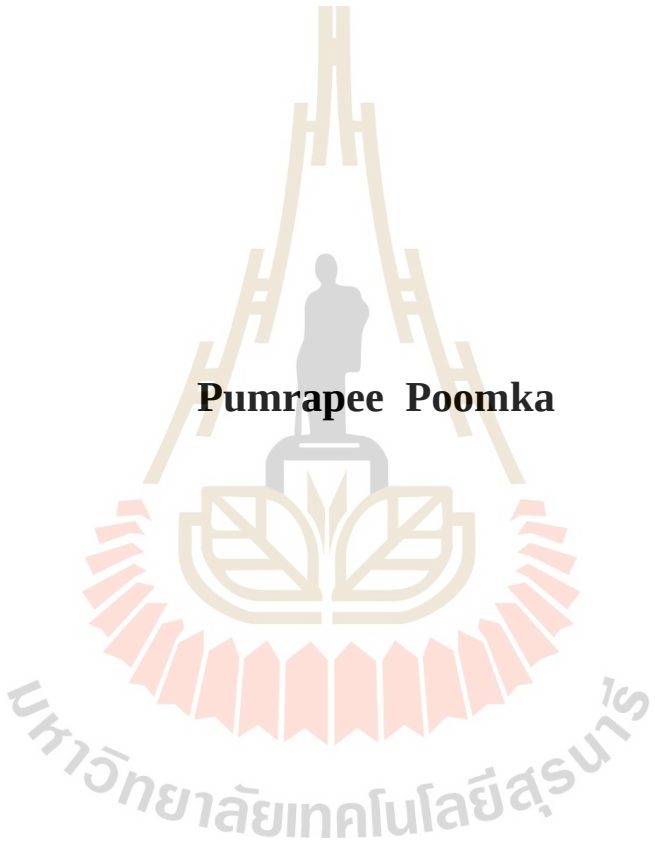
สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2562

DEEP LEARNING TECHNIQUES FOR SENTIMENT ANALYSIS FROM PRODUCT USERS

Pumrapee Poomka



A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Telecommunication and Computer Engineering

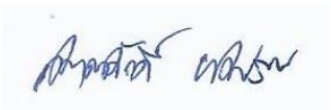
Suranaree University of Technology

Academic Year 2019

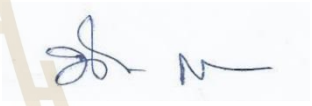
เทคนิคการเรียนรู้เชิงลึกเพื่อวิเคราะห์ความรู้สึกจากผู้ใช้ผลิตภัณฑ์

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

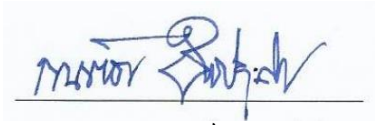


(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ



(รศ. ดร.นิตยา เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อ. ดร.กีระชาติ สุขสุทธิ์)
กรรมการ



(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการ
และพัฒนาความเป็นสากล

(รศ. ดร. พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

ภูมิรพี ภูมิค้า : เทคนิคการเรียนรู้เชิงลึกเพื่อวิเคราะห์ความรู้สึกจากผู้ใช้ผลิตภัณฑ์ (DEEP LEARNING TECHNIQUES FOR SENTIMENT ANALYSIS FROM PRODUCT USERS) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.นิตยา เกิดประสพ, 88 หน้า.

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแบบจำลองเพื่อวิเคราะห์ความรู้สึกจากผู้ใช้ผลิตภัณฑ์ด้วยเทคนิคการเรียนรู้เชิงลึก โดยมีขอบเขตของการวิเคราะห์เฉพาะข้อความภาษาอังกฤษที่แสดงความคิดเห็นต่อผลิตภัณฑ์ แบบจำลองสามารถจำแนกข้อความซึ่งแบ่งตามลักษณะของอารมณ์ได้ 2 กลุ่ม ได้แก่อารมณ์ทางด้านบวกและอารมณ์ทางด้านลบ ขั้นตอนในการพัฒนาแบบจำลองประกอบด้วย 2 ส่วนได้แก่ ส่วนของการแปลงข้อความให้อยู่ในรูปแบบของเวกเตอร์ที่คอมพิวเตอร์สามารถประมวลผลได้ และส่วนของการนำข้อมูลที่ทำการแปลงแล้วมาใช้สร้างแบบจำลองด้วยเทคนิคการเรียนรู้เชิงลึก ประกอบด้วยอัลกอริทึมโครงข่ายประสาทเชิงลึก (Deep Neural Network) อัลกอริทึมโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) อัลกอริทึมหน่วยความจำระยะยาว-ระยะสั้น (Long Short-Term Memory) และอัลกอริทึมหน่วยเวียนกลับแบบมีประตู (Gated Recurrent Unit) การวัดผลการวิจัยจะใช้ค่าความถูกต้องในการวัดประสิทธิภาพของแบบจำลอง โดยใช้ชุดข้อมูลที่ได้จากแหล่งเดียวกันในการทดสอบ จากนั้นทำการสังเกตและบันทึกผลการทดลองเพื่อสรุปงานวิจัย

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา.....

ปีการศึกษา 2562 ลายมือชื่ออาจารย์ที่ปรึกษา.....

PUMRAPEE POOMKA : DEEP LEARNING TECHNIQUES FOR SENTIMENT ANALYSIS FROM PRODUCT USERS. THESIS ADVISOR : ASSOC. PROF. NITAYA KERDPRASOP, Ph.D. 88 PP.

DEEP LEARNING/NATURAL LANGUAGE PROCESSING/SENTIMENT ANALYSIS/TEXT FEATURE EXTRACTION

This research is aimed at developing predictive model for sentiment analysis from product review with Deep Learning algorithms. The scope of this research is to use only study case in English language on product review. We split dataset into 2 classes of sentiment that we want to classify as ether positive or negative. We design experiment into 2 parts. First, we transform text dataset into numerical form that computer can understand. Then, we develop predictive model based on Deep Learning algorithms such as Deep Neural Network, Convolutional Neural Network, Long Short-Term Memory and Gated Recurrent Unit. We evaluate performance of predictive models with accuracy by using the same source of dataset. Finally, we observe and record results of this research.

School of Computer Engineering Student's Signature Pumrapee Poomka

Academic Year 2019 Advisor's Signature Nittaya Kerp

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ ดร.นิตยา เกิดประสพ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และรองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ ที่ให้คำปรึกษาคำแนะนำในการทำงานวิจัย ตลอดจนถึงการจัดรูปแบบวิทยานิพนธ์ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์

คุณวรรธนะ พงษ์เสนา คุณอนุสรา หิรัญวนากุล และคุณอนุพงษ์ บรรจงการ ที่ให้ความช่วยเหลือในการช่วยตรวจทาน ติชม เพื่อปรับปรุงแก้ไขวิทยานิพนธ์ให้สมบูรณ์ยิ่งขึ้น

คุณปราณี กฐินใหม่ และคุณดารณี ทิพย์ทอง เลขานุการ และผู้ช่วยสอนประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานระหว่างศึกษา

ขอขอบคุณนักศึกษาร่วมชั้นเรียนทั้งปริญญาโท และปริญญาเอก ที่ให้คำแนะนำคำปรึกษาด้านวิชาการ และช่วยสนับสนุนด้วยดีมาตลอด

ขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้การสนับสนุนทุนการศึกษา ทุนวิจัย ทั้งยังค่าใช้จ่ายต่าง ๆ

นอกจากนี้ขอขอบคุณ ครู อาจารย์ ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยอย่างมากมายจนประสบความสำเร็จ

ท้ายที่สุดขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดู ส่งเสริมให้ผู้วิจัยมีความรู้ ความสามารถ ทั้งยังเป็นกำลังใจแก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

ภูมิรพี ภูมิค้า

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ซ
สารบัญรูป ..... ฌ
บทที่
1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหาการวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 3
1.4 ประโยชน์ที่ได้รับ 3
2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 4
2.1 การประมวลผลภาษาธรรมชาติ 4
2.2 การวิเคราะห์ความรู้สึก 5
2.2.1 Subjective Classification 6
2.2.2 Sentiment Detection and Classification 6
2.3 การทำเหมืองข้อความ 6
2.3.1 การทำความสะอาดข้อมูล 8
2.3.2 Word Tokenization 9
2.3.3 Bag of Words 11
2.3.4 Term Frequency – Invert Document Frequency 12
2.3.5 Word Embedding 14
2.3.6 Doc2Vec 16

สารบัญ (ต่อ)

หน้า

2.4 เทคนิคการเรียนรู้เชิงลึก 17
2.4.1 Deep Neural Network 20
2.4.2 Convolutional Neural Network 21
2.4.3 Recurrent Neural Network 23
2.4.4 Long Short-Term Memory 25
2.4.5 Gated Recurrent Unit 28
2.5 มาตรวัดประสิทธิภาพ 30
2.5.1 ค่าความถูกต้อง 30
2.5.2 ค่าความเที่ยง 31
2.5.3 ค่าความไว หรือค่าระลึก 32
2.6 งานวิจัยที่เกี่ยวข้อง 32
3 วิธีดำเนินงานวิจัย 37
3.1 กรอบแนวคิดการวิจัย 37
3.1.1 ขั้นตอนที่ 1: การแปลงข้อมูลให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้ 37
3.1.2 ขั้นตอนที่ 2: การสร้างแบบจำลองการวิเคราะห์ความรู้สึกจากข้อความ 38
3.2 ชุดข้อมูล 39
3.3 การออกแบบคุณลักษณะและแบบจำลอง 41
3.3.1 คุณลักษณะ TF-IDF สำหรับแปลงข้อมูลข้อความ 41
3.3.2 คุณลักษณะ Word Embedding สำหรับแปลงข้อมูลข้อความ 41
3.3.3 คุณลักษณะ Doc2Vec สำหรับแปลงข้อมูลข้อความ 42
3.3.4 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Deep Neural Network 42

สารบัญ (ต่อ)

หน้า

3.3.5 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ 44

3.3.6 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Convolutional Neural Network แบบ 2 มิติ 46

3.3.7 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Word Embedding และอัลกอริทึม Long Short-Term Memory.....48

3.3.8 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Gated Recurrent Unit 50

3.3.9 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Deep Neural Network.....52

3.3.10 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ 53

3.4 การฝึกสอนแบบจำลอง 54

3.5 เครื่องมือที่ใช้ในการวิจัย 54

4 ผลการทดลองและอภิปรายผล 56

4.1 ผลการทดสอบประสิทธิภาพ.....56

4.1.1 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Deep Neural Network 56

4.1.2 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ 57

4.1.3 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Convolutional Neural Network แบบ 2 มิติ 58

สารบัญ (ต่อ)

หน้า

4.1.4 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Long Short-Term Memory 58
4.1.5 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Gated Recurrent Unit 59
4.1.6 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และ อัลกอริทึม Deep Neural Network 60
4.1.7 การทดลองอัลกอริทึมวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และ อัลกอริทึม Convolutional Neural Network แบบ 1 มิติ 60
4.2 อภิปรายผล 62
5 สรุปและข้อเสนอแนะ 64
5.1 สรุปผลการวิจัย 64
5.2 การประยุกต์ผลการวิจัย 64
5.3 ข้อเสนอแนะ 65
รายการอ้างอิง 66
ภาคผนวก
ภาคผนวก ก. การใช้งาน Python เพื่อการประมวลผลข้อความและการเรียนรู้เชิงลึก 71
ภาคผนวก ข. บทความวิจัยตีพิมพ์ 79
ประวัติผู้เขียน 88

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางค่าต่าง ๆ ใน Confusion Matrix 31
2.2 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการพัฒนาแบบจำลอง การวิเคราะห์ความรู้สึก 35
3.1 ตัวอย่างชุดข้อมูลที่ใช้ในการวิจัย 40
3.2 แสดงการปรับค่าของคุณลักษณะ TF-IDF 41
3.3 แสดงการปรับค่าของคุณลักษณะ Glove (Word Embedding) 41
4.1 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ TF-IDF และใช้อัลกอริทึม DNN ในการพัฒนา 57
4.2 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ TF-IDF และใช้อัลกอริทึม CNN แบบ 1 มิติ ในการพัฒนา 57
4.3 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม CNN แบบ 2 มิติในการพัฒนา 58
4.4 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม LSTM ในการพัฒนา 59
4.5 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม GRU ในการพัฒนา 59
4.6 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Doc2Vec และใช้อัลกอริทึม DNN ในการพัฒนา 60
4.7 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Doc2Vec และใช้อัลกอริทึม CNN แบบ 1 มิติ ในการพัฒนา 61
4.8 ผลการทดสอบแบบจำลองโดยเฉลี่ยของการใช้คุณลักษณะแบบเมทริกซ์ในการทดลอง 62
4.9 ผลการทดสอบแบบจำลองโดยเฉลี่ยของการใช้คุณลักษณะแบบอนุกรมในการทดลอง 63

สารบัญรูป

รูปที่ **หน้า**

2.1 แสดงตัวอย่างการทำการทำเหมืองข้อความโดยการแปลงข้อมูลข้อความ ให้อยู่ในรูปที่คอมพิวเตอร์สามารถประมวลผลได้ 7

2.2 ตัวอย่างการทำความสะอาดข้อมูลข้อความโดยทำการ Lowercase และทำการลบ Punctuation 8

2.3 ตัวอย่างการทำ Word Tokenization 9

2.4 ตัวอย่างการสร้าง Token ในรูปแบบของ 2-gram 10

2.5 ตัวอย่างการสร้าง Token ในรูปแบบของ 3-gram 11

2.6 ตัวอย่างการเข้ารหัสด้วย One-Hot Encoding ของ Bag of Words 11

2.7 ตัวอย่างการใช้งาน Word Embedding ในการแปลงข้อมูลในรูปแบบ 2 มิติ 14

2.8 ตัวอย่างการคำนวณค่าความคล้ายคลึงกันของคำจากเวกเตอร์คุณลักษณะของ Word Embedding 15

2.9 ตัวอย่างการทำงานของ Doc2Vec ในรูปแบบ Distributed Memory Version of Paragraph Vector (PV-DM) 16

2.10 โครงสร้างโครงข่ายประสาทเทียมโดยมีส่วนย่อยในการคำนวณคือ Node และ การเรียงตัวของ Node เป็นชั้น เรียกว่า Layer 18

2.11 ตัวอย่างโครงสร้างของอัลกอริทึม Deep Neural Network 20

2.12 ตัวอย่างการทำงานของอัลกอริทึม CNN 21

2.13 การนำข้อมูลที่ได้จาก Filter มาผ่าน Max Pooling ขนาด 2×2 22

2.14 การนำข้อมูลที่ได้จาก Filter มาผ่าน Average Pooling ขนาด 2×2 22

2.15 CNN ในรูปแบบ 2 มิติ (ก) และ CNN ในรูปแบบ 1 มิติ (ข) 23

2.16 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ RNN 23

2.17 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ LSTM 25

2.18 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ GRU 28

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.1 กรอบแนวคิดวิธีการแปลงข้อมูล....38
3.2 กรอบแนวคิดการสร้างแบบจำลองสำหรับทำนายผล....39
3.3 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ TF-IDF และอัลกอริทึม DNN....43
3.4 โครงสร้างของอัลกอริทึม DNN ที่ใช้ในการวิจัย....43
3.5 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ TF-IDF และอัลกอริทึม CNN แบบ 1 มิติ....44
3.6 โครงสร้างของอัลกอริทึม CNN แบบ 1 มิติที่ใช้ในการวิจัย....45
3.7 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ Word Embedding และอัลกอริทึม CNN แบบ 2 มิติ....46
3.8 โครงสร้างของอัลกอริทึม CNN แบบ 2 มิติที่ใช้ในการวิจัย....47
3.9 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยอัลกอริทึม Word Embedding และ LSTM....48
3.10 โครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ LSTM ที่ใช้ในการวิจัย....49
3.11 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ Word Embedding และอัลกอริทึม GRU....50
3.12 โครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ GRU ที่ใช้ในการวิจัย....51
3.13 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม DNN....52
3.14 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความ ด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม CNN แบบ 1 มิติ....53
ก.1 ตัวอย่างการใช้งานภาษา Python....72
ก.2 ตัวอย่างการใช้งานฟังก์ชัน การกำหนดเงื่อนไข และการวนรอบ โดยใช้การเว้นระยะในการกำหนดขอบเขตการทำงาน....73
ก.3 ตัวอย่างโปรแกรม Jupyter Notebook....74

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ก.4 ตัวอย่างผลการรันโปรแกรมโดยใช้ Jupyter Notebook 75
ก.5 ตัวอย่างการรันโปรแกรมโดยดึงค่าตัวแปรจากเซลล์ที่ผ่านการรันมาแล้วไปใช้งาน 75
ก.6 ตัวอย่างโปรแกรม Google Colaboratory 76
ก.7 ตัวอย่างการใช้งาน TF-IDF จาก Library Scikit-Learn 77
ก.8 ตัวอย่างชุดคำสั่งสำหรับการสร้างคุณลักษณะ Doc2Vec 77
ก.9 ตัวอย่างการสร้างแบบจำลองโดยใช้ Keras 78

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหางานวิจัย

ปัจจุบันการใช้งานทางด้านเทคโนโลยีการสื่อสารได้มีการพัฒนาไปอย่างก้าวกระโดด ทำให้มีวิธีการติดต่อสื่อสารที่สะดวกและรวดเร็วขึ้นเป็นอย่างมาก โดยเฉพาะการพัฒนาของระบบอินเทอร์เน็ต ซึ่งได้รับความนิยมในการใช้งานในด้านต่าง ๆ ทั้งเป็นช่องทางในการให้ข้อมูลข่าวสาร ช่องทางการติดต่อสื่อสาร และช่องทางในการทำธุรกิจ เป็นต้น ทำให้อินเทอร์เน็ตเป็นช่องทางที่ทุกคนสามารถเข้าถึงข้อมูลข่าวสารจากแหล่งต่าง ๆ ได้อย่างรวดเร็ว

อินเทอร์เน็ตไม่ได้ถูกนำไปใช้งานเฉพาะการติดต่อสื่อสารเพียงเท่านั้น แต่ยังนำไปประยุกต์ใช้เพื่อสร้างมูลค่าทางธุรกิจได้อีกด้วย ตัวอย่างเช่นการเปิดช่องทางในการสั่งสินค้าผ่านทางหน้าเว็บไซต์ โดยที่ลูกค้าไม่ต้องไปเลือกซื้อสินค้าตามห้างสรรพสินค้า (Kowalczyk et al., 2002) ซึ่งเป็นการเพิ่มช่องทางและความสะดวกสบายให้แก่ลูกค้าในการเลือกซื้อสินค้าและบริการโดยที่ไม่ต้องเดินทางมาเลือกสินค้าตามห้างสรรพสินค้าหรือร้านค้าต่าง ๆ และผู้ขายสินค้าสามารถลดค่าใช้จ่ายในการบริหารร้านค้าของตนเอง

อย่างไรก็ตาม การใช้งานร้านค้าบนออนไลน์ผ่านระบบอินเทอร์เน็ตมีข้อจำกัดตรงที่ลูกค้าไม่สามารถทราบได้เลยว่าแท้จริงแล้วสินค้านั้นมีลักษณะอย่างไร ลูกค้าได้เห็นเพียงแต่รูปภาพที่ใช้ในการแนะนำสินค้าเพียงอย่างเดียว ทำให้ยากต่อการตัดสินใจซื้อ เพื่อลดข้อจำกัดนี้ ระบบการซื้อขายออนไลน์ส่วนใหญ่จึงมีส่วนที่ให้ลูกค้าที่ได้ซื้อสินค้าไปแล้วได้ทำการแสดงความคิดเห็นและให้คะแนนต่อสินค้าและบริการ ซึ่งเป็นการให้ข้อมูลเพื่อประกอบการตัดสินใจของลูกค้ารายใหม่ ข้อมูลความคิดเห็นเหล่านี้เมื่อมีปริมาณมากขึ้น การอ่านข้อมูลหรือการวิเคราะห์ข้อมูลโดยมนุษย์อาจใช้เวลามากเกินไป เทคนิคอัตโนมัติเช่นการทำเหมืองข้อมูลสามารถนำมาใช้ช่วยงานได้

การทำเหมืองข้อมูลนั้น เป็นการนำข้อมูลในอดีตมาวิเคราะห์เพื่อสร้างแบบจำลองที่สามารถนำแบบจำลองนั้นไปใช้ประโยชน์ในด้านต่าง ๆ ได้ในอนาคต (Rao et al., 2013) ซึ่งในทางธุรกิจนั้น การนำเสนอในสิ่งที่ลูกค้าสนใจเป็นสิ่งสำคัญต่อการตัดสินใจของลูกค้าที่จะเลือกซื้อผลิตภัณฑ์ การทำเหมืองข้อมูลเป็นการวิเคราะห์ความคิดเห็นต่อผลิตภัณฑ์ว่ามีความโน้มเอียงไปใน

ทิศทางใด เพื่อทำการวางแผนการบริหารและตัดสินใจได้อย่างแม่นยำ เพื่อเพิ่มมูลค่าหรือลดการขาดทุนจากการลงทุนในผลิตภัณฑ์นั้น ๆ

ข้อมูลการแสดงความคิดเห็นต่อผลิตภัณฑ์นั้น ไม่เพียงแต่ปรากฏอยู่บนระบบร้านค้าออนไลน์เท่านั้น แต่ยังรวมไปถึงระบบ Social Network ต่าง ๆ ในปัจจุบันอีกด้วย เช่น Facebook, Instagram และ Twitter เป็นต้น ซึ่งการแสดงความคิดเห็นผ่านช่องทางเหล่านี้สามารถกระจายข้อมูลได้อย่างรวดเร็วกว่าการแสดงความคิดเห็นบนร้านค้าออนไลน์อยู่มาก แต่เนื่องจากไม่มีการวัดคะแนนของการแสดงความคิดเห็น จึงอาจทำให้เกิดความเข้าใจผิดในการตีความต่อสินค้าและบริการได้ ดังนั้นการสำรวจและตรวจจับข้อความเหล่านั้นได้อย่างแม่นยำและทันเวลาจะช่วยในการนำเสนอและแก้ไขผลิตภัณฑ์ให้เหมาะสมกับความต้องการของลูกค้าได้ดียิ่งขึ้น

จากปัญหาข้างต้นสามารถแก้ไขได้โดยการให้คนทำการสังเกตเว็บไซต์ต่าง ๆ ซึ่งต้องใช้กำลังคนจำนวนมากและเสียเวลาในการตรวจจับ จึงได้มีการประยุกต์ใช้เทคนิคการทำเหมืองข้อมูลมาใช้ในการตรวจจับข้อความดังกล่าว เพื่อลดจำนวนกำลังคนที่ใช้ และลดเวลาในการตรวจจับให้มีประสิทธิภาพที่สุด โดยการใช้เทคนิคการทำเหมืองข้อมูลสามารถทำได้โดยการวิเคราะห์ข้อมูลเพื่อสร้างแบบจำลองสำหรับการตรวจจับ ซึ่งแบบจำลองที่ได้นั้น สามารถทำงานได้อย่างอัตโนมัติบนระบบคอมพิวเตอร์ และสามารถทำงานได้รวดเร็วกว่ามนุษย์เป็นอย่างมาก โดยในการสร้างแบบจำลองนั้น การใช้อัลกอริทึมการเรียนรู้ของเครื่องกำลังเป็นที่นิยมอย่างมาก โดยเฉพาะอัลกอริทึมการเรียนรู้เชิงลึก ที่มีประสิทธิภาพสูง อัลกอริทึมเหล่านี้สามารถทำให้คอมพิวเตอร์เรียนรู้สิ่งต่าง ๆ ได้เช่นเดียวกันกับที่มนุษย์เรียนรู้ โดยทำการเรียนรู้จากข้อมูลที่มีอยู่

จากความสำคัญของการวิเคราะห์ข้อความแสดงความคิดเห็นต่อสินค้าและบริการต่าง ๆ และประโยชน์จากการประยุกต์ใช้การเรียนรู้ของเครื่องที่กล่าวมาแล้วข้างต้นนั้น ผู้วิจัยจึงเสนอเทคนิคการสร้างแบบจำลองการทำนายจากข้อมูลที่อยู่ในรูปของข้อความแสดงความคิดเห็น เพื่อประสิทธิภาพและความแม่นยำในการทำนายที่สูง

1.2 วัตถุประสงค์ของการวิจัย

จากแนวคิดในการทำงาน ผู้วิจัยได้ตั้งวัตถุประสงค์ในการวิจัยดังนี้

1) เพื่อศึกษาและพัฒนาแบบจำลองด้วยเทคนิคการเรียนรู้เชิงลึกสำหรับวิเคราะห์ความรู้สึกสำหรับการจำแนกอารมณ์จากข้อความแสดงความคิดเห็นต่อสินค้า

2) เพื่อเปรียบเทียบอัลกอริทึม Deep Neural Network, Convolutional Neural Network, Long Short-Term Memory และ Gated Recurrent Unit ที่ใช้ในการสร้างแบบจำลอง การวิเคราะห์ความรู้สึก จากข้อความที่ผ่านการแปลงสภาพเรียบร้อยแล้ว

1.3 ขอบเขตการวิจัย

งานวิจัยนี้เป็นการศึกษาและพัฒนาแบบจำลอง การวิเคราะห์ความรู้สึก โดยการปรับปรุงทั้งในส่วนของการสร้างคุณลักษณะที่ใช้ในการฝึกสอนจากข้อความ และการออกแบบอัลกอริทึมที่ใช้สำหรับการฝึกสอนแบบจำลอง โดยกำหนดขอบเขตของการวิจัย ดังนี้

1) งานวิจัยนี้จัดทำและพัฒนาโดยใช้ข้อมูลที่อยู่ในรูปแบบของข้อความภาษาอังกฤษเท่านั้น

2) ข้อมูลที่นำมาใช้ในการวิจัย เป็นข้อมูลกรณีศึกษาการแสดงความคิดเห็นต่อสินค้า จาก Amazon.com ซึ่งทำการรวบรวมโดย Biltzer et al. (2007)

3) การเปรียบเทียบประสิทธิภาพการทำงานของแบบจำลอง จะทำการเปรียบเทียบในด้านของเวลาในการฝึกสอน และความแม่นยำของแบบจำลอง โดยใช้ข้อมูลชุดเดียวกัน

1.4 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับจากการศึกษาและพัฒนางานวิจัยนี้ ได้แก่

1) สามารถประยุกต์ใช้อัลกอริทึมในการแปลงข้อความ ให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้

2) การพัฒนาแบบจำลองโดยใช้อัลกอริทึมทางด้านการเรียนรู้เชิงลึก ให้ประสิทธิภาพในการทำนายที่สูง

3) ผู้ใช้สามารถเลือกอัลกอริทึมของการเรียนรู้เชิงลึกสำหรับการวิเคราะห์ ได้อย่างเหมาะสมกับงานของตน

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้ ประกอบด้วยการทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดของการวิเคราะห์ข้อมูลที่อยู่ในรูปของข้อความ ประวัติความเป็นมาของการประมวลผลภาษาธรรมชาติ การวิเคราะห์ความรู้สึก การทำเหมืองข้อความ เทคนิคการเรียนรู้เชิงลึก มาตรวัดประสิทธิภาพ และงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 การประมวลผลภาษาธรรมชาติ

การประมวลผลภาษาธรรมชาติ (Natural Language Processing) คือการวิเคราะห์ข้อมูลที่อยู่ในรูปของข้อความต่าง ๆ ให้คอมพิวเตอร์สามารถเข้าใจข้อความเหล่านั้นได้เช่นเดียวกันกับมนุษย์ (Chowdhury, 2003) ซึ่งประกอบด้วยหลายขั้นตอนย่อย ตั้งแต่ขั้นตอนการจัดเตรียมข้อมูล รวมไปถึงการแปลงข้อมูลให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถนำไปใช้ในการประมวลผลต่อได้ ซึ่งคอมพิวเตอร์สามารถทำความเข้าใจกับข้อมูลที่อยู่ในรูปแบบของตัวเลขที่สามารถนำไปคำนวณได้เท่านั้น

Natural Language Processing เป็นส่วนหนึ่งของงานทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) มีจุดเริ่มต้นโดยเป็นส่วนหนึ่งของภาษาศาสตร์ ใช้การวิเคราะห์ในรูปแบบของการตั้งกฎ (Rule-Based) (Schank and Abelson, 2013) ซึ่งเป็นการตั้งกฎขึ้นมาใช้ในการแก้ปัญหา มาประยุกต์ใช้ในงานด้านต่าง ๆ โดยสามารถสร้างได้โดยไม่ต้องอาศัยชุดข้อมูลจำนวนมาก ซึ่งมีข้อเสียในการวิเคราะห์คำใหม่ ๆ ที่เกิดขึ้นมา ทำให้ระบบไม่รู้จักคำเหล่านั้น และยังไม่สามารถจำแนกได้จะต้องมีการเพิ่มคำนั้นและกำหนดประเภทของคำนั้นเสียก่อน

จากการใช้งานด้วยการตั้งกฎ ต่อมาได้มีการประยุกต์ใช้การเรียนรู้ของเครื่อง (Machine Learning) ในการสร้างตัวจำแนกประเภทของข้อความ (Aone et al., 1998) โดยจะทำงานแตกต่างจากการตั้งกฎ โดยวิเคราะห์คำบางคำและทำการสรุปผลการวิเคราะห์ในทันทีเท่านั้น ซึ่งสามารถสร้างได้จากการรวบรวมข้อมูลและทำการกำหนดประเภทของข้อมูล จากนั้นทำการฝึกสอนโดยใช้อัลกอริทึมของการเรียนรู้ของเครื่อง

