การวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญา

ร.ท.วีรยุทธ ครั่งกลาง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AN ANALYSIS OF NATURAL LANGUAGE TEXT RELATING TO THAI CRIMINAL LAW

Flg.Off. Weerayut Krungklang

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญา
โดย	ร.ท.วีรยุทธ ครั่งกลาง
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

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

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

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

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

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วีรยุทธ ครั่งกลาง : การวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญา. (AN ANALYSIS OF NATURAL LANGUAGE TEXT RELATING TO THAI CRIMINAL LAW) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุกรี สินธุภิญโญ

วิทยานิพนธ์นี้วิเคราะห์การบังคับใช้กฎหมายอาญาของประเทศไทย ในภาค1 บทบัญญัติทั่วไป และภาค2 เฉพาะความผิดเกี่ยวกับชีวิต มาตรา 288 และมาตรา 289 ในลักษณะ10 ความผิดเกี่ยวกับชีวิตและร่างกาย ตามประมวลกฎหมายอาญาของไทย

ส่วนแรกของวิทยานิพนธ์นี้ใช้ความรู้ด้านกฎหมายอาญาและคำพิพากษาของศาลฎีกาในการสร้างกฎในการพิจารณาที่มนุษย์สามารถเข้าใจได้ และส่วนที่สองคือการฝึกฝนแบบจำลองด้วยชุดข้อมูลจากคำพิพากษาด้วยเทคนิคการเรียนรู้เชิงลึก โดยแก้ปัญหาความไม่สมดุลของกลุ่มข้อมูลฝึกสอนด้วยการสังเคราะห์ตัวอย่างข้อมูลในกลุ่มอื่น ๆ ให้มีจำนวนเท่ากับกลุ่มที่มากที่สุด และฝึกสอนด้วยโครงข่ายหน่วยความจำระยะสั้นแบบยาวทิศทางเดียวและสองทิศทาง ซึ่งเป็นโครงข่ายประสาทเทียมแบบวกกลับประเภทหนึ่ง และเมื่อวัดประสิทธิภาพแบบจำลองด้วยค่าเฉลี่ยมหภาคเอฟวัน พบว่าแบบจำลองของหน่วยความจำระยะสั้นแบบยาวสองทิศทางให้ประสิทธิภาพสูงกว่าแบบทิศทางเดียว และการใช้ค่าถ่วงน้ำหนักเริ่มต้นจากเรียนรู้ด้วยคลังข้อมูลขนาดใหญ่อื่น ให้ประสิทธิภาพที่สูงกว่าการใช้เฉพาะข้อมูลฝึกสอน และท้ายสุดทำการทดสอบความแม่นยำของแบบจำลองจากข่าวอาชญากรรมด้วยเทคนิคการหาค่าเฉลี่ยความน่าจะเป็น เพื่อใช้เป็นข้อมูลขาเข้าของกฎการพิจารณา พบว่าสอดคล้องกับความเห็นของนักกฎหมาย 59 %

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2563 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070971521 : MAJOR SOFTWARE ENGINEERING
KEYWORD:

Weerayut Krungklang : AN ANALYSIS OF NATURAL LANGUAGE TEXT RELATING TO THAI CRIMINAL LAW. Advisor: Asst. Prof. SUKREE SINTHUPINYO, Ph.D.

This work analyses Thailand’s criminal law enforcement in book 1, general provisions; book 2, specific offences causing death specified in ; sections 288 and 289 of title 10, offenses affecting life and body under the Thai Criminal Code.

The first part of this work is using criminal law domain knowledge and supreme court judgment results, to be the initial domain information and result is the rules that humans can understand. The second part of this work is bringing training data set from the final judgment to train with deep learning methods. Due to the training set which have severe imbalances, the Synthetic Minority Over-Sampling TEchnique (SMOTE) is used to solve this problem. Models are trained on the training set using unidirectional Long Short-Term Memory (LSTM) networks and bidirectional Long Short-Term Memory (BiLSTM) are type of Recurrent Neural Networks (RNN). BiLSTM macro-average F1 scores are higher than LSTM. Pre-trained word embeddings are then used to make the macro-average F1 scores higher than before. Finally, using models to predict online crime news, the highest average probability of each model is selected by using Soft Voting as input to the rules. The test results compared with the predictions of our methods with the opinion of the lawyer, corresponding 59%.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study:	Software Engineering	Student's Signature .....
Academic Year:	2020	Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดีด้วยความช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร. สุกรี สินธุภิญโญ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความกรุณา ช่วยเหลือ เสียสละ เวลาให้คำแนะนำ คำปรึกษา เกี่ยวกับการทำวิจัยด้วยความเอาใจใส่อย่างดียิ่ง และความเชื่อมั่นที่อาจารย์มีให้ผู้วิจัย ทำให้ผู้วิจัยสามารถดำเนินงานวิจัยสำเร็จได้ด้วยดี

ขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นัททที นิภานันท์ ประธานกรรมการสอบ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ ผู้ช่วยศาสตราจารย์ ดร.ณัฐพงศ์ ชินธเนศ และผู้ช่วยศาสตราจารย์ ดร.เด่นดวง ประดับสุวรรณ คณะกรรมการสอบ ที่กรุณาสละเวลาช่วยชี้แนะและให้คำแนะนำ ทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้ความรู้ซึ่งจำเป็นและเป็นประโยชน์ในการทำวิจัย รวมถึงบุคลากรทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้ข้อมูล คำแนะนำ และความช่วยเหลือในระหว่างที่ผู้วิจัยศึกษาและสอบวิทยานิพนธ์จนสำเร็จด้วยดี

ขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ในภาควิชาวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และในแผนกวิชาเทคโนโลยีสารสนเทศ กองฝึกอบรม กรมการสื่อสารทหาร ที่คอยที่ให้คำแนะนำ ช่วยเหลือและอำนวยความสะดวกในการทำวิจัยตลอดมา และที่สำคัญ นาวาอากาศโทหญิง อินทิรา ทองขาว หัวหน้าแผนกวิชาเทคโนโลยีสารสนเทศ กองฝึกอบรม ที่คอยให้การสนับสนุนและเป็นกำลังใจในการทำวิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ขอขอบคุณ ครอบครัวที่สนับสนุนข้าพเจ้าในทุก ๆ ด้าน และเป็นกำลังใจสำคัญในการทำวิจัย ให้สามารถผ่านพ้นมาได้

วีรยุทธ ครั่งกลาง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

..... ค

บทคัดย่อภาษาไทย..... ค

..... ง

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ..... ฉ

สารบัญตาราง..... ญ

สารบัญรูป ..... ฏ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 ขั้นตอนและวิธีการดำเนินการวิจัย..... 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 บทความวิชาการที่ได้รับการตีพิมพ์ 3

บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง 5

2.1 บทบัญญัติประมวลกฎหมายอาญา [1]..... 5

2.2 โครงสร้างความรับผิดทางอาญา [2] 6

2.3 การใช้และการตีความกฎหมายอาญา [3]..... 14

2.4 ผู้มีส่วนเกี่ยวข้องกับการกระทำผิดและการพิจารณาอัตราส่วนโทษ..... 16

2.5 เหตุลดโทษและเหตุรอการลงโทษ [4] 18

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.6 ศาลฎีกาและคำพิพากษาศาลฎีกา [5] 20
2.7 แบบจำลองโครงข่ายประสาทเทียม (Artificial Neural Networks : ANNs) 22
2.8 Backpropagation [8] 28
2.9 การเรียนรู้เชิงลึก (Deep Learning) [18] 43
2.10 ตัวแทนของข้อความ (Text Representation) [19] 45
2.11 อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimization Algorithms) [20] 51
2.12 โครงข่ายประสาทเทียมแบบวกกลับ (Recurrence Neural Networks : RNN) [18] 60
2.13 หน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory : LSTM) [15] 62
2.14 หน่วยความจำระยะสั้นแบบยาวสองทิศทาง (Bidirectional LSTM) [23] 65
2.15 การประมวลผลข้อความ 66
2.16 การวัดประสิทธิภาพ 68
2.17 เทคนิคการสุ่มเพิ่มตัวอย่างข้างน้อยสังเคราะห์ (Synthetic Minority Over-sampling Technique : SMOTE) [31] 70
2.18 การถ่ายโอนการเรียนรู้ (Transfer Learning) [32] 72
2.19 การหยุดเรียนรู้กลางคัน (Dropout) [38] 76
2.20 ต้นไม้ตัดสินใจ (Decision Tree) [39] 77
2.21 นาอีฟเบย์ส (Naïve Bayes) [41] 79
2.22 Soft-Voting [42] 80
บทที่ 3 งานวิจัยที่เกี่ยวข้อง 81
3.1 งานวิจัย “Research on paten text classification based on Word2Vec and LSTM” .. 81
3.2 งานวิจัย “A Scheme of Criminal Law Knowledge Acquisition Using Ontology” .. 81
3.3 งานวิจัย “A Framework of Multi-Stage Classifier for Identifying Criminal Law Sentences” 82
3.4 งานวิจัย “Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism” 83

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4 แนวคิดและวิธีการดำเนินการ 85
4.1 การเตรียมข้อมูลสำหรับงานวิจัย 86
4.2 การทำความสะอาดข้อมูล (Data Cleansing) 94
4.3 การประมวลผลข้อมูลเบื้องต้น 95
4.4 การคัดเลือกแบบจำลองในงานวิจัย 97
4.5 การออกแบบโครงสร้างและฝึกฝนแบบจำลอง 98
4.6 การบันทึกแบบจำลองและค่าถ่วงนำหนักเริ่มต้น 104
4.7 การประเมินประสิทธิภาพของแบบจำลอง 105
บทที่ 5 การทดลองและผลการทดลอง 107
5.1 สมมติฐานการทดลอง 107
5.2 การออกแบบการทดลอง 108
5.3 การคัดเลือกแบบจำลอง 116
5.4 การคัดเลือกคุณลักษณะสำคัญ 117
5.5 การออกแบบและสร้างกฎการตัดสินใจ 119
5.6 การออกแบบและฝึกฝนแบบจำลองการจำแนกข้อความภาษาธรรมชาติ 122
5.7 การเปรียบเทียบประสิทธิภาพแบบจำลอง LSTM ทิศทางเดียว และสองทิศทาง (BiLSTM) 128
5.8 การปรับปรุงประสิทธิภาพของแบบจำลองด้วย Pretrained Word Embedding 129
5.9 การวัดประสิทธิภาพของแบบจำลองที่ใช้เทคนิคการถ่ายโอนความรู้ 133
5.10 การวัดประสิทธิภาพแบบจำลองเมื่อใช้ทำนายข่าวอาชญากรรม 136
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 140
6.1 สรุปผลการวิจัย 140
6.2 ข้อจำกัดงานวิจัย 141
6.3 แนวทางการวิจัยในขั้นถัดไป 141
บรรณานุกรม 143

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก 147

ภาคผนวก ก 148

ภาคผนวก ข 164

ประวัติผู้เขียน 170

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 1-**1** ตารางแสดงแผนปฏิบัติงานตามวิธีดำเนินการวิจัย 4

ตารางที่ 2-1 ตารางคอนฟิวชันเมทริกต์ 69

ตารางที่ 2-2 แยกคำที่เป็น Positive และ Negative..... 74

ตารางที่ 4-1 รายละเอียดของกลุ่มความรับผิดตามหลักกฎหมาย 90

ตารางที่ 4-2 ข้อเท็จจริงตามโครงสร้างความรับผิดอาญา..... 93

ตารางที่ 4-3 คลังข้อมูลคำศัพท์ที่ใช้ในงานวิจัย 95

ตารางที่ 4-4 การกำหนดค่าไฮเปอร์พารามิเตอร์ในการฝึกฝนแบบจำลอง..... 102

ตารางที่ 5-1 ตัวอย่างข้อมูลคุณลักษณะผลการกระทำ (Causation)..... 108

ตารางที่ 5-2 ตัวอย่างข้อมูลคุณลักษณะเจตนา (Intention)..... 108

ตารางที่ 5-3 ตัวอย่างข้อมูลของคุณลักษณะการพิจารณาโทษ (Impunity)..... 109

ตารางที่ 5-4 ตัวอย่างข้อมูลของคุณลักษณะ พิจารณาความผิด (Justification)..... 109

ตารางที่ 5-5 ตัวอย่างการทำความสะอาดข้อมูลจากคำพิพากษาศาลฎีกา 111

ตารางที่ 5-6 ตัวอย่างการทำความสะอาดข้อมูลจากเนื้อหาข่าวอาชญากรรม 111

ตารางที่ 5-7 การตัดคำด้วยเทคนิค Longest Matching ร่วมกับตัวตัดคำ Deepcut [28]..... 112

ตารางที่ 5-8 เปรียบเทียบข้อมูลในแต่ละคลาสก่อนและหลังการใช้เทคนิค SMOTE..... 114

ตารางที่ 5-9 เปรียบเทียบค่าความถูกต้องของแต่ละแบบจำลอง..... 117

ตารางที่ 5-10 ข้อเท็จจริงตามโครงสร้างความรับผิดอาญาเมื่อผ่านการคัดเลือกคุณสมบัติที่สำคัญ 118

ตารางที่ 5-11 เปรียบเทียบประสิทธิภาพแบบจำลองต้นไม้ตัดสินใจด้วยค่าความแม่นยำ 120

ตารางที่ 5-12 เปรียบเทียบประสิทธิภาพแบบจำลอง LSTM ด้วยค่าเฉลี่ยมหภาคเอฟวัน 134

ตารางที่ 5-13 เปรียบเทียบประสิทธิภาพแบบจำลอง BiLSTM ด้วยค่าเฉลี่ยมหภาคเอฟวัน..... 134

ตารางที่ 5-14 การเปรียบเทียบค่าความแม่นยำและค่าเฉลี่ยมหภาคเอฟวันของทุกแบบจำลอง 136

ตารางที่ 5-15 การวัดประสิทธิภาพแบบจำลองด้วยข่าวอาชญากรรม..... 138

Chulalongkorn University

ตารางที่ ข-1การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย 165

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูป

หน้า

รูปที่ 2-1 การทำงานพื้นฐานของระบบประสาทเทียม [7] 22

รูปที่ 2-2 โครงสร้างพื้นฐานเพอร์เซปตรอน [8] 23

รูปที่ 2-3 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบเพอร์เซปตรอนหลายชั้น [9] 25

รูปที่ 2-4 สถาปัตยกรรมพื้นฐานของ Backpropagation [9] 26

รูปที่ 2-5 แสดงโครงข่ายประสาทแบบวนซ้ำอย่างง่าย [10] 27

รูปที่ 2-6 แสดงโครงข่ายของ Elman [11] 28

รูปที่ 2-7 ชนิดของฟังก์ชันกระตุ้นใน Single Layer Perceptron [12] 30

รูปที่ 2-8 การส่งผ่านข้อมูลในแต่ละชั้นของโครงข่ายประสาท [13] 30

รูปที่ 2-9 แสดงกราฟของฟังก์ชันซิกมอยด์ [14] 31

รูปที่ 2-10 แสดงกราฟของฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ [14] 32

รูปที่ 2- 11แสดงฟังก์ชันเรียงกระแสตรงหรือรีลู (LeRU) [14] 33

รูปที่ 2-12 การเปรียบเทียบฟังก์ชันกระตุ้นในงานจำแนกประเภท [16] 34

รูปที่ 2-13 แสดงฟังก์ชัน Leaky ReLU [14] 35

รูปที่ 2-14 แสดงซอฟต์แมกซ์ฟังก์ชันในชันข้อมูลส่งออก [13] 35

รูปที่ 2-15 โครงข่าย MLP แบบมีผู้สอนและค่าพารามิเตอร์ [17] 36

รูปที่ 2-16 การคำนวณค่าอัตราการผิดพลาดของการแพร่ย้อนกลับในชั้นข้อมูลผลลัพธ์ [17] 41

รูปที่ 2-17 การคำนวณค่าอัตราการผิดพลาดของการแพร่ย้อนกลับในชั้นซ่อน [17] 43

รูปที่ 2-18 แสดง วัน-ฮอท เวกเตอร์ของคำศัพท์ในคลังข้อมูล 46

รูปที่ 2-19 ความสัมพันธ์ของข้อมูลแต่ละคำในเวกเตอร์สเปซ (Vector space) 46

รูปที่ 2-20 Co-occurrence matrix ในระดับคำ [19] 47

รูปที่ 2-21 การหาความเหมือนของคำสองคำ (similarity) [19] 47

รูปที่ 2-22 แสดง PPMI Matrix ระหว่างคำที่เกิดขึ้น [19] 48

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 2-23 ความถี่ของแต่ละคำในแต่ละเอกสาร 49
รูปที่ 2-24 การคำนวณค่า TF-IDF จากเอกสาร 2 ชุด 49
รูปที่ 2-25 แสดงการฝังคำด้วยเทคนิค Word2Vec 51
รูปที่ 2-26 กราฟของฟังก์ชันอานม้า (Saddle Point) [20] 54
รูปที่ 2-27 Contour Plot ของ SGD ที่ไม่มีและมีโมเมนตัม [20] 54
รูปที่ 2-28 การปรับค่าพารามิเตอร์ของโมเมนตัมและโมเมนตัมร่วมกับ Nesterov [21] 55
รูปที่ 2-29 โครงสร้างของโครงข่ายประสาทเทียมแบบวกกลับ [18] 60
รูปที่ 2-30 ปัญหา Vanishing Gradient ของ RNN [15] 61
รูปที่ 2-31 แสดงโครงสร้างของประตูลืมและสูตรการคำนวณ [22] 62
รูปที่ 2-32 แสดงประตูทางเข้าและสูตรการคำนวณ [22] 63
รูปที่ 2-33 โครงสร้างการทำงานของประตูอัพเดตข้อมูล [22] 63
รูปที่ 2-34 แสดงประตูทางออกและสูตรการคำนวณ [22] 64
รูปที่ 2-35 โครงสร้างการทำงานของ Bi-directional LSTM [23] 65
รูปที่ 2-36 การตัดคำที่ไม่มีผลต่อการพิจารณาคดีออกจากข้อมูลสำหรับฝึกฝน 66
รูปที่ 2-37 การตัดคำโดยอาศัยคลังข้อมูลด้านกฎหมายอาญาร่วมกับการเรียนรู้เชิงลึก Deepcut 68
รูปที่ 2-38 5-fold cross-validation และการหาค่าเฉลี่ยความถูกต้อง 68
รูปที่ 2-39 แสดงการทำงานของเทคนิค SMOTE ในการเพิ่มข้อมูลตัวอย่างข้างน้อย 71
รูปที่ 2-40 การสังเคราะห์เพิ่มตัวอย่างข้างน้อย (SMOTE) [31] 72
รูปที่ 2-41 การถ่ายโอนการเรียนรู้ [32] 72
รูปที่ 2-42 เปรียบเทียบการเรียนรู้แบบดั้งเดิม (a) และการใช้ Transfer Learning (b) [33] 73
รูปที่ 2-43 ค่าถ่วงน้ำหนักที่ได้จากการฝึกฝนแบบจำลองถูกใช้ในการกำหนดค่าเริ่มต้นชั้นซ่อน 74
รูปที่ 2-44 Skip-gram Model 74
รูปที่ 2-45 เปรียบเทียบการเรียนรู้แบบตั้งแต่เริ่มต้นเทียบเรียนรู้โดยใช้ ULMFiT [37] 76
รูปที่ 2-46 เปรียบเทียบโครงข่ายทั่วไป และการเพิ่มการหยุดการเรียนรู้บางโหนด [38] 76

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 2-47 คุณลักษณะสำคัญของต้นไม้ตัดสินใจ 78
รูปที่ 4-1 ภาพรวมการสร้างแบบจำลองการจำแนกข้อความและกฎที่ใช้พิจารณา 85
รูปที่ 4-2 การใช้งานแบบจำลอง 86
รูปที่ 4-3 แสดงกระบวนการนำข้อมูลคำพิพากษาศาลฎีกาเข้ามาในระบบ 87
รูปที่ 4-4 การดึงข้อมูลจากเว็บไซต์เผยแพร่คำพิพากษาศาลฎีกา 88
รูปที่ 4- 5 การเลือกส่วนของเนื้อหาที่สำคัญจาก Html files 89
รูปที่ 4-6 ไฟล์ข้อมูลที่ผ่านการคัดเลือกส่วนที่ใช้ในงานวิจัยแล้ว 89
รูปที่ 4-7 แสดงการสกัดข้อเท็จจริงจากฎีกา เพื่อจัดให้อยู่ในโครงสร้างความรับผิดอาญา 94
รูปที่ 4-8 ตัวอย่างคำฟุ่มเฟือยที่ถูกตัดออกในการทำความสะอาดข้อมูล 95
รูปที่ 4-9 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะผลของการกระทำ (Causation) 96
รูปที่ 4-10 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะการพิจารณาโทษ (Impunity) 96
รูปที่ 4-11 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะเจตนา (Intention) 96
รูปที่ 4- 12 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะการพิจารณาความผิด (Justification) 97
รูปที่ 4-13 กระบวนการสร้างกฎเพื่อตัดสินข่าวอาชญากรรม 98
รูปที่ 4-14 โครงสร้างการทำงานของแบบจำลองภาษาธรรมชาติ 100
รูปที่ 4- 15 โครงสร้างภายในของแบบจำลอง LSTM 100
รูปที่ 4-16 โครงสร้างภายในของแบบจำลอง BiLSTM 101
รูปที่ 4-17 FastText Pretrained Word Embedding 103
รูปที่ 4-18 Thai2Vec Pretrained Word Embedding 103
รูปที่ 4-19 Crime2Vec Pretrained Word Embedding 104
รูปที่ 4-20 ไฟล์ที่ใช้ในการกำหนดค่าถ่วงน้ำหนักเริ่มต้นในแบบจำลอง 104
รูปที่ 4 21 แบบจำลองในงานวิจัยที่ทำการบันทึกเป็นไฟล์ .h5 105
รูปที่ 4- 22 การวัดประเมินความแม่นยำของแบบจำลองโดยภาพรวม 106
รูปที่ 5-1 ตัวอย่างการกำหนดรูปแบบในการตัดคำออกโดยการใช้นิพจน์ปกติ 110