ในปัจจุบันการเรียนรู้ของเครื่องได้มีการพัฒนาไปเป็นการเรียนรู้เชิงลึก (Deep Learning) ซึ่งการเรียนรู้เชิงลึกมีจุดเด่นในการเรียนรู้คุณลักษณะต่าง ๆ ได้อย่างหลายหลาย และเรียนรู้ได้จำนวนมาก (Goodfellow et al., 2016) โดยสามารถทำการสร้างแบบจำลองในการจำแนกประเภทของข้อความได้อย่างแม่นยำมากยิ่งขึ้น ซึ่งได้มีการนำมาประยุกต์ใช้ทางด้าน Natural Language Processing ในด้านของการสร้างแบบจำลองสำหรับแปลภาษา (Machine Translation) โดยสามารถแปลได้อย่างถูกต้องและแม่นยำ สืบเนื่องด้วยความสามารถทางเทคโนโลยี และอัลกอริทึมที่สามารถวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพ ทำให้มีการพัฒนาการวิเคราะห์ทางด้านภาษาธรรมชาติเป็นไปอย่างก้าวกระโดด

2.2 การวิเคราะห์ความรู้สึก

การวิเคราะห์ความรู้สึก (Sentiment Analysis) คือการวิเคราะห์ข้อความ เพื่อค้นหาความรู้หรือข้อมูลเชิงลึกที่ซ่อนอยู่ เช่นอารมณ์ของผู้เขียน หัวข้อหรือคำสำคัญในข้อความ เป็นต้น การวิเคราะห์ความรู้สึกเป็นการประยุกต์ใช้งานอย่างหนึ่งของ Natural Language Processing โดยจะเป็นเทคนิคที่ใช้สำหรับการจำแนกข้อความออกเป็นกลุ่ม (Allouch, 2018) ซึ่งส่วนใหญ่จะเป็นการวิเคราะห์เพื่อหาว่าข้อความนั้นกำลังแสดงอารมณ์ในรูปแบบใด ซึ่งความรู้สึกโดยส่วนใหญ่ที่ใช้ในการวิเคราะห์ได้แก่ความรู้สึกในทางที่ดี ความรู้สึกในทางที่ไม่ดี ความรู้สึกปกติ และการคัดเลือกข้อความที่ดีที่สุดมาใช้งาน เป็นต้น การวิเคราะห์ความรู้สึกนั้นจะใช้เทคนิคทางด้านการทำเหมืองข้อความ (Text Mining) ซึ่งเป็นหนึ่งในเทคนิคของ Natural Language Processing ในการวิเคราะห์โดยจะเน้นไปทางด้านของการจำแนกประเภทของข้อความ (Text Classification) การวิเคราะห์ความรู้สึก นั้นได้มีการนำไปประยุกต์ใช้ในงานด้านต่าง ๆ โดยในปัจจุบันนั้นสามารถสังเกตได้อย่างง่ายที่สุดคือ การใช้งาน Platform การซื้อขายบนอินเทอร์เน็ต ที่มีการใช้ การวิเคราะห์ความรู้สึก ในการตัดสินความคิดเห็นต่อสินค้าว่าไปในทิศทางใด เป็นต้น

เนื่องจากระบบอินเทอร์เน็ตที่ได้รับการพัฒนาให้มีความรวดเร็วอย่างมากในปัจจุบัน และได้มีการประยุกต์ใช้อินเทอร์เน็ตในการทำธุรกิจต่าง ๆ เช่นการใช้งาน Social Media Platform (Täuscher and Laudien, 2018) การใช้งาน Platform การซื้อขายต่าง ๆ บนอินเทอร์เน็ต การวิเคราะห์ความรู้สึก จึงมีความสำคัญในการใช้งาน เพื่อประกอบการตัดสินใจในด้านต่าง ๆ ได้อย่างรวดเร็ว เช่นการป้องกันการแพร่ระบาดของข่าวปลอมบนระบบ Social Network หรือการแสดงผลการค้นหาสินค้าที่ดีที่สุดให้กับลูกค้าบนเว็บไซต์ โดยทำการวิเคราะห์จากการแสดงความคิดเห็นของผู้ใช้งานจริง เป็นต้น

การวิเคราะห์ความรู้สึกนั้นมีอยู่หลายประเภท โดยสามารถแบ่งได้ตามประเภทของการใช้งานได้แก่ Subjective Classification และ Sentiment Detection and Classification (AltexSoft, 2020)

2.2.1 Subjective Classification

Subjective Classification เป็นการวิเคราะห์ความรู้สึก เพื่อค้นหาหัวข้อของเนื้อหาข้อความตามที่ผู้วิเคราะห์ได้ทำการออกแบบไว้ เพื่อทำการจำแนกประเภทของเนื้อหาที่อยู่ในหมวดหมู่เดียวกัน โดยแบ่งตามหัวข้อ หรือวัตถุประสงค์ของข้อมูลเป็นส่วนใหญ่ ซึ่งการวิเคราะห์วิธีนี้ ได้ถูกนำไปประยุกต์ใช้ในงานด้านต่าง ๆ เช่น การตรวจจับหัวข้อของคำถามบนเว็บไซต์ StackOverflow เพื่อทำการจำแนกหมวดหมู่ให้ผู้เชี่ยวชาญทางด้านที่เกี่ยวข้องกับคำถามได้รับทราบ หรือการตรวจจับข้อความบทสนทนา เพื่อทำการตรวจสอบคำต่าง ๆ ว่าสื่อไปในทิศทางใด เป็นต้น

2.2.2 Sentiment Detection and Classification

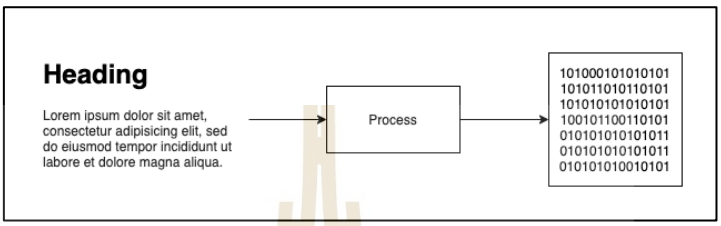
Sentiment Detection and Classification เป็นการตรวจจับความรู้สึกและวิเคราะห์ข้อมูลเพื่อทำการจำแนกตามกฎเกณฑ์ที่ได้ทำการสร้างขึ้นมา จากการวิเคราะห์ของผู้เชี่ยวชาญที่เกี่ยวข้องกับข้อมูลนั้น ๆ โดยจะเน้นไปที่เรื่องของการแสดงออกทางด้านอารมณ์ของผู้สร้างข้อความว่าไปในทิศทางใด โดยสามารถแบ่งย่อยออกเป็นระดับต่าง ๆ ได้ เช่น การจำแนกตามอารมณ์ โดยแบ่งอารมณ์ออกเป็นทางด้านที่ดีและไม่ดี เป็นต้น ซึ่งได้มีการนำไปประยุกต์ใช้กับงานด้านต่าง ๆ ได้แก่ การจำแนกบทความข่าวบนอินเทอร์เน็ตว่าเป็นข่าวจริงหรือเท็จ การจำแนกอารมณ์ของผู้เขียนจากข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ เป็นต้น

2.3 การทำเหมืองข้อความ

การทำเหมืองข้อความ (Text Mining) คือเทคนิคการนำข้อมูลที่อยู่ในรูปของข้อความมาใช้ในการวิเคราะห์ด้วยเทคนิคการทำเหมืองข้อมูล และเทคนิคทางด้านสถิติมาใช้ในการวิเคราะห์เพื่อหาความรู้ หรือสาระสำคัญที่ซ่อนอยู่ในข้อมูล เพื่อประกอบการตัดสินใจที่เหมาะสมกับสถานการณ์ (Tan, 1999) โดยเริ่มต้นมาจากงานทางด้านภาษาศาสตร์ นักภาษาศาสตร์ได้ทำการคิดค้นวิธีการในการวิเคราะห์เพื่อดึงความรู้ออกมาจากข้อความหรือเอกสารต่าง ๆ เพื่อนำมาใช้เป็นข้อมูลในการศึกษาพฤติกรรมเพื่อใช้ในการตัดสินใจ

การทำเหมืองข้อความนั้นเป็นการประยุกต์ใช้ความรู้ทางด้านสถิติและความรู้ทางด้านคอมพิวเตอร์มาใช้ในการวิเคราะห์ข้อความต่าง ๆ เพื่อให้คอมพิวเตอร์สามารถทำความเข้าใจข้อมูลได้เช่นเดียวกับมนุษย์ ยิ่งไปกว่านั้นคอมพิวเตอร์สามารถวิเคราะห์ข้อมูลได้รวดเร็ว และสามารถวิเคราะห์เพื่อหาความรู้ที่มนุษย์ไม่สามารถค้นหาในเวลาอันสั้นได้ โดยเฉพาะในปัจจุบันที่ข้อมูลนั้นได้ถูกสร้างขึ้นเป็นจำนวนมาก และสร้างขึ้นมาอย่างรวดเร็วกว่าในอดีต จึงต้องมีการวิเคราะห์และ

ประมวลผลที่สามารถทำงานได้อย่างรวดเร็วและสะดวกต่อการใช้งาน จึงได้นำคอมพิวเตอร์เข้ามาช่วยในการจัดการข้อมูลจำนวนมากเหล่านี้ ซึ่งการนำคอมพิวเตอร์มาใช้ในการวิเคราะห์นั้น จะต้องทำการจัดการข้อมูลให้อยู่ในรูปที่คอมพิวเตอร์สามารถทำความเข้าใจได้เสียก่อน จึงสามารถนำไปใช้งานต่อได้ ดังแสดงในรูปที่ 2.1



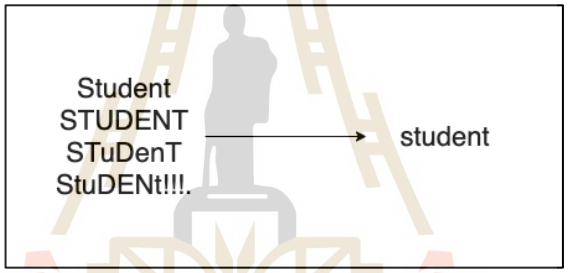
รูปที่ 2.1 แสดงตัวอย่างการทำการทำเหมืองข้อความโดยการแปลงข้อมูลข้อความให้อยู่ในรูปที่คอมพิวเตอร์สามารถประมวลผลได้

การทำเหมืองข้อความได้รับความนิยมเนื่องจากข้อมูลในปัจจุบันอยู่ในรูปของข้อมูลที่มีโครงสร้างไม่แน่นอน (Unstructured data) ถูกสร้างขึ้นเป็นจำนวนมาก สืบเนื่องมาจากความนิยมในการใช้งานเว็บไซต์และเครื่องมือต่าง ๆ บนอินเทอร์เน็ต ทั้งในด้านของการทำธุรกิจ และการติดต่อสื่อสาร ซึ่งทำให้เกิดการสร้างข้อมูลขึ้นเป็นจำนวนมาก ที่สามารถนำมาสร้างประโยชน์ในด้านต่าง ๆ เพื่อตอบสนองต่อความต้องการทางด้านธุรกิจในการเพิ่มมูลค่าให้แก่สินค้าและบริการ โดยอาศัยการวิเคราะห์ผลตอบรับของลูกค้าที่ได้ทำการใช้งาน เพื่อปรับปรุงสินค้าและผลิตภัณฑ์ของตนเองต่อไป ซึ่งวิธีการทำเหมืองข้อความ นั้นจะถูกนำไปประยุกต์ใช้เพื่อค้นหารูปแบบ (Pattern) ในชุดข้อมูลที่อยู่ในรูปแบบของข้อความหรือเอกสารต่าง ๆ ออกมาเป็นความรู้ที่สามารถนำไปใช้งานได้ (Singh and Singh, 2010)

การนำข้อมูลที่อยู่ในรูปแบบข้อความไปใช้ในการวิเคราะห์นั้น จะมีขั้นตอนพื้นฐานในการจัดเตรียมข้อมูลดังเช่นชุดข้อมูลอื่น ๆ ที่ใช้ในการวิเคราะห์ แต่จะมีวิธีการในการจัดเตรียมข้อมูลที่แตกต่างกันออกไป ตามลักษณะงานที่ใช้ในการวิเคราะห์ ซึ่งงานวิจัยนี้เป็นการทำการวิเคราะห์ความรู้สึก จะเป็นการสร้างคุณลักษณะ (Features) เพื่อนำไปใช้ในการวิเคราะห์ โดยทางผู้วิจัยได้ทำการศึกษาขั้นตอนในการจัดเตรียมข้อมูลในรูปแบบต่าง ๆ ได้แก่ การทำความสะอาดข้อมูล (Text Cleaning), Word Tokenize, Bag of Words, Term Frequency – Invert Document Frequency (TF-IDF), Word Embedding และ Doc2Vec เป็นต้น

2.3.1 การทำความสะอาดข้อความ

การทำความสะอาดข้อความ (Text Cleaning) เป็นการจัดเตรียมข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ได้อย่างถูกต้อง หรือการทำความสะอาดชุดข้อมูล ถือว่าเป็นหนึ่งในขั้นตอนสำคัญของการจัดการชุดข้อมูลที่เราจะนำไปใช้ในการวิเคราะห์ เพื่อทำการสร้างประสิทธิภาพในการวิเคราะห์ที่ดี และลดความซับซ้อนในการวิเคราะห์ของแบบจำลอง (Nader, 2019) โดยในการวิเคราะห์รูปแบบของ Text Mining นั้น จะมีขั้นตอนในการทำความสะอาดในรูปแบบต่าง ๆ โดยทางผู้วิจัยได้ทำการเลือกวิธีการทำความสะอาดข้อมูลโดยมีพื้นฐานจากภาษาอังกฤษ ได้แก่ การปรับตัวอักษรให้เป็นตัวพิมพ์เล็กทั้งหมด เนื่องจากคอมพิวเตอร์จะเข้าใจตัวอักษรนั้นในรูปแบบที่ไม่เหมือนกัน เช่น ‘A’ กับ ‘a’ เป็นต้น และการลบ Punctuation หรือสัญลักษณ์ต่าง ๆ ออกไปจากชุดข้อมูล ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างการทำความสะอาดข้อมูลข้อความโดยทำการ Lowercase และทำการลบ Punctuation

การทำ Lowercase หรือว่าการปรับตัวอักษรเป็นตัวพิมพ์เล็กนั้น ถูกนำมาใช้เนื่องจากคอมพิวเตอร์นั้นตีความหมายของตัวอักษรพิมพ์ใหญ่และตัวอักษรพิมพ์เล็กที่แตกต่างกัน ดังเช่นการตีความของตัวอักษร ‘A’ และ ‘a’ ซึ่งมีค่า ASCII ที่แตกต่างกัน โดย ‘A’ มีค่าในรหัส ASCII อยู่ที่ 65 และ ‘a’ มีค่าในรหัส ASCII อยู่ที่ 97 เป็นต้น ซึ่งจะต้องทำการปรับเพื่อให้คอมพิวเตอร์สามารถเข้าใจคำในลักษณะที่เหมือนกันได้ โดยจะต้องทำการแปลงข้อความให้อยู่ในลักษณะเดียวกันทั้งหมด

Punctuation ในภาษาอังกฤษคือ เครื่องหมาย และสัญลักษณ์ต่าง ๆ ที่ใช้งานในบริบทของแต่ละประโยค โดยขั้นตอนนี้เป็นขั้นตอนในการลบ Punctuation ออกไปเนื่องจากเป็น

ส่วนที่มีความหมายและความสำคัญในการวิเคราะห์ที่ต่ำ ไม่เหมาะสมกับการนำมาใช้ในการวิเคราะห์

สำหรับขั้นตอนอื่น ๆ ที่ใช้ในการทำความสะอาดข้อมูลข้อความให้สามารถนำไปใช้งานได้อย่างถูกต้องนั้น จะต้องทำการตรวจสอบในเรื่องของการสะกดคำ เพื่อลบข้อมูลคำศัพท์ที่ไม่ถูกต้องออกไปจากชุดข้อมูล การใช้คำที่เป็นรากศัพท์ หรือคำที่มีความหมายโดยตรง การลบ Prefix-Subfix ออกจากคำหลักในภาษาอังกฤษ เป็นต้น

หลังจากการทำความสะอาดคำไปในบางส่วนแล้ว Word Tokenization เป็นขั้นตอนแรกเริ่มในการนำข้อมูลประเภทข้อความมาใช้ในการวิเคราะห์ โดยทำการแยกคำออกจากประโยค ซึ่งคำที่แยกออกมาเป็นหน่วยย่อยเรียกว่า Token ซึ่งเป็นส่วนย่อยที่สุดในการนำไปใช้ในการวิเคราะห์

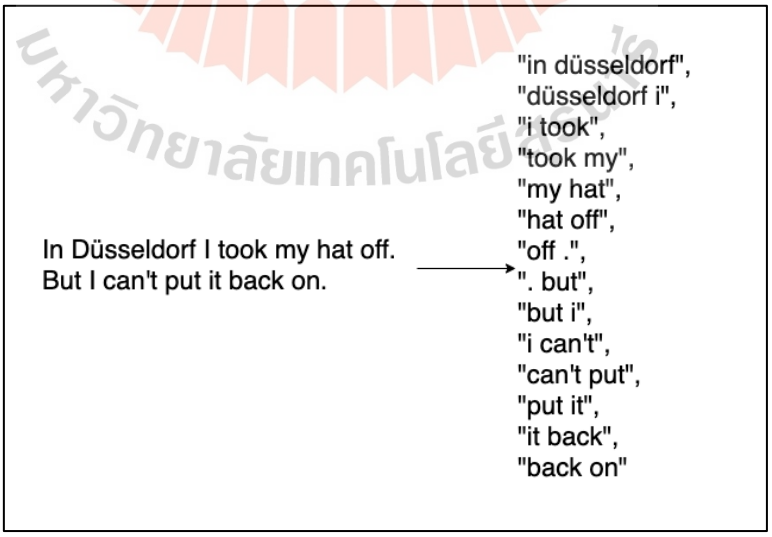
2.3.2 Word Tokenization

Word Tokenization คือขั้นตอนการจัดเตรียมข้อมูลข้อความโดยทำการแยกคำออกจากกันในประโยค ให้อยู่ในรูปของคำเดี่ยวหรือกลุ่มคำที่มีจำนวนตามที่เราสนใจ โดยเรียกว่า Token จากนั้นทำการรวมกลุ่มคำที่แตกต่างกัน และทำการนับจำนวนคำเหล่านั้นที่ปรากฏอยู่ในชุดข้อมูล ซึ่งสามารถนำ Token ที่ได้ไปใช้ในการสกัดความรู้ออกจากชุดข้อมูลในขั้นตอนต่อ ๆ ไป (Nayak et al., 2016) ดังแสดงในรูปที่ 2.3

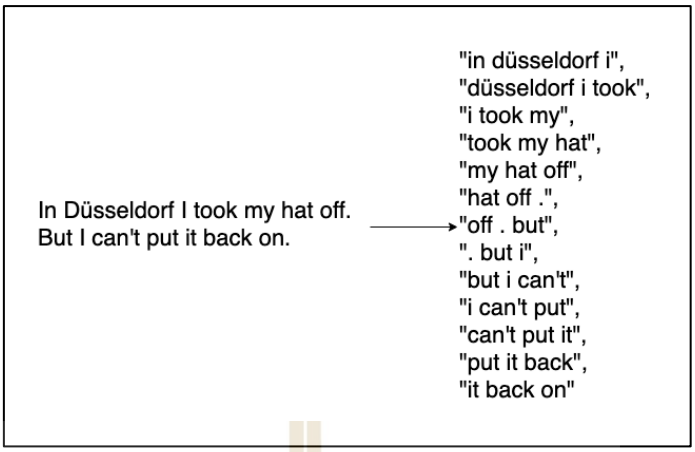
รูปที่ 2.3 ตัวอย่างการทำ Word Tokenization

จากรูปที่ 2.3 เป็นการสร้าง Token ขึ้นจากประโยคหรือชุดข้อมูล โดยทำการแบ่งคำแต่ละคำออกเป็น Token ย่อย โดยในตัวอย่างนี้เป็นการแบ่งคำที่อยู่ในภาษาอังกฤษ ซึ่งสามารถแบ่งคำได้โดยใช้การเว้นช่องว่างของคำแต่ละคำ เป็นตัวกำหนดแบ่ง Token แต่ละตัว โดยผลลัพธ์ที่ได้จากการทำ Word Tokenization คือ Token ของคำศัพท์ในชุดข้อมูลแต่ละตัว แต่ยังไม่สามารถนำไปใช้ในการวิเคราะห์ได้ทันที

การทำ Word Tokenization นั้น พยางค์หรือคำไม่เพียงแต่มีความหมายในตัวมันเองเท่านั้น แต่เมื่อเรานำคำศัพท์มาต่อกัน ซึ่งทำให้เกิดความหมายของคำใหม่ที่แตกต่างจากเดิม ความหมายที่เฉพาะเจาะจงมากขึ้น หรือสามารถเพิ่มความสำคัญให้กับข้อความหรือประโยคนั้นในงานที่เรากำลังสนใจอยู่ ซึ่งเราสามารถทำการวิเคราะห์ชุดรูปแบบเหล่านี้ได้ โดยทำการวิเคราะห์ในรูปแบบของ N-gram (Cavnar and Trenkle, 1994) ซึ่งเป็นการกำหนดการตัดคำในรูปของคำที่อยู่ติดกัน ให้อยู่ในรูปของ Token โดยจะถือว่าคำ 2 คำที่อยู่ติดกันนั้น จะไม่ใช่ตัวเดียวกันกับ Token ที่มาจากคำที่แยกกันและนำมารวมกัน โดย N-gram ที่นิยมใช้ในการวิเคราะห์นั้น ได้แก่ 1-gram หรือ Mono-gram เป็นการวิเคราะห์คำแต่ละคำ ดังแสดงในรูปที่ 2.3, 2-gram หรือ Bi-gram เป็นการวิเคราะห์คำที่อยู่ติดกัน 2 คำ ดังแสดงในรูปที่ 2.4 และ 3-gram หรือ Tri-gram เป็นการวิเคราะห์คำที่อยู่ติดกัน 3 คำ ดังแสดงในรูปที่ 2.5 เป็นต้น โดยในการสร้าง Word Tokenization จะต้องทำการวิเคราะห์ข้อมูลข้อความ ในมุมมองที่หลากหลาย จึงต้องมีการออกแบบคลังคำศัพท์ที่ได้จากการทำ Word Tokenization ทั้งในรูปแบบของ 1-gram, 2-gram และ 3-gram เป็นต้น



รูปที่ 2.4 ตัวอย่างการสร้าง Token ในรูปแบบของ 2-gram



รูปที่ 2.5 ตัวอย่างการสร้าง Token ในรูปแบบของ 3-gram

2.3.3 Bag of Words

Bag of Words เป็นวิธีการทาง Natural Language Processing ในการจัดเตรียมข้อมูลเพื่อนำไปใช้ในการวิเคราะห์บนระบบคอมพิวเตอร์ โดย Bag of Words นั้นเป็นวิธีการในการสร้างคุณลักษณะของข้อความขึ้นมาโดยใช้หลักการของ One-Hot Encoding (Zhang et al., 2010) ในการเข้ารหัสข้อมูลในทุกคำของชุดข้อมูล โดยในการเข้ารหัสนั้นเป็นการเข้ารหัสผ่าน Token ที่ได้ทำการสร้างเอาไว้แล้ว โดยแทนค่า 1 เมื่อคำเหล่านั้นปรากฏขึ้นในชุดข้อมูล และ 0 เมื่อคำเหล่านั้นไม่ปรากฏในชุดข้อมูล ดังแสดงในรูปที่ 2.6

	i	am	a	student	at	medical	school	thais
I am a student at medical school	1	1	1	1	1	1	1	0
I am a medical student	1	1	1	1	0	1	0	0
I am thais student	1	1	1	1	0	0	0	1

รูปที่ 2.6 ตัวอย่างการเข้ารหัสด้วย One-Hot Encoding ของ Bag of Words

ในยุคเริ่มต้นของการวิเคราะห์ Natural Language Processing นั้น การใช้งาน Bag of Words เป็นการประยุกต์ใช้งานการวิเคราะห์ทางด้านสถิติ ในการสร้างคุณลักษณะสำหรับนำไปใช้ในการวิเคราะห์ด้วยระบบคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์สามารถทำความเข้าใจคำศัพท์

โดย Bag of Words นั้นจะเป็นมุมมองของจำนวนครั้งของคำที่มีการนำไปใช้งานในชุดข้อมูล ซึ่งในการสร้างแบบจำลองการจำแนกประเภทนั้น จะต้องทำการค้นหาความสัมพันธ์ของคำต่าง ๆ กับสิ่งที่ต้องการจำแนก โดย Bag of Words สามารถนำไปใช้งานร่วมด้วยได้อย่างมีประสิทธิภาพ

Bag of Words เป็นวิธีการที่ใช้ในการแปลงข้อความให้อยู่ในรูปที่คอมพิวเตอร์นำไปใช้ประมวลผลได้ที่ง่ายที่สุด และยังเป็นพื้นฐานของอัลกอริทึมอื่น ๆ โดย Bag of Words เป็นการนำ Token ที่ได้จากการทำ Word Tokenization มาประยุกต์ใช้ร่วมกับความรู้ทางด้านสถิติในการรวบรวมข้อมูลของคำในข้อความโดยทำการสร้างคลังคำศัพท์จากชุดข้อมูลที่เราจะใช้ในการวิเคราะห์ โดยส่วนใหญ่คลังคำศัพท์นั้นจะถูกจัดเรียงตามลำดับของ Token ที่มีจำนวนมากที่สุดไปน้อยที่สุด จากนั้นทำการกำหนดคีย์หลักสำหรับแทนค่าของ Token นั้น ๆ เพื่อนำไปใช้ในการวิเคราะห์ต่อไป

การนำ Bag of Words ไปใช้ประโยชน์มีหลากหลายรูปแบบด้วยกัน โดยสามารถนำไปใช้ประโยชน์ทั้งในด้านของการวิเคราะห์ข้อมูล โดยทำการสร้างคลังคำศัพท์ขึ้นเพื่อดูว่ามีคำหรือกลุ่มคำใดที่มีการใช้งานมากที่สุด และทำการสร้างแบบจำลองอย่างง่ายจากข้อมูลที่มีอยู่ ด้วยวิธีการใช้ Rule Based ในการสร้างแบบจำลอง

การใช้งาน Bag of Words นั้นยังติดปัญหาในส่วนของการนับจำนวนคำศัพท์อย่างเดียว แต่ไม่ได้นับในส่วนของจำนวนเอกสารที่คำนั้นปรากฏ ซึ่งทำให้ติดปัญหาในส่วนของคำที่ปรากฏเป็นจำนวนมากในเอกสารเพียงแค่ชุดเดียวหรือจำนวนน้อย ๆ ทำให้การวิเคราะห์ข้อมูลออกมาแล้วสูญเสียความรู้บางส่วนในชุดข้อมูล จึงได้มีการคิดค้นอัลกอริทึมขึ้นมาใช้ในการแก้ปัญหาของ Bag of Words โดยมีชื่อว่า Term Frequency – Invert Document Frequency โดยรายละเอียดของอัลกอริทึมนี้อธิบายไว้ในหัวข้อถัดไป

2.3.4 Term Frequency - Invert Document Frequency

Term Frequency – Invert Document Frequency (TF-IDF) ได้มีการพัฒนามาจาก Bag of Words เป็นอัลกอริทึมหนึ่งที่ถูกออกแบบขึ้นมาสำหรับใช้งานในส่วนของการแปลงข้อมูลที่อยู่ในรูปแบบของข้อความ ให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถทำความเข้าใจและนำไปใช้ประโยชน์ต่อได้ โดย TF-IDF นั้นถูกคิดค้นขึ้นมาเพื่อใช้แก้ปัญหาในส่วนของคำที่ปรากฏเป็นจำนวนมากในข้อมูลแค่เพียง 1 ระเบียนที่เกิดขึ้นจากการนำ Bag of Words ไปใช้งาน

TF-IDF ใช้หลักการคล้ายคลึงกับ Bag of Words ในการสร้างคลังคำศัพท์ แต่คุณลักษณะจาก TF-IDF จะมีจุดเด่นที่เหนือกว่าในส่วนที่ TF-IDF ไม่เพียงแต่นับจำนวนคำศัพท์ที่สามารถพบมากที่สุดในชุดข้อมูลเท่านั้น แต่ยังมีการวิเคราะห์ไปถึงความสำคัญของคำที่ปรากฏอยู่