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 5-2 ตัวอย่างการใช้คำที่อยู่ในคลังคำฟุ่มเฟือยเป็นตัวเปรียบเทียบคำที่มีอยู่เพื่อลบออก 110
รูปที่ 5-3 การเรียกใช้ฟังก์ชันในการทำความสะอาดข้อมูลและการตัดคำ 111
รูปที่ 5-4 ตัวอย่างข้อมูลของคุณลักษณะผลการกระทำที่ผ่านกระบวนการตัดคำ 112
รูปที่ 5-5 ตัวอย่างข้อมูลของคุณลักษณะการพิจารณาโทษที่ผ่านกระบวนการตัดคำ 112
รูปที่ 5-6 ตัวอย่างข้อมูลของคุณลักษณะเจตนาที่ผ่านกระบวนการตัดคำ 113
รูปที่ 5-7 ตัวอย่างข้อมูลของคุณลักษณะพิจารณาความผิดที่ผ่านกระบวนการตัดคำ 113
รูปที่ 5-8 การสุ่มเพิ่มตัวอย่างของคลาสทุกคลาสยกเว้นคลาสกลุ่มมาก 113
รูปที่ 5-9 การดูจำนวนข้อมูลของแต่ละคลาสในแบบจำลอง Causation 114
รูปที่ 5-10 การแสดงกราฟจำนวนข้อมูลในแต่ละคลาสของแบบจำลอง 115
รูปที่ 5-11 จำนวนข้อมูลในแต่ละคลาสของ Causation เมื่อผ่านเทคนิค SMOTE 115
รูปที่ 5-12 จำนวนข้อมูลในแต่ละคลาสของ Impunity เมื่อผ่านเทคนิค SMOTE 115
รูปที่ 5-13 จำนวนข้อมูลในแต่ละคลาสของ Intention เมื่อผ่านเทคนิค SMOTE 115
รูปที่ 5-14 จำนวนข้อมูลในแต่ละคลาสของ Justification เมื่อผ่านเทคนิค SMOTE 116
รูปที่ 5-15 การวัดความแม่นยำแบบจำลอง Justification ด้วยแบบจำลองนาอีฟเบย์ 116
รูปที่ 5-16 การวัดความแม่นยำแบบจำลอง Justification ด้วยแบบจำลอง LSTM 116
รูปที่ 5-17 การเรียกใช้ฟังก์ชันการคัดเลือกคุณสมบัติสำคัญในต้นไม้ตัดสินใจ 118
รูปที่ 5-18 แสดงการคัดเลือกคุณลักษณะสำคัญที่ส่งผลต่อการจำแนกกลุ่มความรับผิด 118
รูปที่ 5-19 การแบ่งข้อมูลและฝึกฝนแบบจำลองต้นไม้ตัดสินใจ 119
รูปที่ 5-20 ค่าความแม่นยำชุดข้อมูลทดสอบ 119
รูปที่ 5-21 ค่าความแม่นยำเฉลี่ยของ 5-Fold Cross Validation 119
รูปที่ 5-22 ค่าความแม่นยำเฉลี่ยของ 10-Fold Cross Validation 120
รูปที่ 5-23 คำสั่งที่ใช้แสดงไดอะแกรมของกราฟต้นไม้ตัดสินใจ 120
รูปที่ 5-24 ไดอะแกรมต้นไม้ตัดสินใจด้วยชุดข้อมูลฝึกสอน (Training set) 121
รูปที่ 5-25 แสดงตัวอย่างคำสั่งในการสร้างฟังก์ชันการตัดสินของต้นไม้ตัดสินใจ 121

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 5-26 กฎการพิจารณาที่สกัดจากต้นไม้ตัดสินใจ 122
รูปที่ 5-27 การออกแบบโครงสร้างและค่าพารามิเตอร์สำหรับแบบจำลอง Causation 122
รูปที่ 5-28 ค่าพารามิเตอร์ของโครงสร้างการฝึกฝนแบบจำลองด้วย LSTM 123
รูปที่ 5-29 ความแม่นยำของแบบจำลองระหว่างการฝึกฝนในแต่ละรอบของ BiLSTM 123
รูปที่ 5-30 แนวโน้มค่าความแม่นยำและค่าความผิดพลาดระหว่างการฝึกฝนของ LSTM 124
รูปที่ 5-31 การออกแบบโครงสร้างแบบจำลอง BiLSTM 125
รูปที่ 5-32 ค่าพารามิเตอร์ของโครงสร้างการฝึกฝนแบบจำลองด้วย BiLSTM 125
รูปที่ 5-33 ความแม่นยำของแบบจำลองระหว่างการฝึกฝนในแต่ละรอบของ BiLSTM 125
รูปที่ 5-34 แนวโน้มค่าความแม่นยำและค่าความผิดพลาดระหว่างการฝึกฝนของ BiLSTM 126
รูปที่ 5-35 นวโน้มค่าความแม่นยำระหว่างการฝึกฝนแบบจำลอง LSTM ทั้ง 4 แบบจำลอง 127
รูปที่ 5-36 แนวโน้มค่าความแม่นยำระหว่างการฝึกฝนแบบจำลอง BiLSTM ทั้ง 4 แบบจำลอง 127
รูปที่ 5-37 ค่าเฉลี่ย Macro-average F1 Score แบบจำลอง LSTM 128
รูปที่ 5-38 ค่าเฉลี่ย Macro-average F1 Score แบบจำลอง BiLSTM 128
รูปที่ 5-39 การโหลดค่าถ่วงน้ำหนักของ FastText 129
รูปที่ 5-40 การกำหนดโครงสร้างแบบจำลองที่ใช้ FastText 130
รูปที่ 5-41 การโหลดค่าถ่วงน้ำหนักของ Thai2Vec 130
รูปที่ 5-42 การกำหนดโครงสร้างแบบจำลองที่ใช้ Thai2Vec 130
รูปที่ 5-43 คำสั่งการฝึกฝนแบบจำลองภาษา (Language Model) 131
รูปที่ 5-44 คำสั่งการแสดงแบบจำลองทางภาษา 131
รูปที่ 5-45 ความสัมพันธ์ทางความหมายของแต่ละคำในแบบจำลองทางภาษา 132
รูปที่ 5-46 การโหลดค่าถ่วงน้ำหนักของ Crime2Vec 132
รูปที่ 5-47 การกำหนดโครงสร้างแบบจำลองที่ใช้ Crime2Vec 133
รูปที่ 5-48 ประสิทธิภาพแบบจำลอง Causation ที่ฝึกฝนด้วย LSTM 133
รูปที่ 5-49 ประสิทธิภาพแบบจำลอง Causation ที่ฝึกฝนด้วย BiLSTM 135

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 5-50 การคำนวณข้อมูลนำเข้าของแต่ละแบบจำลองและทำนายกลุ่มการลงโทษ 137
รูปที่ ก-1 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention LSTM 148
รูปที่ ก-2 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention FastText LSTM 148
รูปที่ ก-3 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Thai2Vec LSTM 149
รูปที่ ก-4 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Crime2Vec LSTM 149
รูปที่ ก-5 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation LSTM 150
รูปที่ ก-6 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation FastText LSTM 150
รูปที่ ก-7 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Thai2Vec LSTM 151
รูปที่ ก-8 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Crime2Vec LSTM 151
รูปที่ ก- 9 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity LSTM 152
รูปที่ ก-10 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity FastText LSTM 152
รูปที่ ก-11 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Thai2Vec LSTM 153
รูปที่ ก-12 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Crime2Vec LSTM 153
รูปที่ ก-13 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification LSTM 154
รูปที่ ก-14 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification FastText LSTM 154
รูปที่ ก-15 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Thai2Vec LSTM 155
รูปที่ ก-16 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Crime2Vec LSTM 155
รูปที่ ก-17 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention BiLSTM 156
รูปที่ ก-18 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention FastText BiLSTM 156
รูปที่ ก-19 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Thai2Vec BiLSTM 157
รูปที่ ก-20 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Crime2Vec BiLSTM 157
รูปที่ ก-21 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation BiLSTM 158
รูปที่ ก-22 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation FastText BiLSTM 158
รูปที่ ก-23 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Thai2Vec BiLSTM 159

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ ก-24 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Crime2Vec BiLSTM 159
รูปที่ ก-25 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity BiLSTM 160
รูปที่ ก-26 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity FastText BiLSTM 160
รูปที่ ก-27 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Thai2Vec BiLSTM 161
รูปที่ ก-28 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Crime2Vec BiLSTM 161
รูปที่ ก-29 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification BiLSTM 162
รูปที่ ก-30 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification FastText BiLSTM 162
รูปที่ ก-31 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Thai2Vec BiLSTM 163
รูปที่ ก-32 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Crime2Vec BiLSTM 163

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กฎหมายอาญาเป็นกฎหมายมหาชนผูกพันระหว่างรัฐกับเอกชน ที่มีวัตถุประสงค์เพื่อรักษาไว้ซึ่งความสงบเรียบร้อยของสังคม โดยอาศัยการลงโทษบุคคลซึ่งถูกกล่าวหาว่ากระทำความผิดเป็นมาตรการสำคัญ ดังนั้นจึงต้องมีการบัญญัติว่าการกระทำเช่นใดเป็นความผิดและกำหนดโทษที่จะลงแก่ผู้ที่กระทำผิดนั้น ประเทศไทยใช้ระบบกฎหมายลายลักษณ์อักษร (Civil Law) ซึ่งเป็นระบบที่บัญญัติกฎหมายไว้ในรูปแบบประมวลกฎหมาย [1] การกระทำใดจะเป็นความผิดอาญาหรือไม่นั้นย่อมขึ้นกับว่ามีลายลักษณ์อักษรบัญญัติความผิดไว้หรือไม่ ดังนั้นในการวินิจฉัยความรับผิดทางอาญาจึงต้องวินิจฉัยตามโครงสร้างความรับผิดอย่างเคร่งครัด [2]

เมื่อทุกคนต้องอยู่ภายใต้ข้อบังคับของกฎหมาย จะอ้างความไม่รู้กฎหมายมาเป็นข้อแก้ตัวไม่ได้ [3] ดังนั้นเมื่อเกิดการกระทำผิดอาญาขึ้น บุคคลที่มีส่วนเกี่ยวข้อง เช่น ผู้กระทำความผิด ผู้เสียหาย หรือประชาชนทั่วไปที่ต้องการทราบลักษณะความผิดรวมถึงแนวทางการลงโทษในเบื้องต้น เช่น ฝั่งผู้กระทำความผิดเมื่อทราบแนวทางการลงโทษที่ตัวเองต้องได้รับแล้ว ก็จะทราบข้อเท็จจริงที่เป็นสาเหตุให้ได้รับโทษในลักษณะเช่นนั้น และรวบรวมพยานหลักฐานเพื่อพิสูจน์ให้ได้ว่าข้อเท็จจริงดังกล่าวนั้นที่ถูกต้องเป็นเช่นใด ประกอบด้วยอะไรบ้าง เพื่อจะได้พิสูจน์ว่าตัวเองบริสุทธิ์ไม่ได้กระทำผิด หรือแม้จะได้กระทำผิดแต่ควรได้รับโทษน้อยลง ขณะที่ด้านผู้ได้รับความเสียหายก็เช่นกัน พอได้ทราบแนวทางการกำหนดบทลงโทษแก่ผู้กระทำความผิดแล้ว แต่ยังเชื่อว่าผู้กระทำความผิดควรจะต้องรับโทษหนักขึ้น ซึ่งข้อเท็จจริงใดที่จะทำให้แนวทางการตัดสินผ่อนปรนโทษทางอาญาลง ก็จะได้รวบรวมพยานหลักฐานเพิ่มเติม เพื่อต่อสู้คดีทั้งในชั้นสอบสวนของพนักงานสอบสวนและอัยการ รวมทั้งในชั้นการพิจารณาคดีของศาล หากเป็นผู้ที่อยู่ในกระบวนการยุติธรรมย่อมสามารถทราบแนวทางการวินิจฉัยในเบื้องต้นได้ แต่ประชาชนทั่วไปต้องสอบถามหรือขอความเห็นจากผู้เชี่ยวชาญเฉพาะด้านกฎหมายอาญา หรือการสืบค้นข้อมูลจากแหล่งค้นคว้าต่าง ๆ ผ่านระบบเครือข่ายอินเทอร์เน็ตก็สามารถกระทำได้ แต่การได้มาซึ่งข้อมูลที่กล่าวมาข้างต้น อาจต้องมีค่าใช้จ่ายหรือการสิ้นเปลืองเวลาในการสืบค้น และที่สำคัญผลลัพธ์ที่ได้อาจไม่ตรงตามวัตถุประสงค์ที่แท้จริง

จากข้อจำกัดที่กล่าวมาจึงเป็นที่มาของการจัดทำงานวิจัยนี้ เพื่อนำเสนอแบบจำลองและการวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญานี้ขึ้นมา เพื่อลดความไม่เข้าใจหรือตีความกฎหมายที่ผิดไปจากเจตนารมณ์ของมาตรานั้น รวมถึงช่วยให้การเข้าถึงข้อมูลสะดวก รวดเร็ว ลดขั้นตอน ซึ่งหมายรวมถึงข้อมูลผลลัพธ์จากระบบนี้ เป็นข้อมูลที่มีความถูกต้อง แม่นยำ และสามารถอ้างอิงได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอแบบจำลองและการวิเคราะห์การจำแนกข้อความภาษาธรรมชาติตามกฎหมายอาญาและทำนายแนวทางการตัดสินคดีในเบื้องต้น โดยผู้ใช้งานสามารถนำเข้าข้อมูลที่เป็นข้อเท็จจริงในคดีอาญา หรือตามเนื้อหาข่าวอาชญากรรมเพื่อให้แบบจำลองทำนายว่าจากข้อเท็จจริงที่ได้รับนั้นผู้กระทำความผิดต้องได้รับการลงโทษอย่างไร

1.3 ขอบเขตของการวิจัย

1. พิจารณาตามกฎหมายอาญาในหมวดความผิดเกี่ยวกับชีวิตคือ มาตรา 288 และ มาตรา 289 คือผู้กระทำมีเจตนาประสงค์ต่อชีวิตผู้ถูกกระทำเท่านั้น และนอกจากลงโทษตัวผู้กระทำผิดแล้วยังครอบคลุมบุคคลอื่นที่เกี่ยวข้องในคดี ได้แก่ ตัวการร่วม ผู้ใช้ และผู้สนับสนุน

2. งานวิจัยนี้ไม่รองรับคดีอาญาหรือเหตุการณ์ที่เป็นการก่อการร้าย ก่อความไม่สงบ หรือการวิสามัญฆาตกรรมของเจ้าหน้าที่ที่ได้ปฏิบัติการตามหน้าที่ การทำร้ายร่างกาย และการกระทำโดยประมาทที่เป็นสาเหตุให้ผู้อื่นถึงแก่ความตาย

3. ข้อมูลที่ใช้ในการฝึกฝนแบบจำลอง และวัดประสิทธิภาพ ต้องเป็นข้อความที่ยุติแล้ว รวมทั้งข่าวอาชญากรรมที่ใช้ในการทดสอบความแม่นยำก็หมายความถึงข้อมูลที่เป็นข้อเท็จจริงที่ยุติแล้ว

4. บทลงโทษถูกจำแนกออกเป็น 7 กลุ่ม เรียงตามลำดับจากเบาไปหาหนัก โดยอ้างอิงจากแนวทางการตัดสินของศาลฎีกาจากอดีตที่ผ่านมาถึงปัจจุบัน (พ.ศ.2500 - พ.ศ.2560) ได้แก่ กลุ่ม A คือ ไม่ได้กระทำผิดไม่ต้องรับโทษ กลุ่ม B คือ พยายามฆ่าผู้อื่นโดยเจตนาแต่มีเหตุลดโทษ กลุ่ม C คือ พยายามฆ่าผู้อื่น กลุ่ม D คือ ฆ่าผู้อื่นโดยเจตนาแต่มีเหตุลดโทษ กลุ่ม E คือ ฆ่าผู้อื่นโดยเจตนา กลุ่ม F คือ พยายามฆ่าผู้อื่นโดยไตร่ตรอง และกลุ่ม G คือ ฆ่าผู้อื่นโดยไตร่ตรอง

5. การทดสอบคุณภาพของแบบจำลองและกระบวนการวิเคราะห์ในขั้นสุดท้าย จะใช้ข้อมูลข่าวอาชญากรรมในเว็บไซต์ข่าวออนไลน์ เทียบกับความคิดเห็นของนักกฎหมายจำนวน 100 คดี

1.4 ขั้นตอนและวิธีการดำเนินการวิจัย

1. ศึกษาโครงสร้างความรับผิดกฎหมายอาญา และมาตราที่เกี่ยวข้อง

2. ศึกษาการเรียนรู้ของเครื่องด้วยเทคนิคการเรียนรู้เชิงลึก และการสร้างแบบจำลองด้วยต้นไม้ตัดสินใจ

3. ศึกษางานวิจัยที่เกี่ยวข้องกับคดีอาญาในไทย และงานวิจัยที่เกี่ยวข้องกับการการจำแนกข้อความ

4. รวบรวมข่าวอาชญากรรมและคำพิพากษาศาลฎีกา เพื่อสร้างชุดข้อมูลในการทดลอง

5. ออกแบบการสร้างและทดสอบแบบจำลอง

6. สร้างแบบจำลอง และทดสอบประสิทธิภาพ

7. สอบหัวข้อวิทยานิพนธ์

8. ปรับปรุงเทคนิคการสร้างแบบจำลอง และฟังก์ชันการทดสอบ เพื่อเพิ่มประสิทธิภาพของแบบจำลอง

9. สรุปผลการทดลอง

10.จัดทำบทความทางวิชาการ

11.จัดทำเอกสารวิทยานิพนธ์

12.สอบวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อช่วยให้ผู้ที่เกี่ยวข้องในคดีอาญา ได้แก่ ผู้กระทำความผิด หรือผู้ที่ได้รับความเสียหายได้ทราบแนวทางการต่อสู้คดี เพื่อเป็นประโยชน์แก่ฝ่ายตน และช่วยให้บุคคลทั่วไปที่สนใจในคดีอาญาได้ทราบแนวทางการกำหนดบทลงโทษแก่ผู้กระทำความผิดได้อย่างถูกต้อง

2. สามารถนำแบบจำลองและกระบวนการที่สร้างขึ้น เป็นที่ปรึกษาทางกฎหมายเบื้องต้นในคดีอาญาที่เกี่ยวข้องกับการประสงค์ต่อชีวิตได้

1.6 บทความวิชาการที่ได้รับการตีพิมพ์

งานวิจัยนี้ได้รับคัดเลือกให้ตีพิมพ์เป็นบทความวิชาการ เรื่อง “An Analysis of Natural Language Text Relating to Thai Criminal Law” โดย วีรยุทธ ครั่งกลาง และ สุกรี สินธุภิญโญ ในงานประชุมวิชาการ “12th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2020” ระหว่างวันที่ 25-27 มิถุนายน 2563 ณ มหาวิทยาลัย Politehnica กรุงบูคาเรสต์ ประเทศโรมาเนีย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

*ตารางที่ 1-**1** ตารางแสดงแผนปฏิบัติงานตามวิธีดำเนินการวิจัย*

กิจกรรม	พ.ศ.2562							พ.ศ.2563								
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค. - มิ.ย.			ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาโครงสร้างความรับผิดอาญา																
2. ศึกษาต้นไม้ตัดสินใจและการเรียนรู้เชิงลึก																
3. ศึกษางานวิจัยที่เกี่ยวข้อง																
4. รวบรวมคำพิพากษาศาลฎีกา																
5. ออกแบบและทดสอบแบบจำลอง																
6. ทดสอบประสิทธิภาพแบบจำลอง																
7. สอบหัวข้อวิทยานิพนธ์																
8. ปรับปรุงแบบจำลอง และฟังก์ชัน																
9. สรุปผลการทำวิจัย																
10. จัดทำบทความทางวิชาการ																
11. จัดทำเอกสารวิทยานิพนธ์																
12. สอบวิทยานิพนธ์																

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในการวิเคราะห์และออกแบบงานวิจัยนี้ ประกอบด้วยความรู้ด้านกฎหมายอาญา (Criminal Law) งานด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing) และการเรียนรู้เชิงลึก (Deep Learning) ดังนี้

2.1 บทบัญญัติประมวลกฎหมายอาญา [1]

ประมวลกฎหมายอาญา เป็นบทบัญญัติเกี่ยวของกับความผิดทางอาญาและการกำหนดโทษที่ลงแก่ความผิดนั้น โดยมีการรวบรวมบทบัญญัติแห่งกฎหมายอย่างเป็นระบบ จัดสรรเป็นหมวดหมู่ แบ่งเป็น 3 ภาค ประกอบด้วย 398 มาตรา ได้แก่

2.1.1 ภาค 1 บทบัญญัติทั่วไป วางหลักเกณฑ์ทั่วไปของกฎหมายอาญาที่ใช้กับความผิดตามความผิดในภาค 2 และภาค 3 ด้วย เช่น ประมวลกฎหมายอาญา มาตรา 59 วรรคแรก บัญญัติว่า “บุคคลจะต้องรับผิดในทางอาญาก็ต่อเมื่อได้กระทำโดยเจตนา เว้นแต่จะได้กระทำโดยประมาท ในกรณีที่กฎหมายบัญญัติให้ต้องรับผิดเมื่อได้กระทำโดยประมาท หรือเว้นแต่ในกรณีที่กฎหมายบัญญัติไว้โดยแจ้งชัดให้ต้องรับผิดแม้ได้กระทำโดยไม่มีเจตนา” หมายความว่า การกระทำของบุคคลนั้นจะเป็นความผิดต่อเมื่อได้กระทำโดยเจตนาเป็นหลัก ทั้งนี้ มีข้อยกเว้นให้ผู้กระทำต้องรับผิดในทางอาญาแม้จะกระทำโดยประมาท หรือเว้นแต่ในกรณีที่กฎหมายบัญญัติไว้โดยแจ้งชัดให้ต้องรับผิดแม้ได้กระทำโดยไม่มีเจตนา นอกจากนี้ ประมวลกฎหมายอาญา ภาค 1 ยังนำไปใช้บังคับกับความผิดตามกฎหมายอื่นที่กำหนดความผิดและกำหนดโทษทางอาญาด้วย ดังที่ ประมวลกฎหมายอาญา มาตรา 17 บัญญัติว่า “บทบัญญัติในภาค 1 แห่งประมวลกฎหมายนี้ให้ใช้ ในกรณีแห่งความผิดตามกฎหมายอื่นด้วย เว้นแต่กฎหมายนั้น ๆ จะได้บัญญัติไว้เป็นอย่างอื่น” หลักเกณฑ์ต่าง ๆ ในกฎหมายอาญา ภาค 1 นี้ จึงมีความสำคัญต่อการวินิจฉัยความผิดทางอาญาใน กฎหมายต่าง ๆ ประมวลกฎหมายอาญา ภาค 1 มีบัญญัติไว้ในมาตรา 1 ถึงมาตรา 106

2.1.2 ภาค 2 ความผิด เป็นการศึกษาถึงความผิดทางอาญาฐานต่าง ๆ ที่มีความเสียหายเป็นภัย ลักษณะในลักษณะต่าง ๆ ศึกษาถึงองค์ประกอบความผิดในแต่ละฐาน โดยต้องนำหลักเกณฑ์ในภาค 1 มาใช้ในการวินิจฉัยความผิดฐานต่าง ๆ ด้วย เช่น เมื่อกล่าวถึงความผิดฐานลักทรัพย์ตามมาตรา 334 ซึ่งบัญญัติว่า “ผู้ใดเอาทรัพย์ของผู้อื่น หรือที่ผู้อื่น เป็นเจ้าของรวมอยู่ด้วยไปโดยทุจริต...” แม้จะไม่ปรากฏถ้อยคำถึงองค์ประกอบภายในเรื่องเจตนา แต่ก็ต้องเข้าใจได้ว่า เจตนาย่อมเป็นองค์ประกอบในความผิดฐานลักทรัพย์ด้วยหากคนหลายคนมีส่วนร่วมในการลักทรัพย์ จะถือว่าแต่ละคนต้องรับผิดฐานลักทรัพย์อย่างไร จะเป็นตัวการในความผิดนี้ หรือบางคนเป็นเพียงผู้สนับสนุน อันมีผลต่อการวินิจฉัยว่า การกระทำนั้นต้องรับโทษตาม มาตรา 335 (7) คือลักทรัพย์โดยร่วมกระทำความผิดด้วยกันตั้งแต่ 2 คน

ขึ้นไปหรือไม่ หรือเพียงแต่ รับโทษเพียงผู้สนับสนุนในการกระทำความผิดฐานลักทรัพย์ตามมาตรา 334 ประกอบมาตรา 86 เท่านั้น เป็นต้น ประมวลกฎหมายอาญา ภาค 2 ความผิด แบ่งเป็น 14 ลักษณะ จำนวน 260 มาตรา ตั้งแต่มาตรา 107 – 366 แต่ในงานวิจัยนี้ใช้ฐานความผิดในลักษณะ 10 ความผิดเกี่ยวกับชีวิตและร่างกาย เฉพาะในหมวดความผิดเกี่ยวกับชีวิตโดยเจตนา ได้แก่มาตรา 288 และ 289 เท่านั้น

2.1.3 ภาค 3 ลหุโทษ เป็นการศึกษาถึงความผิดเล็ก ๆ น้อย ๆ ที่ถูกแยก ออกจากภาค 2 ตั้งแต่มาตรา 367 ถึงมาตรา 398 ความผิดลหุโทษนี้ มิได้ถูกจัดแยกออกเป็นลักษณะ

2.2 โครงสร้างความรับผิดทางอาญา [2]

ประมวลกฎหมายอาญาฐานต่าง ๆ ที่บัญญัติไว้ในภาค 2 ความผิด และภาค 3 ลหุโทษนั้น ในแต่ละฐานความผิดจะมีรายละเอียดที่แตกต่างกันออกไป อย่างไรก็ตาม การที่บุคคลจะต้องรับผิดในทางอาญาหรือไม่นั้น ต้องพิจารณาตามโครงสร้างความรับผิดทางอาญาทั้ง 3 โครงสร้าง ดังนี้

2.3.1 โครงสร้างที่ 1 การกระทำครบองค์ประกอบความผิด บุคคลจะมีความรับผิดทางอาญาได้นั้นต้องมีการกระทำครบองค์ประกอบความผิดในแต่ละฐานนั้น ไม่ว่าจะเป็นความผิดในประมวลกฎหมายอาญา หรือกฎหมายอาญาอื่น ต้องมีองค์ประกอบความผิดเสมอ ประกอบด้วยองค์ประกอบภายนอก และองค์ประกอบภายใน ดังนี้

1. องค์ประกอบภายนอก หมายถึงองค์ประกอบที่อยู่ภายนอกของความผิดแต่ละฐาน สามารถพิจารณาและมองเห็นได้ภายนอกได้แก่ ผู้กระทำ คือ ผู้ที่ต้องรับผิดในการกระทำความผิดฐานนั้น ๆ หากกฎหมายบัญญัติถึงผู้กระทำความผิดไว้ว่า “ผู้ใด” ย่อมมีความหมายว่า คนทั่วไปสามารถเป็นผู้กระทำความผิดฐานนั้นได้ แต่หากกฎหมายบัญญัติคุณสมบัติหรือฐานะของผู้กระทำไว้เป็นพิเศษ ผู้ที่จะต้องผิดย่อม ต้องมีคุณสมบัติหรือฐานะนั้น ๆ ด้วย การกระทำ เป็นลักษณะของการกระทำที่แสดงออกมาภายนอก และมีกฎหมายบัญญัติไว้เป็นความผิดและกำหนดโทษไว้ การกระทำที่เป็นความผิดจะมีลักษณะแตกต่างกันไปตามแต่ละฐานความผิด เช่น ฆ่า หรือทำร้าย ตามหลักกฎหมายอาญาบุคคลจะต้องรับผิดต่อเมื่อมี การกระทำ อันได้แก่ การเคลื่อนไหวหรือไม่เคลื่อนไหวร่างกายภายใต้บังคับของจิตใจซึ่งประมวลกฎหมายอาญา มาตรา 59 ใช้คำว่า“กระทำโดยรู้สำนึกในการที่กระทำ” กล่าวคือ จะต้องมีความคิดที่จะกระทำ มีการตกลงใจที่จะกระทำตามที่คิดไว้ และได้กระทำการไปตามที่ตกลงใจนั้น [หยุด แสงอุทัย, 2551] การคิด ตกลงใจ และกระทำตามที่ตกลงใจนั้น หมายความรวมถึง การละเว้น หรือการงดเว้น ด้วย การงดเว้นที่จักต้องกระทำเพื่อป้องกันผลนั้น มาตรา 59 วรรคท้ายนั้น ต้องเป็นกรณีที่มีหน้าที่ตามกฎหมายบัญญัติไว้ เกิดจากการยอมรับโดยเฉพาะเจาะจง เกิดจากการกระทำครั้งก่อน ๆ ของตน และเกิดจากความสัมพันธ์พิเศษเฉพาะเรื่อง แต่หากเป็นการเคลื่อนไหวร่างกายโดยไม่อยู่ภายใต้บังคับของจิตใจ เช่น การละเมอ การชักกระตุก ไม่ใช่

การกระทำ กรรมของการกระทำ คือสิ่งที่กฎหมายบัญญัติไว้ในความผิดแต่ละฐาน ซึ่งอาจเป็นบุคคลหรือวัตถุก็ได้

2. องค์ประกอบภายใน หมายถึงองค์ประกอบที่ไม่สามารถมองเห็นได้จากภายนอก เพราะเป็นส่วนหนึ่งที่อยู่ในจิตใจของผู้กระทำ ซึ่งเราจะพิจารณาส่วนที่อยู่ในจิตใจนี้ได้จากการกระทำ โดยหลักการที่ว่า “กรรมเป็นเครื่องชี้เจตนา” ตามที่บัญญัติไว้ในประมวลกฎหมายอาญา มาตรา 59 วรรคแรกว่า “บุคคลจะต้องรับผิดในทางอาญาก็ต่อเมื่อได้กระทำโดยเจตนา เว้นแต่จะได้กระทำโดยประมาท ในกรณีที่กฎหมายบัญญัติให้ต้องรับผิดเมื่อได้กระทำโดยประมาท หรือเว้นแต่ในกรณีที่กฎหมายบัญญัติไว้โดยแจ้งชัดให้ต้องรับผิดแม้ได้กระทำโดยไม่มีเจตนา” บทบัญญัตินี้ ได้วางหลักว่าบุคคลจะต้องรับผิดต่อเมื่อกระทำการโดยเจตนา โดยแบ่งเจตนาเป็นประเภทต่าง ๆ ได้ดังนี้

2.1 เจตนาตามความเป็นจริง ประมวลกฎหมายอาญา ได้บัญญัติเจตนาไว้ใน มาตรา 59 วรรคสองและสาม การจะค้นหาความหมายของคำว่าเจตนาจึงต้องพิจารณาทั้งสองวรรคประกอบกัน ซึ่งบัญญัติว่า “กระทำโดยเจตนา ได้แก่กระทำโดยรู้สำนึกในการที่กระทำและในขณะเดียวกันผู้กระทำประสงค์ต่อผล หรือย่อมเล็งเห็นผลของการกระทำนั้น ถ้าผู้กระทำมิได้รู้ข้อเท็จจริงอันเป็นองค์ประกอบของความผิด จะถือว่าผู้กระทำประสงค์ต่อผล หรือย่อมเล็งเห็นผลของการกระทำนั้นมิได้” จากบทบัญญัติดังกล่าว ความหมายของการกระทำโดยเจตนาจึง ประกอบด้วย ผู้กระทำต้องรู้ข้อเท็จจริงอันเป็นองค์ประกอบของความผิด คือ รู้ข้อเท็จจริงเกี่ยวกับองค์ประกอบความผิดเช่น ในความผิดฐานฆ่าคนโดยเจตนา ผู้กระทำต้องรู้ว่าการกระทำของตนเป็นการ “ฆ่า” และกรรมของการกระทำเป็น “ผู้อื่น”กล่าวคือรู้ถึงองค์ประกอบภายนอกนั่นเอง แต่ไม่จำเป็นต้องรู้ถึงรายละเอียดเฉพาะเช่นรู้ว่าสิ่งที่ตนเองทำนั้นเป็นการฆ่าและเป็นการฆ่าผู้อื่นก็เพียงพอแล้ว ไม่จำต้องรู้ว่าผู้อื่นที่ถูกฆ่านั้นคือ นาย ก. หรือ นาย ข. ยกตัวอย่างเช่น นาย ก. ต้องการที่จะฆ่า นาย ข. จึงดักยิงที่มุมตึก ระหว่างนาย ข.เดินใกล้ถึงมุมตึก นาย ค. ได้เดินตัด หน้ามาถึงมุมตึกก่อน นาย ก.จึงยิง นาย ค.โดยเข้าใจว่าผู้ที่เดินมาถึงนั้นเป็น นาย ข. เช่นนี้ ถือว่า นาย ก. รู้ข้อเท็จจริงองค์ประกอบความผิดแล้ว เพราะ นาย ก. รู้อยู่ว่ากรรมของการกระทำนั้นเป็นผู้อื่น จะยกเอาความสำคัญผิดในตัวบุคคลเป็นข้อแก้ตัวว่าไม่มีเจตนาไม่ได้ สำหรับข้อเท็จจริงที่เป็นเหตุฉกรรจ์ ถือเป็นข้อเท็จจริงอันเป็นองค์ประกอบความผิดเช่นเดียวกัน ผู้กระทำจึงต้องรู้ข้อเท็จจริงดังกล่าว ซึ่งประมวลกฎหมายอาญาได้ บัญญัติเรื่องนี้ไว้ในมาตรา 62 วรรคท้าย ว่า “บุคคลจะต้องรับโทษหนักขึ้นโดยอาศัยข้อเท็จจริงใด บุคคลนั้นจะต้องได้รู้ข้อเท็จจริงนั้น” เช่น นาย ก. ฆ่า นาง ข. ซึ่งเป็นมารดาของ นาย ก. ซึ่งพลัดพรากกันไปแต่เด็ก นาย ก. ไม่รู้ว่าผู้ที่ตนเป็นมารดาจึงไม่รู้ข้อเท็จจริงอันเป็นองค์ประกอบความผิดตาม มาตรา 289 (1) จึงไม่มีความผิดฐานฆ่าบุพการี คงมีความผิดฐานฆ่าคนโดยเจตนาตามมาตรา 288 เท่านั้น และ ผู้กระทำต้องประสงค์ต่อผล หรือย่อมเล็งเห็นผลของการกระทำนั้น โดยเจตนาประสงค์ต่อผล หมายถึงการประสงค์ให้เกิดผลตามที่ตั้งใจโดยตรง หากเกิดผลตามที่ประสงค์ก็เป็นความผิดสำเร็จ หากไม่เกิดผลก็

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เป็นเพียงฐานพยายามกระทำความผิด ส่วน เจตนาเล็งเห็นผล หมายถึงผู้กระทำมิได้ประสงค์ต่อผล กล่าวคือมิได้มุ่งหมายให้เกิดผลขึ้น แต่ผู้กระทำเล็งเห็นได้ว่าผลนั้นจะเกิดขึ้น ถือว่าผู้กระทำมีเจตนาเช่นกัน เจตนาเล็งเห็นผลนี้ เรียกว่า “เจตนาโดยอ้อม”

2.2 เจตนาโดยผลของกฎหมาย หมายความว่า ผู้กระทำมิได้มีเจตนาตามความเป็นจริงแก่บุคคลที่ได้รับผลร้ายกล่าวคือ มิได้ประสงค์ต่อผล และมิได้เล็งเห็นผล แต่กฎหมายให้ถือว่าผู้กระทำมีเจตนาต่อผู้รับผลร้ายจากการกระทำของตนด้วย หรืออีกนัยหนึ่งให้ถือว่าผู้กระทำประสงค์โดยเจตนาแก่ผู้ที่รับผลร้ายด้วยเจตนาอย่างเดียวกันกับบุคคลที่ตนประสงค์ต่อผลหรือเล็งเห็นผลในการกระทำของตนโดยผลของกฎหมาย จึงเรียกว่าเป็น “เจตนาโดยผลของกฎหมาย” หรือ “เจตนาโอน” แต่คำว่าเจตนาโอนในที่นี้ มิได้หมายความว่ามีเพียงเจตนาต่อผู้รับผลร้ายเท่านั้น แต่หมายความว่ามีเจตนาทั้งกับผู้ที่ผู้กระทำเจตนากระทำต่อและผู้ที่ได้รับผลร้ายด้วย หลักในเรื่องเจตนาโดยผลของกฎหมาย บัญญัติไว้ในประมวลกฎหมาย อาญา มาตรา 60 บัญญัติว่า “ผู้ใดเจตนาที่จะกระทำต่อบุคคลหนึ่ง แต่ผลของการกระทำเกิดแก่อีกบุคคล หนึ่งโดยพลาดไป ให้ถือว่าผู้นั้นกระทำโดยเจตนาแก่บุคคลซึ่งได้รับผลร้ายจากการกระทำนั้น แต่ในกรณีที่กฎหมายบัญญัติให้ลงโทษหนักขึ้น เพราะฐานะของบุคคลหรือเพราะความสัมพันธ์ ระหว่างผู้กระทำกับบุคคลที่ได้รับผลร้าย มิให้นำกฎหมายนั้นมาใช้บังคับเพื่อลงโทษผู้กระทำให้หนักขึ้น” เรียกกันทั่วไปว่าการกระทำโดยพลาด

2.3 เจตนาพิเศษ เป็นเจตนาตามความเป็นจริงอย่างหนึ่งแต่มิใช่เจตนาประสงค์ต่อผลหรือย่อมเล็งเห็นผลแต่เป็นจุดมุ่งหมายแห่งการกระทำความผิดนั้น เช่น ฆ่าผู้อื่นเพื่อตระเตรียมการหรือเพื่อความสะดวกในการที่จะกระทำความผิดอย่างอื่น ตามมาตรา 289 (6) การกระทำใด ๆ ผู้กระทำย่อมต้องมีเหตุจูงใจให้กระทำเช่นนั้น แม้แต่มูลเหตุชักจูงใจ ยังเกิดขึ้นได้เพราะมีเหตุจูงใจอีกอย่างหนึ่งเกิดขึ้นก่อน มูลเหตุชักจูงใจอันเป็นเจตนาพิเศษในที่นี้ จึงหมายเฉพาะที่กฎหมายบัญญัติเป็นองค์ประกอบความผิดเท่านั้นไม่รวมถึงเหตุจูงใจทั่ว ๆ ไป เช่น ผู้กระทำความผิดฐานฆ่าผู้อื่น อาจมีมูลเหตุจูงใจจากการโกรธแค้นกันมา แต่มาตรา 288 มิได้บัญญัติถึงเหตุจูงใจใดไว้ การลงโทษบุคคลฐานฆ่าผู้อื่นโดยเจตนาจึงไม่ต้องคำนึงว่าเป็นการฆ่าด้วยเหตุใด

ข้อยกเว้นที่แม้จะไม่ได้มีเจตนา แต่กฎหมายก็ให้ถือว่ามีความผิดทางอาญา [4] ได้แก่

1. กระทำโดยประมาท เฉพาะในกรณีที่มีกฎหมายบัญญัติให้ต้องรับผิดแม้จะกระทำโดยประมาท เพราะโดยปกติแล้วกฎหมายประสงค์จะลงโทษบุคคลที่กระทำโดยมีเจตนาละเมิด ต่อสิ่งที่กฎหมายประสงค์จะคุ้มครอง หรือที่เรียกว่า “คุณธรรมทางกฎหมาย” แต่การไม่ใช้ความระมัดระวังทั้งที่สามารถกระทำได้ อาจก่อให้เกิดความเสียหาย หรือทำให้สิ่งที่กฎหมายต้องการจะคุ้มครองได้รับอันตราย จึงมีการบัญญัติให้บุคคลต้องรับผิดเมื่อได้กระทำโดยประมาทในบางกรณี

2. กฎหมายบัญญัติไว้โดยชัดแจ้งว่าต้องรับผิดแม้ได้กระทำโดยไม่มีเจตนา เช่น มาตรา 16 แห่งพระราชบัญญัติศุลกากร (ฉบับที่ 9) พุทธศักราช 2482 ซึ่งใช้คำว่า “การกระทำที่

บัญญัติไว้ในมาตรา 27 และมาตรา 99 แห่งพระราชบัญญัติศุลกากร พุทธศักราช 2496 นั้น ให้ถือว่าเป็นความผิดโดยมิพักต้องคำนึงว่าผู้กระทำมีเจตนาหรือกระทำโดยประมาท เลินเล่อหรือหาไม่”

3. ความผิดลหุโทษ ประมวลกฎหมายอาญา มาตรา 104 บัญญัติว่า“การกระทำความผิดลหุโทษตาม ประมวลกฎหมายนี้ แม้กระทำโดยไม่มีเจตนาก็เป็นความผิด เว้นแต่ตามบทบัญญัติความผิดนั้นจะมีความบัญญัติให้เห็นเป็นอย่างอื่น” ที่ว่า “ไม่มีเจตนา” ก็เป็นความผิดนั้น หมายความว่า แม้ผู้กระทำไม่มีเจตนาและไม่ประมาท ก็เป็นความผิด ซึ่งความผิดลหุโทษจะมีโทษจำคุกไม่เกินหนึ่งเดือนหรือปรับไม่เกินหนึ่งพันบาทหรือทั้งจำทั้งปรับ

3. ความสัมพันธ์ระหว่างการกระทำและผล บุคคลจะต้องรับผิดในทางอาญาได้ต่อเมื่อเขาได้กระทำอันเป็นความผิดกฎหมาย หากเขาไม่ได้กระทำหรือไม่ได้เป็นผู้ที่ก่อให้เกิดความเสียหายขึ้น เขาก็ไม่ต้องรับผิดในผลที่เกิดขึ้นนั้น ดังนั้นผลที่เกิดขึ้นต้องมีความสัมพันธ์กับการกระทำของผู้กระทำด้วย แบ่งการพิจารณาออกเป็น 2 ประเภท ได้แก่

3.1 หลักความสัมพันธ์ระหว่างการกระทำและผล คือความผิดที่ต้องมีผลเกิดขึ้น และพิสูจน์ได้ความว่าผลนั้นมีความสัมพันธ์กับการกระทำนั้น ๆ จึงจะเป็นความผิด เช่น นายขาวใช้ปืนยิงนายดำ นายดำได้รับบาดเจ็บและเสียชีวิตในอีก 2 วันต่อมา แม้นายดำจะไม่ได้ตายทันที แต่ความตายก็เป็นผลจากการกระทำของนายแดง ดังนั้นผลและการกระทำจึงมีความสัมพันธ์กัน นายขาวก็ต้องรับผิดในความตายของนายดำ แต่ถ้าเป็นกรณีที่ผลที่เกิดขึ้นไม่มีความสัมพันธ์กับการกระทำ ผู้ที่กระทำก็ไม่ต้องรับผิดในผลที่เกิดขึ้น ในเรื่องความสัมพันธ์ระหว่างการกระทำและผล เมื่อมีปัญหาว่าผลที่เกิดขึ้นนั้นเกิดจากการกระทำนั้นหรือไม่ จะใช้ทฤษฎีทางกฎหมายอาญามาพิจารณา 2 ทฤษฎี ได้แก่ ทฤษฎีเงื่อนไข มีหลักว่าถ้าไม่มีการกระทำอันใดอันหนึ่งแล้ว ผลจะไม่เกิด ถือว่าผลเกิดจากการกระทำอันนั้น แม้ว่าผลจะมาจากการกระทำอื่น ๆ ด้วยก็ตาม [2] กล่าวคือในเหตุแต่ละเหตุที่ประกอบกันขึ้นผลนั้น จะแยกออกจากกันไม่ได้ ทุก ๆ เหตุถือเป็นเหตุที่ทำให้เกิดผลได้เท่ากันทุกเหตุ หากในกระบวนการที่ก่อให้เกิดผลนั้น เกิดจากความผิดพลาดของผู้ใดแทรกอยู่ ผู้นั้นย่อมต้องรับผิดชอบในผลที่เกิดขึ้นทั้งหมด และ ทฤษฎีมูลเหตุที่เหมาะสม เป็นการพิจารณาการกระทำที่เป็นเหตุว่ามีความเหมาะสมที่จะก่อให้เกิดผลเช่นนั้นหรือไม่กล่าวคือ พิจารณาว่าการกระทำและผลมีความเกี่ยวพันกันพอสมควรหรือไม่ ในกรณีที่ผลของการกระทำความผิดทำให้ผู้กระทำต้องรับโทษหนักขึ้นหมายความว่า เป็นผลที่ผู้กระทำสามารถ “คาดเห็น”ความเป็นไปได้ของผลนั้น โดยในการวินิจฉัยความสามารถในการคาดเห็น ใช้หลักมาตรฐานของคนทั่วไปหรือ “วิญญูชน” เช่น ทะเลาะวิวาทกัน ชกโดนเบ้าตาอีก 5 วันต่อมาตาอักเสบและบอด คนทั่วไปย่อม “คาดเห็น” ได้ว่าการที่ตาบอดนั้น ย่อมเป็นผลที่เกิดขึ้นได้ ดังนั้นในการวินิจฉัยความผิดของผู้กระทำต้องใช้ทั้งใช้ทฤษฎีเงื่อนไขและทฤษฎีเหตุที่เหมาะสมประกอบกัน [2]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 ไม่ต้องพิจารณาหลักความสัมพันธ์ระหว่างการกระทำและผล คือความผิดที่ไม่ต้องมีผลเกิดขึ้นก็เป็นความผิดเช่น ความผิดฐานแจ้งความความเท็จแก่เจ้าพนักงานตามมาตรา 137 เมื่อผู้กระทำความผิดได้แจ้งข้อความอันเป็นเท็จแก่เจ้าพนักงาน ก็เป็นความผิดแล้ว