ในชุดข้อมูลอีกด้วย (Ramos, 2003) ซึ่งการคำนวณคะแนนความสำคัญของแต่ละ Token นั้น จะมีการคำนวณดังสมการที่ 2.1 และ 2.2 ดังนี้

$$TF(t,d) = log(1 + freq(t,d)) \tag{2.1}$$

เมื่อ $TF(t,d)$ แทนฟังก์ชัน Term Frequency

t แทนคำที่ปรากฏอยู่ใน Document

d แทนคำศัพท์ใน Document ที่เราสนใจ

$freq(t,d)$ แทนความถี่ของคำศัพท์ที่เราสนใจ ต่อคำศัพท์ทั้งหมดใน Document

$$IDF(t,D) = log\left(\frac{|D|}{|\{d \in D | t \in d\}|}\right) \tag{2.2}$$

เมื่อ $IDF(t,D)$ แทนฟังก์ชัน Invert Document Frequency

t แทนคำที่ปรากฏอยู่ใน Document

d แทนคำศัพท์ใน Document ที่เราสนใจ

D แทน Document ทั้งหมดในชุดข้อมูล

สมการข้างต้นสามารถแบ่งการคำนวณออกเป็น 2 ส่วนได้แก่ ส่วน Term Frequency ดังสมการที่ 2.1 และส่วน Invert Document Frequency ดังสมการที่ 2.2 โดยในส่วนของ Term Frequency นั้นจะทำการคำนวณคะแนนของ Token แต่ละตัวที่ได้ทำการสร้างขึ้นมา เพื่อทำการลำดับความสำคัญของ Token แต่ละตัวว่ามีความสำคัญอย่างไร จากนั้นทำการคำนวณ Invert Document Frequency เพื่อค้นหาว่าคำศัพท์นั้นไปปรากฏอยู่มากน้อยเท่าใดในชุดข้อมูลทั้งหมด ของ Token จากนั้นจะทำการนำค่าที่ได้จากทั้ง 2 สมการมาทำการคำนวณเพื่อหาค่าคะแนนความสำคัญของ Token เพื่อทำการจัดลำดับความสำคัญ ดังสมการที่ 2.3

$$TFIDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (2.3)$$

เมื่อ $TFIDF(t, d, D)$ แทนฟังก์ชัน Term Frequency – Invert Document Frequency

$TF(t, d)$ แทนฟังก์ชัน Term Frequency

$IDF(t, D)$ แทนฟังก์ชัน Invert Document Frequency

t แทนคำที่ปรากฏอยู่ใน Document

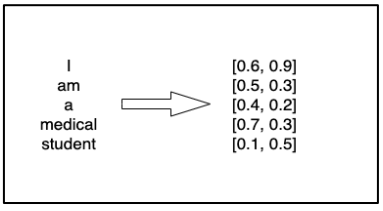
d แทนคำศัพท์ใน Document ที่เราสนใจ

D แทน Document ทั้งหมดในชุดข้อมูล

2.3.5 Word Embedding

Word Embedding คือการสร้างเวกเตอร์คุณลักษณะ (Feature Vector) ขึ้นมาจาก Token ที่เราได้ทำการสร้างไว้ โดยทำการสร้างเวกเตอร์คุณลักษณะขึ้นมาจากประโยคหรือเอกสารที่มีอยู่ในข้อมูลของเรามาเพื่อทำการสร้างคุณลักษณะที่อยู่ในรูปของตัวเลขที่สามารถนำไปใช้ในการคำนวณต่อได้ ซึ่งจุดเด่นของเวกเตอร์คุณลักษณะที่ได้จาก Word Embedding นั้นจะสามารถนำไปใช้คำนวณความคล้ายคลึงกับคำอื่น ๆ ในบริบทของคำที่แตกต่างกันได้ (Ganguly et al., 2015)

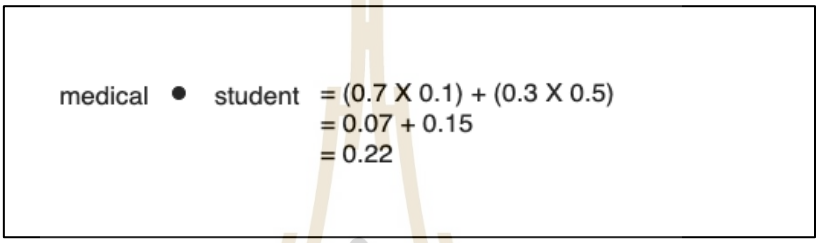
Word Embedding จะทำการสร้างเวกเตอร์คุณลักษณะโดยเริ่มจากการเข้ารหัสคำแต่ละคำให้อยู่ในรูปที่คอมพิวเตอร์สามารถทำความเข้าใจได้ก่อนด้วยวิธีการ One-Hot Encoding โดย One-Hot Encoding นั้นจะทำงานโดยการนำจำนวนคำที่ปรากฏขึ้นมาในชุดข้อมูล และนำประโยคในชุดข้อมูลหรือเอกสารที่เราได้ทำการกำหนดไว้ในชุดข้อมูลมาเข้ารหัส ทำให้ได้เวกเตอร์ตามจำนวนคำในประโยคที่กำหนดให้อยู่ในรูปของบิต จากนั้นทำการรวมเวกเตอร์ที่ได้จาก One-Hot Encoding ซึ่งสามารถกำหนดจำนวน Dimension หรือ Features ของ Vector ได้ ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการใช้งาน Word Embedding ในการแปลงข้อมูลในรูปแบบ 2 มิติ

จากรูปที่ 2.7 ผลของการทำ Word Embedding นั้นจำได้มาจากการคำนวณโดยนำคำศัพท์แต่ละคำมาทำการ One-Hot Encoding จากนั้นทำการทำการรวมคำหรือส่วนซ้ำกัน เพื่อสร้างเวกเตอร์คุณลักษณะโดยเราทำการกำหนดมิติของเวกเตอร์ที่เราต้องการได้

จากผลของเวกเตอร์คุณลักษณะที่ได้จาก Word Embedding นั้น ในการวิเคราะห์ความคล้ายคลึงกันหรือความใกล้เคียงของคำนั้น จะสามารถหาความสัมพันธ์ได้ผลคูณของเวกเตอร์คุณลักษณะของคำ เพื่อหาค่าของความคล้ายคลึงกันของคำ (Word Similarity) ดังแสดงในรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างการคำนวณค่าความคล้ายคลึงกันของคำจากคุณลักษณะของเวกเตอร์ของ Word Embedding

จุดอ่อนของ Word Embedding คือการที่คำหนึ่งคำสามารถมีเวกเตอร์คุณลักษณะได้เพียงแค่ 1 ชุดเท่านั้น แต่คำนั้นสามารถมีความหมายในการนำเสนอได้หลายความหมาย เวกเตอร์คุณลักษณะที่ได้จาก Word Embedding จึงไม่สามารถนำเสนอความหมายของคำในทุกกรณีได้

ในปัจจุบันได้มีการพัฒนาแบบจำลองการทำ Word Embedding ต่าง ๆ ขึ้นมาให้เลือกใช้งานมากมาย ซึ่งแต่ละแบบจำลองนั้นจะมีจุดเด่นต่าง ๆ ที่ไม่เหมือนกัน โดยผู้วิจัยได้ทำการยกตัวอย่างมา 2 แบบ ได้แก่ Word2Vec และ GloVe

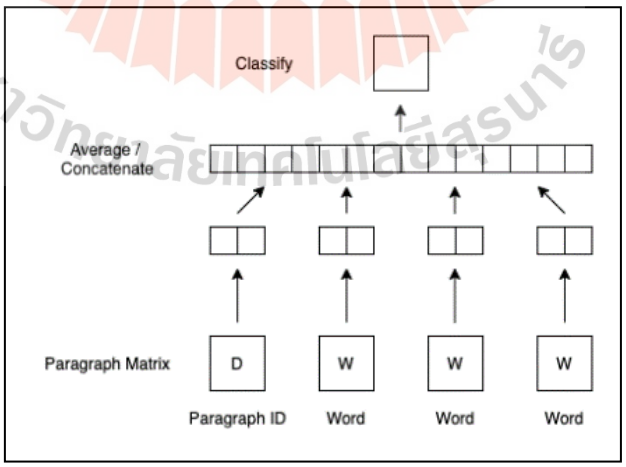
Word Embedding แบบแรกที่ได้ทำการพัฒนาขึ้นได้แก่ Word2Vec ซึ่งพัฒนาโดย Google โดยมี Mikolov เป็นหัวหน้าในการพัฒนา (Mikolov et al, 2013) โดย Word2Vec ได้รับการออกแบบโครงสร้างให้มีโครงข่ายประสาทเทียม 2 ชั้น โดยมีการนำข้อมูลจำนวนมากมาใช้ในการฝึกสอนให้ได้เวกเตอร์คุณลักษณะออกมา โดยถูกปล่อยออกมาให้ได้ใช้งานครั้งแรกในปี 2013 Word2Vec ส่วนมากถูกนำไปใช้กับงานทางด้าน Language Modeling ซึ่งเป็นแบบจำลองที่ใช้ในการทำนายคำถัดไปของประโยคว่าควรที่จะเป็นคำใด โดยอาศัยหลักการของ Continuous Bag of Words สำหรับการใช้คำหลาย ๆ คำต่อกัน เพื่อทำนายคำที่อยู่ถัดไป และ Continuous Skip-Gram สำหรับการใช้คำหนึ่งคำในการทำนายคำอื่น ๆ ที่มีโอกาสเป็นคำถัดไปจากคำนี้

จากความสำเร็จในการสร้าง Word2Vec ขึ้น ทางมหาวิทยาลัย Stanford ได้ทำการพัฒนาต่อยอดโดยใช้ชื่อว่า GloVe หรือ Global Vectors (Pennington et al., 2014) ในการสร้างเวกเตอร์คุณลักษณะโดยใช้การฝึกสอนในรูปแบบการเรียนรู้แบบไม่มีผู้ฝึกสอนสำหรับเรียนรู้คำศัพท์ในพัฒนาแบบจำลองและสร้างเวกเตอร์คุณลักษณะขึ้นมา โดยใช้หลักการของ Nearest Neighbors ในการนำ Feature ที่ได้ไปใช้ค้นหาคำที่มีความหมายใกล้เคียงกัน และ Linear Substructures เพื่อทำการค้นหาคำที่ใช้ในหมวดหมู่เดียวกัน หรือมีความสัมพันธ์กันอยู่

2.3.6 Doc2Vec

Doc2Vec เป็นอัลกอริทึมที่ได้พัฒนาต่อยอดขึ้นมาจาก Word2Vec ของ Word Embedding โดยเปลี่ยนจากการเข้ารหัสคำแต่ละคำเพื่อสร้างเวกเตอร์คุณลักษณะ มาเป็นการเข้ารหัสข้อความหรือเอกสารทั้งหมด ให้อยู่ในรูปของเวกเตอร์คุณลักษณะ โดยตัวเอกสารนั้นจะมีโครงสร้างที่แตกต่างกับคำในส่วนที่เอกสารนั้น ไม่มีโครงสร้างทางตรรกะเช่นเดียวกันกับคำ ซึ่งเหมาะสมกับการนำไปใช้งานในด้านของการจำแนกประเภทของเอกสารต่าง ๆ

Doc2Vec นั้นใช้หลักการของ Word2Vec ในการเข้ารหัสข้อมูลเพื่อสร้างเวกเตอร์คุณลักษณะ แต่ได้มีการเพิ่มเติมในส่วนของ Paragraph ID เข้ามาใช้ในการสร้าง โดย Paragraph ID นั้นสร้างมาจากตัวเอกสารและทำให้มีเลขเฉพาะตัวในการนำไปใช้วิเคราะห์ จากนั้นจะทำงานโดยการทำงานในลักษณะดังกล่าวเรียกว่า Distributed Memory Version of Paragraph Vector (PV-DM) ดังแสดงในรูปที่ 2.9 (Le & Mikolov, 2014)



รูปที่ 2.9 ตัวอย่างการทำงานของ Doc2Vec ในรูปแบบ Distributed Memory Version of Paragraph Vector (PV-DM)

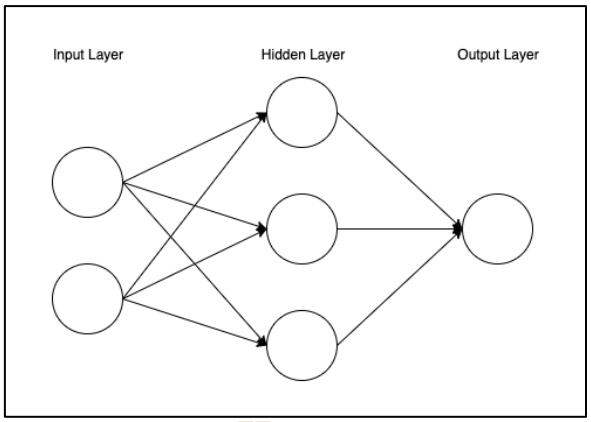
จุดเด่นของ Doc2Vec นั้นจะเป็นการแก้ปัญหาในด้านของการจำกัดจำนวนคุณลักษณะที่ใช้สำหรับการเรียนรู้ของแบบจำลอง เช่น Bag of Word และ TF-IDF ที่จะต้องกำหนดจำนวนคำศัพท์ที่สนใจในการสร้างเวกเตอร์คุณลักษณะ และการใช้งาน Word Embedding ที่จะต้องกำหนดความยาวของอนุกรม และจำนวนคำศัพท์ที่สนใจในการนำไปวิเคราะห์ เป็นต้น โดย Doc2Vec สามารถเรียนรู้คำต่าง ๆ เพิ่มเติมได้เองโดยอัตโนมัติ โดยไม่ต้องทำการกำหนดจำนวนคำศัพท์ที่สนใจในชุดข้อมูล เพราะ Doc2Vec นั้นเป็นการเข้ารหัสทั้งข้อความหรือเอกสารในชุดข้อมูล Doc2Vec นั้นได้รับความนิยมอย่างมากในการนำไปใช้งานทั้งในด้านของการจำแนกประเภทของข้อความเอกสาร หรือการประยุกต์ใช้ในงานด้านเครื่องจักรแปลภาษา เป็นต้น

2.4 เทคนิคการเรียนรู้เชิงลึก

เทคนิคการเรียนรู้เชิงลึก (Deep Learning) เป็นส่วนหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นอัลกอริทึมที่ใช้สำหรับการเรียนรู้ที่สามารถทำให้เครื่องจักรสามารถตัดสินใจได้เช่นเดียวกับมนุษย์ โดยการเรียนรู้ของเครื่องเป็นการประยุกต์ใช้ความรู้ทางด้านสถิติ ในการวิเคราะห์ข้อมูลและสร้างแบบจำลองสำหรับทำนายผลลัพธ์จากข้อมูล

จุดเริ่มต้นของการเรียนรู้เชิงลึก นั้นเริ่มมาจากโครงข่ายประสาทเทียม (Neural Network) เป็นอัลกอริทึมที่คิดค้นขึ้นมาจากการเลียนแบบการทำงานของสมองของมนุษย์ ซึ่งสมองของมนุษย์มีการทำงานที่ซับซ้อนและสามารถวิเคราะห์สิ่งต่าง ๆ เป็นจำนวนมากได้อย่างมีประสิทธิภาพ โดยโครงข่ายประสาทเทียมนั้นได้ทำการจำลองการทำงานของเซลล์ประสาทขึ้นมา ซึ่งแต่ละเซลล์ก็มีการเชื่อมต่อเพื่อส่งข้อมูลไปหาแต่ละเซลล์เพื่อใช้ในการตัดสินใจ จุดเด่นของสมองของมนุษย์คือการที่เซลล์ประสาทแต่ละเซลล์สามารถเชื่อมต่อกันได้อย่างทั่วถึง และมีการกระจายตัวในการวิเคราะห์ข้อมูลให้แต่ละเซลล์อย่างชัดเจน (Hecht-Nielsen, 1992)

โครงข่ายประสาทเทียมนั้นถูกออกแบบมาให้มีการทำงานคล้ายกับสมองของมนุษย์ โดยการทำงานเบื้องหลังของโครงข่ายประสาทเทียมนั้นมีหน่วยย่อยที่ทำงานคล้ายกับเซลล์ประสาทของมนุษย์เรียกว่า Node ซึ่ง Node สามารถรวมตัวกันจำนวนหนึ่งเรียงตัวเป็นชั้น จะเรียกว่า Layer โดยแต่ละ Node จะมีขั้นตอนการทำงานแบ่งหน้าที่ตาม Layer เช่น Input Layer, Hidden Layer และ Output Layer เป็นต้น ดังแสดงในรูปที่ 2.10



รูปที่ 2.10 โครงสร้างโครงข่ายประสาทเทียมโดยมีส่วนย่อยในการคำนวณคือ Node และ การเรียงตัวของ Node เป็นชั้น เรียกว่า Layer

จากรูปที่ 2.10 การทำงานของ Node นั้นมีพื้นฐานการทำงานจากทั้งในส่วนของ Linear Regression โดยสามารถแทนค่าได้ดังสมการที่ 2.4 โดย Node นั้นจะมีส่วนประกอบย่อยเรียกว่า Weight ซึ่งเปรียบเทียบได้กับค่า Intercept จาก Linear Regression ที่เอาไว้ใช้ในการกำหนดน้ำหนักของค่าตัวแปรแต่ละตัวที่ใช้ในการวิเคราะห์ และ Bias ซึ่งเปรียบได้กับค่า Coefficient ใน Linear Regression ซึ่งเปรียบ Node ได้เหมือนกับการทำงานของ Linear Regression โดยในขั้นตอนของการฝึกสอนนั้น จะทำการฝึกสอนเพื่อหาค่าพารามิเตอร์ที่เหมาะสมกับข้อมูลที่ได้ทำการนำมาใช้ฝึกสอน จากนั้นนำค่าพารามิเตอร์ที่ได้ไปใช้ในการสร้างแบบจำลองเพื่อใช้ในการทำนายผล โดยการฝึกสอน Neural Network นั้นจะต้องทำการฝึกสอนเป็นรอบ โดย 1 รอบนั้นจะต้องทำการฝึกสอนกับข้อมูลทุกตัวในข้อมูลที่ใช้สำหรับฝึกสอน ซึ่งเรียกจำนวนรอบในการฝึกสอนว่า Epoch

$$Neural\ Network(m,n) = activation(W_{mn}i_n + b_m) \tag{2.4}$$

เมื่อ $Neural\ Network(m,n)$ แทนผลลัพธ์ของแบบจำลองโครงสร้างโครงข่ายประสาทเทียม

$activation$ แทนฟังก์ชันที่ใช้ในการแปลงค่าที่ได้จากการคำนวณค่าน้ำหนักของโครงสร้างโครงข่ายประสาทเทียม

W_{mn}	แทนค่าน้ำหนักของ Node ในโครงสร้างโครงข่ายประสาทเทียม
i_n	แทนค่า Input ของโครงสร้างโครงข่ายประสาทเทียม
b_m	แทนค่า bias ของโครงสร้างโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมนั้นมีอยู่ด้วยกันหลายชนิด ซึ่งสามารถแบ่งออกได้ด้วยขั้นตอนการฝึกสอน โดยมีตัวอย่างของประเภทโครงสร้างโครงข่ายประสาทเทียมที่นิยมใช้ได้แก่ Feedforward Neural Network ที่มีการป้อนข้อมูลจากด้านหน้าไปด้านหลัง โดยมีตัวอย่างได้แก่ Perceptron เป็นต้น และ Backpropagation Neural Network ที่มีการป้อนข้อมูลและทำการเรียนรู้โดยป้อนค่าผลลัพธ์ที่ได้แบบย้อนกลับ

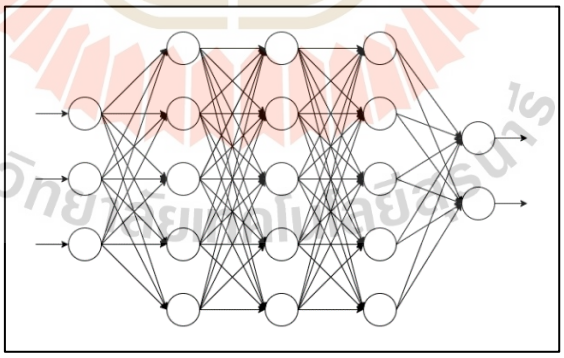
การเรียนรู้เชิงลึกเป็นหนึ่งในอัลกอริทึมที่ถูกพัฒนาต่อยอดมาจากโครงข่ายประสาทเทียม ซึ่งเป็นการเลียนแบบการเรียนรู้ของระบบประสาทของมนุษย์ โดยทำการต่อยอดด้วยการวางโครงสร้างในรูปแบบวางซ้อนกันหลาย ๆ ชั้น โดยมีทั้งการวางซ้อนในรูปแบบของประเภทเดียวกันในทุกชั้น และการวางซ้อนกันโดยแต่ละชั้นนั้นทำงานไม่เหมือนกัน ซึ่งเกิดจากการประยุกต์ใช้ความสามารถของแต่ละโครงสร้างที่นำมาใช้ร่วมกันเพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ที่ดียิ่งขึ้น (LeCun et al., 2015) ในปัจจุบันการเรียนรู้เชิงลึก ได้ให้ประสิทธิภาพในการวิเคราะห์และทำนายผลได้ดีกว่าการใช้อัลกอริทึมของการเรียนรู้ของเครื่องแบบเดิมเป็นอย่างมาก ทั้งในด้านของจำนวนข้อมูลที่เพิ่มขึ้นอย่างก้าวกระโดด และหน่วยประมวลผลของคอมพิวเตอร์ที่มีประสิทธิภาพสูงขึ้น ทำให้การเรียนรู้เชิงลึกนั้นสามารถใช้ตัวแปรจำนวนมากในการวิเคราะห์ เพื่อเพิ่มประสิทธิภาพในการวิเคราะห์ได้

ไม่เพียงแต่การเพิ่มจำนวนชั้นเท่านั้น การเรียนรู้เชิงลึกได้ทำการปรับความสามารถในการเรียนรู้ของอัลกอริทึม โดยทำการนำการเรียนรู้แบบ Backpropagation มาปรับปรุงโดยแยกอัลกอริทึมที่ใช้ในการเรียนรู้ออกเป็น 2 ตัว ได้แก่ Loss Function และ Optimize Function โดย Loss Function ใช้ในการคำนวณเพื่อหาค่าความผิดพลาดที่ได้จากการเปรียบเทียบระหว่างผลที่ได้จากแบบจำลอง และผลลัพธ์ที่ใช้ในการฝึกสอน จากนั้นนำค่า Loss ที่ได้มาใช้กับ Optimize Function ซึ่งเป็นฟังก์ชันสำหรับการปรับค่าพารามิเตอร์ที่ใช้ในการเรียนรู้ของแบบจำลองที่สร้างขึ้นมา โดยมีการคิดค้น Loss Function และ Optimize Function ต่าง ๆ ขึ้นมาให้เลือกใช้งานในปัจจุบัน

การเรียนรู้เชิงลึกนั้นได้มีการคิดค้นอัลกอริทึมและโครงสร้างต่าง ๆ ขึ้นมา เพื่อใช้ในการตอบสนองต่อความต้องการในการวิเคราะห์รูปแบบต่าง ๆ โดยอัลกอริทึมที่ได้รับความนิยมในปัจจุบันได้แก่ Deep Neural Network, Convolutional Neural Network, Recurrent Neural Network, Long Short-Term Memory และ Gated Recurrent Units เป็นต้น โดยความนิยมในการใช้งาน Deep Learning ที่เพิ่มขึ้นนั้นสืบเนื่องมาจากประสิทธิภาพในการวิเคราะห์ที่สูง และยังรวมไปถึงเทคโนโลยีทางด้าน Hardware ที่พัฒนาขึ้นไปอย่างก้าวกระโดดในปัจจุบัน ทั้งหน่วยประมวลผลที่มีประสิทธิภาพสูง และยังรวมไปถึงการประยุกต์ใช้ Graphic Processing Unit ในการใช้งานการเรียนรู้เชิงลึก ซึ่งสามารถประมวลผลได้รวดเร็วกว่าถูกประมวลผลด้วย Central Processing Unit (CPU) เนื่องจาก GPU นั้นประมวลผลในรูปแบบของ Matrix Parallels แต่ CPU นั้นประมวลผลในรูปแบบของ Serial (Dean et al., 2012)

2.4.1 Deep Neural Network

โครงข่ายประสาทเชิงลึก (Deep Neural Network: DNN) เป็นการนำอัลกอริทึมการเรียนรู้เชิงลึกไปใช้งานในรูปแบบที่ง่ายที่สุด โดยเป็นการต่อยอดมาจากโครงข่ายประสาทเทียมด้วยการเพิ่มจำนวนชั้นเข้าไปจำนวนหนึ่ง ทำให้มีตัวแปรที่ใช้สำหรับการคัดแยกคุณลักษณะที่มากขึ้นเพื่อเพิ่มประสิทธิภาพในการวิเคราะห์และทำนายผล (Han et al., 2016) ซึ่งในการทำงานของตัวโครงสร้างนั้นมีลักษณะเช่นเดียวกับโครงข่ายประสาทเทียมดังแสดงในรูปที่ 2.11



รูปที่ 2.11 ตัวอย่างโครงสร้างของอัลกอริทึม Deep Neural Network

DNN มีการทำงานและโครงสร้างที่คล้ายกันกับโครงข่ายประสาทเทียมมากที่สุด แต่ได้มีการเพิ่มจำนวนชั้นสำหรับการวิเคราะห์คุณลักษณะของข้อมูล โดยจำนวนที่มากขึ้นทำให้

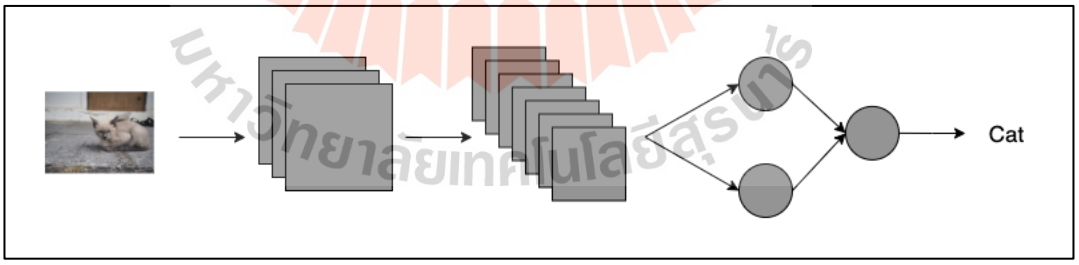
สามารถวิเคราะห์คุณลักษณะได้ละเอียดขึ้น และทำการฝึกสอนโดยใช้หลักการของ Backpropagation ในการฝึกสอน

DNN ได้ถูกนำไปใช้ในการวิเคราะห์ข้อมูลในรูปที่มีโครงสร้างแน่นอนโดยในการวิเคราะห์ข้อมูลของ DNN นั้น จะทำการวิเคราะห์โดยไม่ได้ทำลายตัวข้อมูลดั้งเดิม ทำให้ยังสามารถวิเคราะห์ได้อย่างแม่นยำและมีประสิทธิภาพที่สูง

2.4.2 Convolutional Neural Network

โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network: CNN) เป็นอัลกอริทึมของการเรียนรู้เชิงลึก โดยทำงานคล้ายกับการกวาดสายตามองของมนุษย์โดยจะทำการแบ่งกลุ่มของคุณลักษณะออกไปวิเคราะห์ และทำการนำคุณลักษณะที่ได้ใหม่ไปใช้ในการทำนายผล โดย CNN นั้นมีจุดเด่นในด้านของการทำ Feature Extraction จากชุดข้อมูล โดยเน้นไปที่การหาคุณลักษณะจากชุดข้อมูลในรูปแบบของกลุ่มของข้อมูล (Kalchbrenner et al., 2014)

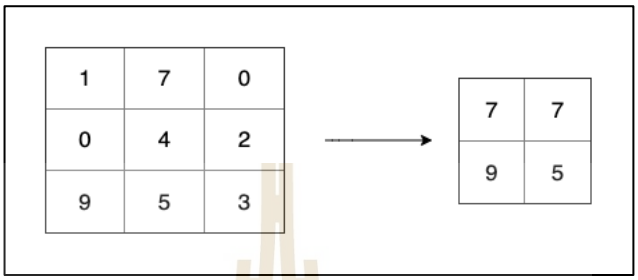
อัลกอริทึม CNN มีการแบ่งการทำงานออกเป็น 2 ส่วนได้แก่ Feature Extraction และ Classification โดย Feature Extraction เป็นการทำงานเพื่อคัดเลือกคุณลักษณะสำหรับการนำไปใช้ในการทำนายผลที่ขั้นตอน Classification ซึ่งเป็นขั้นตอนต่อไป สำหรับการทำ Feature Extraction ของ CNN เป็นการใช้ Filter ในการคัดเลือก Feature โดยทำการกำหนดขนาดของ Filter ที่ใช้สำหรับการคัดเลือกข้อมูล ซึ่ง Filter นี้อยู่ในรูปของ Matrix ทำงานโดยการวางลงไปบนชุดข้อมูลเพื่อกำหนดบริเวณที่ใช้ในการวิเคราะห์ และทำการประมวลผลออกมา ดังแสดงในรูปที่ 2.12



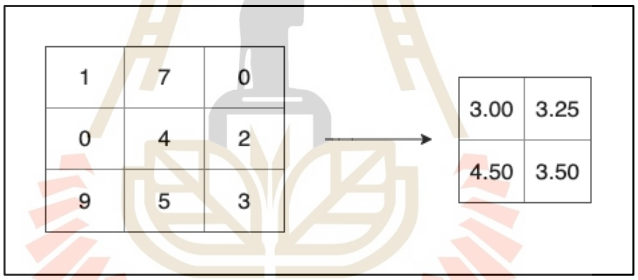
รูปที่ 2.12 ตัวอย่างการทำงานของอัลกอริทึม CNN

จากรูปที่ 2.12 อัลกอริทึม CNN ทำการใช้ Filter ในการสร้างชุดคุณลักษณะขึ้นมาใหม่ เมื่อได้คุณลักษณะขึ้นมาใหม่แล้ว เราสามารถทำการลดขนาดของคุณลักษณะที่ได้มาและยังคงเอกลักษณ์ของข้อมูลเดิมโดยไม่ทำให้ผิดเพี้ยนได้ โดยมีอัลกอริทึมให้เลือกใช้งาน 2 แบบ ได้แก่ Max Pooling และ Average Pooling โดย Max Pooling นั้นเป็นการสร้าง Filter อีกตัวขึ้นมาเพื่อ

นำไปใช้ในการวิเคราะห์ข้อมูล จากนั้นทำการดึงค่าที่มากที่สุดที่อยู่ใน Filter ออกมาใช้งาน ดังแสดงในรูปที่ 2.13 ส่วน Average Pooling เป็นการสร้าง Filter เช่นเดียวกันกับ Max Pooling แต่เป็นการดึงค่าเฉลี่ยของค่าต่าง ๆ ของ Filter ออกมา ดังแสดงในรูปที่ 2.14



รูปที่ 2.13 การนำข้อมูลที่ได้จาก Filter มาผ่าน Max Pooling ขนาด 2×2

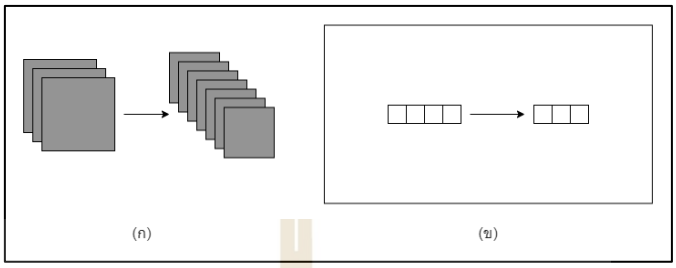


รูปที่ 2.14 การนำข้อมูลที่ได้จาก Filter มาผ่าน Average Pooling ขนาด 2×2

CNN ถูกนำไปประยุกต์ใช้งานในด้านต่าง ๆ โดยส่วนใหญ่ได้มีการนำไปประยุกต์ใช้กับงานทางด้านคอมพิวเตอร์วิทัศน์ ซึ่ง CNN สามารถวิเคราะห์รูปภาพได้ดีว่าอัลกอริทึมจำพวกการเรียนรู้ของเครื่องอื่น ๆ เนื่องจากการทำ Feature Extraction ของ CNN ที่ทำการดึงจุดเด่นของรูปภาพออกมาวิเคราะห์ และสามารถทำให้ชุดคุณลักษณะนั้นเล็กลงโดยไม่เสียรายละเอียดของชุดข้อมูลเดิม โดยมีตัวอย่างในการนำ CNN ไปใช้งานได้แก่การจำแนกประเภทของรูปภาพ การตรวจจับวัตถุในรูปภาพ หรือการสร้างรูปภาพเสมือน เป็นต้น

CNN ไม่เพียงแต่นำไปใช้ในการวิเคราะห์ข้อมูลที่อยู่ในรูปแบบของรูปภาพเท่านั้น แต่ยังสามารถนำไปใช้ในการวิเคราะห์ข้อมูลที่อยู่ในรูปของข้อความได้อีกด้วย โดยสามารถวิเคราะห์ได้ทั้งในรูปแบบของ 1 มิติ และ 2 มิติ ขึ้นอยู่กับวิธีในการแปลงข้อมูลของข้อความ ซึ่ง

โครงสร้างของ CNN 1 มิติและ 2 มิตินั้นมีโครงสร้างที่แตกต่างกันในส่วนของการทำ Filter และส่วนของการ Pooling ข้อมูล (Mao et al., 2017) ดังแสดงในรูปที่ 2.15

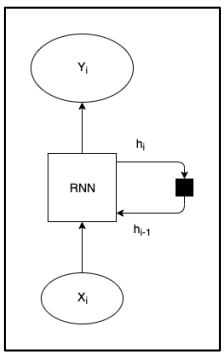


รูปที่ 2.15 CNN ในรูปแบบ (ก) 2 มิติ และ (ข) 1 มิติ

2.4.3 Recurrent Neural Network

Recurrent Neural Network (RNN) เป็นอัลกอริทึมของการเรียนรู้เชิงลึกซึ่งใช้หลักการการวิเคราะห์ข้อมูลในรูปแบบของอนุกรมลำดับเหตุการณ์ (Sequence) ซึ่งข้อมูลในรูปแบบนี้จะมีลำดับการเกิดของเหตุการณ์ที่ชัดเจน และสามารถเปลี่ยนบริบทของเหตุการณ์ตามลำดับได้ โดยตัวอย่างข้อมูลในลักษณะนี้ได้แก่ข้อมูลหุ้น ที่มีการเก็บบันทึกรายวัน ข้อมูลที่อยู่ในรูปแบบของอนุกรมเวลา (Time Series) และข้อมูลข้อความต่าง ๆ เป็นต้น (Mikolov et al., 2010)

RNN ได้รับการพัฒนาเพิ่มเติมขึ้นมาจากเดิมที่โครงข่ายประสาทเทียมนั้นจะมีการป้อนค่า Input เข้าไปแล้วได้ค่า Output ออกมา โดย RNN ได้ทำการออกแบบใหม่ให้สามารถนำค่า Output ไปคำนวณย้อนกลับเป็น Input ได้อีกครั้ง ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ RNN

จากรูป 2.16 โครงข่ายประสาทแบบ RNN นั้นจะมีโครงสร้างที่แตกต่างจากโครงข่ายประสาทเทียม โดยโครงสร้างของ RNN ที่เล็กที่สุดจะเรียกว่า Cell โดย Cell ของ RNN นั้นจะมีทั้งในส่วนของ Hidden State ที่ใช้ในการเก็บข้อมูลการวิเคราะห์จากข้อมูลในชุดที่ผ่านมา เพื่อใช้ในการวิเคราะห์ในตัวถัดไป

ในการปรับค่าพารามิเตอร์ต่าง ๆ นั้น RNN จะใช้วิธีการป้อนข้อมูลแบบย้อนกลับเพื่อทำการปรับค่าพารามิเตอร์ต่าง ๆ ในแบบจำลองที่ได้ทำการพัฒนาขึ้น โดยใช้หลักการของกฎลูกโซ่ (Chain Rule) ในการปรับค่าพารามิเตอร์ โดยทำการคำนวณเพื่อหาค่า Gradient ที่ได้จาก Error สำหรับการปรับพารามิเตอร์ของแบบจำลอง ดังสมการที่ 2.5 และ 2.6 (Premjith et al., 2018)

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.5}$$

$$\hat{y_t} = g(W_{yh}h_t + b_y) \tag{2.6}$$

เมื่อ h_t แทนค่า Hidden State ของหน่วยเวลา

f แทนฟังก์ชันสำหรับคำนวณค่า Hidden State ของหน่วยเวลา

W_{xh} แทนค่าน้ำหนักสำหรับคำนวณค่า Input ใน Hidden State

x_t แทนค่า Input ของข้อมูล

W_{hh} แทนค่าน้ำหนักสำหรับคำนวณค่า Hidden จากหน่วยเวลาก่อนหน้าใน Hidden State

h_{t-1} แทนค่า Hidden State จากหน่วยเวลาก่อนหน้า

b_h แทนค่า Bias สำหรับคำนวณค่า Hidden State

$\hat{y}_t$ แทนค่า Output ที่ได้จากแบบจำลอง RNN

g แทนฟังก์ชันสำหรับคำนวณค่า Output

W_{yh} แทนค่าน้ำหนักสำหรับคำนวณค่า Hidden State

b_y แทนค่า Bias สำหรับคำนวณค่า Output

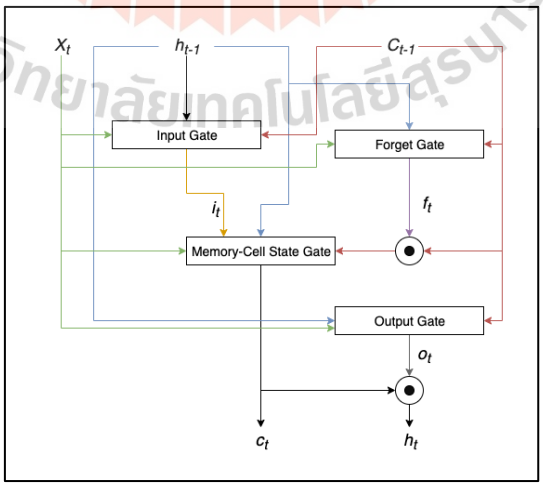
จากสมการที่ 2.5 ในการปรับค่าพารามิเตอร์ในส่วนของ Hidden State นั้น ได้เกิดปัญหาขึ้นจากการใช้งานโครงข่ายประสาทแบบ RNN ได้เนื่องจากการกำหนดลำดับของข้อมูลที่มีขนาดที่ยาวเกินไป ทำให้ค่า Gradient มีค่าลดลงจนไม่สามารถเห็นการเปลี่ยนแปลงได้ จึงทำให้โครงข่ายประสาทแบบ RNN นั้นมีข้อจำกัดในการใช้งานวิเคราะห์ข้อมูลที่มีความยาวที่มากเกินไป

โครงข่ายประสาทแบบ RNN ได้รับความนิยมนำไปใช้งานกับข้อมูลที่อยู่ในรูปของข้อความเป็นอย่างมาก โดยเฉพาะการประยุกต์ใช้ทางด้านของ Language Modeling และ Machine Translation ซึ่งเป็นเพราะการมองข้อความให้อยู่ในรูปแบบข้อมูลแบบลำดับ และทำการวิเคราะห์โดยคำนึงถึงตำแหน่งของ Input ทำให้ RNN สามารถวิเคราะห์ข้อมูลข้อความได้อย่างมีประสิทธิภาพ

2.4.4 Long Short-Term Memory

จากหัวข้อของ RNN ในการนำไปใช้งานจริงพบว่าเกิดปัญหาขึ้นจากการใช้ลำดับของข้อมูลในการวิเคราะห์ที่มีจำนวนมากเกินไป ทำให้ค่า Gradient ที่ได้นั้นมีการเปลี่ยนแปลงจนไม่สามารถสังเกตการเปลี่ยนแปลงได้ จนไม่สามารถนำมาใช้งานในการวิเคราะห์ได้ ซึ่งปัญหาข้างต้นคือปัญหา Gradient Vanishing จึงได้ทำการปรับปรุงโครงข่ายประสาทแบบ RNN ให้สามารถแก้ปัญหาของ Gradient Vanishing ออกไปให้ได้ โดยการเพิ่มฟังก์ชันการทำงานเพิ่มเติม คือ อัลกอริทึม Long Short-Term Memory

หน่วยความจำระยะยาว-ระยะสั้น (Long Short-Term Memory: LSTM) เป็นอัลกอริทึมที่ถูกพัฒนาต่อยอดมาจาก RNN โดยทำการแก้ปัญหาในส่วนของ Gradient Vanishing ด้วยการออกแบบการทำงานในส่วนของ Cell ใหม่ ให้สามารถเก็บสถานะของการคำนวณได้ (Hochreiter and Schmidhuber, 1997) โดยใน Cell ของ LSTM นั้นมีหน่วยคำนวณย่อยเรียกว่า Gate ซึ่งประกอบด้วย Input Gate, Forget Gate, Memory Cell State Gate และ Output Gate ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ LSTM

Input Gate เป็นหน่วยย่อยในการกำหนดข้อมูลที่จะนำเข้ามาวิเคราะห์ใน Cell โดยรับข้อมูลเข้ามาเพื่อทำการเขียนค่าลงไปในแต่ละ Cell ดังสมการที่ 2.7 (Premjith et al., 2018)

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2.7)$$

เมื่อ i_t แทนผลลัพธ์ที่ได้จาก Input Gate

σ แทนฟังก์ชัน Sigmoid

W_{xi} แทนค่าน้ำหนักสำหรับคำนวณ Input ใน Input Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{hi} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Input Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{ci} แทนค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Input Gate

c_{t-1} แทนค่า Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_i แทนค่า Bias ที่ใช้ในการคำนวณใน Input Gate

Forget Gate เป็นหน่วยย่อยที่ใช้ในการกำหนดข้อมูลที่จะนำเข้ามาวิเคราะห์ใน Cell โดยทำการกำหนดว่าข้อมูลนั้นควรที่จะถูกบันทึกหรือถูกลืม โดยสามารถกำหนดได้จากสมการที่ 2.8 (Premjith et al., 2018)

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2.8)$$

เมื่อ f_t แทนผลลัพธ์ที่ได้จาก Forget Gate

σ แทนฟังก์ชัน Sigmoid

W_{xf} แทนค่าน้ำหนักสำหรับคำนวณ Input ใน Forget Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{hf} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Forget Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{cf} แทนค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Forget Gate

c_{t-1} แทนค่า Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_f แทนค่า Bias ที่ใช้ในการคำนวณใน Forget Gate

Memory Cell State Gate เป็นหน่วยย่อยในการกำหนดข้อมูลที่นำเข้ามาวิเคราะห์ใน Cell และทำการคำนวณค่าสถานะ เพื่อใช้ในการคำนวณในครั้งถัดไป โดยมีสมการดังสมการที่ 2.9 (Premjith et al., 2018)

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (2.9)$$

เมื่อ c_t แทนค่า Memory Cell State ในช่วงหน่วยเวลา

f_t แทนผลลัพธ์ที่ได้จาก Forget Gate

c_{t-1} แทนค่า Memory Cell State จากหน่วยเวลาก่อนหน้า

i_t แทนผลลัพธ์ที่ได้จาก Input Gate

$tanh$ แทนฟังก์ชัน Hyperbolic tangent

W_{xc} แทนค่าน้ำหนักสำหรับคำนวณค่า Input จาก Memory Cell State Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{hc} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Memory Cell State Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_c แทนค่า Bias ที่ใช้ในการคำนวณใน Forget Gate

Output Gate เป็นหน่วยย่อยสำหรับการคำนวณ Output ของ Cell ซึ่งผลลัพธ์ที่ได้จาก Cell นี้จะมีอยู่ 2 อย่าง ได้แก่ Output และ Hidden State สำหรับใช้ในการคำนวณครั้งถัดไป โดยมีสมการดังสมการที่ 2.10 และ 2.11 ตามลำดับ (Premjith et al., 2018)

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (2.10)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (2.11)$$

เมื่อ o_t แทนผลลัพธ์ที่ได้จาก Output Gate

σ แทนฟังก์ชัน Sigmoid

W_{xo} แทนค่าน้ำหนักสำหรับคำนวณ Input ใน Output Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{ho} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Output Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{co} แทนค่าน้ำหนักสำหรับคำนวณ Memory Cell State ใน Output Gate

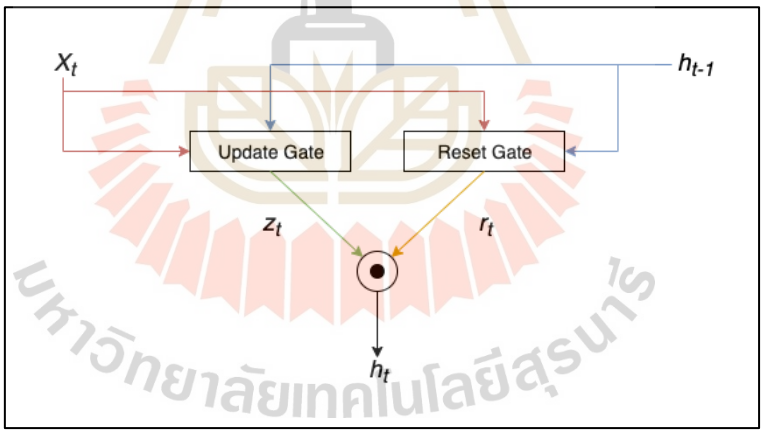
c_{t-1} แทนค่า Memory Cell State ที่ได้จากการคำนวณในหน่วยเวลาก่อนหน้า

b_o แทนค่า Bias ที่ใช้ในการคำนวณใน Output Gate

h_t แทนค่า Hidden State จากการคำนวณ

2.4.5 Gated Recurrent Unit

หน่วยเวียนกลับแบบมีประตู (Gated Recurrent Unit: GRU) ถูกพัฒนาต่อยอดมาจาก LSTM ซึ่งพัฒนาในส่วนของการลดความซับซ้อนในการทำงานของโครงข่ายประสาทแบบ LSTM เนื่องจากจำนวนหน่วยย่อยใน Cell จำนวนมาก ซึ่งมีผลต่อประสิทธิภาพในการวิเคราะห์และทำนายผล (Chung et al., 2014) โดย GRU ได้ทำการลดความซับซ้อนในการทำงานของโครงข่ายประสาทแบบ LSTM โดยการลดหน่วยย่อยใน Cell เหลือเพียง 2 ส่วน ได้แก่ Update Gate และ Reset Gate ดังแสดงในรูปที่ 2.18



รูปที่ 2.18 ตัวอย่างการทำงานในเซลล์ของอัลกอริทึมโครงข่ายประสาทแบบ GRU

Update Gate เป็นหน่วยย่อยที่ทำการนำข้อมูลไปคำนวณเพื่อกำหนดสถานะของ Cell สำหรับใช้ในการคำนวณในขั้นถัดไป โดยทำการคำนวณในทุก ๆ รอบที่มีข้อมูลเข้ามา ดังสมการที่ 2.12 (Premjith et al., 2018)

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (2.12)$$

เมื่อ z_t แทนค่าที่ได้จาก Update Gate

σ แทนฟังก์ชัน Sigmoid

W_{xz} แทนค่าน้ำหนักสำหรับคำนวณ Input ใน Update Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{hz} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Update Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_z แทนค่า Bias ที่ใช้ในการคำนวณใน Update Gate

Reset Gate เป็นหน่วยย่อยที่ใช้ในการกำหนดข้อมูลว่าควรที่จะเก็บค่าสถานะที่ได้จากการคำนวณในครั้งที่แล้วมากน้อยเพียงใด ดังสมการที่ 2.13 (Premjith et al., 2018)

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (2.13)$$

เมื่อ r_t แทนค่าที่ได้จาก Update Gate

σ แทนฟังก์ชัน Sigmoid

W_{xr} แทนค่าน้ำหนักสำหรับคำนวณ Input ใน Reset Gate

x_t แทนค่า Input ที่นำเข้ามาคำนวณ

W_{hr} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State ใน Reset Gate

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

b_r แทนค่า Bias ที่ใช้ในการคำนวณใน Reset Gate

สำหรับการคำนวณหาค่า Output และ Hidden State ของ GRU นั้น หน่วยย่อยสำหรับใช้คำนวณที่ตายตัว โดยทำการนำค่าผลลัพธ์ที่ได้จากทั้ง 2 Gate มาทำการคำนวณด้วยฟังก์ชัน Tanh จากนั้นค่าที่ได้จาก Reset Gate จะกำหนดว่าจะทำการจำค่าเดิม หรือทำการล้างค่าเดิมออกไป และทำการควบคุมปริมาณของข้อมูลด้วยค่าจาก Update Gate ดังสมการที่ (2.14) (Premjith et al., 2018)

$$h_t = (1 - z_t) \cdot tanh(r_t \cdot W_{hh} h_{t-1} + W_{xh} x_t) + z_t h_{t-1} \quad (2.14)$$

เมื่อ h_t แทนค่า Hidden State จากการคำนวณ

z_t แทนค่าที่ได้จาก Update Gate

$tanh$ แทนฟังก์ชัน Hyperbolic tangent

r_t แทนค่าที่ได้จาก Update Gate

W_{hh} แทนค่าน้ำหนักสำหรับคำนวณ Hidden State จากหน่วยเวลาก่อนหน้า

h_{t-1} แทนค่า Hidden State ที่ได้มาจากการคำนวณในหน่วยเวลาก่อนหน้า

W_{xh} แทนค่าน้ำหนักสำหรับคำนวณค่า Input

ในการทำงานของโครงข่ายประสาทแบบ GRU นั้นมีความแตกต่างจากโครงข่ายประสาทแบบ LSTM ในส่วนของการที่ Cell ของโครงข่ายประสาทแบบ GRU นั้นจะไม่ทำการเก็บค่าสถานะจากการคำนวณในครั้งก่อนหน้า มาใช้ในการวิเคราะห์

โครงข่ายประสาทแบบ GRU แสดงให้เห็นในส่วนของประสิทธิภาพในการวิเคราะห์ข้อมูลเมื่อทำการเปรียบเทียบกับโครงข่ายประสาทแบบ LSTM แต่โครงข่ายประสาทแบบ GRU นั้นสามารถแสดงจุดเด่นในด้านของการลดจำนวนพารามิเตอร์ในการฝึกสอนลง ทำให้แบบจำลองนั้นสามารถทำงานได้อย่างรวดเร็วยิ่งขึ้น

2.5 มาตรวัดประสิทธิภาพ

ในการทดลองทางด้านของการจำแนกประเภทของข้อมูล (Classification) จะต้องมีการวัดประสิทธิภาพการทำงานของแบบจำลอง เพื่อประเมินว่าแบบจำลองสามารถนำไปใช้งานจริงได้หรือไม่ โดยทางผู้วิจัยได้ทำการศึกษามาตรวัดต่าง ๆ ดังนี้

2.5.1 ค่าความถูกต้อง

ค่าความถูกต้อง (Accuracy) คือค่าทางสถิติใช้สำหรับเปรียบเทียบข้อมูลการทำนายผลระหว่างผลลัพธ์เป้าหมาย กับผลลัพธ์ที่ทำนายได้ ว่ามีความสัมพันธ์กันอย่างไร โดยกรคำนวณค่าความถูกต้องนั้น จะต้องทำการสร้าง เมทริกซ์วัดประสิทธิภาพ (Confusion Matrix) ขึ้นมา โดยมีรายละเอียด ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ตารางค่าต่าง ๆ ใน Confusion Matrix

		Actual	
		Positive	**Negative**
Prediction	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

จากตารางที่ 2.1 เมื่อแบ่งให้ข้อมูลมี 2 ประเภท ได้แก่ข้อมูลด้านบวก (Positive) และข้อมูลด้านลบ (Negative) สามารถอธิบายค่าในตัวแปรต่าง ๆ ของ Confusion Matrix ออกได้เป็น 4 กรณี ได้แก่

1. True Positive (TP) หมายถึงแบบจำลองสามารถทำนายข้อมูลเป็นด้านบวก และตรงกับผลลัพธ์เป้าหมายเป็นด้านบวก

2. False Positive (FP) หมายถึงแบบจำลองสามารถทำนายข้อมูลเป็นด้านบวก แต่ผลลัพธ์เป้าหมายเป็นด้านลบ

3. False Negative (FN) หมายถึงแบบจำลองสามารถทำนายข้อมูลเป็นด้านลบ แต่ผลลัพธ์เป้าหมายเป็นด้านบวก

4. True Negative (TN) หมายถึงแบบจำลองสามารถทำนายข้อมูลเป็นด้านลบ และตรงกับผลลัพธ์เป้าหมายเป็นด้านลบ

จากข้อมูลข้างต้น สามารถนำมาคำนวณค่าความแม่นยำ เพื่อหาค่าความถูกต้องของแบบจำลอง ได้ดังสมการที่ 2.15

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.15}$$

2.5.2 ค่าความเที่ยง

ค่าความเที่ยง (Precision) เป็นค่าที่ใช้สำหรับการวัดประสิทธิภาพในการทำนายของแบบจำลอง โดยคำนวณจาก TP เทียบกับกลุ่มของข้อมูลที่ทำนายเป็นด้านบวกทั้งหมด (TP + FP) ดังสมการที่ 2.16

$$Precision = \frac{TP}{TP + FP} \tag{2.16}$$

2.5.3 ค่าความไว หรือค่าระลึก

ค่าความไว หรือค่าระลึก (Recall) เป็นค่าที่ใช้สำหรับการวัดประสิทธิภาพในการทำนายของแบบจำลอง โดยคำนวณจาก โดยคำนวณจากค่า TP เทียบกับกลุ่มของข้อมูลที่มีผลลัพธ์เป้าหมายเป็นจริงทั้งหมด ดังสมการที่ 2.17

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

2.6 งานวิจัยที่เกี่ยวข้อง

จากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่ามีงานวิจัยที่ศึกษาเกี่ยวกับการทำแบบจำลองสำหรับการวิเคราะห์ความรู้สึก โดยการสร้างแบบจำลองนั้นได้มีการศึกษาทั้งในส่วนของขั้นตอนในการออกแบบและพัฒนาอัลกอริทึมสำหรับการแปลงข้อมูลให้อยู่ในรูปที่คอมพิวเตอร์ สามารถทำความเข้าใจได้ และการออกแบบและพัฒนาอัลกอริทึมที่ใช้สำหรับสร้างแบบจำลอง โดยงานวิจัยที่เกี่ยวข้องที่ได้ทำการศึกษา มีรายละเอียดดังนี้

งานวิจัยของ Cai et al. (2019) ได้ทำการศึกษาและพัฒนาแบบจำลองสำหรับระบบการแจ้งเตือนภัยสำหรับโรงงานอุตสาหกรรม โดยในงานวิจัยนี้ได้เน้นการพัฒนาระบบสำหรับใช้ในโรงงานรูปแบบสมัยใหม่ ที่เน้นการทำงานในรูปแบบอัตโนมัติ และสามารถเพิ่มประสิทธิภาพในการทำงานในด้านต่าง ๆ โดยทางผู้วิจัยได้ทำการพัฒนาแบบจำลองการแจ้งเตือนอุทกภัยโดยรอบ ซึ่งการแจ้งเตือนที่รวดเร็วจะทำให้ลดความสูญเสียในด้านต่าง ๆ ได้เป็นอย่างมาก การพัฒนาแบบจำลองนั้นจะใช้หลักการของ Deep Learning และ Natural Language Processing ในการพัฒนาแบบจำลอง โดยใช้ Word2Vec ในการสร้าง Feature สำหรับนำไปใช้ในการฝึกสอน และใช้งาน LSTM ในการสร้างแบบจำลองการทำนายผล โดยสามารถทำนายผลได้อย่างแม่นยำสูงถึง 80%

งานวิจัยของ Shabani และ Sokhn (2018) ได้ทำการสร้างแบบจำลองสำหรับตรวจจับข่าวปลอม เนื่องจากข่าวสารต่าง ๆ ในปัจจุบันนั้นเผยแพร่ไปได้รวดเร็วผ่านเครือข่ายอินเทอร์เน็ต โดยข่าวปลอมสามารถส่งผลกระทบในระดับนานาชาติได้ ซึ่งทางผู้วิจัยได้ทำการพัฒนาแบบจำลองเพื่อทดแทนการทำงานของมนุษย์ในการจำแนกข่าว ซึ่งใช้เวลาในการจำแนกที่นาน โดยทางผู้วิจัยได้ทำการพัฒนาแบบจำลองจากข้อมูลข่าวปลอมเกี่ยวกับการเมือง โดยมีข่าวปลอมจำนวน 283 ข่าว และข่าวจริงจำนวน 203 ข่าว จากนั้นทำการคัดเลือก Feature โดยใช้ TF-IDF ในการสร้าง Feature จากนั้นทำการสร้างแบบจำลองโดยใช้ Machine Learning ด้วยอัลกอริทึม Logistic Regression,

Support Vector Machine และ Neural Networks แบบ Hybrid ซึ่งให้ประสิทธิภาพในการทำนายได้แม่นยำถึง 84% จากนั้นทางผู้วิจัยได้พัฒนาเพิ่มเติมด้วยการนำมนุษย์มาช่วยในการคัดแยกร่วมกับแบบจำลองที่ทำงานอย่างอัตโนมัติ ซึ่งให้ประสิทธิภาพในการทำนายเพิ่มขึ้นจากเดิม โดยให้ค่าความถูกต้องอยู่ที่ 87%

งานวิจัยของ Zhou et al. (2019) ได้ทำการศึกษาพฤติกรรมการใช้งานของผู้ใช้งานต่อเว็บไซต์ StackOverflow ซึ่งเป็นเว็บไซต์สังคมของนักพัฒนาที่ใช้สำหรับแลกเปลี่ยนประสบการณ์เกี่ยวกับการพัฒนาซอฟต์แวร์ มีทั้งการตั้งคำถามและการค้นหาความรู้ต่าง ๆ ทางด้านการพัฒนาซอฟต์แวร์ โดยจะทำการแบ่งประเภทของคำถามตามหัวข้อที่เกี่ยวข้อง โดยการคัดแยกหัวข้อที่ถูกต้องนั้น จะช่วยเพิ่มประสิทธิภาพ ทั้งในด้านของการส่งคำถามเพื่อถามความรู้จากคนที่เชี่ยวชาญทางด้านนี้ได้ และด้านของการคัดแยกคำถามเฉพาะหมวดหมู่ที่ผู้ใช้สนใจ โดยทางผู้วิจัยได้ทำการคิดค้นและพัฒนาแบบจำลองสำหรับการคัดแยกหมวดหมู่ของคำถาม โดยทำการฝึกสอนด้วยแบบจำลอง TagCNN และ TagRCNN ซึ่งทางผู้วิจัยได้ทำการคิดค้นหลักการ โดยเป็นการนำ CNN มาประยุกต์ใช้ในการวิเคราะห์ข้อมูลข้อความ โดยวัดประสิทธิภาพในการทำนายจากค่า F-Measure ซึ่งเป็นค่าที่คำนวณจาก Precision และ Recall อยู่ที่ประมาณ 0.7 หรือคิดเป็น 70%