2.3.2 โครงสร้างที่ 2 การไม่มีกฎหมายยกเว้นความผิด การกระทำใดที่ครบองค์ประกอบความผิดในฐานนั้น ๆ แล้ว จะต้องพิจารณาต่อไปอีกว่า การกระทำความผิดนั้น มีกฎหมายยกเว้นความผิดไว้หรือไม่ หากมีกฎหมายยกเว้นความผิดไว้ การกระทำนั้นแม้จะครบองค์ประกอบความผิดแล้ว ก็ถือว่าไม่เป็นความผิดอาญา ซึ่งเหตุยกเว้นความผิดมีหลายสาเหตุ ทั้งเหตุยกเว้นความรับผิดที่อยู่ในประมวลกฎหมายอาญาและกฎหมายอื่น รวมถึงเหตุยกเว้นความผิดตามหลักกฎหมายทั่วไป ที่ไม่ได้บัญญัติไว้เป็นลายลักษณ์อักษร ดังนี้

1. กฎหมายหรือจารีตประเพณีให้อำนาจกระทำได้ ดังนี้

1.1 ให้อำนาจแก่พนักงานเจ้าหน้าที่ในการปฏิบัติการตามหน้าที่ เช่น ให้อำนาจตำรวจในการจับกุมผู้กระทำผิดเป็นต้น

1.2 ให้อำนาจแก่บุคคลโดยผ่านการกลั่นกรองของพนักงานเจ้าหน้าที่ก่อน เช่น บุคคลที่ต้องการจะทำไม้หวงห้ามต้องขออนุญาตจากพนักงานเจ้าหน้าที่ก่อน

1.3 ให้อำนาจแก่บุคคลในการที่จะกระทำกิจการเพื่อประโยชน์สาธารณะหรือเพื่อผู้อื่น เช่น สมาชิกสภาผู้แทนราษฎรมีเอกสิทธิ์ในการแสดงความคิดเห็นต่าง ๆ ในสภาผู้แทนราษฎร บุคคลใดจะนำไปฟ้องร้องไม่ได้

1.4 ให้อำนาจแก่บุคคลเพื่อการปกป้องคุ้มครองสิทธิหรือประโยชน์สุขส่วนตน เช่น คู่ความซึ่งแสดงความคิดเห็นในกระบวนการพิจารณาคดีในศาลไม่มีความผิดฐานหมิ่นประมาท

2. กระทำเพื่อป้องกันสิทธิของตนเองหรือผู้อื่นพอสมควรแก่เหตุ ตามประมวลกฎหมายอาญา มาตรา 68 ว่าด้วยการป้องกัน ซึ่งบัญญัติไว้ว่า “ผู้ใดจำต้องกระทำการใดเพื่อป้องกันสิทธิของตนหรือของผู้อื่นให้พ้นภยันตรายซึ่งเกิดจากการประทุษร้ายอันละเมิดต่อกฎหมายและเป็นภยันตรายที่ใกล้จะถึง ถ้าได้กระทำพอสมควรแก่เหตุ การกระทำนั้นเป็นการป้องกันโดยชอบด้วยกฎหมาย ผู้นั้นไม่มีความผิด” จากบทบัญญัติดังกล่าว สามารถจำแนกองค์ประกอบของการป้องกัน ได้ดังนี้

2.1 มีภยันตรายซึ่งเกิดจากการประทุษร้ายอันละเมิดต่อกฎหมาย หมายถึง ภัยที่เป็นความเสียหายต่อชีวิต ร่างกาย ชื่อเสียง เสรีภาพ ทรัพย์สิน ซึ่งเป็นสิทธิของบุคคล ภัยที่เป็นอันตรายต่อสิ่งเหล่านี้ล้วนแต่เป็นสิ่งที่กฎหมายคุ้มครองอยู่ เมื่อมีผู้ใดมาประทุษร้ายย่อมสามารถป้องกันได้ แต่หากยังไม่มีภยันตรายก็ไม่อาจอ้างป้องกันได้ และ

2.2 ภยันตรายนั้นใกล้จะถึง หมายถึง ภยันตรายที่ปรากฏอยู่เฉพาะหน้า หรือเป็นภยันตรายที่ได้เริ่มเกิดขึ้นแล้ว รวมตลอดถึงภยันตรายที่ยังปรากฏอยู่ต่อไปอีก และผู้รับภัยไม่จำเป็นต้องหลบหนีภัยที่ใกล้จะถึงนั้น แม้ว่าหากหลบหนีเสียก็จะพ้นภัยก็ตาม รวมถึงการป้องกัน

ภยันตรายไว้ล่วงหน้าก็สามารถทำได้ เว้นแต่การป้องกันไว้ล่วงหน้านั้นไม่ใช่ผู้ที่เป็นผู้ก่อภยันตรายขึ้น ได้รับอันตราย และ

2.3 ได้กระทำเพื่อป้องกันสิทธิของตนหรือของผู้อื่นให้พ้นภยันตราย เพื่อป้องกันสิทธิแสดงว่ามีเจตนาพิเศษหรือมูลเหตุชักจูงใจ เพื่อป้องกันสิทธิของตนเองหรือผู้อื่นหากขาดเจตนาพิเศษอ้างป้องกันไม่ได้ และการป้องกันต้องเป็นการกระทำโดยเจตนาเท่านั้น หากกระทำโดยประมาทจะอ้างป้องกันไม่ได้ การป้องกันโดยชอบด้วยกฎหมาย อาจจะเป็นการป้องกันสิทธิของผู้อื่นได้ด้วย และยังไม่จำกัดว่าจะเป็นบุคคลธรรมดาหรือนิติบุคคล และการกระทำโดยป้องกันนั้นต้องกระทำต่อผู้ก่อภัยขึ้น แม้จะพลาดไปถูกบุคคลที่ 3 ก็ยังเป็นป้องกันอยู่ และ

2.4 การกระทำโดยป้องกันนั้นต้องไม่เกินขอบเขต ตามมาตรา 69 “ในกรณีที่บัญญัติไว้ในมาตรา 67 และมาตรา 68 นั้น ถ้าผู้กระทำได้กระทำไปเกินสมควรแก่เหตุ หรือเกินกว่ากรณีแห่งความจำเป็นหรือเกินกว่ากรณีแห่งการจำต้องกระทำเพื่อป้องกัน ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้แต่ถ้าการกระทำนั้นเกิดขึ้นจากความตื่นเต้นความตกใจหรือความกลัว ศาลจะไม่ลงโทษผู้กระทำก็ได้” จากบทบัญญัติในมาตรานี้ การกระทำที่ถือว่าไม่เกินขอบเขตนั้น สามารถพิจารณาได้จาก ทฤษฎี “วิถีทางน้อยที่สุด” คือผู้กระทำต้องกระทำไปพอสมควรแก่เหตุ โดยพิจารณาว่าผู้ป้องกันได้กระทำเพื่อป้องกันสิทธิของตนเองหรือของผู้อื่นให้พ้นจากภยันตรายนั้นด้วยวิธีทางที่น้อยที่สุดเท่าที่จำต้องกระทำ ตามที่หลักของมาตรา 68 ที่บัญญัติไว้ว่า “ผู้ใดจำต้องกระทำ...” การกระทำโดยป้องกันนั้นต้องกระทำโดยใช้มาตรการขั้นต่ำที่สุดในการกระทำเพื่อให้พ้นภัย และตามทฤษฎี “สัดส่วน” ผู้ป้องกันได้กระทำโดยป้องกันโดยได้สัดส่วนกับภยันตราย เช่น คำพิพากษาฎีกาที่ 606/2510 คนตายเข้ามาชกจำเลย จำเลยล้มลง คนตายเงื้อมีดจะแทง จำเลยยิงสวนทันที เป็นป้องกันพอสมควรแก่เหตุ

3. ผู้เสียหายยินยอมให้กระทำการนั้น หรือเป็นการปฏิบัติตามคำสั่งโดยชอบของเจ้าพนักงาน ในบางกรณีถ้าผู้เสียหายยินยอมให้กระทำการอย่างหนึ่งอย่างใดและยินยอมนั้นไม่ขัดต่อความสงบเรียบร้อยหรือศีลธรรมอันดีของประชาชน ผู้กระทำก็ไม่มีความผิดทางอาญา หรือถ้าบุคคลมีหน้าที่ที่ปฏิบัติตามคำสั่งโดยชอบของเจ้าพนักงาน บุคคลนั้นไม่ต้องรับผิดในทางอาญา ความยินยอมที่เป็นเหตุยกเว้นความผิดมาจากหลัก “ความยินยอมไม่ก่อให้เกิดความเสียหายหรือไม่ทำให้เป็นละเมิด” ซึ่งหลักการนี้ถูกนำมาใช้ยกเว้นความรับผิดทางอาญา โดยที่หลักความยินยอมนี้แม้ไม่ถูกบัญญัติไว้เป็นลายลักษณ์ แต่ก็ถูกนำมาใช้ในฐานะที่เป็นหลักกฎหมายทั่วไป เพราะเป็นคุณต่อผู้กระทำความผิด ดังนั้นแม้หลักความยินยอมจะไม่ได้บัญญัติไว้ในประมวลกฎหมายอาญาก็สามารถนำมาใช้ได้ โดยหลักการยินยอมประกอบด้วยองค์ประกอบ ดังนี้

3.1 ผู้ให้ความยินยอมจะต้องมีความสามารถที่จะให้ความยินยอมได้ หากเป็นบุคคลที่ไม่รู้ถึงการกระทำของตนเพราะขาดความสามารถ เช่น เด็ก หรือคนวิกลจริต ผู้กระทำความผิดก็อ้างความยินยอมของบุคคลดังกล่าวไม่ได้

3.2 ต้องเป็นความยินยอมที่เกิดขึ้นโดยสมัครใจ ปราศจากการบังคับ ขู่เข็ญหรือกระทำด้วยประการใด ๆ ก็ตามที่ทำให้ผู้เสียหายต้องยินยอมโดยไม่สมัครใจ

3.3 เวลาที่ให้ความยินยอม โดยหลักแล้วความยินยอมที่เป็นเหตุยกเว้นความรับผิดทางอาญาจะต้องให้ก่อนการกระทำความผิด ไม่ว่าความยินยอมนั้นจะไว้ล่วงหน้านานเท่าใดก็ตาม จนกว่าจะมีการถอนหรือเลิกให้ความยินยอม

3.4 ความยินยอมนั้นต้องไม่ขัดกับความสงบเรียบร้อยและศีลธรรมอันดีของประชาชน ความยินยอมอันเป็นเหตุยกเว้นความผิดนั้น หากเป็นความยินยอมที่ขัดต่อความสงบเรียบร้อย เช่น ยินยอมให้ผู้อื่นทำร้ายร่างกายเพื่อทดลองวิชาอาคม หรือความยินยอมที่ขัดต่อศีลธรรมอันดี เช่น ยินยอมให้ผู้อื่นทำให้ตนแท้งลูก คนที่ทำให้หญิงแท้งลูกแม้หญิงจะยอมให้ทำก็มีความผิดและไม่อาจอ้างความยินยอมให้พ้นจากความรับผิดไปได้

3.5 การให้ความยินยอมอาจกระทำได้หลายวิธี การให้ความยินยอมโดยชัดแจ้งหรืออาจให้ความยินยอมโดยปริยายก็ได้ เช่น ก่อนแพทย์ที่ทำการรักษาจะผ่าตัดจะให้ผู้ป่วยลงลายมือในหนังสือยินยอมให้ผ่าตัด ซึ่งทำให้การผ่าตัดของแพทย์นั้นไม่เป็นความผิดฐานทำร้ายร่างกาย เนื่องจากได้รับความยินยอมจากผู้ป่วย แต่ในกรณีที่ผู้ป่วยไม่สามารถให้ความยินยอมได้ก็จะขอความยินยอมจากญาติของผู้ป่วย หากไม่มีบุคคลที่ให้ความยินยอมได้เลย แต่มีความจำเป็นที่แพทย์ต้องทำการผ่าตัดเพื่อรักษาชีวิตของผู้ป่วยไว้ แพทย์ก็อ้างการกระทำด้วยความจำเป็นซึ่งเป็นเหตุยกเว้นโทษได้

2.3.3 โครงสร้างที่ 3 การไม่มีกฎหมายยกเว้นโทษ การกระทำซึ่งครบองค์ประกอบที่กฎหมายบัญญัติตามโครงสร้างที่ 1 และไม่มีกฎหมายยกเว้นความผิดตามโครงสร้างที่ 2 แล้ว ต่อไปพิจารณาว่าต้องไม่มีกฎหมายยกเว้นโทษให้แก่ผู้กระทำความผิด แต่ถ้ามีกฎหมายยกเว้นโทษแล้วก็ไม่ต้องรับโทษในทางอาญา กฎหมายที่ยกเว้นโทษมีหลายกรณี ได้แก่

1. การกระทำความผิดด้วยความจำเป็น ด้วยสาเหตุดังต่อไปนี้

1.1 อยู่ในที่บังคับหรือภายใต้อำนาจซึ่งไม่สามารถหลีกเลี่ยงขัดขืนได้ และได้กระทำไปพอสมควรแก่เหตุ ตามประมวลกฎหมายอาญา มาตรา 67 บัญญัติไว้มีใจความว่า “ผู้ใดกระทำความผิดด้วยความจำเป็นเพราะอยู่ในที่บังคับหรือ ภายใต้อำนาจซึ่งไม่สามารถหลีกเลี่ยงได้ ถ้าการกระทำนั้นไม่เกินสมควรแก่เหตุ ผู้นั้นไม่ต้องรับโทษ”

1.2 เพื่อให้พ้นจากภยันตรายที่ใกล้จะถึง และไม่สามารถหลีกเลี่ยงให้พ้นโดยวิธีอื่นได้ หากกระทำไปพอสมควรแก่เหตุบุคคลนั้นไม่ต้องได้รับโทษเช่นเดียวกัน

2. ความสามารถและความอ่อนอายุของผู้กระทำผิดที่ได้รับการยกเว้นโทษ ได้แก่

2.1 กระทำความผิดในขณะไม่สามารถรับผิดชอบหรือบังคับตนเองได้ เพราะมีจิตบกพร่อง โรคจิต หรือจิตฟั่นเฟือน บุคคลจิตไม่ปกติที่กระทำความผิดอาจไม่ต้องรับโทษหรือรับโทษน้อยลง เนื่องจากประมวลกฎหมายอาญา มาตรา 65 บัญญัติว่า “ผู้ใดกระทำความผิดในขณะไม่สามารถรู้ผิดชอบ หรือไม่สามารถบังคับตนเองได้ เพราะมีจิตบกพร่อง โรคจิตหรือจิตฟั่นเฟือน ผู้นั้นไม่ต้องรับโทษสำหรับความผิดนั้น แต่ถ้าผู้กระทำความผิดยังสามารถรู้ผิดชอบอยู่บ้าง หรือยังสามารถบังคับตนเองได้บ้าง ผู้นั้นต้องรับโทษสำหรับความผิดนั้น แต่ศาลจะโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดเพียงใดก็ได้”

2.2 เสพสุราหรือสิ่งมึนเมา เป็นกรณีที่ไม่รู้ว่าสิ่งนั้นจะทำให้มึนเมา หรือถูกขืนใจให้เสพและได้กระทำความผิดในขณะที่ไม่สามารถรู้ผิดชอบหรือบังคับตนเองได้เพราะความมึนเมา ประมวลกฎหมายอาญาบัญญัติไว้ในมาตรา 66 บัญญัติว่า "ความมึนเมาเพราะเสพสุราหรือสิ่งมึนเมาอย่างอื่นจะยกขึ้นเป็นข้อแก้ตัวตามมาตรา 65 ไม่ได้ เว้นแต่ความมึนเม่นั้นจะเกิดขึ้นโดยผู้เสพไม่รู้ว่าสิ่งนั้นจะทำให้มึนเมา หรือได้เสพโดยถูกขืนใจให้เสพ และได้กระทำความผิดในขณะไม่สามารถรู้ผิดชอบหรือไม่สามารถบังคับตนเองได้ ผู้กระทำความผิดจึงจะได้รับยกเว้นโทษสำหรับความผิดนั้น แต่ถ้าผู้นั้นยังสามารถรู้ผิดชอบอยู่บ้าง หรือยังสามารถบังคับตนเองได้บ้าง ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้"

2.3 เด็กอายุไม่เกินสิบสองปี โดยสภาพจิตใจถือว่ายังไม่อาจจะรู้ผิดชอบชั่วดี หมายความว่าเด็กอายุต่ำกว่า 12 ปี ได้กระทำความผิดทางอาญา ให้ถือว่าเด็กคนนั้นมีความผิด แต่ไม่ต้องรับโทษทางอาญา ตามประมวลกฎหมายอาญา มาตรา 73 วรรคแรก บัญญัติว่า “เด็กอายุยังไม่เกินสิบสองปี กระทำการอันกฎหมายบัญญัติเป็นความผิดเด็กนั้นไม่ต้องรับโทษ”

3. การกระทำตามคำสั่งที่มิชอบด้วยกฎหมาย และความเป็นสามีภริยา

ถ้าบุคคลมีหน้าที่ต้องปฏิบัติตามคำสั่งของเจ้าพนักงาน แม้คำสั่งนั้นจะมิชอบด้วยกฎหมาย ผู้นั้นก็ไม่ควรรับโทษจากการกระทำของตน เว้นแต่จะรู้ว่าคำสั่งนั้นมิชอบด้วยกฎหมาย ตามประมวลกฎหมายอาญา มาตรา 70 บัญญัติว่า “ผู้ใดกระทำตามคำสั่งของเจ้าพนักงาน แม้คำสั่งนั้นจะมิชอบด้วยกฎหมาย ถ้าผู้กระทำมีหน้าที่หรือเชื่อโดยสุจริตว่ามีหน้าที่ต้องปฏิบัติตาม ผู้นั้นไม่ต้องรับโทษ เว้นแต่จะรู้ว่าคำสั่งนั้นเป็นคำสั่งซึ่งมิชอบด้วยกฎหมาย” เนื่องจากผู้กระทำมีหน้าที่หรือเชื่อโดยสุจริตว่ามีหน้าที่ต้องปฏิบัติตาม และเพื่อไม่ให้ผู้อยู่ใต้บังคับบัญชาต้องเกิดความไม่แน่ใจในความถูกต้องของคำสั่งของผู้บังคับบัญชา อันจะเกิดผลเสียต่อการปฏิบัติงานราชการทั่ว ๆ ไปได้

สามีภริยากระทำความผิดต่อกันในความผิดบางประเภท ซึ่งไม่กระทบต่อความสงบเรียบร้อยหรือศีลธรรมอันดีของประชาชนควรได้รับการยกเว้นโทษ หมายความว่า สามีหรือภริยาเป็นผู้กระทำความผิดโดยลำพังผู้เดียว เป็นตัวการร่วมกันกับผู้อื่น เป็นผู้ก่อให้ผู้อื่นกระทำความผิดหรือเป็น

ผู้สนับสนุนผู้อื่นกระทำความผิด และภริยาหรือสามีเป็นผู้เสียหายโดยตรงในการกระทำความผิดนั้นได้รับยกเว้นโทษตามมาตรา 71

2.3 การใช้และการตีความกฎหมายอาญา [3]

2.2.1 การใช้กฎหมายอาญา ตามประมวลกฎหมายอาญา มาตรา 2 บัญญัติว่า “บุคคลจักต้องรับโทษทางอาญาต่อเมื่อได้กระทำการอันกฎหมายที่ใช้ในขณะกระทำนั้นบัญญัติเป็นความผิดและกำหนดโทษไว้ และโทษที่จะลงแก่ผู้กระทำความผิดนั้นต้องเป็นโทษที่บัญญัติไว้ในกฎหมาย

ถ้าตามบทบัญญัติของกฎหมายที่บัญญัติในภายหลัง การกระทำเช่นนั้นไม่เป็นความผิดต่อไป ให้ผู้ที่ได้กระทำการนั้นพ้นจากการเป็นผู้กระทำความผิด และถ้าได้มีคำพิพากษาถึงที่สุดให้ลงโทษแล้ว ก็ให้ถือว่าผู้นั้นไม่เคยต้องคำพิพากษาว่าได้กระทำความผิดนั้น ถ้ารับโทษอยู่ก็ให้ การลงโทษนั้นสิ้นสุดลง” จากมาตรา 2 นี้ คำว่ากฎหมาย หมายถึงเฉพาะกฎหมายที่มีโทษทางอาญาที่จะต้องนำมาลงโทษแก่ผู้กระทำผิด ไม่ได้หมายถึงกฎหมายอื่น สำหรับกฎหมายที่มีโทษทางอาญาจะบัญญัติถึงการกระทำที่เป็นองค์ประกอบของความผิดทั้งองค์ประกอบภายนอกและองค์ประกอบภายใน และระวางโทษที่จะลงแก่ผู้กระทำความผิดได้แก่ ประมวลกฎหมายอาญา และกฎหมายอื่นที่มีโทษทางอาญา เช่น พระราชบัญญัติยาเสพติดให้โทษ พระราชบัญญัติอาวุธปืน เครื่องปืนและวัตถุระเบิด เป็นต้น

ซึ่งหลักการใช้กฎหมายเพื่อการรับโทษทางอาญามีองค์ประกอบ ดังนี้

1. ต้องมีกฎหมายบัญญัติเป็นความผิด และกำหนดโทษ

2. ต้องบัญญัติกฎหมายนั้นในขณะกระทำความผิด เพราะกฎหมายอาญาไม่มีผลย้อนหลังเป็นโทษแก่ผู้กระทำความผิด แต่ย้อนหลังเพื่อเป็นคุณได้ หมายความเฉพาะถึงโทษตามประมวลกฎหมายอาญา มาตรา 18 เท่านั้น ซึ่งโทษที่ลงแก่ผู้กระทำความผิดได้แก่ ประหารชีวิต จำคุก กักขัง ปรับ และริบทรัพย์สิน แต่มาตรการทางแพ่ง เช่น การยึดทรัพย์สินให้ตกเป็นของแผ่นดินตามพระราชบัญญัติการฟอกเงิน ไม่ใช่โทษทางอาญามีผลย้อนหลังได้

3. ต้องตีความโดยเคร่งครัด โดยอ้างอิงตามคำพิพากษาฎีกาที่ 356/2509 (ประชุมใหญ่) วางหลักในการตีความกฎหมายอาญาว่า การตีความนั้นต้องตีความโดยเคร่งครัดจากมาตรา 2 คำว่ากฎหมาย หมายถึงเฉพาะกฎหมายที่มีโทษทางอาญาที่จะต้องนำมาลงโทษแก่ผู้กระทำผิด ไม่ได้หมายถึงกฎหมายอื่น สำหรับกฎหมายที่มีโทษทางอาญาจะบัญญัติถึงการกระทำที่เป็นองค์ประกอบของความผิดทั้งองค์ประกอบภายนอกและองค์ประกอบภายใน และระวางโทษที่จะลงแก่ผู้กระทำความผิดได้แก่ ประมวลกฎหมายอาญา และกฎหมายอื่นที่มีโทษทาง

กฎหมายอาญา คือกฎหมายที่บัญญัติว่า การกระทำหรือไม่กระทำอย่างใดเป็นความผิดและกำหนดโทษที่จะลงแก่ผู้กระทำความผิดไว้ด้วย กล่าวอีกนัยหนึ่ง กฎหมายอาญาคือกฎหมายที่บัญญัติ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ห้ามมิให้มีการกระทำอย่างหนึ่งอย่างใด หรือ บังคับให้มีการกระทำอย่างหนึ่งอย่างใด โดยผู้ที่ฝ่าฝืนหรือไม่ปฏิบัติตามจะต้องได้รับโทษ [2]

ลักษณะของการบัญญัติกฎหมายที่มีโทษทางอาญานั้น มี 2 ลักษณะได้แก่

1. บัญญัติองค์ประกอบของความผิดและระวางโทษที่จะลงแก่ผู้กระทำความผิดในมาตราเดียวกัน เช่นมาตรา 288 บัญญัติว่า “ผู้ใดฆ่าผู้อื่น ต้องระวางโทษประหารชีวิต จำคุกตลอดชีวิต หรือจำคุกตั้งแต่สิบห้าปีถึงยี่สิบปี” ส่วนแรกที่บัญญัติการกระทำที่เป็นองค์ประกอบภายนอกของความผิด ได้แก่ ผู้กระทำเป็นใครก็ได้ ผู้ถูกกระทำคือคนทั่วไปที่ไม่ทำให้เป็นเหตุฉกรรจ์ตามมาตรา 289 (บุคคลที่ถูกกระทำหากเป็น บุพการี หรือเจ้าหน้าที่ที่ปฏิบัติการตามหน้าที่จะต้องรับโทษตาม มาตรา 289) การกระทำคือฆ่า (ฆ่า หมายถึง การกระทำอันเป็นเหตุให้คนตาย หรือการกระทำให้บุคคลที่มีชีวิตอยู่ถึงแก่ความตาย กฎหมายไม่ได้จำกัดลักษณะของการกระทำในการฆ่าไว้ ผู้กระทำจะกระทำด้วยวิธีการใด ๆ ก็ได้ เช่น ใช้ปืนยิง ใช้มีดแทง ขับรถยนต์ชน กดให้จมน้ำ หรือบังคับให้ดื่มยาพิษ เป็นต้น รวมทั้งการทำร้ายทางจิตใจ เช่น แกล้งบอกข่าวร้ายแก่คนป่วยจวนจะตายให้ตกใจตาย) องค์ประกอบภายในคือ มีเจตนากระทำ หมายถึง ผู้กระทำต้องมีเจตนาตามมาตรา 59 วรรคสอง ที่จะทำให้ผู้ถูกฆ่าถึงแก่ความตาย โดยเหตุนี้ในตำราจึงเรียกความผิดนี้ว่า ความผิดฐานฆ่าคนตายโดยเจตนา เมื่อกระทำการที่เข้าองค์ประกอบภายนอกและภายในครบถ้วนแล้ว ถือว่าเป็นการกระทำที่เป็นความผิดตามความหมายของมาตรา 2 คือผิดตามประมวลกฎหมายอาญามาตรา 288 และส่วนที่สองคือการระวางโทษ ซึ่งระวางโทษประหารชีวิต จำคุกตลอดชีวิต หรือจำคุกตั้งแต่สิบห้าปีถึงยี่สิบปี

2. กฎหมายที่มีโทษทางอาญาที่บัญญัติส่วนที่เป็นบทห้ามและบทกำหนดโทษแยกคนละมาตรากัน ได้แก่ ความผิดตามพระราชบัญญัติต่าง ๆ ซึ่งจะบัญญัติส่วนที่เป็นบทห้ามไว้ส่วนหนึ่ง และบัญญัติส่วนที่เป็นบทกำหนดโทษไว้อีกส่วนหนึ่ง เช่น พระราชบัญญัติอาวุธปืน เครื่องกระสุน วัตถุระเบิด ดอกไม้เพลิง และสิ่งเทียมอาวุธปืน พ.ศ.2490 ในส่วนที่1 ว่าด้วยอาวุธปืนและเครื่องกระสุนปืนส่วนบุคคล มีมาตรา 7 บัญญัติว่า ห้ามมิให้ผู้ใดทำ ซื้อ มี ใช้ สั่ง หรือนำเข้า ซึ่งอาวุธหรือเครื่องกระสุนปืน เว้นแต่จะได้รับใบอนุญาตจากนายทะเบียนท้องที่ จะเห็นว่ามาตรา 7 ถือว่าเป็นบทห้าม คือห้ามบุคคลใดทำอาวุธปืนหรือเครื่องกระสุนปืน ฯลฯ โดยไม่ได้รับใบอนุญาตจะนายทะเบียนท้องที่ ผู้ที่ฝ่าฝืนมาตรา 7 จะมีโทษตามมาตรา 72 ซึ่งบัญญัติว่า ผู้ใดฝ่าฝืนมาตรา 7 ต้องระวางโทษจำคุกตั้งแต่หนึ่งถึงสิบปี และปรับตั้งแต่สองพันบาทถึงสองหมื่นบาท มาตรา 72 บัญญัติไว้ในหมวด 6 ว่าด้วยบทกำหนดโทษ ถือว่าเป็นบทกำหนดโทษสำหรับผู้ที่ฝ่าฝืนมาตรา 7

2.2.2 การตีความกฎหมายอาญาในประเทศไทย จะต้องตีความตามตัวอักษรและตามเจตนารมณ์ของกฎหมายนั้น และมีหลักการตีความหมายพิเศษเพิ่มขึ้นอีกนั่นคือ กฎหมายอาญาต้องตีความโดยเคร่งครัด เพื่อเป็นหลักประกันว่าประชาชนผู้อยู่ใต้การใช้บังคับกฎหมายจะได้รับความเป็นธรรม ดังนั้นกฎหมายอาญาจึงมีหลักเกณฑ์ในการตีความ ดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1. กฎหมายอาญาตองตีความโดยเครงครัด ประมวลกฎหมายอาญา มาตรา 2 ซึ่งแสดงให เห็นชัดวา กฎหมายอาญาต้องตีความโดยเครงครัดตามตัวอักษร กลาวอีกนัยหนึ่งก็คือ ตัวบทกฎหมาย บัญญัติไวเชนใด ต้องตีความเชนนั้น

2. กฎหมายอาญาจะตีความใหเปนการลงโทษหรือเพิ่มโทษผูกระทำใหหนักขึ้นไมได

3. กฎหมายอาญาในกรณีเปนที่สงสัยตองตีความใหเปนผลดีแกผูตองหา ในกรณีที่ บทบัญญัติที่มีโทษอาญาอาจตีความไดหลายนัย ศาลจะตองตีความตามนัยที่เปนผลดีแกจำเลย

4. กฎหมายอาญาตองตีความตามตัวอักษรและเจตนารมณ์ของกฎหมายควบคูกันไป

2.4 ผู้มีส่วนเกี่ยวข้องกับการกระทำผิดและการพิจารณาอัตราส่วนโทษ

2.4.1 ตัวการ หมายถึง บุคคลที่มีการกระทำร่วมกันและเจตนาร่วมกับบุคคลอื่นในการกระทำ ความผิดอาญา ตามประมวลกฎหมายอาญา มาตรา 83 บัญญัติว่า “ในกรณีความผิดใดเกิดขึ้นโดยการ กระทำของบุคคลตั้งแต่สองคนขึ้นไป ผู้ที่ได้ร่วมกระทำความผิดด้วยกันนั้นเป็นตัวการ ต้องระวางโทษ ตามที่กฎหมายกำหนดไว้สำหรับความผิดนั้น” ดังนั้นการกระทำร่วมกันอาจเป็นกรณีบุคคลตั้งแต่ 2 คนขึ้นไป ร่วมกระทำส่วนใดส่วนหนึ่งอันเป็นองค์ประกอบความผิด หรือเป็นการแบ่งหน้าที่กันทำ หรือ ร่วมอยู่ในที่เกิดเหตุในลักษณะที่สามารถเข้าช่วยเหลือผู้กระทำผิดคนอื่นทันทีก็ได้ และมีเจตนาร่วมกัน หมายถึงผู้กระทำผิดทุกคนได้รู้ถึงการกระทำของกันและกันและต่างถือเอาการกระทำของคนอื่นเป็น การกระทำของตนเอง กฎหมายกำหนดให้ตัวการต้องรับโทษสำหรับความผิดนั้น

2.4.2 ผู้ใช้ ตามประมวลกฎหมายอาญา มาตรา 84 บัญญัติว่า มาตรา 84 บัญญัติว่า "ผู้ใดก่อให้ ผู้อื่นกระทำความผิดไม่ว่าด้วยการใช้ บังคับ ขู่เข็ญ จ้าง วานหรือยุยงส่งเสริม หรือด้วยวิธีอื่นใด ผู้นั้น เป็นผู้ใช้ให้กระทำความผิด ถ้าผู้ถูกใช้ได้กระทำความผิดนั้นผู้ใช้ต้องรับโทษเสมือนเป็นตัวการ ถ้า ความผิดมิได้กระทำลง ไม่ว่าจะเป็นเพราะผู้ถูกใช้ไม่ยอมกระทำ ยังไม่ได้กระทำหรือเหตุอื่นใด ผู้ใช้ต้อง ระวางโทษเพียงหนึ่งในสามของโทษที่กำหนดไว้สำหรับความผิดนั้น” ดังนั้น ผู้ใช้ จึงหมายถึงบุคคลที่ ทำให้บุคคลอื่นซึ่งยังไม่มีเจตนาที่จะกระทำความผิด ตกลงใจที่จะกระทำความผิดนั้น โดยผู้ใช้จะต้องมี เจตนาก่อให้ผู้อื่นกระทำผิดด้วย

2.4.3 ผู้สนับสนุน มาตรา 86 บัญญัติว่า "ผู้ใดกระทำด้วยประการใดๆ อันเป็นการช่วยเหลือ หรือให้ความสะดวกในการที่ผู้อื่นกระทำความผิดก่อนหรือขณะกระทำความผิด แม้ผู้กระทำความผิด จะมิได้รู้ถึงการช่วยเหลือหรือให้ความสะดวกนั้นก็ตาม ผู้นั้นเป็นผู้สนับสนุนการกระทำความผิด ต้อง ระวางโทษสองในสามส่วนของโทษที่กำหนดไว้สำหรับความผิดที่สนับสนุนนั้น" การกระทำอันเป็นการ ช่วยเหลือให้ความสะดวกนั้น จะเป็นด้วยวิธีใดก็ได้ แต่จะต้องมีขึ้นก่อน และไม่ว่าผู้กระทำความผิดจะรู้ ถึงการช่วยเหลือนั้นหรือไม่ก็ตามไม่เป็นข้อสำคัญ และแม้ความผิดจะไม่สำเร็จก็เป็นการสนับสนุนให้

พยายามได้ ข้อสำคัญคือ ผู้กระทำความผิดจะต้องเข้ารับความช่วยเหลือนั้น มิฉะนั้นแล้วการสนับสนุนก็ไม่เกิด

ในงานวิจัยนี้สถานะของผู้กระทำความผิดแบ่งตามประเภทของความรับผิดทางอาญา เพื่อให้ตรงตามวัตถุประสงค์ของงานวิจัย ได้ดังนี้

1) กระทำด้วยตนเอง ตัวการร่วม และผู้ใช้ รับโทษทางอาญาเท่ากัน ตามฐานความผิดนั้น ๆ แต่สำหรับผู้ใช้ถ้าผู้ถูกใช้ไม่ได้กระทำความผิดให้รับโทษเพียงหนึ่งในสาม

2) ผู้กระทำอายุไม่เกินสิบสองปี มึนเมาเพราะถูกบังคับ หรือจิตฟั่นเฟือน ที่ได้รับการยกเว้นโทษ

3) ผู้สนับสนุนให้ผู้อื่นกระทำความผิด รับโทษสองในสามของฐานความผิดนั้น ๆ

2.4.4 การพยายามกระทำความผิด กฎหมายอาญาจะถือว่าจุดเริ่มต้นของความผิดคือได้ลงมือกระทำ เว้นแต่ในบางฐานความผิดร้ายแรงเพียงอยู่ในขั้นตอนการวางแผน หรือเตรียมการก็รับโทษทางอาญาแล้ว เช่น เรื่องการวางเพลิงเผาทรัพย์ ประมวลกฎหมายอาญา มาตรา 217 ประกอบมาตรา 219 ในกรณีที่ความผิดไม่สำเร็จผู้กระทำรับโทษ “พยายามกระทำความผิด ตามฐานความผิดนั้น ๆ” โดยสามารถแบ่งประเภทของการพยายามกระทำความผิด ได้ดังนี้

1. การพยายามกระทำความผิดในกรณีทั่วไป

ประมวลกฎหมายอาญา มาตรา 80 วรรคแรก บัญญัติว่า “ผู้ใดลงมือกระทำความผิดแต่กระทำไปไม่ตลอด หรือกระทำไปตลอดแล้วแต่การกระทำนั้นไม่บรรลุผล ผู้นั้นพยายามกระทำความผิด” การจะเป็นการพยายามกระทำความผิดตามบทบัญญัติมาตรา 80 มีองค์ประกอบดังนี้

1) ผู้กระทำผิดต้องมีเจตนา ตามความหมายในโครงสร้างความรับผิด และ

2) ดำเนินการถึงขั้น “ลงมือ” กระทำความผิด โดยใช้หลักความใกล้ชิดต่อผล เช่น แนวการวินิจฉัยของศาลไทยเกี่ยวกับการใช้อาวุธปืน “ลงมือ” ต่อเมื่อได้มีการจ้องปืนเล็งไปยังผู้เสียหายด้วยเจตนาฆ่า ไม่จำเป็นต้องถึงขั้นนก หรือลั่นไกปืน และ

3) การกระทำไม่บรรลุผล โดยกระทำไปไม่ตลอด คือไม่สามารถจะกระทำครบถ้วนตามที่ประสงค์ได้ด้วยสาเหตุอื่น ๆ ที่ไม่ใช่ความสมัครใจ หรือกระทำไปตลอดแล้วแต่ไม่บรรลุผลเพราะวิธีการที่ใช้กระทำผิดหรือเหตุแห่งวัตถุที่มุ่งกระทำต่อการพยายามกระทำผิดตามมาตรา 80 กฎหมายกำหนดโทษไว้ว่าให้รับโทษ 2 ใน 3 ส่วน ของโทษที่กฎหมายกำหนดไว้สำหรับความผิดนั้น

2. การพยายามกระทำความผิดที่เป็นไปไม่ได้อย่างแน่แท้

ประมวลกฎหมายอาญา มาตรา 81 บัญญัติว่า “ผู้ใดกระทำการโดยมุ่งต่อผลซึ่งกฎหมายบัญญัติเป็นความผิด แต่การกระทำนั้นไม่สามารถจะบรรลุผลได้อย่างแน่แท้ เพราะเหตุปัจจัยซึ่งใช้ในการกระทำหรือเหตุแห่งวัตถุที่มุ่งหมายกระทำต่อ ให้ถือว่าผู้นั้นพยายามกระทำความผิด แต่ให้ลงโทษไม่เกินกึ่งหนึ่งของโทษที่กฎหมายกำหนดไว้สำหรับความผิดนั้น ถ้าการกระทำดังกล่าวในวรรค

แรกได้กระทำไปโดยความเชื่ออย่างงมงาย ศาลจะไม่ลงโทษก็ได้" การพยายามกระทำความผิดตามมาตรา 81 และมาตรา 80 นั้น สิ่งที่เหมือนกัน 2 ประการ ได้แก่ ผู้กระทำมีเจตนากระทำความผิดและความผิดได้ลงมือกระทำไปแล้ว แต่ความแตกต่างคือ การไม่บรรลุผลตามมาตรา 80 เกิดขึ้นเพราะเหตุบังเอิญ และจากปัจจัยซึ่งใช้ในการกระทำหรือวัตถุที่มุ่งหมายกระทำต่อ หรือเพราะเหตุอื่นๆ ก็ได้ ส่วนการไม่บรรลุผลตามมาตรา 81 นั้น เกิดขึ้นอย่างแน่แท้เด็ดขาด และเกิดเพราะเหตุปัจจัยซึ่งใช้ในการกระทำ หรือวัตถุที่มุ่งหมายกระทำต่อเท่านั้น เช่น คำพิพากษาศาลฎีกาที่ 1188/2536 จำเลยใช้อาวุธลูกซองสั้นยิงผู้เสียหาย 1 นัด ขณะจำเลยอยู่ห่างผู้เสียหายประมาณ 1 เมตร มีเพียงเขม่าดินปืนไปกระทบที่คางกับคอของผู้เสียหายบาดแผลมีเลือดซึมเล็กน้อย รักษา 7 วันหาย กระสุนปืนที่จำเลยใช้ยิงเป็นกระสุนที่ไม่มีเม็ดกระสุนบรรจุ ย่อมแสดงว่ากระสุนปืนดังกล่าวไม่อาจทำให้ผู้เสียหายถึงแก่ความตายได้โดยแน่แท้ การกระทำของจำเลยจึงมีความผิด ตามประมวลกฎหมายอาญา มาตรา 288 ประกอบมาตรา 81 วรรคแรก

มาตรา 81 กฎหมายกำหนดโทษไว้ว่าให้รับโทษไม่เกินกึ่งหนึ่งของโทษที่กฎหมายกำหนดไว้สำหรับความผิดนั้น เว้นแต่การกระทำได้ทำไปเพราะความเชื่อ งมงาย อาจไม่ต้องรับโทษก็ได้

3. การพยายามกระทำความผิด เพราะการยับยั้งกลับใจ

ประมวลกฎหมายอาญา มาตรา 82 บัญญัติว่า "ผู้ใดพยายามกระทำความผิด หากยับยั้งเสียเองไม่กระทำการให้ตลอด หรือกลับใจแก้ไขไม่ให้การกระทำนั้นบรรลุผล ผู้นั้นไม่ต้องรับโทษสำหรับการพยายามกระทำความผิดนั้น แต่ถ้าการที่ได้กระทำไปแล้วต้องตามบทกฎหมายที่บัญญัติเป็นความผิด ผู้นั้นต้องรับโทษสำหรับความผิดนั้น ๆ" การที่ผู้กระทำจะได้รับการยกเว้นโทษในความผิดฐานพยายามได้นั้น ต้องเป็นการที่ได้ยับยั้งโดยเจตนาและสมัครใจไม่ให้การกระทำนั้นบรรลุผล แต่ถ้าได้กระทำความผิดไปแล้วยังคงต้องรับโทษสำหรับการกระทำความผิดนั้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.5 เหตุลดโทษและเหตุรอการลงโทษ [4]

เหตุลดโทษ คือ เหตุที่อาจทำให้ผู้กระทำได้รับโทษน้อยลง ซึ่งขึ้นอยู่กับดุลพินิจของศาลที่จะลดโทษให้แก่ผู้กระทำหรือไม่ก็ได้ กฎหมายมักจะบัญญัติเกี่ยวกับการลดโทษไว้ในทำนองว่า "ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้ หรือบางกรณีก็จำกัดขอบเขตการลดโทษของศาลไว้ เช่น ถ้าศาลเห็นสมควรจะลดโทษไม่เกินกึ่งหนึ่งของโทษที่จะลงแก่ผู้กระทำความผิดนั้นก็ได้" โดยมีเหตุลดโทษ ดังต่อไปนี้

2.5.1 การกระทำความผิดโดยบันดาลโทสะ

ประมวลกฎหมายอาญา มาตรา 72 ได้บัญญัติว่า "ผู้ใดบันดาลโทสะ โดยถูกข่มเหงอย่างร้ายแรงด้วยเหตุอันไม่เป็นธรรม จึงกระทำความผิดต่อผู้ข่มเหงในขณะนั้น ศาลจะลงโทษผู้นั้นน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้" โดยหลักเกณฑ์ของการที่จะใช้เหตุบันดาล

โทสะได้ เพื่อได้รับการลดโทษหรือละเว้นโทษนั้น ประกอบด้วย 3องค์ประกอบได้แก่ การถูกข่มเหงอย่างร้ายแรงด้วยเหตุอันไม่เป็นธรรม การที่ถูกข่มเหงเช่นนั้นเป็นเหตุให้ผู้กระทำบันดาลโทสะ และผู้กระทำได้กระทำความผิดต่อผู้ข่มเหงในขณะบันดาลโทสะ ผลของการกระทำโดยบันดาลโทสะ ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้เพียงใดก็ได้ โดยไม่ต้องคำนึงถึงโทษขั้นต่ำ แต่จะไม่ลงโทษเลยไม่ได้