งานวิจัยของ Rashkin et al. (2017) ได้ทำการศึกษาวิจัยเกี่ยวกับการวิเคราะห์บทความข่าวปลอมทางด้านการเมือง โดยกำหนดเกณฑ์การวิเคราะห์ให้คะแนนอยู่ที่ 6 ระดับ โดยนำชุดข้อมูลจาก PolitiFact.com ในการวิเคราะห์ โดยชุดข้อมูลดังกล่าวได้นำมาใช้ในการสร้างแบบจำลองสำหรับการให้คะแนนบทความข่าวว่าอยู่ในระดับใด โดยได้ทำการสร้าง Feature จาก GloVe ซึ่งเป็นแบบจำลอง Word Embedding ที่ได้รับความนิยม และทำการฝึกสอนโดยใช้อัลกอริทึม LSTM โดยให้ค่าความถูกต้องอยู่ที่ 22 %

งานวิจัยของ Haddi et al. (2013) ได้ทำการศึกษาวิธีการสกัด Feature จากชุดข้อมูลข้อความสำหรับการสร้างแบบจำลองจากชุดข้อมูลที่มาจาก Social Network ซึ่งเป็นชุดข้อมูลขนาดใหญ่ที่ต้องการสร้างระบบการทำ Opinion Mining หรือ การวิเคราะห์ความรู้สึก แบบอัตโนมัติ ผู้วิจัยได้ทำการศึกษาการสร้าง Feature ด้วยหลักการของ TF-IDF โดยเป็นการวิเคราะห์ในรูปแบบของการตรวจสอบทั้งจำนวนความถี่ของคำที่พบ และความถี่ของคำที่พบในแต่ละเอกสาร และทำการฝึกสอนแบบจำลองโดยใช้อัลกอริทึม Support Vector Machine โดยใช้ชุดข้อมูลการแสดงความคิดเห็นต่อภาพยนตร์เรื่องต่าง ๆ บนระบบอินเทอร์เน็ต จากนั้นทำการวัดประสิทธิภาพในการสร้าง

พบว่า การสร้าง Feature ด้วย TF-IDF และสร้างแบบจำลองโดยใช้ Support Vector Machine นั้นให้ประสิทธิภาพในการทำนายด้วยค่าความถูกต้องที่ 93.5%, Precision 94%, Recall 93.06% และ F-Measure 93.53%

งานวิจัยของ Liu et al. (2017) ได้ทำการทดลองและสร้างแบบจำลองสำหรับการวิเคราะห์ความรู้สึก จากชุดข้อมูลต่าง ๆ โดยเน้นไปที่ข้อมูลการแสดงความคิดเห็นบนเว็บไซต์ต่าง ๆ ได้แก่ การแสดงความคิดเห็นต่อสินค้าบน Amazon.com และการรีวิวภาพยนตร์บน IMDB และ Rotten Tomato มาใช้ในการวิเคราะห์ โดยสร้างแบบจำลองใช้อัลกอริทึมในกลุ่ม Deep Learning ได้แก่ LSTM, Bidirectional LSTM (BiLSTM), Stacked LSTM (sLSTM) และเลือกใช้อัลกอริทึม GloVe ซึ่งเป็น Word Embedding ในการแปลงข้อมูล ซึ่งในการทดลองใช้งาน Deep Learning นั้นเป็นการทดลองในส่วนการทำงาน Single Task โดยให้ค่าความผิดพลาดในการทำนายอยู่ที่ 18 % หรือคิดเป็นค่าความถูกต้องที่ 82%

งานวิจัยของ Le และ Mikolov (2014) ได้พัฒนาอัลกอริทึม Doc2Vec โดยทำการปรับปรุงการทำงานจากอัลกอริทึม Word2Vec ด้วยการปรับเปลี่ยนการทำงานจากการเข้ารหัสคำแบบจำกัดขนาดของคำศัพท์ เป็นการเข้ารหัสเอกสาร โดยไม่จำกัดจำนวนคำศัพท์ จากนั้นทำการทดลองการจำแนกประเภทของการแสดงความคิดเห็นต่อภาพยนตร์บนเว็บไซต์ IMDB พบว่าคุณลักษณะที่ได้จาก Doc2Vec นั้นให้ค่าความผิดพลาดที่ต่ำ อยู่ที่ 7.42% ส่งผลให้การทำนายมีความแม่นยำคิดเป็น 92.58%

จากงานวิจัยต่าง ๆ ที่ผู้วิจัยได้ทำการศึกษาพบว่าการสร้างแบบจำลองทางด้านการวิเคราะห์ความรู้สึกด้วยข้อมูลข้อความนั้น ต้องใช้ความรู้ด้าน Natural Language Processing มาประยุกต์ใช้งาน โดยจะต้องทำการสร้าง Feature สำหรับนำไปใช้ในการเรียนรู้ของอัลกอริทึมทางด้านการเรียนรู้ของเครื่องซึ่งเป็นส่วนสำคัญของงานวิจัยต่าง ๆ ที่ได้ศึกษา โดยในส่วนของการคัดแยกคุณลักษณะนั้น จะสามารถแบ่งออกได้เป็น 2 กลุ่ม โดยเป็นกลุ่มของการสร้างคุณลักษณะสำหรับใช้งานกับอัลกอริทึมทางด้านของการเรียนรู้ของเครื่องเช่น TF-IDF และ การสร้างคุณลักษณะสำหรับใช้งานกับอัลกอริทึมทางด้านของการเรียนรู้เชิงลึกเช่น Word Embedding เป็นต้น นอกจากนี้ในส่วนของอัลกอริทึมที่ใช้ในการสร้างแบบจำลองสำหรับการทำนายผลของการวิเคราะห์ความรู้สึกนั้น อัลกอริทึม Support Vector Machine สามารถให้ประสิทธิภาพในการทำนายที่สูงกับงานในอดีต แต่ปัจจุบันนั้นอัลกอริทึมจำพวก Deep Learning เช่น Neural Networks, Convolutional Neural

Network และ Long Short-Term Memory เป็นต้น ได้ให้ประสิทธิภาพในการทำนายที่สูงกว่าเป็นอย่างมากและยังสามารถนำไปใช้วิเคราะห์ข้อมูลได้หลากหลายยิ่งกว่าการใช้งาน Machine Learning แบบดั้งเดิม

บทความวิจัยที่เกี่ยวข้องประกอบไปด้วย

“ก” แทนงานวิจัยของ Cai, et al. (2018)

“ข” แทนงานวิจัยของ Shabani และ Sokhn (2018)

“ค” แทนงานวิจัยของ Zhou et al. (2019)

“ง” แทนงานวิจัยของ Rashkin et al. (2017)

“จ” แทนงานวิจัยของ Haddi et al. (2013)

“ฉ” แทนงานวิจัยของ Liu et al. (2017)

“ช” แทนงานวิจัยของ Le และ Mikolov (2014)

“ซ” แทนงานวิจัยของวิทยานิพนธ์ฉบับนี้

ตารางที่ 2.2 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการพัฒนาแบบจำลองการวิเคราะห์ความรู้สึก

กระบวนการทำงาน	**งานวิจัยที่เกี่ยวข้อง**							
	ก	**ข**	**ค**	**ง**	**จ**	**ฉ**	**ช**	**ซ**
ฐานความรู้ของระบบ								
ฐานความรู้เกี่ยวข้องกับกระบวนการทางคณิตศาสตร์			✓			✓	✓	
ฐานความรู้ทางด้านการวิเคราะห์ข้อมูลข้อความ	✓	✓	✓	✓	✓	✓	✓	✓
ฐานความรู้การทำเหมืองข้อมูล	✓	✓	✓	✓	✓	✓	✓	✓
ระบบการคัดแยกคุณลักษณะจากข้อความ								
ความสามารถของระบบ								
ระบบสามารถใช้งานได้กับข้อมูลขอบเขตใดขอบเขตหนึ่ง	✓	✓	✓	✓	✓	✓	✓	✓

ตารางที่ 2.2 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการพัฒนาแบบจำลองการวิเคราะห์ความรู้สึก (ต่อ)

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง							
	ก	ข	ค	ง	จ	ฉ	ช	ซ
ปรับรูปแบบให้เหมาะสมด้วยหลักการทางคณิตศาสตร์			✓			✓	✓	
อัลกอริทึมที่ใช้ในการพัฒนา								
TF-IDF		✓			✓			✓
Word2Vec	✓							✓
GloVe				✓		✓		✓
Doc2Vec							✓	✓
ระบบ การวิเคราะห์ความรู้สึก								
อัลกอริทึมที่ใช้ในการพัฒนาแบบจำลอง								
Statisticak-based Machine Learning		✓			✓		✓	
Neural Network		✓						✓
CNN			✓					✓
RNN (LSTM and GRU)	✓			✓		✓		✓

บทที่ 3
วิธีดำเนินงานวิจัย

3.1 กรอบแนวคิดการวิจัย

แนวคิดหลักของงานวิจัยนี้คือการนำเทคนิคการเรียนรู้เชิงลึกมาประยุกต์ใช้กับเทคนิคการวิเคราะห์ความรู้สึกในการจำแนกข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์และบริการต่าง ๆ สามารถแบ่งกรอบแนวคิดการวิจัยออกเป็น 2 ขั้นตอนหลัก ได้แก่ ขั้นตอนที่ 1 การแปลงข้อมูลให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้ และขั้นตอนที่ 2 การออกแบบและพัฒนาอัลกอริทึมสำหรับสร้างแบบจำลองเพื่อใช้ในการทำนายผล

3.1.1 ขั้นตอนที่ 1: การแปลงข้อมูลให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้

ขั้นตอนแรกของงานวิจัยคือการแปลงชุดข้อมูลข้อความที่ได้ ให้อยู่ในรูปของตัวเลขเข้ารหัส ที่คอมพิวเตอร์สามารถนำไปใช้ประมวลผลต่อได้ และทำการนำข้อมูลที่ได้ไปใช้ในการวิเคราะห์ต่อไป โดยจะต้องแปลงสภาพของข้อมูลให้ได้คุณลักษณะที่เหมาะสมกับอัลกอริทึมที่เลือกใช้ด้วย ซึ่งมีขั้นตอนแสดงดังรูปที่ 3.1

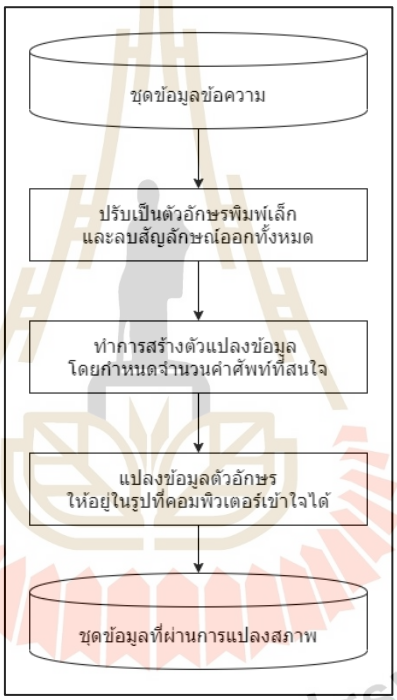
จากรูปที่ 3.1 กรอบแนวคิดวิธีการแปลงข้อมูลประกอบไปด้วยขั้นตอนย่อย 3 ขั้นตอนดังนี้

1) การทำความสะอาดข้อมูล (Text Cleaning) เป็นขั้นตอนในการทำความสะอาดข้อความต่าง ๆ สำหรับข้อมูลที่อยู่ในรูปแบบข้อความนั้น จะเป็นการแก้ไขรายละเอียดคำในส่วนที่สะกดผิดเป็นส่วนใหญ่ หรือการปรับตัวอักษรของคำให้อยู่ในรูปแบบเดียวกัน เช่นตัวพิมพ์เล็กในภาษาอังกฤษเนื่องจากคอมพิวเตอร์จะเข้าใจตัวอักษรพิมพ์ใหญ่ และพิมพ์เล็กว่าแตกต่างกัน ไม่ใช่ตัวเดียวกัน เช่น ‘A’ กับ ‘a’ เป็นต้น และการลบสัญลักษณ์ต่าง ๆ ที่อยู่ในชุดข้อมูลออกไป

2) การสร้างคลังคำศัพท์ (Vocabulary) เป็นขั้นตอนในการสร้างคลังคำศัพท์สำหรับใช้ในการวิเคราะห์ โดยทำการรวบรวมคำที่มีอยู่ในประโยคและทำการสร้างคลังคำศัพท์โดยทำการนำคำที่แตกต่างกันมาใช้ในการสร้าง ซึ่งสามารถ

กำหนดขอบเขตของคำที่ใช้ในการศึกษาได้โดยการกำหนดจำนวนคำที่แสดงผลมากที่สุดในชุดข้อมูล เพื่อลดข้อมูลในส่วนที่ไม่จำเป็นในการวิเคราะห์ออกไป

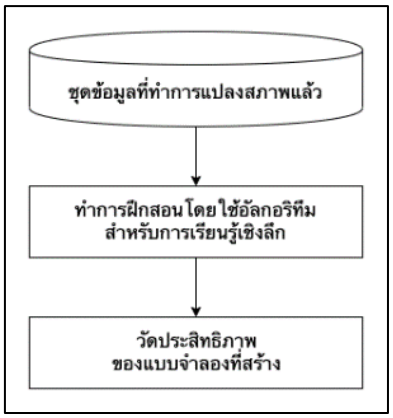
3) การแปลงข้อความ (Text Transformation) เป็นขั้นตอนที่ใช้ในการแปลงคำต่าง ๆ ให้อยู่ในรูปที่คอมพิวเตอร์สามารถนำไปใช้ในการประมวลผลต่อได้ ซึ่งจะต้องทำการแปลงข้อมูลก่อนที่จะนำมาใช้วิเคราะห์ด้วยอัลกอริทึมของการเรียนรู้เชิงลึก



รูปที่ 3.1 กรอบแนวคิดวิธีการแปลงข้อมูล

3.1.2 ขั้นตอนที่ 2: การสร้างแบบจำลองการวิเคราะห์ความรู้สึกจากข้อความ

การดำเนินการในขั้นตอนที่ 2 จะเป็นการดำเนินงานในส่วนของการนำข้อความที่ได้ทำการแปลงสภาพให้อยู่ในรูปที่คอมพิวเตอร์สามารถทำความเข้าใจได้แล้ว นำมาใช้ในการวิเคราะห์และทำนายผล โดยมีกรอบแนวคิดดังรูปที่ 3.2



รูปที่ 3.2 กรอบแนวคิดการสร้างแบบจำลองสำหรับทำนายผล

จากกรอบแนวคิดการใช้ การวิเคราะห์ความรู้สึก ในการจำแนกประเภทของข้อความ จะประกอบไปด้วย 3 ขั้นตอนย่อย ดังนี้

1) นำข้อมูลที่ผ่านการแปลงสภาพแล้ว มาทำการเตรียมข้อมูลเพื่อที่จะนำไปใช้ในการวิเคราะห์และสร้างแบบจำลอง

2) ทำการสร้างแบบจำลอง การวิเคราะห์ความรู้สึก สำหรับจำแนกข้อความ และทำการทดสอบ

3) นำผลลัพธ์ที่ได้จากแบบจำลอง มาทำการวัดประสิทธิภาพการทำนาย ด้วยค่าความถูกต้อง ค่าความเที่ยง ค่าความไว และเวลาที่ใช้การฝึกสอน

3.2 ชุดข้อมูล

งานวิจัยนี้ได้ทำการเลือกใช้ชุดข้อมูลกรณีศึกษาของการแสดงความคิดเห็นต่อสินค้าบนเว็บไซต์ amazon.com ซึ่งได้ทำการเก็บรวบรวมโดย Biltzer et al. (2007) ซึ่งชุดข้อมูลดังกล่าวนั้นเปิดให้เข้าใช้งานในรูปแบบของสาธารณะ ภายในชุดข้อมูลประกอบด้วยชุดข้อมูลการแสดงความคิดเห็นในรูปแบบทั้งด้านดี จำนวน 1,000 ระเบียน และด้านลบจำนวน 1,000 ระเบียน โดยแสดงออกมาเป็นตารางที่มีจำนวน 2 คอลัมน์ โดยคอลัมน์ text_review แทนข้อความแสดงความคิดเห็น และคอลัมน์ rating แสดงการจำแนกประเภทของข้อความ โดยเลข 0 แทนข้อความที่ไปในทางด้านลบ และเลข 1 แทนข้อความที่ไปในทิศทางบวก ตัวอย่างข้อมูลแสดงดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างชุดข้อมูลที่ใช้ในงานวิจัย

rating	review_text
0	The content is excellent. The digital format is largely useless.I bought the digital 5th edition to supplement my paper 4th edition. But, the digital rights management restrictions mean that I can't print selected pages to stick in a folder, read on the train, or scribble notes on. I can't even cut and paste particularly relevant bits into an electronic 'notes' file. The onerous digital restrictions are not suitable for an academic book like this one
0	The stationery is cute and colorful, but the pages are very cluttered. While the envelopes have spaces to write addresses and put return labels, the pages are too colorful and bold. A ballpoint or ink pen doesn't look good on the paper.
1	The purpose of Nessus is to provide an Open Source Solution for network auditing on all Unix like systems. This book not only details using Nessus but also comes with a CD containing the program, as well as Ethereal, Snort, and Newt (a port of the program to the Windows environment).What is a network assessment? At its basic level it is an attempt to detect a live system and then identify the computing environment, services, applications, and vulnerabilities on that system. Basically there are two types of assessment - internal and external. An internal assessment is done over the local network and external is done from outside the LAN. Nessus will do both types and the book details how to do either, or both of them.The authors do an excellent job of detailing installation, setup, and how to interpret the results of a scan as well as various factors that can affect the report. One of the parts not to be missed is the discussion of not only the benefits but also the potential problems of scanning your system. Some of the vulnerability types scanned for include buffer overflows, default passwords, backdoors, information leaks, and denial of service.The Nessus scripting language is covered in detail in Appendix A instead of the main portion of the book; a choice I appreciated very much as it allowed the flow of the book to not be interrupted by such a highly technical section. With Open Source products there generally is no organized technical support phone number you can call of help. So, the authors include information on how to get help via the Nessus User Community, mailing lists, and archives.Nessus Network Auditing is a highly recommended book for anyone interested in auditing their network to find potential problems before they become reality
1	This is our favorite baby book for reading with our baby/young child. Each of our children, from ages 9 months to three years, has loved this book. We buy this as a gift for all new babies

3.3 การออกแบบคุณลักษณะและแบบจำลอง

ในการวิจัยนี้ ผู้วิจัยได้ทำการแบ่งการทดลองออกเป็น 7 รูปแบบ โดยแบ่งออกเป็นการสร้างตัวแปลงข้อความให้อยู่ในรูปของคุณลักษณะที่ใช้ประมวลผลได้ ได้แก่ TF-IDF, Word Embedding และ Doc2Vec จากนั้นทำการเลือกใช้อัลกอริทึมในการพัฒนาแบบจำลองได้แก่ Deep Neural Network, Convolutional Neural Network แบบ 1 มิติ, Convolutional Neural Network แบบ 2 มิติ, Long Short-Term Memory และ Gated Recurrent Unit โดยมีขั้นตอนในการทดลอง ดังนี้

3.3.1 คุณลักษณะ TF-IDF สำหรับแปลงข้อมูลข้อความ

ในงานวิจัยนี้ทางผู้วิจัยได้ประยุกต์ใช้อัลกอริทึม TF-IDF สำหรับการแปลงข้อความเพื่อใช้ในการฝึกสอนอัลกอริทึม DNN และ CNN แบบ 1 มิติ ซึ่งทางผู้วิจัยได้ออกแบบกรอบการดำเนินงานโดยใช้คลังคำศัพท์สำหรับเก็บ Token จำนวน 5,000 คำ ทำการเลือกโดยใช้คะแนนที่ได้จากการคำนวณด้วย TF-IDF ในการคัดเลือก โดยทำการกำหนด Token ให้อยู่ในรูปของ N-gram โดยมีความกว้างอยู่ที่ 1-3 gram ในการสร้าง TF-IDF ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 แสดงการปรับค่าของคุณลักษณะ TF-IDF

Max Vocabulary	N-Gram range	Output Vector
5000	1-3	5000

3.3.2 คุณลักษณะ Word Embedding สำหรับแปลงข้อมูลข้อความ

ในการดำเนินงานวิจัยนี้ทางผู้วิจัยได้ทำประยุกต์ใช้อัลกอริทึม Word Embedding สำหรับการแปลงข้อความเพื่อใช้ในการฝึกสอนอัลกอริทึม CNN แบบ 2 มิติ LSTM และ GRU ซึ่งทางผู้วิจัยได้ทำการออกแบบการดำเนินงานโดยใช้คลังคำศัพท์สำหรับเก็บ Token จำนวน 5,000 คำ เพื่อทำการเลือกโดยใช้ความถี่ของ Token ที่พบจากมากไปน้อย ตามลำดับ ในขั้นตอนนี้เลือกแบบจำลอง Pre-Trained จาก GloVe มาใช้ในการสร้าง Word Embedding โดยทำการกำหนดคุณลักษณะเพิ่มเติมเกี่ยวกับจำนวนมิติของ Embedding และ ขนาด Output จาก Word Embedding ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 แสดงการปรับค่าของคุณลักษณะ Glove (Word Embedding)

Max Vocabulary	Max Length	Embedding Size	Output Vector
5000	500	200	500 x 200

3.3.3 คุณลักษณะ Doc2Vec สำหรับแปลงข้อมูลข้อความ

ในการวิจัยนี้ทางผู้วิจัยได้ประยุกต์ใช้อัลกอริทึม Doc2Vec สำหรับการแปลงข้อความเพื่อใช้ในการฝึกสอนอัลกอริทึม DNN และ CNN แบบ 1 มิติ เนื่องจาก Doc2Vec นั้นเป็นการเข้ารหัสทั้งชุดข้อมูลแต่ละระเบียน จึงไม่จำเป็นที่จะต้องกำหนดจำนวนคำศัพท์ที่สนใจ แต่จะต้องกำหนดจำนวนคุณลักษณะที่ต้องการนำไปใช้ในการวิเคราะห์ต่อ โดยทางผู้วิจัยได้กำหนดพารามิเตอร์ของอัลกอริทึมโดยกำหนดจำนวนเวกเตอร์คุณลักษณะที่ได้จำนวน 5,000 ตัว สำหรับการทดลองนี้

3.3.4 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Deep Neural Network

อัลกอริทึม TF-IDF จะทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้น และจำนวนของ N-gram ของ Token ที่เราได้ทำการกำหนดไว้ เมื่อได้ตัวคุณลักษณะของข้อมูลแล้ว จากนั้นทำการแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และสร้างแบบจำลองการทำนายผลการวิเคราะห์ความรู้สึกด้วยอัลกอริทึม Deep Neural Network จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดงดังรูปที่ 3.3

รูปที่ 3.3 แสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยอัลกอริทึม TF-IDF และ DNN โดยมีขั้นตอนการทำงาน ดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และสร้างแบบจำลอง

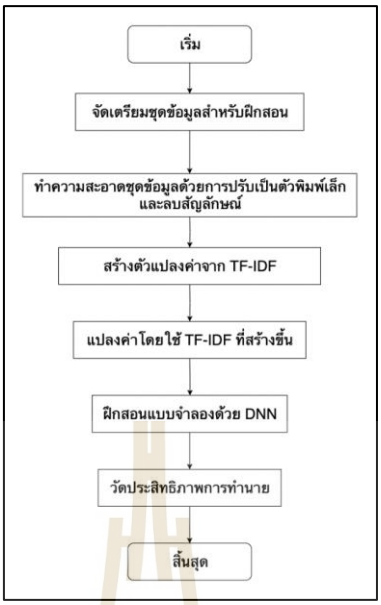
2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็ก และลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม TF-IDF

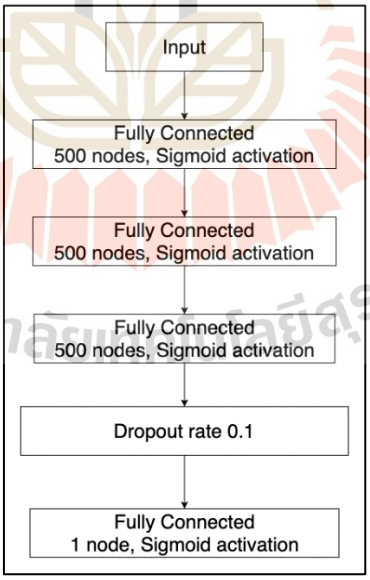
4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง TF-IDF ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม DNN ที่ได้ออกแบบไว้ดังรูปที่ 3.4

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย



รูปที่ 3.3 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ TF-IDF และอัลกอริทึม DNN

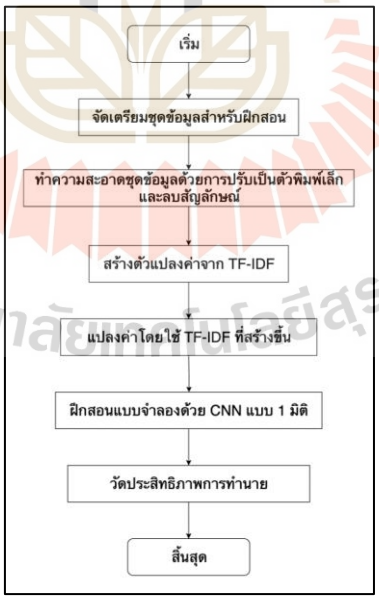


รูปที่ 3.4 โครงสร้างของอัลกอริทึม DNN ที่ใช้ในการวิจัย

รูปที่ 3.4 แสดงโครงสร้างของอัลกอริทึม DNN ที่ใช้ในการวิจัย โดยทำการรับ Input ข้อความ และทำการแปลงโดยใช้ TF-IDF จากนั้นนำคุณลักษณะที่ได้มาใช้ในการวิเคราะห์ด้วย DNN โดยใช้ชั้น Fully Connected ซึ่งเป็นโครงสร้างแบบเรียงตัวกันเป็นจำนวน 3 ชั้น ชั้นละ 500 Node และใช้ Sigmoid Function เป็น Activation Function จากนั้นทำการผ่าน Dropout เพื่อทำการป้องกัน Overfitting และสุดท้ายทำการผ่านชั้น Fully Connected ที่ได้ออกแบบไว้สำหรับการจำแนกประเภท โดยในชั้นนี้จะมี 1 Node และใช้ Sigmoid Function เป็น Activation Function

3.3.5 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ

อัลกอริทึม TF-IDF ทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้นและจำนวนของ N-gram ของ Token ที่เราได้ทำการกำหนดไว้ เมื่อได้ตัวคุณลักษณะของข้อมูลแล้ว จากนั้นทำการแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่นำไปใช้ในการวิเคราะห์ได้และทำการสร้างแบบจำลองการวิเคราะห์ความรู้สึกด้วยอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดงในรูปที่ 3.5



รูปที่ 3.5 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ TF-IDF และอัลกอริทึม CNN แบบ 1 มิติ

รูปที่ 3.5 แสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ TF-IDF และอัลกอริทึม CNN แบบ 1 มิติ โดยมีขั้นตอนการทำงานอธิบายได้ ดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

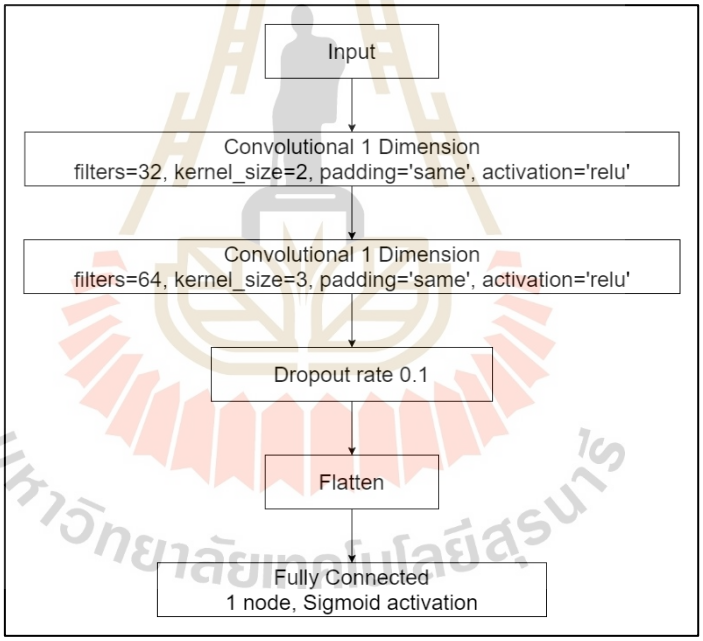
2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็ก และลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม TF-IDF

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง TF-IDF ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม CNN แบบ 1 มิติ ที่ได้ออกแบบไว้ดังรูปที่ 3.6