2.5.2 เหตุลดโทษอื่น เหตุลดโทษที่กฎหมายบัญญัติไว้มีหลายกรณี ได้แก่

1. ความไม่รู้กฎหมาย มาตรา 64 บัญญัติไว้ว่า “บุคคลจะแก้ตัวว่าไม่รู้กฎหมาย เพื่อให้พ้นจากความรับผิดในทางอาญาไม่ได้ แต่ถ้าศาลเห็นว่าตามสภาพและพฤติการณ์ ผู้กระทำความผิดอาจจะไม่รู้ว่ากฎหมายบัญญัติว่าการกระทำนั้นเป็นความผิด ศาลอาจอนุญาตให้แสดงพยานหลักฐานต่อศาล และถ้าศาลเชื่อว่า ผู้กระทำไม่รู้ว่ากฎหมายบัญญัติไว้เช่นนั้น ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้” หลักของมาตรานี้ คือ บุคคลจะแก้ตัวว่าไม่รู้กฎหมายเพื่อไม่ต้องรับผิดทางอาญาไม่ได้ ยกเว้น ยอมให้แก้ตัวได้ ถ้าศาลเห็นว่าตามสภาพของความผิดซึ่งหมายความว่ากรณีที่เป็นความผิดเพราะกฎหมายห้าม มิใช่ความผิดในตัวเองและพฤติการณ์ ศาลอาจอนุญาตให้แสดงพยานหลักฐาน ถ้าศาลเชื่อความไม่รู้กฎหมาย ศาลอาจลดโทษให้แก่ผู้กระทำความผิดก็ได้ หรือไม่ลดโทษก็ได้ แต่ศาลจะไม่ลงโทษเลยไม่ได้

2. การกระทำความผิดเกี่ยวกับทรัพย์บางฐานระหว่างญาติสนิท มาตรา 71 วรรค 2 บัญญัติว่า “ความผิดดังระบุมานี้ ถ้าเป็นการกระทำที่ผู้บุพการีกระทำต่อผู้สืบสันดาน ผู้สืบสันดานกระทำต่อผู้บุพการี หรือพี่หรือน้องร่วมบิดามารดาเดียวกันกระทำต่อกัน แม้กฎหมายมิได้บัญญัติให้เป็นความผิดอันยอมความได้ ก็ให้เป็นความผิดอันยอมความได้ และนอกจากนั้น ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้” ความผิด ดังระบุในมาตรานี้ ได้แก่ ความผิดตามที่บัญญัติไว้ในมาตรา 334 ถึงมาตรา 336 วรรคแรก และมาตรา 341 ถึงมาตรา 364 เช่น ความผิดเกี่ยวกับการลักทรัพย์ในมาตรา 334

3. ความอ่อนอายุ มาตรา 75 บัญญัติว่า “ผู้ใดอายุกว่า 15 ปี แต่ยังไม่เกิน 18 ปี กระทำการอันกฎหมายบัญญัติเป็นความผิด ให้ศาลพิจารณาถึงความรู้ผิดชอบและสิ่งอื่นทั้งปวงเกี่ยวกับผู้นั้นในอันที่จะควรวินิจฉัยว่าสมควรพิพากษาลงโทษผู้นั้นหรือไม่ ถ้าศาลเห็นว่าไม่สมควรพิพากษาลงโทษก็ให้จัดการตามมาตรา 74 หรือศาลเห็นว่าสมควรพิพากษาลงโทษ ก็ให้ลดมาตราส่วนโทษที่กำหนดไว้สาหรับความผิดลงกึ่งหนึ่ง” มาตรา 76 บัญญัติว่า “ผู้ใดอายุตั้งแต่สิบแปดปีแต่ยังไม่เกินยี่สิบปี กระทำการอันกฎหมายบัญญัติเป็นความผิด ถ้าศาลเห็นสมควรจะลดมาตราส่วนโทษที่กำหนดไว้สำหรับความผิดนั้นลงหนึ่งในสามหรือกึ่งหนึ่งก็ได้

4. เหตุบรรเทาโทษ กฎหมายให้เป็นดุลพินิจของศาลในการพิจารณาลดโทษที่จะลงแก่จำเลยได้อีกไม่เกินกึ่งหนึ่งของโทษที่จะลงนั้น ไม่ว่าคดีนั้นจะได้มีการเพิ่มโทษหรือลดโทษด้วยเหตุอื่น ๆ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตามประมวลกฎหมายอาญาหรือตามกฎหมายอื่นใดมาแล้วหรือไม่ก็ตาม เหตุบรรเทาโทษตามมาตร 78 มีหลายประการ มาตรา 78 วรรคแรก บัญญัติว่า “เมื่อปรากฏว่ามีเหตุบรรเทาโทษ ไม่ว่าจะได้มีการเพิ่มหรือการลดโทษตามบทบัญญัติแห่งประมวลกฎหมายนี้หรือกฎหมายอื่นแล้วหรือไม่ ถ้าศาลเห็นสมควรจะลดโทษไม่เกินกึ่งหนึ่งของโทษที่จะลงแก่ผู้กระทำความผิดนั้นก็ได้” การจะเข้าเงื่อนไขมีเหตุบรรเทาโทษ ได้พิจารณาจากพฤติกรรมของผู้กระทำผิดก่อนการกระทำผิด เช่น เคยมีคุณงามความดีเป็นที่ปรากฏมาก่อน ถูกหลอกให้กระทำผิดเพราะความโง่เขลาหรือรู้เท่าไม่ถึงการณ์ รวมทั้งมีความทุกข์ยากแสนสาหัสจำต้องกระทำความผิด และพฤติกรรมหลังการกระทำผิด เช่น การรู้สึกผิดต่อการกระทำที่เกิดขึ้นแล้วพยายามบรรเทาผลร้ายนั้น หรือการให้ความรู้แก่ศาลในการพิจารณาคดี เช่น ให้การรับสารภาพต่อศาล เว้นแต่การจำนนต่อพยานหลักฐาน การสารภาพไม่ถือเป็นเหตุบรรเทาโทษ การลดโทษเพราะมีเหตุบรรเทาโทษตามมาตรา 78 นี้ เป็นดุลยพินิจของศาล ดังนั้นแม้มีเหตุบรรเทาโทษศาลอาจไม่ลดโทษให้ก็ได้

ในงานวิจัยนี้การกระทำผิดอาญาที่จะได้รับการลดโทษ หรือมีเหตุลดโทษตามการจำแนกกลุ่มการลงโทษนั้น คือการที่กระทำผิดด้วยความจำเป็นตามมาตรา 72 และการป้องกันเกินสมควรกว่าเหตุ หรือเกินกรณีแห่งความจำเป็น ตามมาตรา 69 และนอกจากนี้เหตุลดโทษตามความหมายในงานวิจัยนี้ คือการไม่ได้รับอัตราโทษตามที่กฎหมายกำหนดในฐานความผิดนั้น ๆ เช่น การรับโทษสองในสามส่วนของการเป็นผู้สนับสนุนเมื่อความผิดนั้นสำเร็จ หรือการรับโทษหนึ่งในสามส่วนของผู้ใช้เมื่อผู้ถูกใช้ไม่ได้กระทำความผิดนั้น และการให้รับโทษไม่เกินกึ่งหนึ่งของฐานความผิดนั้น สำหรับการกระทำผิดที่ไม่สามารถบรรลุผลได้อย่างแน่แท้ เพราะเหตุลดโทษที่กล่าวมาทั้งหมดอยู่ในบทบัญญัติของกฎหมาย แต่เหตุลดโทษเพราะความอ่อนอายุของผู้กระทำผิดตามมาตรา 75 และมาตรา 76 หรือเพราะมีเหตุบรรเทาโทษตามมาตรา 78 นั้น ไม่จัดอยู่ในกลุ่มที่มีการลดโทษ เพราะเป็นการใช้ดุลยพินิจของศาล โดยมุ่งเน้นฐานความผิดที่เกี่ยวกับชีวิตโดยเจตนา ในมาตรา 288 และ 289 ทำให้เหตุลดโทษในเรื่องความไม่รู้กฎหมาย และความผิดเกี่ยวกับทรัพย์บางฐานระหว่างญาติสนิทไม่ได้มีการพิจารณา

2.6 ศาลฎีกาและคำพิพากษาศาลฎีกา [5]

ปัจจุบันประเทศไทยมีการปกครองระบอบประชาธิปไตย อันมีพระมหากษัตริย์ทรงเป็นประมุขและรับหลักการแบ่งแยกการใช้อำนาจอธิปไตยมาใช้อย่างเป็นรูปธรรม โดยให้ประชาชนเป็นเจ้าของอำนาจอธิปไตย ตามที่รัฐธรรมนูญแห่งราชอาณาจักรไทยพุทธศักราช 2560 มาตรา 3 บัญญัติว่า “อำนาจอธิปไตยเป็นของปวงชนชาวไทย พระมหากษัตริย์ผู้ทรงเป็นประมุขทรงใช้อำนาจนั้นทางรัฐสภา คณะรัฐมนตรี และศาล ตามบทบัญญัติแห่งรัฐธรรมนูญ” ประเทศไทยใช้ระบบศาลคู่ในการอำนวยความยุติธรรมให้กับประชาชน โดยมีศาลยุติธรรมเป็นผู้มีอำนาจพิจารณาพิพากษาคดีทั้งปวง

เว้นแต่คดีที่รัฐธรรมนูญหรือกฎหมายบัญญัติให้อยู่ในอำนาจของศาลอื่น ศาลยุติธรรมมี 3 ชั้นศาล คือ ศาลชั้นต้น ศาลอุทธรณ์ และศาลฎีกา

ศาลฎีกาเป็นศาลยุติธรรมสูงสุด มีเขตอำนาจทั่วราชอาณาจักรมีอำนาจพิจารณาพิพากษาบรรดาคดีที่อุทธรณ์ คำพิพากษาหรือคำสั่งของศาลอุทธรณ์และศาลอุทธรณ์ภาค ภายใต้เงื่อนไขของกฎหมายว่าด้วยการฎีกา และมีอำนาจพิจารณาพิพากษาคดีที่อุทธรณ์คำพิพากษาหรือคำสั่งของศาลชั้นต้นโดยตรงต่อศาลฎีกาตามกฎหมายเฉพาะ และคดีที่รัฐธรรมนูญหรือกฎหมายอื่นบัญญัติให้ศาลฎีกามีอำนาจพิจารณาพิพากษา รวมทั้งมีอำนาจวินิจฉัยชี้ขาดหรือสั่งคำร้องคำขอที่ยื่นต่อศาลฎีกาตามกฎหมาย คำสั่งหรือคำพิพากษาของศาลฎีกาเป็นที่สุด พระธรรมนูญศาลยุติธรรม มาตรา 23 วรรคหนึ่ง (ฉบับที่ 3) พ.ศ. 2551 ได้กำหนดเขตอำนาจศาลไว้ว่า "ศาลฎีกามีอำนาจพิจารณาพิพากษาคดีที่รัฐธรรมนูญหรือกฎหมายบัญญัติให้เสนอต่อศาลฎีกาได้โดยตรง และคดีที่อุทธรณ์หรือฎีกาคำพิพากษาหรือคำสั่งของศาลชั้นต้น ศาลอุทธรณ์ หรือศาลอุทธรณ์ภาคตามที่กฎหมายบัญญัติ เว้นแต่กรณีที่ศาลฎีกาเห็นว่าข้อกฎหมายหรือข้อเท็จจริงที่อุทธรณ์หรือฎีกานั้นจะไม่เป็นสาระอันควรแก่การพิจารณา ศาลฎีกามีอำนาจไม่รับคดีไว้พิจารณาพิพากษาได้ ทั้งนี้ตามระเบียบที่ที่ประชุมใหญ่ศาลฎีกากำหนดโดยประกาศในราชกิจจานุเบกษา" ประเทศไทยใช้ระบบศาลสามชั้น โดยศาลฎีกาเป็นศาลยุติธรรมสูงสุด มีหน้าที่ควบคุมดูแลการใช้ดุลพินิจของศาลล่าง (ศาลชั้นต้น และศาลชั้นอุทธรณ์) วินิจฉัยปัญหาข้อกฎหมายที่สำคัญ รวมทั้งมีอำนาจพิจารณาคดีใหม่ และมีคำพิพากษาเป็นของตนเอง

จะเห็นว่าคำพิพากษาของศาลฎีกามีความสำคัญต่อการวินิจฉัยคดี แม้ว่าประเทศไทยจะไม่ได้ใช้หลักกฎหมายจารีตประเพณี (Common Law) ที่จะถือเอาคำพิพากษาของศาลเป็นที่มาของกฎหมายและถือเป็นบรรทัดฐานที่ศาลต้องยึดถือ เมื่อมีกรณีอย่างเดียวกันเกิดขึ้นก็ตาม แต่คำพิพากษาของศาลฎีกามีสภาพเด็ดขาดทางกฎหมายในการวางหลักเกณฑ์การวินิจฉัยทั้งประเด็นข้อกฎหมายและข้อเท็จจริง ไม่อาจขอให้แก้ไขหรือเปลี่ยนแปลงคำชี้ขาดในคำพิพากษานั้นได้อีกต่อไป นอกจากนี้หลักกฎหมายบางอย่างอาจเกิดจากแนวทางการตัดสินของศาลฎีกา เพราะในประมวลกฎหมายอาญาไม่ได้ระบุไว้ เช่น การใช้หลักกรรมเป็นเครื่องชี้เจตนา เพื่อวินิจฉัยผู้กระทำผิดว่ามีเจตนาทำร้ายหรือเจตนาฆ่า โดยสามารถพิจารณาจากอาวุธที่ใช้ อวัยวะที่ถูกกระทำ ลักษณะของบาดแผล และจากพฤติการณ์อื่น ๆ [2] ดังนั้นการนำคำพิพากษาของศาลฎีกาที่ได้มีการวางหลักเกณฑ์และทำคำตัดสินไว้แล้วมาเป็นแนวทางทางในการพิจารณาข้อเท็จจริงที่เกิดขึ้นร่วมกับการใช้ประมวลกฎหมายอย่างเคร่งครัดจึงเป็นสิ่งสำคัญในการกำหนดแนวทางการตัดสินคดีได้อย่างถูกต้อง

ในงานวิจัยนี้ใช้แนวทางการพิจารณาคดีของศาลฎีกา [5] ในการเป็นข้อมูลฝึกสอน (Training set) และทดสอบประสิทธิภาพของงแบบจำลองด้วยการใช้ข่าวอาชญากรรมออนไลน์ [6]

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

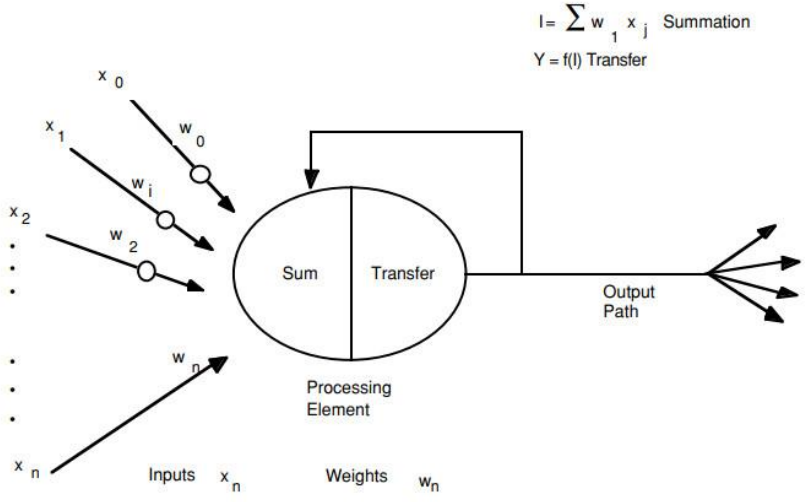
2.7 แบบจำลองโครงข่ายประสาทเทียม (Artificial Neural Networks : ANNs)

โครงข่ายประสาทเทียม (Artificial Neural Networks, ANNs) เป็นเทคนิคการเรียนรู้หนึ่งในศาสตร์การเรียนรู้ของเครื่อง (Machine learning) หรือที่มักจะเรียกสั้น ๆ ว่า โครงข่ายประสาท (Neural Networks) โดยการพยายามเลียนแบบการทำงานตามระบบประสาทของมนุษย์ ในการแก้ปัญหาต่าง ๆ โดยการเรียนรู้ของแบบจำลอง ANNs แบ่งได้เป็น 2 รูปแบบ [7] ได้แก่

1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นการใช้รูปแบบ (Pattern) ของชุดข้อมูลตัวแปรขาเข้า (Input) และผลลัพธ์ (Output) สอนโครงข่ายให้เรียนรู้ความสัมพันธ์ระหว่างตัวแปรขาเข้า และตัวแปรผลลัพธ์ โดยแบบจำลองจะปรับค่าถ่วงน้ำหนักที่เชื่อมโยงเพื่อลดความผิดพลาดในการทำนายค่าผลลัพธ์

2) การเรียนรู้แบบไม่มีการสอน (Unsupervised Learning) จะไม่มีการตรวจสอบ คำตอบว่าถูกหรือผิด ซึ่งโครงข่าย ANNs จะจัดเรียงโครงสร้างด้วยตัวเองตามลักษณะของข้อมูล โดยจะสร้างค่าถ่วงน้ำหนัก (Weight) ที่เชื่อมโยงตัวแปรขาเข้า ข้อมูลใหม่มีลักษณะควรจะจัดเข้ากลุ่มใด หลังจากนั้นจึงมีการปรับคุณลักษณะของกลุ่ม โดยการนำลักษณะของข้อมูลใหม่มาช่วยกำหนดแนวทางการจัดด้วย ในการตัดสินว่าข้อมูลผลลัพธ์ที่ได้นี้ควรจัดรวมเข้ากลุ่มใด

การทำงานของโครงข่ายประสาทเทียมคือ เมื่อมีข้อมูลขาเข้าหลายค่าเข้ามายังโครงข่าย แทนด้วยสัญลักษณ์ $X(n)$ จากนั้นนำแต่ละค่าคูณกับค่าถ่วงน้ำหนักของแต่ละขา ซึ่งแทนด้วยสัญลักษณ์ $W(n)$ แล้วนำค่าที่ได้รวมกันและส่งเข้าสู่ฟังก์ชันส่งผ่าน (Transfer Function) เพื่อหาคำตอบในชั้นผลลัพธ์ต่อไป ตามที่แสดงในรูปที่ 2-1



รูปที่ 2-1 การทำงานพื้นฐานของระบบประสาทเทียม [7]

ชนิดของโครงข่ายประสาทเทียมแบ่งตามลักษณะการเรียนรู้ได้เป็นออกเป็นหลายประเภท แต่ในงานวิจัยนี้แบ่งได้เป็น 2 ประเภทที่เกี่ยวข้อง ได้แก่

2.7.1. โครงข่ายประสาทแบบป้อนไปข้างหน้า (Feedforward neural network) คือโครงข่ายที่มีการไหลเวียนข้อมูลจากชั้นข้อมูลขาเข้าไปถึงชั้นข้อมูลผลลัพธ์ โดยไม่มีการย้อนกลับไปยังชั้นก่อนหน้า เพราะผลของข้อมูลในแต่ละชั้นไม่มีการคำนวณซ้ำ แบ่งออกได้ทั้งแบบชั้นเดียวและหลายชั้น ดังนี้

1. โครงข่ายประสาทเทียมชั้นเดียว (Single Layer Perceptron , SLP) [8] ประกอบด้วยหน่วยที่ย่อยที่สุดเรียกว่า เพอร์เซปตรอน (Perceptron) มีลักษณะโครงสร้างเป็นเซลล์ประสาทจำลองเซลล์หนึ่งที่รับสัญญาณจำนวนหนึ่งมาจากชั้นข้อมูลขาเข้า แล้วพิจารณาว่าสัญญาณทั้งหมดรวมแล้วเกินค่าที่เรียกว่าค่าขีดแบ่ง (Threshold) หรือไม่ โดยปกติจะถูกเขียนในรูปของค่าไบแอส (Bias) หรือค่าถ่วงน้ำหนักพิเศษ ถ้าเกินก็จะทำการส่งสัญญาณต่อไป โครงสร้างและการทำงานตามรูปที่ 2-2 แต่เนื่องจากข้อมูลต่าง ๆ ที่ป้อนเข้าไปจะมีค่าถ่วงน้ำหนักไม่เท่ากัน จึงต้องทำการคูณด้วยค่าถ่วงน้ำหนักก่อนจึงค่อยนำมาบวกกัน โดยสามารถสรุปขั้นตอนการทำงานของเพอร์เซปตรอน ได้ดังนี้

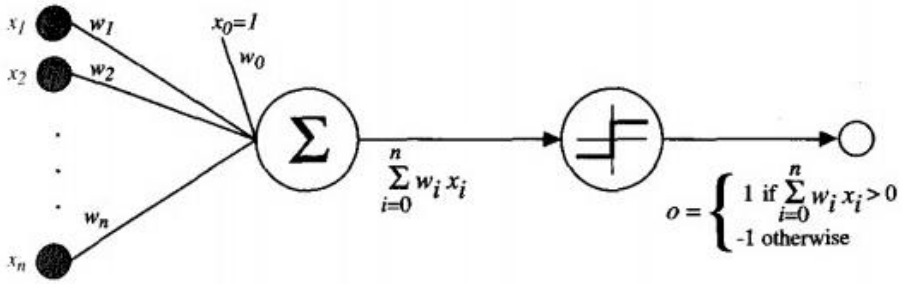
1) เพอร์เซปตรอนรับข้อมูลขาเข้าเป็นตัวเลขจำนวน n ค่า ตั้งแต่ $x_1, x_2, \ldots, x_n$

2) นำค่าจาก 1) แต่ละค่าไปคูณด้วยค่าถ่วงน้ำหนัก $w_1, w_2, \ldots, w_n$

3) นำค่าจาก 2) แต่ละค่าบวกกันทั้งหมด และรวมกับค่าถ่วงน้ำหนักพิเศษ w_0 หรืออาจมองเป็นข้อมูลนำเขาตัวแรกเลยว่าเป็น x_0 โดยกำหนดค่าให้เป็น 1

4) นำค่าจาก 3) เข้าฟังก์ชันกระตุ้น (Activate function) ที่มีลักษณะเป็นขั้นบันได (Step function) ให้ผลลัพธ์เป็น 1 หรือ -1 เท่านั้น โดยขั้นตอนที่กล่าวมาทั้งหมดแสดงด้วยสมการ 2.1

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2-2 โครงสร้างพื้นฐานเพอร์เซปตรอน [8]

เพื่อลดความซับซ้อนของการใช้สัญลักษณ์ในฟังก์ชัน อาจเขียนให้ฟังก์ชันของเพอร์เซปตรอนที่ให้ค่าผลลัพธ์เป็น 1 หรือ -1 ว่าเป็นฟังก์ชันเครื่องหมาย (Sign function, sgn) ตามสมการที่ 2.2 และ 2.3

$$o(x_i, \dots, x_n) = \begin{cases} 1 \; if \; w_0 + w_0 x_1 + w_2 x_2 + \dots + w_n x_n > 0 \\ -1 \; if \; otherwise \end{cases} \quad (2.1)$$

$$o(\vec{x}) = sgn(\vec{w} \cdot \vec{x}) \quad (2.2)$$

$$sgn(y) = \begin{cases} 1 \text{ if } y > 0 \\ -1 \text{ otherwise} \end{cases} \quad (2.3)$$

กฎการเรียนรู้ของเพอร์เซปตรอน หรือแนวคิดในการปรับค่าพารามิเตอร์ มีดังนี้ กำหนดให้ w_i คือค่าถ่วงน้ำหนักของเพอร์เซปตรอนตัวที่ i ตามสมการที่ 2.4 โดย o คือคำตอบที่คำนวณได้จากตัวแปร x ตัวที่ i ส่วน t คือคำตอบจริงตัวที่ i โดยค่า η เรียกว่า อัตราการเรียนรู้ (Learning rate) มีค่าตั้งแต่ 0 ถึง 1 เป็นค่าที่กำหนดว่าในแต่ละครั้งควรจะมีการปรับค่าพารามิเตอร์ไปมากน้อยเพียงใด ตามสมการที่ 2.5 แต่ข้อจำกัดของเพอร์เซปตรอนคือไม่สามารถใช้แก้ปัญหาแบบไม่เป็นเชิงเส้นได้ และผลลัพธ์ที่ได้มีลักษณะสองค่าเท่านั้น ตามที่แสดงในรูปที่ 2-3 จะเห็นว่าผลลัพธ์สุดท้ายในชั้นผลลัพธ์ให้ค่าเป็น 1 หรือ -1 เท่านั้น เมื่อผ่านฟังก์ชันกระตุ้นซึ่งเป็นแบบขั้นบันได้ (Step function) หรือฟังก์ชันตามค่าขีดแบ่ง (Threshold-based activation function) หรือสามารถจำแนกข้อมูลขาเข้าที่เป็นแบบเชิงเส้นเท่านั้น (Linear classifier) ทำให้การเรียนรู้ของเพอร์เซปตรอนเพียงอย่างเดียวไม่เหมาะในการนำไปใช้จริง

$$w_i \leftarrow w_i + \Delta w_i \quad (2.4)$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\Delta w_i \leftarrow \eta(t - o)x_i \quad (2.5)$$

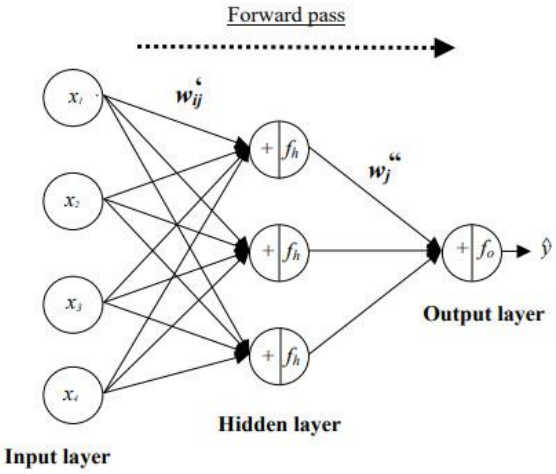
2. โครงข่ายประสาทเทียมเพอร์เซปตรอนหลายชั้น (Multilayer perceptron, MLP) หรือโครงข่ายแบบป้อนไปข้างหน้าโดยมีการเรียนรู้ย้อนกลับ (Feed-forward backpropagation neural network) เป็นต้นแบบของโครงข่ายประสาทเทียมที่มีความซับซ้อนแต่มีประสิทธิภาพ เพราะโจทย์ปัญหา XOR (Exclusive-OR) ซึ่งเป็นปัญหาแบบไม่เป็นเชิงเส้น (Non-linear) ไม่สามารถแก้ได้ด้วยเพอร์เซปตรอนเพียงชั้นเดียว ดังนั้นจึงต้องเพิ่มชั้นของเพอร์เซปตรอนเข้าไปอีก 1 ชั้น และเพิ่มความสามารถด้วยเทคนิคการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation) ที่มีการนำข้อมูลขาเข้าสู่โครงข่ายที่ได้รับการฝึกสอนแล้ว จากนั้นนำค่าผลลัพธ์ที่ได้จากการคำนวณเปรียบเทียบค่าที่ถูกต้องในชั้นข้อมูลขาเข้า หรือเรียกว่าหาค่าความแตกต่างหรือค่าความผิดพลาด [7] เพื่อปรับค่าถ่วงน้ำหนักและไบแอสในชั้นก่อนหน้า ในการใช้งานจะใช้ฟังก์ชันกระตุ้นแบบไม่เป็นเชิงเส้น (Non-Linear

activation functions) แทนฟังก์ชันขั้นบันได (Step function) โดยผลลัพธ์ของเพอร์เซปตรอนของแต่ละชั้นจะไม่ย้อนกลับมาเป็นข้อมูลขาเข้าของชั้นก่อนหน้าอีก แต่เป็นการส่งต่อข้อมูลแบบเดินหน้าทางเดียว โครงสร้างของโครงข่ายประสาทนี้แบ่งออกเป็น 3 ส่วน ตามรูปที่ 2-3 ได้แก่

2.1 ชั้นข้อมูลขาเข้า (Input Layer) เป็นชั้นแรกในโครงสร้างของแบบจำลอง ANNs ประกอบด้วยโหนดข้อมูล (Input Neuron) ซึ่งจะมีค่าเท่ากับจำนวนข้อมูลขาเข้าของข้อมูล หรือจำนวนคุณลักษณะ (Features) ของข้อมูลฝึกสอน (Training set) เป็นปัจจัยที่ในการเข้าสู่โครงข่าย

2.2 ชั้นซ่อน (Hidden Layer) ทำหน้าที่ในการรวมอินพุตที่คูณด้วยค่าถ่วงน้ำหนักเพื่อแปลงให้เป็นผลลัพธ์และส่งไปสู่ชั้นผลลัพธ์ (Output) โดยค่าถ่วงน้ำหนักจะส่งผลให้ทุก ๆ โหนด (Node) หรือนิวรอน (Neuron) มีค่าที่ไม่เท่ากัน ทำให้สามารถแยกว่าข้อมูลนี้เป็นคลาสอะไรได้ด้วยการดูตัวเลขที่ผลลัพธ์ และพบว่ายังไม่มีกฎที่กำหนดจำนวนชั้นซ่อนและโหนดที่เหมาะสม แต่ต้องมีอย่างน้อย 1 ชั้น

2.3 ชั้นข้อมูลผลลัพธ์ (Output Layer) เป็นชั้นสุดท้ายในโครงสร้างของแบบจำลอง ANNs ประกอบด้วย นิวรอนหรือโหนดแสดงผลข้อมูล (Output Neuron) โดยจำนวนนิวรอนจะเท่ากับจำนวนตัวแปร ที่ทำหน้าที่แสดงผลลัพธ์ (Output) ของ แบบจำลอง ANNs เมื่อข้อมูลผ่านไปสู่ชั้นข้อมูลผลลัพธ์แล้ว นิวรอนแต่ละตัวจะมีค่าไม่เท่ากัน นิวรอนของคลาสใดมีค่ามากที่สุดแสดงว่าผลลัพธ์คือคลาสนั้น

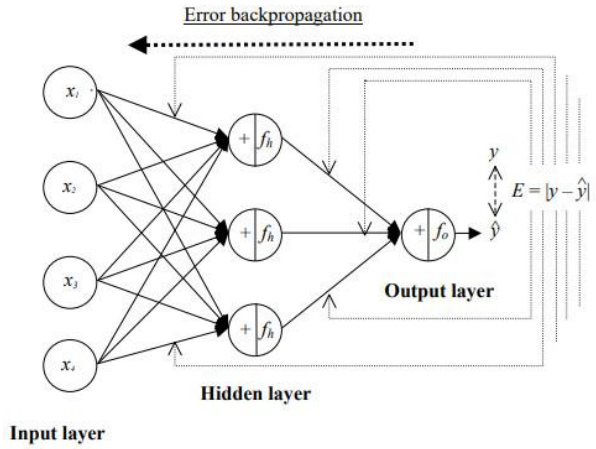


รูปที่ 2-3 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบเพอร์เซปตรอนหลายชั้น [9]

ตามรูปที่ 2-3 ประกอบด้วย 4 ข้อมูลขาเข้า และหนึ่งข้อมูลผลลัพธ์ สามารถเขียนให้อยู่ในรูปสมการของการหาผลลัพธ์ได้ตามสมการที่ 2.6 [9]

$$\hat{y} = f_o\left(\ddot{\theta} + \sum_{j=1}^{nh} w''_j \; fh\left(\sum_{i=1}^{nd} w'_{ij}\, x_i + \dot{\theta}\right)\right) \tag{2.6}$$

เมื่อ nd และ nh คือจำนวนโหนดของข้อมูลขาเข้าและจำนวนโหนดในชั้นซ่อนตามลำดับ และมีค่าถ่วงน้ำหนักระหว่างชั้นข้อมูลขาเข้าและชั้นซ่อนคือ w'_{ij} โดย i คือโหนดที่ i ในชั้นข้อมูลขาเข้า ส่วน j คือโหนดที่ j ในชั้นซ่อน เริ่มจากการคูณค่าข้อมูลขาเข้าของแต่ละโหนด x_i กับค่าถ่วงน้ำหนักจากชั้นข้อมูลขาเข้ามา รวมผลคูณทั้งหมด และบวกกับค่าไบแอส $\dot{\theta}$ ของโหนดในชั้นซ่อน จากนั้นเข้าสู่ฟังก์ชันกระตุ้น (Activation function , f_h) ซึ่งเป็นฟังก์ชันที่ไม่เป็นเชิงเส้น (Non-linear function) ได้ผลลัพธ์ส่งไปยังโหนดชั้นผลลัพธ์ เมื่อได้ค่าในแต่ละโหนดที่ผ่านฟังก์ชันกระตุ้นในชั้นซ่อนแล้วก็นำแต่ละค่าที่ได้คูณกับค่าถ่วงน้ำหนัก w''_j ที่อยู่ระหว่างชั้นซ่อนและชั้นข้อมูลผลลัพธ์ และบวกกับค่าไบแอส $\ddot{\theta}$ ของโหนดในชั้นข้อมูลผลลัพธ์ และเข้าสู่ฟังก์ชันกระตุ้น (f_o) ที่อาจเป็นฟังก์ชันเส้นตรงหรือไม่ก็ได้ โดยผลลัพธ์สุดท้ายที่คำนวณได้คือ $\hat{y}$ และในขั้นตอนสุดท้ายคือการค่าผิดพลาดระหว่างค่าที่คำนวณได้ $\hat{y}$ กับค่าผลลัพธ์หรือผลเฉลยที่แท้จริง y ในชั้นข้อมูลผลลัพธ์ ตามรูปที่ 2-4 [9] โดยเมื่อได้ค่าผิดพลาด (Error) ก็จะแพร่กลับไป เพื่อปรับค่าถ่วงน้ำหนักและค่าไบแอสเพื่อให้ค่าความผิดพลาดลดน้อยลงในการฝึกสอนรอบถัดไป โดยขั้นตอนการทำงานของ Backpropagation จะได้กล่าวถึงโดยละเอียดในหัวข้อถัดไป



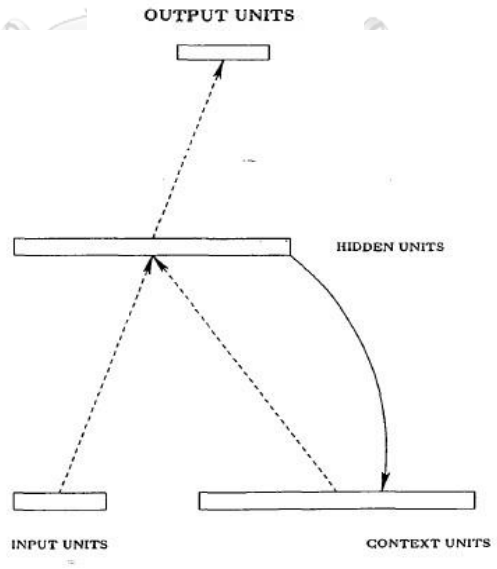
รูปที่ 2-4 สถาปัตยกรรมพื้นฐานของ Backpropagation [9]

2.7.2 โครงข่ายประสาทแบบคำนวณวนกลับ (Recurrent Networks) [10]

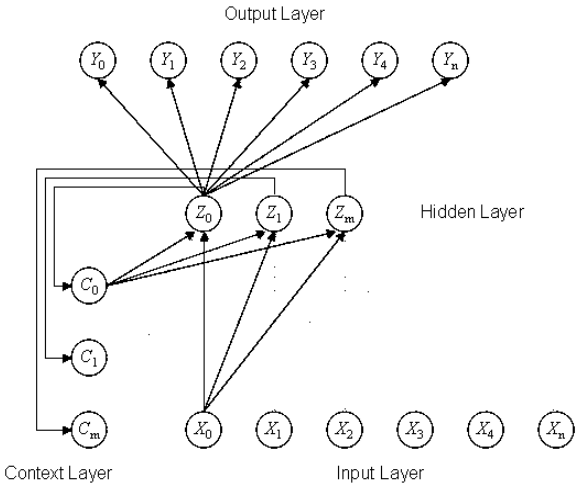
โครงข่ายแบบนี้ต่างจากแบบคำนวณไปข้างหน้า และมีความซับซ้อนมากกว่า โดยโครงสร้างการคำนวณจะเป็นสองทิศทาง โดยมีการคำนวณไปข้างหน้าจากข้อมูลขาเข้า และขณะเดียวกันโครงข่ายทำการคำนวณวนกลับโดยเริ่มจากข้อมูลผลลัพธ์กลับมายังข้อมูลขาเข้าวนซ้ำหลาย ๆ ครั้ง จนได้ผลลัพธ์ออกมาตัวอย่างของโครงข่ายประเภทนี้ เช่น โครงข่ายประสาทวนซ้ำอย่างง่าย (Simple Recurrent Networks : SRN) ซึ่งเป็นโครงข่ายประสาทแบบหลายชั้น (MLP) คือ

ประกอบด้วยชั้นข้อมูลขาเข้า ชั้นซ่อนหนึ่งชั้น และชั้นข้อมูลผลลัพธ์ และเพิ่มเติมชั้นของบริบท (Context Layer) โดยในแต่ละรอบการคำนวณปรับค่าถ่วงน้ำหนักและค่าไบแอส ค่าของข้อมูลขาเข้าใหม่จะถูกส่งเข้าโครงข่ายประสาทวนซ้ำ เพื่อทำการคำนวณและส่งไปยังชั้นบริบท ก่อนจะส่งกลับมายังชั้นซ่อนอีกครั้ง ซึ่งขั้นตอนการทำงานคล้ายกับการทำงานแบบแพร่ย้อนกลับ และเหมาะกับข้อมูลขาเข้าที่เป็นลักษณอนุกรมหรือมีความต่อเนื่องของข้อมูล และการจำแนกประเภท (Classification) โครงสร้างการทำงานของโครงข่ายประสาทอย่างง่ายแสดงในรูปที่ 2-5

โครงข่ายวนซ้ำอย่างง่าย (Simple Recurrent Network : SRN) ได้รับความนิยม เพราะมีชั้นเก็บสถานะก่อนเข้าเข้าสู่โครงข่ายในลำดับถัดไป จึงสามารถนำไปใช้กับรูปแบบข้อมูลที่มีเวลามาเกี่ยวข้องได้หลากหลายโดยแสดงให้อยู่ในรูปโครงข่ายตามรูปที่ 2-6 ซึ่งจะเห็นว่าโครงข่ายทั่วไปของ Elman ในชั้นบริบทซึ่งรับข้อมูลจากชั้นซ่อน และส่งคืนค่ากลับไปเมื่อผ่านการคำนวณแล้ว ซึ่งพบว่าข้อมูลเข้ามายังชั้นซ่อนนอกจากสถานะปัจจุบันแล้วยังรับมาจากสถานะก่อนหน้า เพื่อให้ข้อมูลที่เกิดขึ้นก่อนถูกนำเข้ามาคำนวณด้วย และนอกจากนี้สามารถรองรับข้อมูลขาเข้าที่มีความยาวไม่จำกัดได้ และโครงข่ายประสาทแบบวนกลับนี้รองรับการทำงานของการแพร่ย้อนกลับขณะทำการฝึกสอนได้ เหมาะกับการนำไปใช้กับข้อมูลที่เป็นลำดับ (Sequence recognition) หรือการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP)



รูปที่ 2-5 แสดงโครงข่ายประสาทแบบวนซ้ำอย่างง่าย [10]



รูปที่ 2-6 แสดงโครงข่ายของ Elman [11]

ในงานวิจัยนี้เลือกใช้โครงข่ายประสาทแบบวนกลับ แต่ประยุกต์ใช้การเรียนรู้เชิงลึก (Deep Learning) ที่มีการเพิ่มจำนวนโหนดของชั้นซ่อน เพราะลักษณะเป็นข้อความที่ลำดับของคำมีความสำคัญและส่งผลต่อผลลัพธ์ในการจำแนก และนอกจากนี้ยังใช้การเรียนรู้แบบแพร่ย้อนกลับเพื่อปรับลดความผิดงานของการทำนายผลในชั้นข้อมูลผลลัพธ์

2.8 Backpropagation [8]

เป็นอัลกอลิทึมที่ใช้ในการฝึกสอนแบบจำลองในโครงข่ายประสาทเทียมหลายชั้น (MLP) เพื่อให้ได้ค่าถ่วงน้ำหนักที่เหมาะสม โดยผลลัพธ์ที่ได้เข้าใกล้ค่าเป้าหมายมากที่สุด ในกระบวนการเรียนรู้นั้นใช้เกรเดียนดีเซนต์ (Gradient descent) เพื่อพยายามลดข้อผิดพลาดระหว่างค่าผลลัพธ์ของที่ได้และค่าเฉลยที่แท้จริง

Chulalongkorn University

จากแนวคิดของการแพร่ย้อนกลับ เมื่อได้ค่าความผิดพลาด (error) ที่คำนวณจากค่าถ่วงน้ำหนักในรอบปัจจุบัน จากการคำนวณค่าเป้าหมายเทียบกับค่าผลลัพธ์ ทำให้ทราบว่าแต่ละพารามิเตอร์ (Weight , Bias) ส่งผลต่อความผิดพลาดมากน้อยเพียงใด จากนั้นก็ทำการปรับค่าพารามิเตอร์เหล่านั้น เพื่อให้ค่าความผิดพลาดน้อยที่สุด โดยขั้นตอนการเรียนรู้ของการแพร่ย้อนกลับ (Backpropagation) มี 3 ขั้นตอนหลัก ได้แก่ การคำนวณไปข้างหน้า การคำนวณย้อนกลับและการปรับค่าพารามิเตอร์ (Weight, Bias) โดยใช้เทคนิคการทำงานของเกรเดียนดีเซนต์ (Gradient Descent: GD) ดังนี้

2.8.1 การส่งผ่านไปข้างหน้า (Forward Pass)

กำหนดให้ $\vec{x}$ คือเวกเตอร์ของข้อมูลขาเข้า w_{0j} คือค่าถ่วงน้ำหนักพิเศษหรือค่าไบแอส (Bias) ในชั้นซ่อน w_{0k} คือค่าถ่วงน้ำหนักพิเศษหรือค่าไบแอส (Bias) ในชั้นข้อมูลผลลัพธ์ n_i เป็นจำนวนโหนดข้อมูลเข้า n_j จำนวนโหนดชั้นซ่อน และ n_k เป็นจำนวนโหนดชั้นผลลัพธ์ จากนั้นสุ่มค่าถ่วงน้ำหนักทั้งหมดทั้งของชั้นซ่อนและชั้นผลลัพธ์ โดยใช้ฟังก์ชันในการสุ่มตัวเลขขึ้นมา เพื่อกำหนดเป็นค่าเริ่มต้น จากนั้นทำตามขั้นตอนต่อไปนี้จนกว่าจะเข้าเงื่อนไงให้หยุดการเรียนรู้หรือครบรอบตามที่กำหนดไว้

1. กำหนดค่าข้อมูลขาเข้า x_i ($i=1,\ldots,n$) ให้กับแต่ละโหนด เพื่อส่งไปยังชั้นซ่อน

2. คำนวณค่าที่ได้จากชั้นข้อมูลขาเข้า ในโหนดของชั้นซ่อน โดยรวมค่าถ่วงน้ำหนักและค่าไบแอส จากนั้นเข้าฟังก์ชันกระตุ้น ตามสมการที่ 2.7

$$h_j(\vec{x}) = f\left(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i\right) \tag{2.7}$$

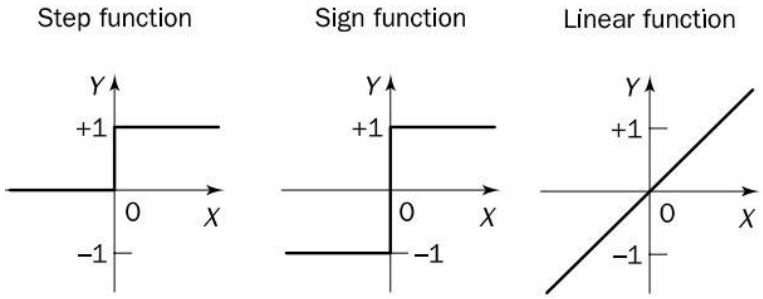
โดย h_j เป็นสมการของหน่วยย่อยตัวที่ j ในชั้นซ่อน จากนั้นส่งค่าที่คำนวณได้เป็นข้อมูลขาเข้าของชั้นผลลัพธ์ w_{0j} คือค่าถ่วงน้ำหนักพิเศษของโหนดในชั้นซ่อน w_{ij} คือค่าถ่วงน้ำหนักของแต่ละข้อมูลขาเข้ามายังโหนดในชั้นซ่อน และ f คือฟังก์ชันกระตุ้น (Activation function) โดยเหตุผลหลักที่ต้องมีฟังก์ชันกระตุ้นในงานโครงข่ายประสาทเทียม เพราะในเพอร์เซปตรอนแต่ละตัวมีการหาผลรวมเชิงเส้นแล้ว (Linear Combination) ดังนั้นเพื่อเป็นการเพิ่มประสิทธิภาพของโครงข่ายประสาทจึงมีฟังก์ชันกระตุ้น (Activation function) โดยขึ้นอยู่กับลักษณะข้อมูลผลลัพธ์ในชั้นข้อมูลผลลัพธ์ แบ่งเป็น 3 ประเภทหลัก ดังนี้