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย



รูปที่ 3.6 โครงสร้างของอัลกอริทึม CNN แบบ 1 มิติที่ใช้ในการวิจัย

รูปที่ 3.6 แสดงโครงสร้างของอัลกอริทึม CNN แบบ 1 มิติที่ใช้ในการวิจัย โดยทำการรับ Input ข้อความ และทำการแปลงโดยใช้ TF-IDF จากนั้นนำคุณลักษณะที่ได้มาใช้ในการวิเคราะห์ด้วย ของ CNN แบบ 1 มิติ โดยใช้โครงสร้างโครงข่ายจำนวน 2 ชั้นเพื่อทำการคัดเลือกคุณลักษณะที่สำคัญของข้อมูล โดยชั้นแรกนั้นใช้ Filter ขนาด 32 หน่วย Kernel ขนาด 2 หน่วย ชั้น

ที่สองใช้ Filter ขนาด 64 หน่วย Kernel ขนาด 3 หน่วย และทั้งสองชั้นใช้ Activation Function คือ Rectified Linear Unit (RELU) จากนั้นทำการผ่าน Dropout เพื่อลดการ Overfitting และทำการผ่านตัว Flatten เพื่อลบมิติของข้อมูล และสุดท้ายทำการผ่านชั้น Fully Connected ที่ได้ออกแบบไว้สำหรับการจำแนกประเภท โดยในชั้นนี้จะมี 1 Node และใช้ Sigmoid Function เป็น Activation Function

3.3.6 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Convolutional Neural Network แบบ 2 มิติ

อัลกอริทึม Word Embedding จะทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่ได้ทำการกำหนดขึ้นและจำนวนมิติของ Word Embedding เมื่อได้ตัวคุณลักษณะของข้อมูลแล้ว จากนั้นทำการแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และทำการสร้างแบบจำลองการวิเคราะห์ความรู้สึกด้วยอัลกอริทึม Convolutional Neural Network แบบ 2 มิติ จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดงดังรูปที่ 3.7

รูปที่ 3.7 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Word Embedding และอัลกอริทึม CNN แบบ 2 มิติ

รูปที่ 3.7 แสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากด้วยคุณลักษณะ Word Embedding และอัลกอริทึม CNN แบบ 2 มิติ โดยมีขั้นตอนการทำงานดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

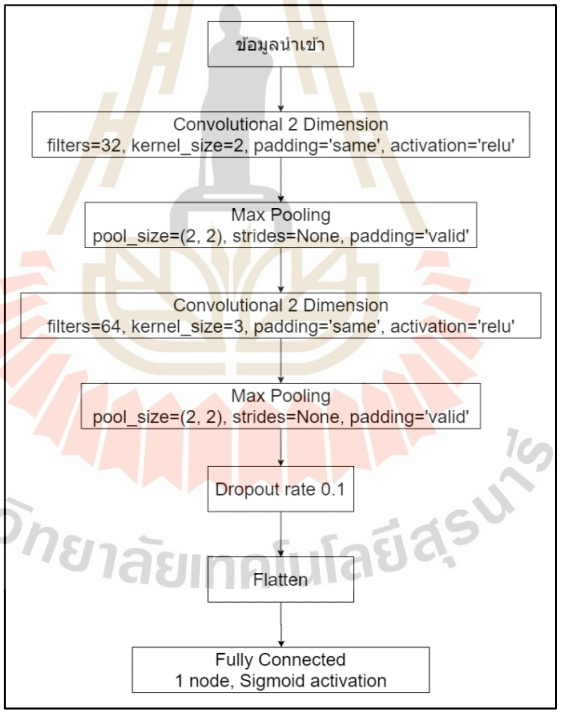
2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็ก และลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม Word Embedding

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง Word Embedding ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม CNN แบบ 2 มิติ ที่ได้ออกแบบไว้ดังรูปที่ 3.8

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย



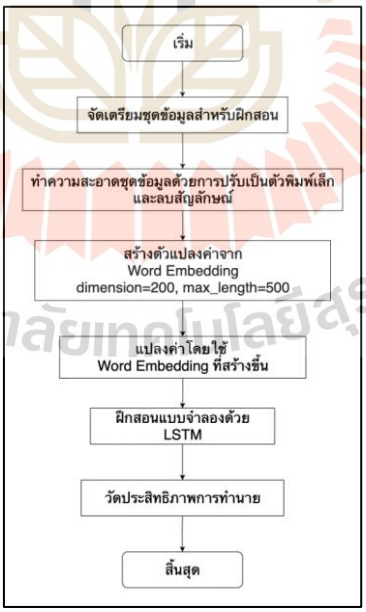
รูปที่ 3.8 โครงสร้างของอัลกอริทึม CNN แบบ 2 มิติที่ใช้ในการวิจัย

รูปที่ 3.8 แสดงโครงสร้างของอัลกอริทึม CNN แบบ 2 มิติที่ใช้ในการวิจัย โดยทำการรับ Input ข้อความ และทำการแปลงโดยใช้ Word Embedding จากนั้นนำคุณลักษณะที่ได้มาใช้ในการวิเคราะห์ด้วย CNN แบบ 2 มิติ โดยใช้โครงสร้างโครงข่ายจำนวน 2 ชั้นเพื่อทำการคัดเลือก

คุณลักษณะที่สำคัญของข้อมูล โดยชั้นแรกนั้นใช้ Filter ขนาด 32 หน่วย Kernel ขนาด 2 หน่วย ชั้นที่สองใช้ Filter ขนาด 64 หน่วย Kernel ขนาด 3 หน่วย และทั้งสองชั้นใช้ Rectified Linear Unit (RELU) เป็น Activation Function โดยการทำงานของทั้ง 2 ชั้นนั้นจะถูกคั่นด้วยชั้น Max Pooling เพื่อทำการลดขนาดของคุณลักษณะ โดยคัดเลือกตามค่าที่มากที่สุด จากนั้นทำการผ่าน Dropout เพื่อลดการ Overfitting และทำการผ่านตัว Flatten เพื่อลบมิติของข้อมูล และสุดท้ายทำการผ่านชั้น Fully Connected ที่ได้ออกแบบไว้สำหรับการจำแนกประเภท โดยในชั้นนี้จะมี 1 Node และใช้ Sigmoid Function เป็น Activation Function

3.3.7 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Long Short-Term Memory

อัลกอริทึม Word Embedding ทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้นและจำนวนมิติของ Word Embedding เมื่อได้คุณลักษณะของข้อมูลแล้ว แปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และทำการสร้างแบบจำลองการทำนายผล การวิเคราะห์ความรู้สึก ด้วยอัลกอริทึม Long Short-Term Memory จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดงดังรูปที่ 3.9



รูปที่ 3.9 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยอัลกอริทึม Word Embedding และ LSTM

รูปที่ 3.9 คือผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Word Embedding และอัลกอริทึม LSTM โดยมีขั้นตอนการทำงาน ดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็กและลบสัญลักษณ์ออก

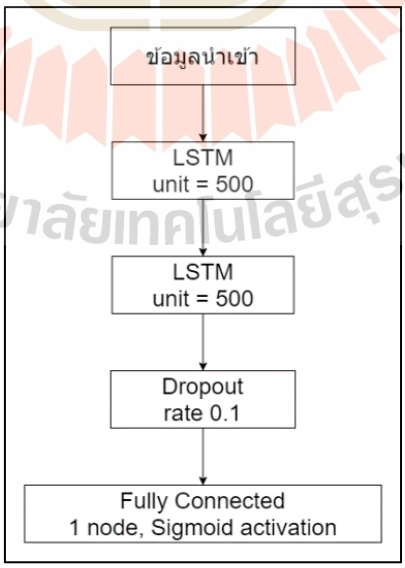
3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม Word Embedding

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง Word Embedding ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม LSTM ที่ได้ออกแบบไว้ดังรูปที่ 3.10

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย

รูปที่ 3.10 แสดงโครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ LSTM ที่ใช้ในการวิจัย โดยทำการรับ Input ข้อความ และทำการแปลงโดยใช้ Word Embedding จากนั้นนำคุณลักษณะที่ได้มาใช้ในการวิเคราะห์โครงข่ายประสาทแบบ LSTM โดยใช้โครงสร้างที่กำหนดให้ Unit มีจำนวน 500 หน่วย เป็นจำนวน 2 ชั้น จากนั้นทำการผ่านชั้น Dropout เพื่อลด Overfitting และสุดท้ายทำการจำแนกประเภทด้วย Fully Connected 1 Node โดยใช้ Sigmoid Function เป็น Activation Function



รูปที่ 3.10 โครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ LSTM ที่ใช้ในการวิจัย

3.3.8 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Gated Recurrent Unit

อัลกอริทึม Word Embedding ทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้นและจำนวนมิติของ Word Embedding เมื่อได้ตัวคุณลักษณะของข้อมูลแล้วจากนั้นทำการแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และทำการสร้างแบบจำลองการทำนายผล การวิเคราะห์ความรู้สึก ด้วยอัลกอริทึม Gated Recurrent Unit จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดง ดังรูปที่ 3.11

รูปที่ 3.11 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Word Embedding และอัลกอริทึม GRU

รูปที่ 3.11 คือผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยอัลกอริทึม Word Embedding และ GRU โดยมีขั้นตอนการทำงานดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

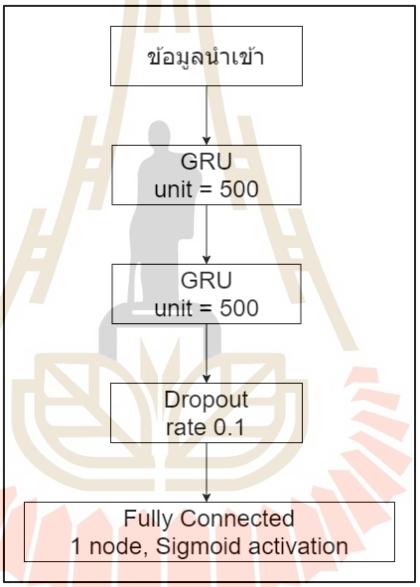
2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็กและลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม Word Embedding

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง Word Embedding ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม GRU ที่ได้ออกแบบไว้ดังรูปที่ 3.12

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย

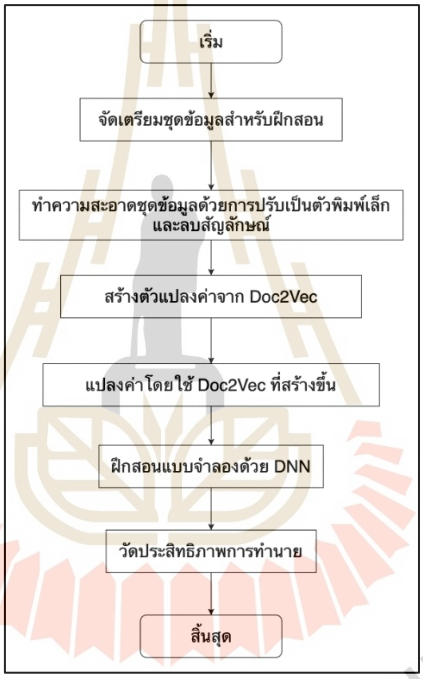


รูปที่ 3.12 โครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ GRU ที่ใช้ในการวิจัย

รูปที่ 3.12 แสดงโครงสร้างของอัลกอริทึมโครงข่ายประสาทแบบ GRU ที่ใช้ในการวิจัย โดยทำการรับ Input ข้อความ และทำการแปลงโดยใช้ Word Embedding จากนั้นนำคุณลักษณะที่ได้มาใช้ในการวิเคราะห์โครงข่ายประสาทแบบ GRU โดยใช้โครงสร้าง GRU ที่กำหนดให้ Unit มีจำนวน 500 หน่วย เป็นจำนวน 2 ชั้น จากนั้นทำการผ่านชั้น Dropout เพื่อลด Overfitting และสุดท้ายทำการจำแนกประเภทด้วย Fully Connected 1 Node โดยใช้ Sigmoid Function เป็น Activation Function

3.3.9 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Deep Neural Network

อัลกอริทึม Doc2Vec ทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้นและจำนวนของคุณลักษณะ Doc2Vec เมื่อได้ตัวคุณลักษณะของข้อมูลแล้วจากนั้นแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และทำการสร้างแบบจำลองการทำนายผล การวิเคราะห์ความรู้สึก ด้วยอัลกอริทึม Deep Neural Network จากนั้นทำการทดสอบประสิทธิภาพการทำนายและบันทึกผล โดยขั้นตอนในการทำงานแสดง ดังรูปที่ 3.13



รูปที่ 3.13 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม DNN

รูปที่ 3.13 คือผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม DNN โดยมีขั้นตอนการทำงานดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็กและลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม Doc2Vec

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง Doc2Vec ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม DNN ที่ได้ออกแบบไว้ดังรูปที่ 3.4

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย

เนื่องจากคุณลักษณะที่ได้จากการใช้อัลกอริทึม Doc2Vec นั้นมีลักษณะเดียวกันกับ TF-IDF ซึ่งอยู่ในลักษณะของเมทริกซ์ ดังนั้นทางผู้วิจัยจึงทำการออกแบบโครงสร้างของแบบจำลองของ DNN เช่นเดียวกันกับโครงสร้างดังแสดงในรูปที่ 3.4

3.3.10 แบบจำลองการวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ

อัลกอริทึม Doc2Vec ทำการสร้างคุณลักษณะของข้อความขึ้นมาตามจำนวนคำศัพท์ที่เราได้ทำการกำหนดขึ้นและจำนวนของคุณลักษณะ Doc2Vec เมื่อได้ตัวคุณลักษณะของข้อมูลแล้วจากนั้นทำการแปลงข้อมูลให้อยู่ในรูปของคุณลักษณะที่ได้และทำการสร้างแบบจำลองการทำนายผล การวิเคราะห์ความรู้สึก ด้วยอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ จากนั้นทำการทดสอบประสิทธิภาพการทำนายและทำการบันทึกผล โดยขั้นตอนในการทำงานได้แสดงดังรูปที่ 3.14

รูปที่ 3.14 ผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม CNN แบบ 1 มิติ

รูปที่ 3.14 คือผังงานแสดงขั้นตอนการทำงานของการวิเคราะห์ความรู้สึกจากข้อความด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม CNN แบบ 1 มิติ โดยมีขั้นตอนการทำงานดังนี้

1) เริ่มทำการทดลองโดยการจัดเตรียมข้อมูลสำหรับการวิเคราะห์และพัฒนาแบบจำลอง

2) ทำความสะอาดชุดข้อมูลด้วยการปรับเป็นตัวพิมพ์เล็กและลบสัญลักษณ์ออก

3) สร้างตัวแปลงข้อมูลด้วยอัลกอริทึม Doc2Vec

4) แปลงสภาพชุดข้อมูลโดยใช้ตัวแปลง Doc2Vec ที่สร้างขึ้น

5) นำข้อมูลที่ผ่านการแปลงสภาพแล้วมาทำการฝึกสอนด้วยอัลกอริทึม CNN แบบ 1 มิติที่ได้ออกแบบไว้ดังรูปที่ 3.6

6) บันทึกผลการทดลองและนำไปคำนวณประสิทธิภาพการทำนาย

เนื่องจากคุณลักษณะที่ได้จากการใช้อัลกอริทึม Doc2Vec นั้นมีลักษณะเช่นเดียวกันกับ TF-IDF ซึ่งอยู่ในลักษณะของเมทริกซ์ ดังนั้นทางผู้วิจัยจึงออกแบบโครงสร้างของแบบจำลอง CNN แบบ 1 มิติ เช่นเดียวกันกับโครงสร้างดังแสดงในรูปที่ 3.6

3.4 การฝึกสอนแบบจำลอง

ในการฝึกสอนแบบจำลองจากอัลกอริทึมเทคนิคการเรียนรู้เชิงลึกต่าง ๆ นั้น ทางผู้วิจัยได้ทำการออกแบบการฝึกสอนแบบจำลองต่าง ๆ โดยฝึกสอนด้วย Epoch จำนวน 20 รอบและกำหนด Optimize Function ด้วยอัลกอริทึม Adam Optimizer ด้วย Learning Rate เท่ากับ 0.001 จากนั้นนำแบบจำลองที่ได้ มาทำการคำนวณประสิทธิภาพของการทำนาย

3.5 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ ในส่วนซอฟต์แวร์ใช้ Google Colaboratory เป็นหลัก ซึ่งนำ Jupyter Notebook ไปสร้างบนระบบ Cloud Computer ของ Google ที่สามารถใช้งานได้ฟรี ซึ่งมีรายละเอียดฮาร์ดแวร์และซอฟต์แวร์ ดังนี้

1) ฮาร์ดแวร์ที่ใช้สำหรับงานวิจัย ประกอบด้วย

- หน่วยประมวลผลกลาง Intel Xeon @ 2x 2GHz
- หน่วยความจำหลัก 13022 MB
- หน่วยประมวลผลกราฟิก Nvidia Tesla T4

2) ระบบปฏิบัติการและโปรแกรมประยุกต์สำหรับพัฒนาแบบจำลอง ประกอบด้วย

- ระบบปฏิบัติการ Ubuntu 18.04 bionic
- เครื่องมือที่ใช้ในการสร้างตัวคัดเลือกคุณลักษณะของข้อความได้แก่ Scikit-learn, Gensim และ Keras
- เครื่องมือที่ใช้ในการพัฒนาแบบจำลอง ได้แก่ Keras

บทที่ 4
ผลการทดลองและอภิปรายผล

การทดสอบประสิทธิภาพแบบจำลองสำหรับจำแนกอารมณ์ของข้อความการแสดงความคิดเห็นต่อสินค้า จะทดสอบด้วยค่าความแม่นยำในการจำแนก (Accuracy) ค่าความเที่ยง (Precision) ค่าความไวหรือค่าระลึก (Recall) และระยะเวลาที่ใช้ในการฝึกสอนแบบจำลอง โดยเปรียบเทียบการจำแนกของแบบจำลองที่ถูกฝึกสอนด้วยคุณลักษณะในรูปแบบต่าง ๆ ตามที่ผู้วิจัยได้ทำการออกแบบไว้ สำหรับเนื้อหาในบทนี้จะประกอบไปด้วย ผลการทดสอบประสิทธิภาพแบบจำลองต่าง ๆ และการอภิปรายผล

4.1 ผลการทดสอบประสิทธิภาพ

สำหรับการทดสอบประสิทธิภาพการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อสินค้า ผู้วิจัยได้ทำการเลือกข้อมูลบางหมวดหมู่จากชุดข้อมูลมาใช้ในการศึกษารวมทั้งสิ้น 6 หมวดหมู่ ได้แก่ หนังสือ, ดีวีดี, เครื่องใช้ไฟฟ้า, เครื่องครัว, เพลง และวีดีโอ โดยมีผลการทดลอง ดังนี้

4.1.1 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Deep Neural Network

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม TF-IDF และสร้างแบบจำลองโดยใช้อัลกอริทึม DNN โดยให้ผลการทดลองดังตารางที่ 4.1

จากตารางที่ 4.1 จะเห็นได้ว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน โดยให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.84 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.84 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.83 และเวลาที่ใช้ในการฝึกสอนโดยเฉลี่ย 5.12 วินาที

ตารางที่ 4.1 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ TF-IDF และใช้อัลกอริทึม DNN ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.82	0.84	0.79	5.19 s
dvd	0.83	0.83	0.82	5.08 s
electronics	0.83	0.83	0.82	5.14 s
kitchen	0.85	0.85	0.85	5.05 s
music	0.85	0.85	0.85	5.10 s
video	0.85	0.85	0.85	5.07 s
Average	**0.84**	**0.84**	**0.83**	**5.12 s**

4.1.2 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ TF-IDF และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม TF-IDF และสร้างแบบจำลองโดยใช้อัลกอริทึม CNN แบบ 1 มิติ ให้ผลการทดลองดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ TF-IDF และใช้อัลกอริทึม CNN แบบ 1 มิติในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.81	0.82	0.78	18.9 s
dvd	0.82	0.83	0.81	18.7 s
electronics	0.82	0.83	0.81	18.7 s
kitchen	0.83	0.82	0.85	18.6 s
music	0.83	0.82	0.85	18.6 s
video	0.83	0.82	0.85	18.7 s
Average	**0.82**	**0.82**	**0.83**	**18.7 s**

จากตารางที่ 4.2 จะเห็นได้ว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน โดยให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.82 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.82 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.83 และเวลาที่ใช้ในการฝึกสอนโดยเฉลี่ย 18.7 วินาที

4.1.3 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Convolutional Neural Network แบบ 2 มิติ

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม Word Embedding และสร้างแบบจำลองโดยใช้อัลกอริทึม CNN แบบ 2 มิติ ให้ผลการทดลองดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม CNN แบบ 2 มิติ ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.82	0.81	0.83	1:55 min
dvd	0.77	0.77	0.77	1:54 min
electronics	0.77	0.77	0.77	1:55 min
kitchen	0.77	0.76	0.8	1:54 min
music	0.77	0.76	0.8	1:54 min
video	0.77	0.76	0.8	1:55 min
Average	**0.78**	**0.77**	**0.80**	**1:55 min**

จากตารางที่ 4.3 จะเห็นได้ว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน โดยให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.82 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.82 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.83 และใช้เวลาในการฝึกสอนโดยเฉลี่ยอยู่ที่ 1 นาที 55 วินาที

4.1.4 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Long Short-Term Memory

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม Word Embedding และสร้างแบบจำลองโดยใช้อัลกอริทึม LSTM ให้ผลการทดลองดังตารางที่ 4.4

ตารางที่ 4.4 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม LSTM ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.61	0.66	0.46	3:25 min
dvd	0.73	0.72	0.74	3:27 min
electronics	0.73	0.72	0.74	3:26 min
kitchen	0.75	0.73	0.79	3:26 min
music	0.75	0.73	0.79	3:26 min
video	0.75	0.73	0.79	3:25 min
Average	**0.72**	**0.72**	**0.72**	**3:26 min**

จากตารางที่ 4.4 สังเกตได้ว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน และให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.72 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.72 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.72 และใช้เวลาในการฝึกสอนโดยเฉลี่ยอยู่ที่ 3 นาที 26 วินาที

4.1.5 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Word Embedding และอัลกอริทึม Gated Recurrent Unit

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม Word Embedding และสร้างแบบจำลองโดยใช้อัลกอริทึม GRU ให้ผลการทดลองดังตารางที่ 4.4

ตารางที่ 4.5 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Word Embedding และใช้อัลกอริทึม GRU ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.68	0.68	0.69	2:52 min
dvd	0.71	0.72	0.69	2:52 min
electronics	0.71	0.72	0.69	2:51 min
kitchen	0.75	0.73	0.79	2:51 min
music	0.75	0.73	0.79	2:51 min
video	0.75	0.73	0.79	2:51 min
Average	**0.73**	**0.72**	**0.74**	**2:51 min**

จากตารางที่ 4.5 พบว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน และให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.73 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.72 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.74 และใช้เวลาในการฝึกสอนโดยเฉลี่ยอยู่ที่ 3 นาที 26 วินาที

4.1.6 การทดลองแบบจำลองวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Deep Neural Network

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม Doc2Vec และสร้างแบบจำลองโดยใช้อัลกอริทึม DNN ให้ผลการทดลองดังตารางที่ 4.6

ตารางที่ 4.6 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Doc2Vec และใช้อัลกอริทึม DNN ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.74	0.77	0.69	9.50 s
dvd	0.71	0.84	0.52	11.2 s
electronics	0.71	0.79	0.58	8.55 s
kitchen	0.69	0.66	0.79	8.63 s
music	0.70	0.79	0.54	8.73 s
video	0.76	0.75	0.76	5.90 s
Average	**0.72**	**0.77**	**0.65**	**8.75 s**

จากตารางที่ 4.6 พบว่าการพัฒนาแบบจำลองในแต่ละชุดข้อมูลนั้นใช้เวลาในการฝึกสอนที่แตกต่างกันในบางชุดข้อมูล และให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.72 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.77 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.65 และใช้เวลาในการฝึกสอนโดยเฉลี่ยอยู่ที่ 8.75 วินาที

4.1.7 การทดลองอัลกอริทึมวิเคราะห์ความรู้สึกด้วยคุณลักษณะ Doc2Vec และอัลกอริทึม Convolutional Neural Network แบบ 1 มิติ

จากการทดลองสร้างแบบจำลองการจำแนกอารมณ์ของผู้ใช้ผลิตภัณฑ์ผ่านข้อความการแสดงความคิดเห็นต่อผลิตภัณฑ์ โดยใช้คุณลักษณะจากอัลกอริทึม Doc2Vec และสร้างแบบจำลองโดยใช้อัลกอริทึม CNN แบบ 1 มิติ ให้ผลการทดลองดังตารางที่ 4.7

ตารางที่ 4.7 ผลการทดสอบแบบจำลองที่ใช้คุณลักษณะ Doc2Vec และใช้อัลกอริทึม CNN แบบ 1 มิติ ในการพัฒนา

Dataset	Accuracy	Precision	Recall	Time
book	0.72	0.79	0.61	16.1 s
dvd	0.77	0.74	0.85	17.5 s
electronics	0.70	0.70	0.71	14.5 s
kitchen	0.72	0.67	0.86	14.5 s
music	0.73	0.75	0.69	14.8 s
video	0.76	0.76	0.76	7.98 s
Average	**0.73**	**0.74**	**0.75**	**14.23 s**

จากตารางที่ 4.7 สังเกตได้ว่าการพัฒนาแบบจำลองในชุดข้อมูลส่วนใหญ่ใช้เวลาในการฝึกสอนที่ใกล้เคียงกัน และให้ค่าความถูกต้องโดยเฉลี่ยอยู่ที่ 0.73 ค่าความเที่ยงโดยเฉลี่ยอยู่ที่ 0.74 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.75 และใช้เวลาในการฝึกสอนโดยเฉลี่ยอยู่ที่ 14.23 วินาที

ในการออกแบบการทดลอง ทางผู้วิจัยได้ทำการแบ่งการทดลองออกเป็น 3 ส่วน โดยแบ่งตามการเลือกใช้อัลกอริทึมในการสร้างคุณลักษณะของข้อมูลในรูปแบบต่าง ๆ ได้แก่ TF-IDF, Word Embedding และ Doc2Vec โดยช่วงแรกจะเป็นการทดลองโดยใช้ข้อมูลจากคุณลักษณะ TF-IDF กับอัลกอริทึม DNN และอัลกอริทึม CNN แบบ 1 มิติสำหรับสร้างแบบจำลอง พบว่าประสิทธิภาพด้านความถูกต้อง ความเที่ยง และความไวเป็นไปในทิศทางเดียวกัน โดยแบบจำลองอัลกอริทึม DNN นั้น ให้ประสิทธิภาพในการทำนายสูงที่สุดในทุก ๆ การทดลอง

การทดลองโดยใช้ข้อมูลจากคุณลักษณะ Word Embedding ซึ่งเป็นคุณลักษณะที่อยู่ในรูปแบบของอนุกรมเวลา ทำให้ข้อจำกัดในการเลือกใช้อัลกอริทึมสำหรับการสร้างแบบจำลอง โดยทางผู้วิจัยได้เลือกอัลกอริทึม CNN แบบ 2 มิติ อัลกอริทึม GRU และ อัลกอริทึม LSTM สำหรับสร้างแบบจำลอง โดยผลลัพธ์ที่ได้จากแบบจำลองอัลกอริทึม CNN แบบ 2 มิติ ให้ประสิทธิภาพที่สูงเมื่อวัดจากค่าประสิทธิภาพต่าง ๆ จากแบบจำลองที่ใช้ข้อมูลในลักษณะเดียวกัน

การทดลองโดยใช้ข้อมูลจากคุณลักษณะ Doc2Vec ซึ่งเป็นการนำเอกสารหรือข้อความทั้งหมด มาเข้ารหัสให้อยู่ในคุณลักษณะของเวกเตอร์กับอัลกอริทึม DNN และ อัลกอริทึม CNN แบบ 1 มิติสำหรับสร้างแบบจำลอง โดยผลลัพธ์ที่ได้จากแบบจำลอง CNN แบบ 1 มิติ ให้ประสิทธิภาพที่สูงเมื่อวัดจากค่าประสิทธิภาพต่าง ๆ จากแบบจำลองที่ใช้ข้อมูลในลักษณะเดียวกัน

จากนั้นผู้วิจัยได้ทำการนำข้อมูลที่ได้จากการทดลองแต่ละครั้ง มาทำการสรุป โดยหาค่าเฉลี่ยของค่าประสิทธิภาพต่าง ๆ และได้ทำการแบ่งผลสรุปออกเป็นการทดลองที่ใช้คุณลักษณะในรูปแบบของเมทริกซ์ได้แก่คุณลักษณะ TF-IDF และ Doc2Vec ดังแสดงในตารางที่ 4.8 และคุณลักษณะแบบอนุกรมได้แก่ Word Embedding ดังแสดงในตารางที่ 4.9 เมื่อทำการนำผลลัพธ์ที่ได้จากการทดลองในแต่ละชุดข้อมูล มาหาค่าเฉลี่ย พบว่าแบบจำลอง DNN ที่ใช้คุณลักษณะของ TF-IDF นั้นให้ประสิทธิภาพในการทำนายในด้านของค่าความถูกต้องสูงที่สุดในทุกชุดข้อมูลที่ใช้ในการทดสอบ

4.2 อภิปรายผล

จากผลการทดสอบประสิทธิภาพของแบบจำลองการจำแนกอารมณ์ของข้อความการแสดงความคิดเห็นต่อสินค้า ซึ่งจำแนกออกเป็น 2 ประเภทได้แก่ด้านบวกและด้านลบ โดยมีกระบวนการทำงานอยู่ 2 ขั้นตอนได้แก่การสร้างคุณลักษณะจากข้อมูลข้อความตัวอักษร และการสร้างแบบจำลองสำหรับจำแนกอารมณ์ สามารถอภิปรายผลการทดลองได้ดังนี้

1) การทดลองส่วนของอัลกอริทึมการเรียนรู้เชิงลึกที่ใช้นั้น ประกอบไปด้วยอัลกอริทึมที่ใช้วิเคราะห์กับคุณลักษณะในรูปแบบของเมทริกซ์ ได้แก่อัลกอริทึม DNN อัลกอริทึม CNN แบบ 1 มิติ และอัลกอริทึม CNN แบบ 2 มิติ และ อัลกอริทึมที่ใช้วิเคราะห์คุณลักษณะในรูปของอนุกรมได้แก่อัลกอริทึม LSTM อัลกอริทึม GRU จากผลการทดลองพบว่าแบบจำลองในกลุ่มที่ใช้คุณลักษณะในรูปแบบของเมทริกซ์ให้ประสิทธิภาพในการทำนายได้ดีกว่าแบบจำลองที่ใช้คุณลักษณะ TF-IDF โดยอัลกอริทึมที่มีประสิทธิภาพในการทำนายที่สูงที่สุด ได้แก่อัลกอริทึม DNN

ตารางที่ 4.8 ผลการทดสอบแบบจำลองโดยเฉลี่ยของการใช้คุณลักษณะแบบเมทริกซ์ในการทดลอง

Algorithms		**Accuracy**	**Precision**	**Recall**	**Time to Train**
Feature Extraction	**Model Development**				
TF-IDF	DNN	**0.84**	**0.84**	**0.83**	**5.12 s**
TF-IDF	CNN1D	0.82	0.82	**0.83**	18.7 s
Doc2Vec	DNN	0.72	0.77	0.65	8.75 s
Doc2Vec	CNN1D	0.73	0.74	0.75	14.23 s

ตารางที่ 4.9 ผลการทดสอบแบบจำลองโดยเฉลี่ยของการใช้คุณลักษณะแบบอนุกรมในการทดลอง

Algorithms		Accuracy	Precision	Recall	Time to Train
Feature Extraction	Model Development				
Word Embedding	CNN2D	**0.78**	**0.77**	**0.80**	**1:55 min**
Word Embedding	LSTM	0.72	0.72	0.72	3:26 min
Word Embedding	GRU	0.73	0.72	0.74	2:51 min

2) เมื่อเปรียบเทียบระยะเวลาที่ใช้ในการฝึกสอนแบบจำลอง พบว่าแบบจำลองในกลุ่มของเมทริกซ์นั้นใช้เวลาในการฝึกสอนที่น้อยกว่าแบบจำลองที่อยู่ในกลุ่มของอนุกรม เนื่องจากแบบจำลองในกลุ่มของอนุกรมนั้นมีจำนวนตัวแปรที่ใช้ในการเรียนรู้มากกว่า และวิธีการเรียนรู้ในรูปแบบของอนุกรม ทำให้ใช้เวลาในการฝึกสอนที่นานกว่าแบบจำลองในกลุ่มของเมทริกซ์ โดยแบบจำลองที่ใช้เวลาในการเรียนรู้น้อยที่สุดได้แก่แบบจำลอง DNN

3) การทดลองในส่วนการสร้างคุณลักษณะของข้อมูลข้อความเพื่อใช้ในการเรียนรู้ของแบบจำลองจำแนกอารมณ์ สำหรับการทดลองนี้ ผู้วิจัยได้ทำการศึกษาเทคนิค TF-IDF ที่เป็นการวิเคราะห์คลังคำศัพท์ที่มีอยู่ ทั้งในด้านของความถี่ของคำและจำนวนคำศัพท์ที่สามารถพบได้ยาก และเทคนิคการเข้ารหัสเวกเตอร์อย่าง Word Embedding พบว่าคุณลักษณะของข้อมูลจากเทคนิค TF-IDF มีประสิทธิภาพในการทำนายที่สูงกว่า Word Embedding

4) เมื่อพิจารณาแล้วพบว่าการใช้ GloVe ในการใช้งาน Word Embedding นั้นมีการสร้างเวกเตอร์ขึ้นมาโดยเป็นการสร้างความสัมพันธ์ของคำแต่ละคำเท่านั้น โดยข้อความแต่ละข้อความนั้นมีความยาวที่แตกต่างกัน จึงได้ทำการนำ Doc2Vec ซึ่งเป็นการใช้เทคนิค Word Embedding ในการเข้ารหัสข้อมูลที่อยู่ในรูปแบบเอกสารและทำการแปลงข้อมูลออกมาให้อยู่ในรูปของเวกเตอร์ของเอกสาร จากนั้นทำการทดสอบเพื่อเปรียบเทียบประสิทธิภาพกับ TF-IDF พบว่า TF-IDF นั้นยังให้ประสิทธิภาพในการทำนายที่สูงกว่า

5) จากผลการทดลองข้างต้นที่ได้อภิปรายไปนั้น พบว่าการใช้คุณลักษณะที่ได้จาก TF-IDF และการใช้อัลกอริทึม DNN ในการสร้างแบบจำลอง ให้ประสิทธิภาพที่ดีที่สุดสำหรับการวิจัยนี้ โดยมีประสิทธิภาพการทำนายซึ่งวัดด้วยค่าเฉลี่ยของค่าประสิทธิภาพต่าง ๆ จากการทดลองในแต่ละครั้ง ซึ่งให้ค่าความถูกต้องโดยเฉลี่ยที่ 0.84 ค่าความเที่ยงโดยเฉลี่ยที่ 0.84 ค่าความไวโดยเฉลี่ยอยู่ที่ 0.83 และเวลาที่ใช้ในการฝึกสอนโดยเฉลี่ยที่ 5.12 วินาที

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการทดลองพัฒนาแบบจำลองวิเคราะห์ความรู้สึกจากผู้ใช้ผลิตภัณฑ์ด้วยเทคนิคการเรียนรู้เชิงลึก ซึ่งการทดลองนี้ได้ทำการกำหนดเป้าหมายในการทำนายอยู่ 2 เป้าหมาย ได้แก่ข้อความที่แสดงความรู้สึกด้านบวกและข้อความที่แสดงความรู้สึกด้านลบ โดยได้ทำการทดลองทั้งในส่วนของอัลกอริทึมการสร้างคุณลักษณะจากข้อความ และอัลกอริทึมสำหรับการสร้างแบบจำลองด้วยเทคนิคการเรียนรู้เชิงลึก ซึ่งการที่จะสร้างแบบจำลองจากข้อความนั้น จะต้องเริ่มต้นด้วยการทำให้คอมพิวเตอร์เข้าใจภาษาของมนุษย์เสียก่อน จึงต้องมีการสร้างอัลกอริทึมที่ใช้สำหรับแปลงข้อความให้อยู่ในรูปของตัวเลขที่นำไปใช้ในการคำนวณต่อได้ โดยผู้วิจัยได้ทำการเลือกใช้อัลกอริทึม TF-IDF สำหรับการสร้างคุณลักษณะ เนื่องจาก TF-IDF สามารถวิเคราะห์ทั้งในด้านของความถี่ของคำและโอกาสในการพบเจอคำในเอกสารต่าง ๆ อีกด้วย สำหรับการสร้างแบบจำลองผู้วิจัยเลือกใช้อัลกอริทึม DNN ซึ่งเป็นหนึ่งในอัลกอริทึมการเรียนรู้เชิงลึก ซึ่งมีข้อดีในด้านของการจัดการคุณลักษณะจำนวนมากได้ดีจากการที่มีหน่วยในการเรียนรู้และพารามิเตอร์จำนวนมาก ทำให้วิเคราะห์ข้อมูลได้ถูกต้องมากยิ่งขึ้น

ต่อมาทางผู้วิจัยได้ทำการวัดประสิทธิภาพการทำนายของแบบจำลอง พบว่าแบบจำลอง DNN ที่ใช้คุณลักษณะของ TF-IDF ในการฝึกสอนนั้น ให้ประสิทธิภาพในการทำนายที่สูงกว่าแบบจำลองอื่น ๆ โดยมีประสิทธิภาพการทำนายซึ่งวัดด้วยค่าเฉลี่ยของค่าประสิทธิภาพต่าง ๆ จากการทดลองในแต่ละครั้ง ซึ่งให้ค่าความถูกต้องโดยเฉลี่ยที่ 0.84 ค่าความเที่ยงโดยเฉลี่ยที่ 0.84 ค่าความไวโดยเฉลี่ยอยู่ที่ ที่ 0.83 และเวลาที่ใช้ในการฝึกสอนโดยเฉลี่ยที่ 5.12 วินาที

5.2 การประยุกต์ผลการวิจัย

จากผลการวิจัยแสดงให้เห็นว่าอัลกอริทึมการเรียนรู้เชิงลึกนั้นสามารถขยายขอบเขตการทำงานทางด้านการเรียนรู้ของเครื่องได้อย่างหลากหลาย โดยในงานวิจัยนี้ได้ทำการนำมาประยุกต์ใช้ในการวิเคราะห์ข้อมูลที่อยู่ในรูปแบบของข้อความเพื่อจำแนกอารมณ์ ซึ่งสามารถนำไป

ประยุกต์ใช้ร่วมกับการจำแนกข้อความในด้านอื่น ๆ ได้เช่น การจำแนกข่าวปลอม หรือการจำแนกผลตอบรับต่อผลิตภัณฑ์โดยใช้ข้อมูลบนเครือข่ายสังคมออนไลน์เป็นต้น โดยสามารถให้ประสิทธิภาพในการทำงานที่สูง และมีความรวดเร็วในการทำงาน

5.3 ข้อเสนอแนะ

จากการทดลองพบว่า การที่แบบจำลองจะให้ประสิทธิภาพในการทำนายที่สูงนั้น ไม่เพียงแค่การเลือกอัลกอริทึมที่เหมาะสมสำหรับการสร้างแบบจำลอง แต่รวมไปถึงการคัดเลือกหรือทำการสร้างคุณลักษณะของข้อมูลที่สามารถนำไปใช้ในการวิเคราะห์ ให้เหมาะสมกับประเภทงานที่จะวิเคราะห์ด้วย ซึ่งการที่จะนำไปใช้งานกับชุดข้อมูลประเภทอื่น ๆ นั้น จะต้องทำการวิเคราะห์ข้อมูลและสร้างคุณลักษณะที่เหมาะสมสำหรับการใช้งานด้วย

รายการอ้างอิง

Allouch, N. (2018). Sentiment and Emotional Analysis: The Absolute Difference. Emojics Blog. Retrieved from **https://www.emojics.com/blog/emotional-analysis-vs-sentiment-analysis/#:~:text=While%20sentiment%20analysis%20helps%20you,wide%20universe%20of%20human%20emotions/**

AltexSoft. (2020). Sentiment Analysis: Types, Tools, and Use Cases. Retrieved January 5, 2020, from **https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/**

Aone, C., Okurowski, M. E., & Gorlinsky, J. (1998, August). Trainable, scalable summarization using robust NLP and machine learning. In **Proceedings of the 17th International Conference on Computational Linguistics-Volume 1** (pp. 62-66). Association for Computational Linguistics.

Bisong, E. (2019). Google Colaboratory. In **Building Machine Learning and Deep Learning Models on Google Cloud Platform** (pp. 59-64). Apress, Berkeley, CA.

Blitzer, J., Dredze, M., & Pereira, F. (2007, June). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In **Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics** (pp. 440-447).

Cai, S., Palazoglu, A., Zhang, L., & Hu, J. (2019). Process alarm prediction using deep learning and word embedding methods. **ISA Transactions**, 85, 274-283.

Cavnar, W. B., & Trenkle, J. M. (1994, April). N-gram-based text categorization. In **Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval** (Vol. 161175).

Chollet, F. (2015). Keras documentation**. Keras. io**.

Chowdhury, G. G. (2003). Natural language processing**. Annual Review of Information Science and Technology**, 37(1), 51-89.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. **arXiv preprint arXiv:1412.3555**.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Le, Q. V. (2012). Large scale distributed deep networks. In **Advances in Neural Information Processing Systems** (pp. 1223-1231).

Ganguly, D., Roy, D., Mitra, M., & Jones, G. J. (2015, August). Word embedding based generalized language model for information retrieval. In **Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval** (pp. 795-798).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT press.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. **Procedia Computer Science**, *17*, 26-32.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., & Dally, W. J. (2016). EIE: efficient inference engine on compressed deep neural network. A**CM SIGARCH Computer Architecture News**, 44(3), 243-254.

Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In **Neural Networks for Perception** (pp. 65-93). Academic Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. **Neural Computation**, 9(8), 1735-1780.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. **arXiv preprint arXiv**:1404.2188.

Kowalczyk, R., Ulieru, M., & Unland, R. (2002, October). Integrating mobile and intelligent agents in advanced e-commerce: A survey. In **Net. ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World** (pp. 295-313). Springer, Berlin, Heidelberg.

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In **Proceedings of International Conference on Machine Learning** (pp. 1188-1196).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. **Nature**, 521(7553), 436-444.

Liu, P., Qiu, X., & Huang, X. (2017). Adversarial multi-task learning for text classification. **arXiv preprint arXiv:1704.05742**.

Mao, H., Han, S., Pool, J., Li, W., Liu, X., Wang, Y., & Dally, W. J. (2017). Exploring the regularity of sparse structure in convolutional neural networks. **arXiv preprint arXiv:1705.08922**.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In **Proceeding of Eleventh Annual Conference of the International Speech Communication Association**.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In **Advances in Neural Information Processing Systems** (pp. 3111-3119).

Nader, R. (2019). Creating a Custom Classifier for Text Cleaning. Retrieved January 5, 2020, **from https://towardsdatascience.com/creating-a-custom-classifier-for-text-cleaning-a2a1fc818935**

Nayak, A. S., Kanive, A. P., Chandavekar, N., & Balasubramani, R. (2016). Survey on pre-processing techniques for text mining. **International Journal of Engineering and Computer Science**, 5(6), 16875-16879.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. **The Journal of Machine Learning Research**, 12, 2825-2830.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)** (pp. 1532-1543).

Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. **Nature**, 563(7732), 145-147.

Premjith, B., Soman, K. P., & Kumar, M. A. (2018). A deep learning approach for Malayalam morphological analysis at character level. **Procedia Computer Science**, 132, 47-54.

Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In **Proceedings of the First Instructional Conference on Machine Learning** (Vol. 242, pp. 133-142).

Rao, T. R. K., Khan, S. A., Begum, Z., & Divakar, C. (2013, December). Mining the E-commerce cloud: A survey on emerging relationship between web mining, E-commerce and cloud computing. In **Proceeding of 2013 IEEE International Conference on Computational Intelligence and Computing Research** (pp. 1-4).

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017, September). Truth of varying shades: Analyzing language in fake news and political fact-checking**.** In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing** (pp. 2931-2937).

Řehůřek, R., & Sojka, P. (2011). Gensim—statistical semantics in python. Retrieved from **genism. org.**

Rossum, G. (1995). **Python Reference Manual**. Retrieved from https://docs.python.org/2.0/ref/ ref.html

Schank, R. C., & Abelson, R. P. (2013). **Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures**. Psychology Press.

Shabani, S., & Sokhn, M. (2018, October). Hybrid machine-crowd approach for fake news detection. In **Proceeding of 2018 IEEE 4th International Conference on Collaboration and Internet Computing** (CIC) (pp. 299-306).

Singh, B., & Singh, H. K. (2010, December). Web data mining research: a survey. In **Proceeding of 2010 IEEE International Conference on Computational Intelligence and Computing Research** (pp. 1-10).

Tan, A. H. (1999, April). Text mining: The state of the art and the challenges. In **Proceedings of the PAKDD 1 9 9 9 Workshop on Knowledge Disocovery from Advanced Databases** (Vol. 8, pp. 65-70).

Täuscher, K., & Laudien, S. M. (2018). Understanding platform business models: A mixed methods study of marketplaces. **European Management Journal**, 36(3), 319-329.

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. **International Journal of Machine Learning and Cybernetics**, 1(1-4), 43-52.

Zhou, P., Liu, J., Liu, X., Yang, Z., & Grundy, J. (2019). Is deep learning better than traditional approaches in tag recommendation for software information sites?. **Information and Software Technology**, 109, 1-13

ภาคผนวก ก

การใช้งาน Python เพื่อการประมวลผลข้อความและการเรียนรู้เชิงลึก

การใช้งานโปรแกรม

1. Python

Python เป็นภาษาสำหรับการเขียนโปรแกรม มีต้นกำเนิดมาตั้งแต่ปี ค.ศ. 1989 (Rossum, 1995) โดยเป็นภาษาที่อยู่ในประเภทของ Interpreter และเขียนในรูปแบบของสคริป (Script) ซึ่งสามารถทำการเขียนและแสดงผลออกมาได้อย่างทันที ซึ่งทำให้มีความเร็วในการประมวลผลที่สูงกว่าภาษาโปรแกรมในรูปแบบของ Compiler ซึ่งจะทำการนำชุดคำสั่งที่เขียนไปทำการประมวลผลก่อนจึงจะสามารถทำงานได้ โดยภาษา Python นั้นถูกออกแบบขึ้นมาให้อยู่ในรูปแบบที่มนุษย์สามารถทำความเข้าใจกับชุดคำสั่งได้ โดย Python มีการทำงานดังแสดงในรูปที่ ก.1

รูปที่ ก.1 ตัวอย่างการใช้งานภาษา Python

Python เป็นภาษาที่จัดอยู่ในกลุ่มของ Dynamically Type หมายถึงการที่ Python สามารถกำหนดชนิดของตัวแปรได้อย่างอัตโนมัติ สามารถปรับเปลี่ยนได้ตลอดการทำงานของโปรแกรม และสามารถรองรับคำสั่งการทำงานของโปรแกรม ทั้งในรูปแบบของ Procedure, Object Oriented และ Functional Programming เป็นต้น

Python นั้นมี Syntax ที่แตกต่างจากภาษาอื่น ๆ ในส่วนของการเว้นระยะ (Indentation) ซึ่งเป็นตัวกำหนดส่วนต่าง ๆ ของโปรแกรมได้ เช่นการกำหนดฟังก์ชัน การกำหนดเงื่อนไข หรือการกำหนดการวนลูป เป็นต้น ดังแสดงในรูปที่ ก.2 โดยการเว้นระยะนั้นทำให้มนุษย์สามารถอ่านโค้ดเข้าใจได้ง่าย และเพิ่มความเป็นระเบียบให้กับชุดคำสั่งของโปรแกรม

```
def sample():
    if(True):
        print("Hello World")
        for i in range(1,5):
            print(i)
    else:
        print("Nothing")
```

รูปที่ ก.2 ตัวอย่างการใช้งานฟังก์ชัน การกำหนดเงื่อนไข และการวนรอบโดยใช้การเว้นระยะในการกำหนดขอบเขตการทำงาน

Python ถูกปล่อยให้ใช้งานเป็นครั้งแรกปี ค.ศ. 1991 โดยในปี 2000 ได้ทำการปล่อย Python 2 ออกมา ซึ่งได้ทำการเพิ่มการใช้งาน list comprehensions และ garbage collection และในปี ค.ศ. 2008 ได้ทำการปล่อย Python 3 ออกมา ซึ่งเป็นเวอร์ชั่นอัพเดทที่ไม่สามารถทำการใช้ชุดคำสั่งเช่นเดียวกันกับที่ใช้ใน Python 2 ได้ โดยในปัจจุบันเวอร์ชั่นที่ยังใช้งานอยู่ของภาษา Python ได้แก่ Python 2.7 และ Python 3 โดย Python 2.7 นั้นจะทำการยุติการสนับสนุนในวันที่ 1 มกราคม ค.ศ. 2020

Python นั้นเป็นภาษาโปรแกรมที่สามารถใช้งานได้บนหลายหลายระบบปฏิบัติการ และได้ทำการพัฒนาอย่างต่อเนื่องโดย Python Software Foundation ซึ่งเป็นองค์กรไม่แสวงหาผลกำไร โดย Python นั้นถูกจัดอยู่ในกลุ่มของ Opensource ซึ่งทุกคนสามารถใช้งานภาษาโปรแกรมนี้ได้ฟรี และเปิดชุดคำสั่งของภาษาให้กับกลุ่มนักพัฒนาในการพัฒนาและปรับปรุงตัวภาษา Python ให้มีความปลอดภัยและมีประสิทธิภาพกรใช้งานที่สูงอยู่ตลอด

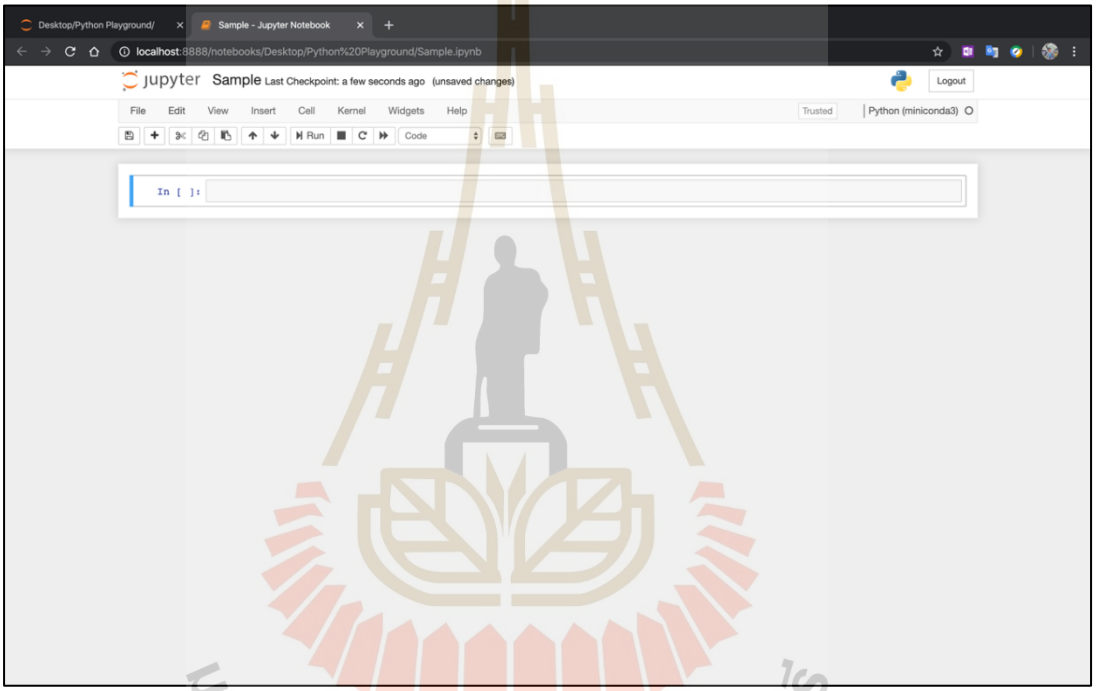
Python นั้นไม่เพียงแต่ได้รับความนิยมในการใช้งานทางด้านการพัฒนาโปรแกรมต่าง ๆ แต่ Python นั้นยังได้รับความนิยมในการใช้งานทางด้านวิทยาศาสตร์ข้อมูลอีกด้วย โดยภาษา Python นั้นทำงานในรูปแบบของ Interpreter ซึ่งสามารถทำการแยกผลการรันโปรแกรมออกเป็นส่วน ๆ ได้ และยังรวมไปถึงการมีเครื่องมือต่าง ๆ ที่ถูกพัฒนามาให้ใช้บน Python ที่มีประสิทธิภาพในการวิเคราะห์ข้อมูลที่สูง เช่น Jupyter, Keras เป็นต้น

2. Jupyter

Jupyter เป็นเครื่องมือที่ถูกพัฒนาขึ้นสำหรับการใช้งานทางด้านของการวิเคราะห์ข้อมูล (Perkel, 2018) โดย Jupyter นั้น ถูกออกแบบให้ใช้งานในรูปแบบ Interactive Computing หมายถึง

การมีหน่วยการรันย่อยในรูปแบบของ Cell ซึ่งแต่ละ Cell ที่ถูกรันคำสั่ง จะสามารถแสดงผลของการรันออกมาได้ทันที โดยที่ไม่จำเป็นต้องรันชุดคำสั่งที่อยู่ในไฟล์ทั้งหมด ซึ่งเพิ่มความสะดวกในการวิเคราะห์ข้อมูลเป็นอย่างมาก ด้วยการวิเคราะห์ข้อมูลเป็นส่วน ๆ

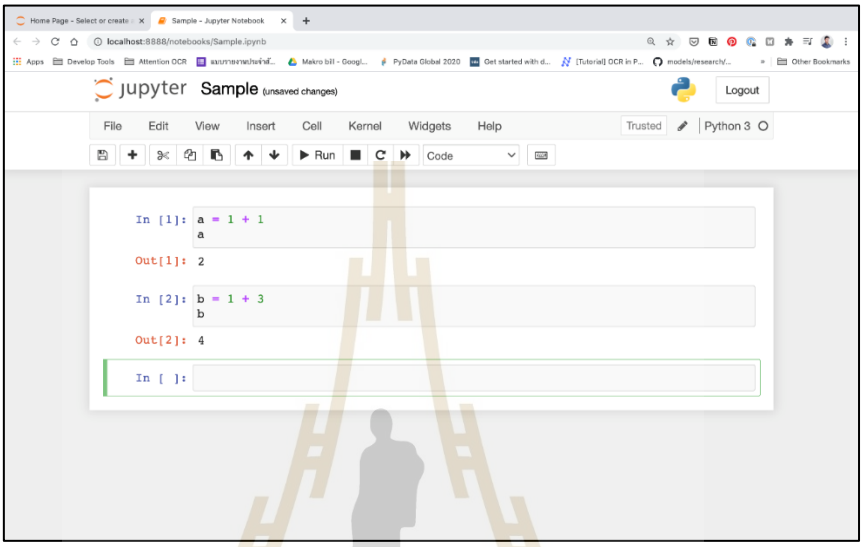
Jupyter ปล่อยให้ใช้งานในรูปแบบ Open source โดยเริ่มต้นพัฒนาขึ้นมาโดยใช้ชื่อโปรเจคว่า IPython โดยทำการแยกโปรเจค ออกโดยใช้ชื่อว่า Project Jupyter โดยในการพัฒนานั้นได้ทำการนำ IPython มาใช้งานเป็นแก่น (Kernel) ของตัว Jupyter



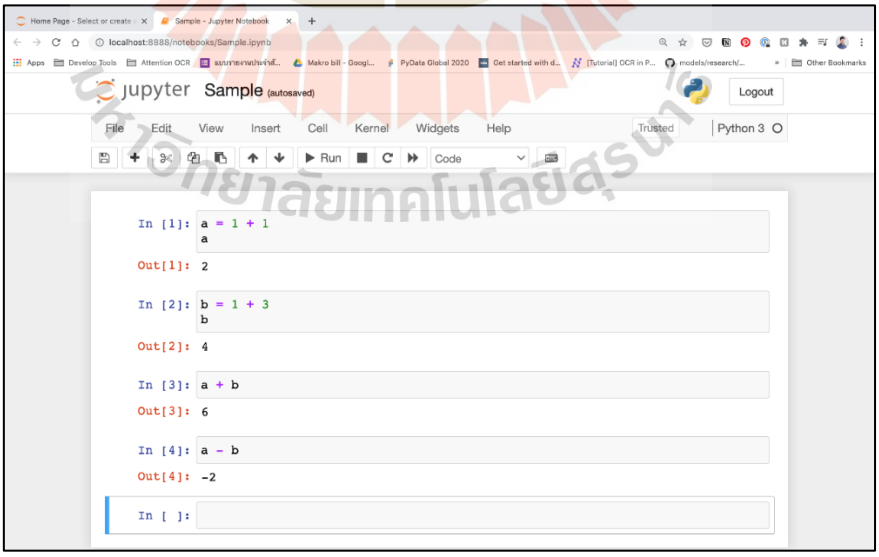
รูปที่ ก.3 ตัวอย่างโปรแกรม Jupyter Notebook

จากรูปที่ ก.3 Jupyter นั้นถูกสร้างขึ้นมาในรูปแบบของ Web Application ซึ่งสามารถใช้งานผ่าน Browser ได้ทันที โดย Jupyter มีเครื่องมือในการใช้งานที่หลากหลายได้แก่ Jupyter Notebook, JupyterHub และ JupyterLab เป็นต้น และยังสามารถรองรับภาษาที่สามารถใช้งานใน Jupyter ได้แก่ Python, R และ Julia เป็นต้น โดย การป้อนคำสั่งและทำการรันเพื่อแสดงผล ดังรูปที่ ก.4

ผลการรันที่ได้จากแต่ละเซลล์ (Cell) นั้นสามารถนำไปใช้ในเซลล์ถัดไปได้ เพราะ Jupyter นั้นสามารถเก็บค่าตัวแปรที่ได้จากกการรันในแต่ละเซลล์ไว้ที่แก่นของระบบ ทำให้เซลล์อื่น ๆ สามารถดึงข้อมูลเหล่านั้นไปใช้ต่อได้ทันที ไม่ต้องสร้างตัวแปรใหม่ขึ้นมา ดังรูปที่ ก.5



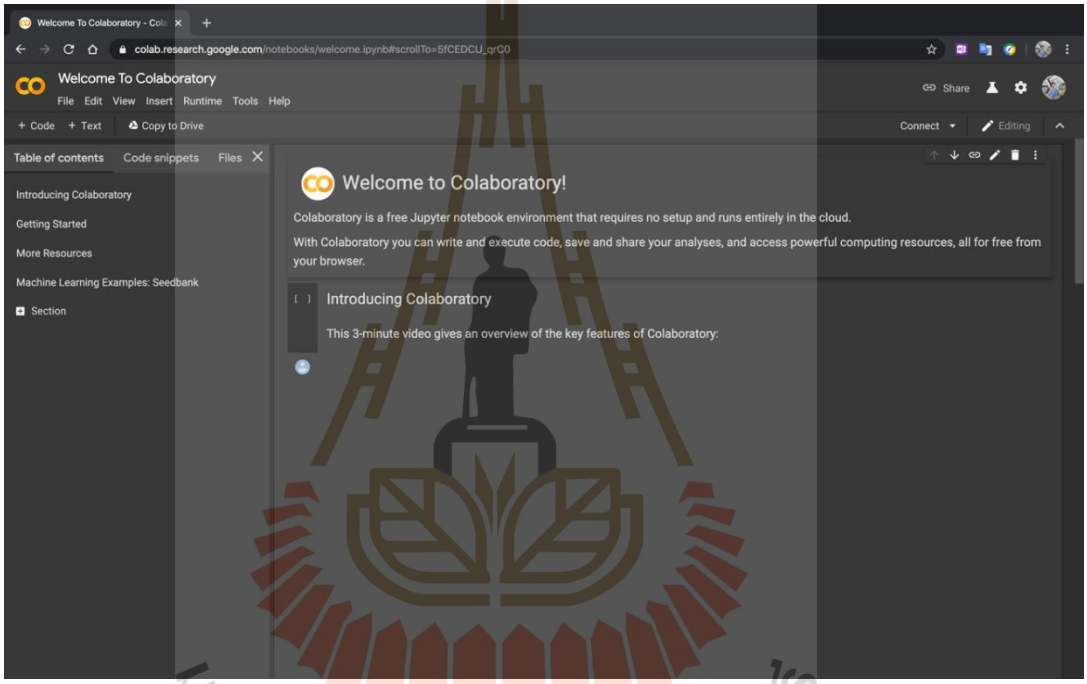
รูปที่ ก.4 ตัวอย่างผลการรันโปรแกรมโดยใช้ Jupyter Notebook



รูปที่ ก.5 ตัวอย่างการรันโปรแกรมโดยดึงค่าตัวแปรจากเซลล์ที่ผ่านการรันมาแล้วไปใช้งาน

3. Google Colaboratory

Google Colaboratory (Bisong, 2019) เป็นการนำ Jupyter Notebook มาประยุกต์ใช้งานบนระบบ Cloud Computer ของ Google ซึ่งมีจุดมุ่งหมายในการนำไปใช้ในการศึกษาการเรียนรู้ของเครื่อง และนำไปใช้ในการวิจัย สำหรับบุคคลที่ไม่มีคอมพิวเตอร์ที่มีสเปกเครื่องที่สูงในการใช้งาน Google Colaboratory สามารถใช้งานได้ฟรี ซึ่งสามารถใช้งานโดยใช้ภาษา Python ในการใช้งานเป็นหลัก โดยมีตัวอย่างหน้าตาการใช้งานดังแสดงในรูปที่ ก.6



รูปที่ ก.6 ตัวอย่างโปรแกรม Google Colaboratory

4. Scikit-Learn

Scikit-Learn (Pedregosa et al., 2011) นั้นเป็นเครื่องมือที่ใช้สำหรับการสร้างและพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning)โดย Scikit-Learn นั้นได้มีเครื่องมีต่าง ๆ สำหรับการใช้งาน โดยมีเครื่องมือสำหรับการจัดเตรียมข้อมูล อัลกอริทึมสำหรับสร้างและพัฒนาแบบจำลองการเรียนรู้ของเครื่อง และอัลกอริทึมที่ใช้สำหรับการวัดประสิทธิภาพแบบจำลองที่ได้ทำการสร้างขึ้น โดย Scikit-Learn นั้นถูกพัฒนาขึ้นมาบนภาษา Python และยังเป็น Open Source ให้ใช้งานได้ฟรี

สำหรับการใช้งาน Scikit-Learn นั้น ทางผู้วิจัยได้มีตัวอย่างการนำ Scikit-Learn ไปใช้งานในส่วนของการจัดเตรียมข้อมูล ด้วยอัลกอริทึม TF-IDF ดังแสดงในรูปที่ ก.7

```
73
74 def create_tfidf(data, min_df=5, max_features=5000, ngram_range=(1,3)):
75     tf_idf = TfidfVectorizer(min_df=min_df,
76                              max_features=max_features,
77                              ngram_range=ngram_range)
78     data = tf_idf.fit_transform(data)
79     data = data.toarray()
80     return data, tf_idf
81
```

รูปที่ ก.7 ตัวอย่างการใช้งานคุณลักษณะ TF-IDF จาก Library Scikit-Learn

5. Gensim

Gensim เป็น Library บน Python (Řehůřek and Sojka, 2011) ที่ถูกพัฒนาเพื่อการใช้งานเกี่ยวกับภาษาธรรมชาติ หรือภาษาของมนุษย์ที่ใช้ในการสื่อสาร ให้คอมพิวเตอร์สามารถทำความเข้าใจได้เช่นเดียวกันกับมนุษย์ โดย Gensim นั้นจะเน้นการใช้งานไปในส่วนของ Word Embedding ซึ่งได้มีการสร้างพัฒนาชุดคำสั่งการใช้งาน Word2Vec ให้สามารถใช้งานได้อย่างเรียบง่าย จากนั้น Gensim ได้มีการสร้างชุดคำสั่ง Doc2Vec ซึ่งเป็นชุดคำสั่งที่ได้ทำการนำ Word2Vec มาปรับปรุงเพื่อให้ใช้งานกับเอกสารได้ดียิ่งขึ้น โดยมีตัวอย่างการใช้งานดังรูปที่ ก.8

```
1 # Tag document with word
2 def tag_document(x, y):
3     sample_document = [TaggedDocument(words=word, tags=[tag]) for
4                        word, tag in zip(x, y)]
5     return sample_document
6
7 # Create Doc2Vec model
8 model_d2v = Doc2Vec(train_tagged, vector_size=NUM_WORDS, window=5,
9                      min_count=5, workers=4)
10
11 # Embedding document and create feature
12 def document_embedding(document):
13     results = np.array([model_d2v.infer_vector(record) for record in document])
14     return results
```

รูปที่ ก.8 ตัวอย่างชุดคำสั่งสำหรับการสร้างคุณลักษณะ Doc2Vec

6. Keras

Keras (Chollet, 2015) นั้นเป็น Library ถูกพัฒนาบนภาษา Python สำหรับการพัฒนาการเรียนรู้เชิงลึก (Deep Learning)โดย Keras เป็น Library ที่ทำงานบน Library ของการเรียนรู้เชิงลึกตัวอื่น ๆ เช่น Tensorflow, Theano และ Microsoft Cognitive Toolkit เป็นต้น โดย Keras ทำการสร้างขึ้นเพื่อทำให้การพัฒนา Deep Learning เป็นไปอย่างง่ายที่สุด โดยที่ไม่จำเป็นต้องใช้งาน Library ตัวอื่น ๆ ที่มีความซับซ้อนในการใช้งาน ดังแสดงการใช้งาน ดังตัวอย่างในรูปที่ ก.8

```
def dnn():
    model = Sequential()
    model.add(Dense(500, activation='sigmoid'))
    model.add(Dense(500, activation='sigmoid'))
    model.add(Dense(500, activation='sigmoid'))
    model.add(Dropout(0.1))
    model.add(Dense(1, activation='sigmoid'))
    optimizer = Adam(lr=0.001)
    model.compile(loss='binary_crossentropy',
                  optimizer=optimizer,
                  metrics=[tf.keras.metrics.Precision(),
                           tf.keras.metrics.Recall(),
                           'accuracy'])
    return model
```

รูปที่ ก.9 ตัวอย่างการสร้างแบบจำลองโดยใช้ Keras

โดยในปี ค.ศ. 2017 นั้น Google ได้ทำการประกาศให้การสนับสนุนให้ Keras นั้นเป็นแกนหลักในการใช้งาน Tensorflow ที่เป็น Library การเรียนรู้เชิงลึกของ Google ซึ่งจะช่วยเพิ่มความง่ายในการสร้างแบบจำลองของ Tensorflow

ภาคผนวก ข

บทความวิจัยตีพิมพ์

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

Pumrapee Poomka, Kittisak Kerdprasop and Nittaya Kerdprasop. (2020). Deep learning techniques for sentiment analysis from product users. In **Proceeding of SUT International Virtual Conference on Science and Technology**, (pp. 149-155).

SUT International Virtual Conference on Science and Technology
Nakhon-Ratchasima, Thailand
28th August 2020

EAT0026

Deep Leaning Techniques for Sentiment Analysis from Product Users

Pumrapee Poomka*, Kittisak Kerdprasop and Nittaya Kerdprasop

School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand.

* Corresponding Author: pumrapee.p@outlook.com

Abstract. Nowadays, online business such as online retailer is very popular. Customers can order products from the internet and waiting for delivery to their place. With the online platform that customers cannot see what the real products look like, they thus rely on the product review posted by other customers and decide whether to buy or not. The problem of decision making process is that comments can be positive or negative based on their context not on the given rating. Hence, we propose to develop sentiment analysis model to classify the comment message as either positive or negative. We use product review on amazon.com as the study case. We select reviews of 6 product categories including book, dvd, electronics, kitchen, music, and video. We develop multiple models based on various techniques of feature selection and deep learning. Feature extraction for the text includes Term Frequency - Invert Document Frequency (TF-IDF), and Doc2Vec, whereas deep learning algorithms are Deep Neural Network (DNN) and Convolutional Neural Network (CNN). The models are tested on each category of products. Finally, we compare the results and found that DNN model with TF-IDF feature extraction outperform all other models in terms of accuracy, precision, recall, and time to train the model.

Keywords: Sentiment Analysis, Text Classification, Deep Learning.

1. Introduction

Presently, the internet-based business is very popular among customers because it is very comfortable for the customers. The rising popularity drives much change in many businesses. For example, from the retails store, Amazon creates the frontend shop in an online platform. The social media platform such as Facebook, Twitter, or Instagram has been widely adopted to sell the online products. This kind of platform provides the channel to the retail stores for selling their product directly to their customers [1]. But the major problem of online shopping is that the customers do not see how the real product looks like. Most customers search on the internet for the reviews of the product they are interested. The opinions of other customers can help them making decision to buy or not buy the product. The review comment can be either positive or negative. Customers will confuse with some product that has top comments unclear of being positive or negative. Moreover, the score that is given to comments may conflict with

the context in the comment. Such conflict can cause difficulty on making correct decision for new customers. This problem can be solved with sentiment analysis.

Sentiment analysis is the subject for analyzing data to get insight into the emotions of the writer. The results from sentiment analysis can help the retail stores for evaluating their product satisfying among customers. Moreover, sentiment analysis can help the customer to convey other opinions on the product they are interested in. Mostly sentiment analysis is based on textual data and handle by humans to classify the text as either positive or negative [2]. This process is however taken too much time. Nowadays, machine learning is an algorithm that can mimic how humans think and can act like a human. Machine learning is used in many businesses and its performance improves exponentially. One major improvement is deep learning. The success of deep learning is because of the performance of the Graphics Processing Unit (GPU) that adopted by most deep learning algorithms [3]. Before applying deep learning to analyze reviews, we need to do feature extraction from textual data because the computer cannot understand text in natural language as humans do. This process transforms the text into the vector with numerical representation. Previous work on sentiment analysis by many researchers provide many use cases to work on textual feature extraction such as applying Doc2Vec feature extraction on product description and train classify model [4], use TF-IDF with CNN to detect cyberbully text from social media [5], and use TF-IDF with support vector machine algorithm to classify movie review [6].

In this research, we aim to develop sentiment analysis models to classify product opinions. We choose several feature extraction algorithms and deep learning algorithms to build the models. We compare the performance of the models in terms of accuracy, precision, recall, and time to train the model.

2. Literature Review

For the last few year, Sentiment Analysis is very popular method to use for analyzing the text dataset. This method uses on many fields of work but mostly on online business that have people to discuss on such as Social Network, Online forum, and Online Retailer. Nowadays, Machine learning and deep learning can make the Sentiment Analysis very

Before we use text data to analyze and develop model, we need to transform text dataset with feature extraction methods to make computer understandable like human. In traditional, Bag of Word is very popular method to use for feature extraction, but it cannot find how importance of rarely word that impact on dataset. Hence, TF-IDF is create with the improve from Bag of Word by having the statistical that show how importance of rarely word that impact on dataset [5].

With the emerging technology on deep learning, Word Embedding is the new method of feature extraction that extract word to feature vector that can calculate the similarity of each word. However, Word Embedding can only extract only word and cannot deal with long document well. Hence, Doc2Vec is the method that can create vector feature from long document. Doc2Vec can take the advantage of deep learning and outperform on text classification compared with Word Embedding [4].

3. Research Framework

In this work, we design research framework as shown in Figure 1. The objective is to compare the results obtained from different strategies on modeling building based on various techniques of feature extraction and deep learning for sentiment analysis. The same framework has been applied for each category of the dataset. We preprocess data before doing feature extraction by transforming all characters to be lowercase and removing punctuation. Then, we apply datasets into the feature extraction and model development.

Firstly, we perform feature extraction using Term Frequency - Invert Document Frequency (TF-IDF) and Doc2Vec methods. After that, we develop model based on Deep Neural Network (DNN) and Convolutional Neural Network (CNN). Finally, we compare the result of all models.

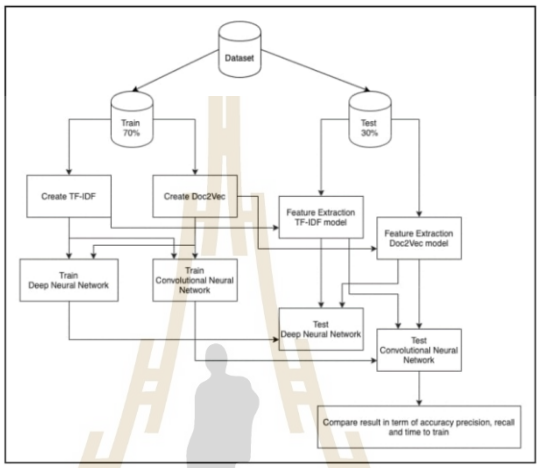


Figure 1. Research framework

4. Dataset

We use the study case of product review from amazon.com collected by Blitzer et al [7] to develop models for sentiment analysis. We select 6 categories of this dataset including book, dvd, electronics, kitchen, music, and video. The data examples are shown in Table 1.

Data in each category contain 1000 records of positive opinions and 1000 records of negative ones. We split the dataset in each category into 70% to be the training set and the rest 30% to be the test as summarized in Table 2 and 3.

Table 1. Example of some review records in dataset.

Rating	Text
Positive	This is our favorite baby book for reading with our baby/young child. Each of our children, from ages 9 months to three years, has loved this book. We buy this as a gift for all new babies
Negative	The stationery is cute and colorful, but the pages are very cluttered. While the envelopes have spaces to write addresses and put return labels, the pages are too colorful and bold. A ballpoint or ink pen doesn't look good on the paper.

Table 2. Quantity of data records in book, dvd, and electronics categories.

	book			**dvd**			**electronics**		
	Positive	**Negative**	**Total**	**Positive**	**Negative**	**Total**	**Positive**	**Negative**	**Total**
Training set	700	700	**1400**	700	700	**1400**	700	700	**1400**
Test set	300	300	**600**	300	300	**600**	300	300	**600**
Total	**1000**	**1000**	**2000**	**1000**	**1000**	**2000**	**1000**	**1000**	**2000**

Table 3. Quantity of data records in kitchen, music, and video categories.

	kitchen			music			video		
	Positive	**Negative**	**Total**	**Positive**	**Negative**	**Total**	**Positive**	**Negative**	**Total**
Training set	700	700	**1400**	700	700	**1400**	700	700	**1400**
Test set	300	300	**600**	300	300	**600**	300	300	**600**
Total	**1000**	**1000**	**2000**	**1000**	**1000**	**2000**	**1000**	**1000**	**2000**

5. Feature Extraction and Deep Learning Methods

4.1. Term Frequency – Invert Document Frequency

Term Frequency - Invert Document Frequency (TF-IDF) is the algorithm for counting and computing importance of words in the document compared to the entire dataset. TF-IDF can solve the problem of Bag of Word (BoW) algorithm in terms of the importance of the word in the document by using the statistical method for calculating the importance called Term Frequency (TF) and Invert Document Frequency (IDF) [8].

TF is the summary of the words from each document calculated from the frequency of word or phase in token representation for each document, as shown in Equation (1). IDF calculates the importance of the word from the document compared with the entire dataset, as shown in Equation (2).

$$TF(t,d) = \log(1+freq(t,d)) \quad (1)$$

When $TF(t,d)$ is the Term Frequency function,
t is each term of word or phrase in document,
d is each document,
$freq(t,d)$ is the frequency of term of word or phrase t in document d.

$$IDF(t,D) = \ln\left(\frac{|D|}{|\{d \in D \mid t \in d\}|}\right) \quad (1)$$

When $IDF(t,D)$ is the Invert Document Frequency function,
t is each term of word or phrase in document,
d is each document,
D is the entire dataset.

After we calculate TF and IDF, we can compute TF-IDF that represents the vector feature by the product of 2 functions, as shown in equation (3). This vector feature can represent the word that is rarely found but important in the document datasets.

$$TFIDF(t,d,D) = TF(t,d) \times IDF(t,d) \quad (2)$$

When $TFIDF(t,d,D)$ is the TF-IDF function,
$TF(t,d)$ is the value from Term Document Frequency function,
$IDF(t,d)$ is the value from Invert Document Frequency function,
t is each term of word or phrase in document,
d is each document,
D is the entire dataset.

4.2. Doc2Vec

Doc2Vec is part of the word embedding technique. It is an algorithm that uses Continuous Bag of Words and Skip-Gram model to calculate the vector that represents the documents. Doc2Vec is developed based on Word2Vec. Unlike the Word2Vec, Doc2Vec does not come

with the logic or similarity in word. Doc2Vec model that we use in this research is called Distributed Memory version of Paragraph Vector (PV-DM) that uses for creating features for classifying document [9].

4.3. Deep Neural Network

Deep Neural Network (DNN) is the simplest form of deep learning. DNN is developed from neural networks by using a stack of layers of computing nodes that make network deep and increase the performance to analyze feature in dataset (Figure 3a [10]).

4.4. Convolutional Neural Network

Convolutional Neural Network (CNN) is the algorithm that mimics behavior of human when looking at object. CNN uses the same behavior by creating the filter that looks into the data. Filter moves all over the dataset and construct the new feature by combining filtering results and selecting only the important information in the dataset. Mostly, CNN is used in the computer vision field but it can also be used for text classification with 1 dimension CNN as shown in Figure 3b [11].

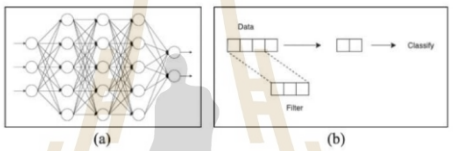


Figure 3. The example network in Deep Neural Network (a) and the process of 1 dimension Convolutional Neural Network (b).

6. Experiments

In our experiment, we firstly compare the results obtained from different feature extraction techniques: TF-IDF and Doc2Vec. Then, we develop predictive models using DNN and CNN methods. Performances are compared in terms of accuracy, precision, recall, and time to train.

For TF-IDF, we fix the vocabulary size to be 5000 words with N-gram in range 1-3. We use data from the training set to create the TF-IDF feature extraction result. For Doc2Vec, we fix the vector feature-length to be 5000. Then, we use data from the training set to create the Doc2Vec feature extraction result.

In the model development process, we apply the DNN and CNN algorithms with parameters as shown in Figure 4. We design this structure for sample experiments to develop model from DNN and CNN. We design DNN with 3 layers of internal feature extraction with 500 nodes for each layer then we use dropout rate 0.1 to avoid overfitting. We design first CNN layer with 32 of filter size and 2 of kernel size and second layer with 64 of filter size and 3 of kernel size. Both layers use same padding method and rectifier activation function then use dropout rate 0.1 to avoid overfitting and flatten to make liner feature for classification. We design to add 3 layers for feature learning in DNN and CNN model because we get over parameter and overfitting problem after we add more layers on each model. We use Google Colab as the main environment as it is free for using Nvidia Tesla T4. We use Python language to run on Google Colab, NLTK library for preprocessing text data, Scikit-learn for using TF-IDF function, Gensim for Doc2Vec, and Keras for deep learning models.

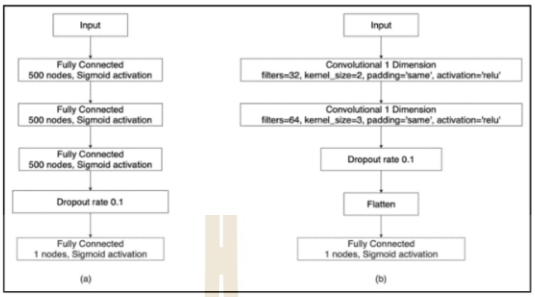


Figure 4. Structure and parameter of DNN (a) and CNN (b) model building.

7. Results and Discussion

Our main focus is to observe the performances of two feature extraction techniques (TF-IDF and Doc2Vec) and two deep learning methods (DNN and CNN). Therefore, the four comparative scenarios are TF-IDF with DNN, TF-IDF with CNN, Doc2Vec with DNN, and Doc2Vec with CNN. The comparison has been done on six product categories (book, dvd, electronics, kitchen, music, and video). The predictive performances on book, dvd, electronics datasets are displayed in Table 4, whereas the performances on the rest of datasets are shown in Table 5.

From the results, it can be noticed that DNN model from TF-IDF feature extraction outperforms other model building methods. But on some categories, such as dvd and kitchen, the CNN model with Doc2Vec feature extraction has slightly higher recall value. This means the CNN model with Doc2Vec feature extraction can classify the positive class more correctly than TF-IDF. The key point that make models from TF-IDF feature more accurately than Doc2Vec because TF-IDF can analyze, select frequencies words and let rarely word show importance of its word than Doc2Vec that feature is based on whole document.

Table 4. Results from book, dvd, electronics datasets.

Category		book			dvd			electronics		
Feature	Model	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
TF-IDF	DNN	**0.82**	**0.84**	**0.79**	**0.83**	0.83	0.82	**0.83**	**0.83**	**0.83**
	CNN	0.81	0.82	0.78	0.82	0.83	0.81	0.82	0.82	0.82
Doc2Vec	DNN	0.74	0.77	0.69	0.71	**0.84**	0.52	0.71	0.79	0.58
	CNN	0.72	0.79	0.61	0.77	0.74	**0.85**	0.7	0.7	0.71

Table 5. Results from kitchen, music, video datasets.

Category		kitchen			music			video		
Feature	Model	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
TF-IDF	DNN	**0.85**	**0.85**	0.85	**0.8**	**0.82**	0.77	**0.84**	**0.84**	**0.83**
	CNN	0.83	0.82	0.85	0.8	0.81	**0.80**	0.83	0.83	0.83
Doc2Vec	DNN	0.69	0.66	0.79	0.7	0.79	0.54	0.76	0.75	0.76
	CNN	0.72	0.67	**0.86**	0.73	0.75	0.69	0.76	0.76	0.76

Table 6. Time to train model (in second) for each category of datasets

Feature	Model	book	dvd	electronics	kitchen	music	video
TF-IDF	DNN	*5.19 s*	*5.08 s*	*5.14 s*	*5.05 s*	*5.10 s*	*5.07 s*
	CNN	18.90 s	18.70 s	18.70 s	18.60 s	18.60 s	18.70 s
Doc2Vec	DNN	9.50 s	11.20 s	8.55 s	8.63 s	8.73 s	5.90 s
	CNN	16.10 s	17.50 s	14.50 s	14.50 s	14.80 s	7.98 s

When considering time to train the models as shown in Table 6, we can see that the time to train DNN model combined with the TF-IDF feature extraction takes less time the other combination methods. Models from Doc2Vec feature extraction take more time to train in both DNN and CNN. In terms of the learning algorithm, DNN has lower complexity of the model structure than CNN. Therefore, it can be trained faster than the CNN model.

8. Conclusion

This research aims to develop a sentiment analysis model to classify product review comments on Amazon.com as wither positive or negative. Two feature extraction techniques (TF-IDF and Doc2Vec) with two deep learning algorithms (DNN and CNN) are studied. We experiment with 6 categories of the dataset including book, dvd, electronics, kitchen, music, and video. Then, we compare the results in terms of accuracy, precision, recall, and time to train the model.

Form the experimentation results, we found that TF-IDF outperforms the Doc2Vec when considering model accuracy of the overall datasets. TF-IDF feature extraction technique can be trained with DNN and CNN faster than the Doc2Vec because TF-IDF is based on statistical computation having lower complexity than Doc2Vec. TF-IDF with DNN outperforms other methods in terms of accuracy, precision, recall, and the fastest training time.

References

[1] Kowalczyk R, Ulieru M, and Unland R 2002 *Proc. Int. Conf. on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World* (Berlin: Springer) p. 295-313.

[2] TäuscherK and Laudien S M 2018 J. *European Management Journal* 36(3) pp 319-329

[3] Zhang H, Zheng Z, Xu S, Dai W, Ho Q, Liang X, Hu Z, Wei J, Xie P and Xing E P 2017 *Proc. Int. Conf. on USENIX Annual Technical Conference USENIXATC 17* pp 181-193.

[4] Lee H and Yoon Y 2018 Engineering doc2vec for automatic classification of product descriptions on O2O applications J. *Electronic Commerce Research* 18(3) pp 433-456.

[5] Chen J, Yan S, and Wong K C 2018 J. *Neural Computing and Applications* pp 1-10.

[6] Zin H M, Mustapha N, Murad M A A, and Sharef N M 2018 J. *Advanced Science Letters* 24(2) pp 933-937.

[7] Blitzer J, Dredze M, and Pereira F 2007 *Proc. Int. Conf. on the 45th Annual Meeting of the Association of Computational Linguistics* pp 440-447.

[8] Ramos J 2003 *Proc. Int. Conf. on the first instructional conference on Machine Learning* vol 242 pp 133-142.

[9] Le Q and Mikolov T 2014 *Proc. Int. Conf. on Machine Learning* pp 1188-1196.

[10] LeCun Y, Bengio Y, and Hinton G 2015 Deep learning *Nature* 521(7553) pp 436-444.

[11] Kalchbrenner N, Grefenstette E, and Blunsom P 2014 *arXiv preprint arXiv:1404.2188*.

ประวัติผู้เขียน

นายภูมิรพี ภูมิค้า เกิดเมื่อวันที่ 30 กันยายน พ.ศ. 2538 ที่อำเภอเมือง จังหวัดนครราชสีมา เริ่มเข้าศึกษาชั้นประถมศึกษาปีที่ 1 ที่โรงเรียนเหรียญทองวิทยา อำเภอครบุรี จังหวัดนครราชสีมา จนจบการศึกษาระดับประถมศึกษาชั้นปีที่ 6 จากนั้นศึกษาต่อในระดับมัธยมศึกษาตอนต้นที่โรงเรียนครบุรี อำเภอครบุรี จังหวัดนครราชสีมา และศึกษาในระดับมัธยมศึกษาตอนปลายที่โรงเรียนเตรียมอุดมศึกษาน้อมเกล้านครราชสีมา อำเภอเมือง จังหวัดนครราชสีมา ในปีการศึกษา 2557 ได้เข้าศึกษาระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี อำเภอเมือง จังหวัดนครราชสีมา โดยได้รับการคัดเลือกเข้ารับทุนการศึกษาเฉลิมพระเกียรติ 84 พรรษา ตลอดทั้งหลักสูตร และได้สำเร็จการศึกษาเมื่อปี 2561

ปี พ.ศ. 2561 ได้เข้ารับการศึกษาในระดับปริญญาโท สาขาวิชาวิศวกรรมโทรคมนาคมและคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี โดยทำการวิจัยในด้านของการนำภาษาของมนุษย์มาวิเคราะห์บนระบบคอมพิวเตอร์ด้วยเทคนิคการเรียนรู้เชิงลึก โดยในระหว่างการศึกษาได้รับการอนุเคราะห์เป็นอย่างดีจากอาจารย์ที่ปรึกษาและอาจารย์ประจำรายวิชาต่าง ๆ และได้ทำการตีพิมพ์บทความวิชาการซึ่งมีรายละเอียดสามารถดูได้ที่ภาคผนวก ข