2.1 ฟังก์ชันขั้นบันไดแบบสองคำตอบ (Binary Step Function) เป็นฟังก์ชันที่ให้ผลลัพธ์ในลักษณะเป็น “ใช่” หรือ “ไม่ใช่” จะต้องใช้ Threshold function เช่น ฟังก์ชันขั้นบันได (Step function) ที่ให้ผลลัพธ์ {0,1} ฟังก์ชันเครื่องหมาย (Sign function) ที่ให้ผลลัพธ์ {-1,1} ฟังก์ชันขั้นบันไดและฟังก์ชันเครื่องหมายสามารถถูกเรียกว่า Hard Limit function เหมาะในการแก้ปัญหาเกี่ยวกับการตัดสินใจ แยกแยะข้อมูล หรือการรู้จำ (Pattern Recognition) และต้องการคำตอบที่ชัดเจน แต่มีข้อจำกัดคือไม่สามารถใช้กับการจำแนกข้อมูลออกเป็นหลายประเภทหรือให้คำตอบที่หลากหลายได้ และนอกจากนี้ไม่สามารถใช้กับการหาค่าความลาดชันได้ (Gradient Descent) ในกระบวนการแพร่ย้อนกลับได้

2.2 ฟังก์ชันเส้นตรง (Linear function) เหมาะสำหรับงานที่ประมาณค่าความสัมพันธ์ที่เป็นเชิงเส้นระหว่างข้อมูลขาเข้า (input) และข้อมูลผลลัพธ์ (Output) หรือการวิเคราะห์การถดถอยเชิงเส้น (Linear regression) โดยมีความสามารถเหนือกว่าฟังก์ชันขั้นบันไดเพราะสามารถให้ผลลัพธ์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ได้หลายประเภท แต่มีข้อจำกัดคือไม่สามารถใช้การแพร่ย้อนกลับ (Backpropagation) ด้วยเทคนิคการคำนวณหาค่าความลาดชันเพื่อฝึกโมเดลได้ เพราะอนุพันธ์ของฟังก์ชันเป็นค่าคงที่และไม่มีความสัมพันธ์กับข้อมูลขาเข้า และที่สำคัญถ้าใช้ฟังก์ชันกระตุ้นเป็นแบบเชิงเส้นในชั้นซ่อนแล้ว การคำนวณในแต่ละโหนดจะไม่เพิ่มความสามารถให้กับโครงข่าย เพราะทุกชั้นซ่อนในโครงข่ายทำงานเหมือนมีเพียงชั้นข้อมูลผลลัพธ์เท่านั้น ซึ่งเหมือนการทำงานของโครงข่ายประสาทเทียมชั้นเดียว (Single Layer Perceptron) ตามที่แสดงในรูปที่ 2-7

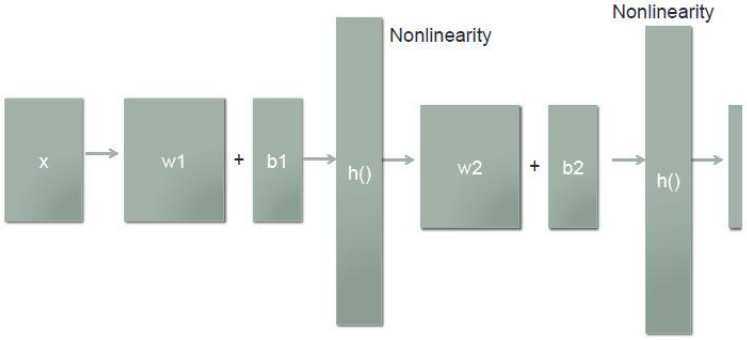


รูปที่ 2-7 ชนิดของฟังก์ชันกระตุ้นใน Single Layer Perceptron [12]

ตามสมการเมทริกซ์ที่ 2.8 [13] ของการส่งผ่านข้อมูลในแต่ละชั้นของโครงข่ายประสาท (Computation Graph) ทำให้สามารถนำโหนดมาต่อ ๆ กันได้ตามรูปที่ 2-8 เพราะถ้า h ในสมการเมทริกซ์ ตามสมการที่ 2.8 เป็นฟังก์ชันเชิงเส้น (Linear Function) ก็สามารถคูณกันได้รวมเป็นชั้นเดียว ไม่สามารถทำเป็นชั้น ๆ ได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$h(w_2^T h(w_1^T x + b_1) + b_2) \tag{2.8}$$

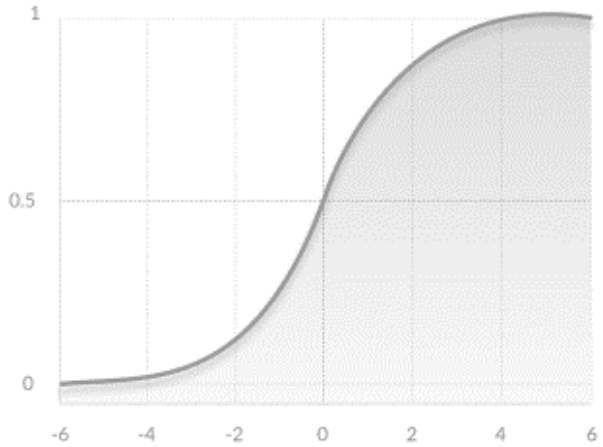


รูปที่ 2-8 การส่งผ่านข้อมูลในแต่ละชั้นของโครงข่ายประสาท [13]

2.3 ฟังก์ชันกระตุ้นแบบไม่เป็นเชิงเส้น (Non-Linear Activation function) ใช้ในการแก้ปัญหาที่ซับซ้อนมากขึ้นและไม่เป็นลักษณะเชิงเส้น ที่ได้รับความนิยมในปัจจุบัน ดังนี้

1) ฟังก์ชันซิกมอยด์ (Sigmoid function) ใช้สำหรับงานที่มีค่าตัวเลขต่อเนื่อง จึงเหมาะในการหาความน่าจะเป็นของผลลัพธ์ในชั้นข้อมูลผลลัพธ์ (Output Layer) โดยสามารถแปลงค่าจำนวนข้อมูลขาเข้าที่เข้ามาแบบไม่จำกัดให้เป็นข้อมูลผลลัพธ์ ให้อยู่ในช่วง 0 ถึง 1 ได้ตามที่แสดงในรูปที่ 2-9 ในลักษณะความน่าจะเป็น ทำให้มีคุณสมบัติที่เวลานำไปคำนวณค่าอนุพันธ์แล้วค่าที่ได้ไม่เป็น 0 จึงสนับสนุนกระบวนการแพร่ย้อนกลับได้ ในการคำนวณหาค่าความลาดชัน แต่มีข้อจำกัดคืออาจเกิดปัญหาติดขัดระหว่างการฝึกฝนแบบจำลอง ถ้าข้อมูลขาเข้ามีค่าน้อยมากๆ หรือสูงมากๆ ความชันจะเข้าใกล้ 0 จน Gradient หายไปหมด ทำให้ไม่มีการเปลี่ยนแปลงระหว่างการฝึกสอนแบบจำลอง เรียกว่าปัญหา Vanishing Gradient คือพบว่าในระหว่างการเทรน Gradient มีขนาดเล็กลงเรื่อย ๆ จนเท่ากับ 0 ทำให้ค่าถ่วงน้ำหนักไม่ถูกอัพเดทอีกต่อไป ทำให้โมเดลเทรนต่อไม่ได้ และปัญหาผลลัพธ์ ไม่สมดุลเพราะมีค่ากึ่งกลางไม่เท่ากับ 0 (Not Zero Centered) ทำให้การหาจุดสมดุล (Optimize) ยาก เพราะระหว่างการฝึกสอนทำให้ค่าความลาดชันซิกแซกไปมา ส่งผลให้ได้แบบจำลองที่ต้องการช้า (Converge) ดังนั้นจึงเหมาะในการนำฟังก์ชันซิกมอยด์ไปเป็นฟังก์ชันกระตุ้นในชั้นข้อมูลผลลัพธ์ และให้ลักษณะของฟังก์ชันตามสมการที่ 2.9 [12] เมื่อ x คือผลลัพธ์ที่ได้จากผลรวมของผลคูณระหว่างข้อมูลขาเข้ากับค่าถ่วงน้ำหนักและค่าไบแอสที่โหนดนั้นได้รับ

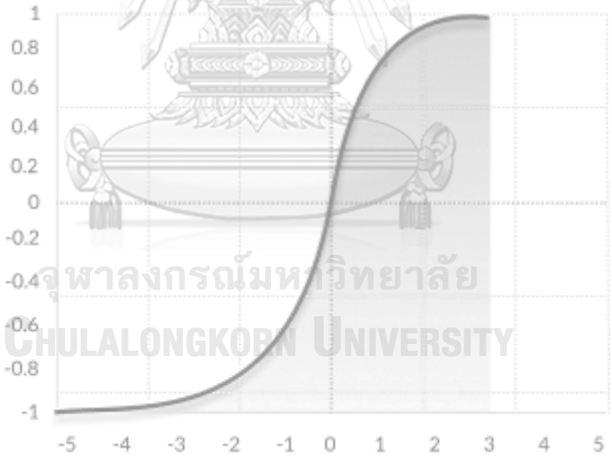
$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \tag{2.9}$$



รูปที่ 2-9 แสดงกราฟของฟังก์ชันซิกมอยด์ [14]

2) ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ (Hyperbolic tangent function หรือ tanh) [15] มีลักษณะเช่นเดียวกับฟังก์ชันซิกมอยด์คือเป็นรูป s-curve เหมือนกันตามที่แสดงในรูปที่ 2-10 แต่แปลงค่าจำนวนจริงให้อยู่ในช่วงระหว่าง -1 ถึง +1 และมีคุณสมบัติหาอนุพันธ์แล้วไม่เป็น 0 เหมือนอย่างฟังก์ชันซิกมอยด์ และเมื่อค่ากึ่งกลางของผลลัพธ์มีค่าเป็น 0 (Zero Centered) ทำให้แก้ปัญหาของการซิกแซกไปมาระหว่างการฝึกสอนเพื่อหาค่าความลาดชันที่มีในฟังก์ชันซิกมอยด์ได้ และนอกจากนี้ถ้าข้อมูลขาเข้าใกล้ 0 แล้วเมื่อหาค่าอนุพันธ์จะทำให้ความชันมากกว่าฟังก์ชันซิกมอยด์ ส่งผลให้หาค่าความลาดชันได้มากกว่า จึงฝึกสอนแบบจำลองได้เร็วกว่า แต่ปัญหา Vanishing Gradient Descent เมื่อข้อมูลมีค่าน้อยมากๆหรือสูงมากๆ (ไกลจาก 0) ยังคงเป็นเป็นปัญหาเช่นเดียวกับฟังก์ชันซิกมอยด์ แต่การได้แบบจำลองที่ต้องการจะทำได้เร็วกว่า (Converge) ฟังก์ชันซิกมอยด์ แต่ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ไม่สามาถให้ผลลัพธ์เป็นค่าความน่าจะเป็น ดังนั้นจึงไม่สามารถใช้เป็นฟังก์ชันกระตุ้นในชั้นข้อมูลผลลัพธ์แต่สามารถใช้ในซ่อนได้ ตามสมการ 2.10

$$f(x) = tanh(x) = 2\sigma(2x) - 1 = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.10}$$



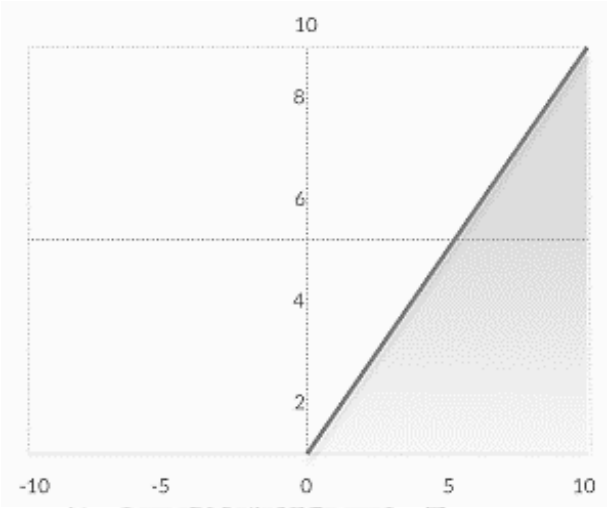
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 2-10 แสดงกราฟของฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ [14]

3) ฟังก์ชันเรียงกระแสตรง (Rectifier) หรือฟังก์ชันรีลู (ReLU : Rectified Linear Unit) [15] โดยจะแปลงข้อมูลขาเข้าให้เป็นค่า 0 สำหรับค่าที่เป็นลบ ส่วนข้อมูลขาเข้าจำนวนอื่นก็เป็นตามเดิม ข้อดีของฟังก์ชันนี้คือไม่มีปัญหา Vanishing Gradient descent เพราะขอบล่าง (Lower bound) เป็น 0 แต่ด้านบนไม่มีนั่นเอง แต่ก็มีข้อเสียคือปัญหาค่ากึ่งกลางข้อมูลส่งออก (Output centered) ไม่เป็น 0 ทำให้การ Optimize ทำได้ยาก และจัดการลำบากเพราะผลลัพธ์มีค่า [0,∞) เพราะเป็นช่วงที่กว้าง และค่าเป็น 0 ที่ออกมาจากโหนดที่ขอบล่างจะเยอะมากที่เรียกว่าเกิด

Inactive Neural แต่ปัญหาที่สำคัญของฟังก์ชันนี้คือ เมื่อข้อมูลขาเข้า (Input) มีค่าเป็นลบแต่ข้อมูลผลลัพธ์ (Output) มีค่าเป็นบวก ทำให้เกิดปัญหาการแพร่ย้อนกลับ เพราะโครงข่ายไม่สามารถเรียนรู้ค่าได้ แต่ข้อดีเรื่องของ Vanishing Gradient descent ที่ทำให้เวลาในฝึกสอนแบบจำลองลดลง และเมื่อใช้แล้วทำให้แบบจำลองสามารถ Converge ได้เร็วขึ้น เมื่อเทียบกับอีกสองฟังก์ชัน ตามที่แสดงในรูปที่ 2-11 และฟังก์ชันรีลูมีรูปแบบตามสมการที่ 2.11

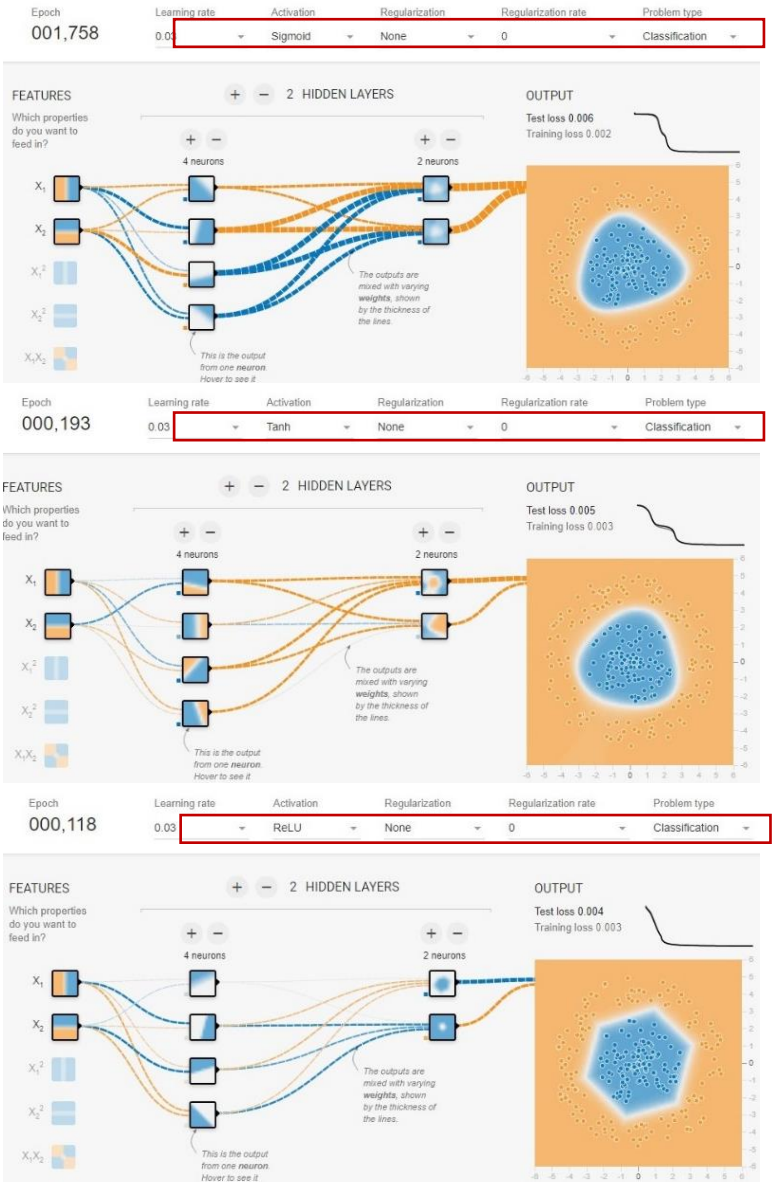
$$f(x) = ReLU(x) = max\ (0, x) \qquad 2.11$$



รูปที่ 2- 11แสดงฟังก์ชันเรียงกระแสตรงหรือรีลู (LeRU) [14]

ฟังก์ชันกระตุ้นทั้ง 3 ได้แก่ Sigmoid, Tanh และ ReLU เมื่อทำการเปรียบเทียบด้วยการตั้งค่าอัตราการเรียนรู้ (Learning rate) จำนวนชั้นซ่อน (Hidden Layer) และจำนวนโหนดของข้อมูลขาเข้า (Input) ในงานจำแนกประเภท (Classification) เท่ากันแล้ว พบว่าจำนวนรอบของการฝึกฝน (Epoch) ที่ทำให้ผลลัพธ์แสดงความผิดพลาดมีค่าใกล้เคียงกัน (Test loss, Training loss) คือ Tanh 193 รอบ และ ReLU 118 รอบ ส่วน Sigmoid ต้องใช้ 1,758 รอบ ตามที่แสดงในรูปที่ 2-12 ในงานวิจัย Tanh ถูกใช้ในการฝึกฝนแบบจำลอง เพราะใช้งานแบบจำลอง LSTM ที่สามารถแก้ปัญหาการเลือนหาย (Vanishing Gradient) ได้อยู่แล้ว แต่ผู้วิจัยเลือกใช้เพราะการสมดุลของข้อมูล แทนการการใช้ ReLU

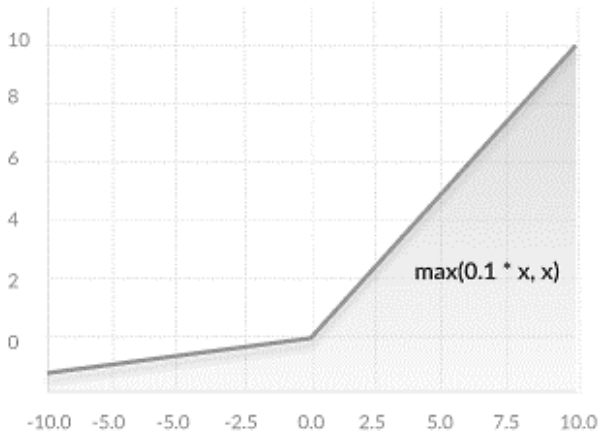
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 2-12 การเปรียบเทียบฟังก์ชันกระตุ้นในงานจำแนกประเภท [16]

จากปัญหาเรื่องขอบล่างที่มีค่าเป็น 0 ของรีลู จึงมีผู้คิดค้น Leaky ReLU [14] ขึ้นมา เพื่อแก้ไขปัญหาที่เกิดขึ้น โดยข้อมูลส่งออกจากเดิมถ้าเป็นลบจะให้ค่าเป็น 0 ก็เปลี่ยนเป็นให้ค่าติดลบเช่นเดิม แต่ค่าจะลดน้อยลง เพื่อที่จะยังสามารถให้มีการเรียนรู้ข้อมูลในกระบวนการแพร่ย้อนกลับได้ แต่พบว่าการเรียนรู้ที่เกิดขึ้นค่าที่ได้จากค่าติดลบที่น้อยลงไม่สอดคล้องกับข้อมูลขาเข้าจริงๆ ดังนั้นวิธีนี้จึงไม่ค่อยเป็นที่นิยมตาม ตามที่แสดงในรูปที่ 2-13 ฟังก์ชัน Leaky ReLU มีรูปแบบตามสมการที่ 2.12

$$f(x) = Leaky\ ReLu(x) = \max(0.1x, x) \qquad 2.12$$

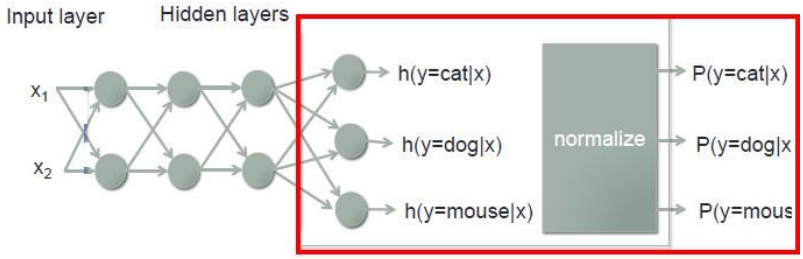


รูปที่ 2-13 แสดงฟังก์ชัน Leaky ReLU [14]

4) ฟังก์ชันซอฟต์แมกซ์ (SoftMax function) [15] ทำหน้าที่แปลงข้อมูลขาเข้าที่มีลักษณะเป็นเวกเตอร์จำนวนจริง ให้เป็นข้อมูลผลลัพธ์ที่เป็นจำนวนจริงระหว่าง 0 ถึง 1 [0,1] แล้วทำการ Normalize ค่าให้ออกมาเป็นค่าความน่าจะเป็นที่มีผลรวมเป็น 1 ดังนั้นฟังก์ชันนี้จึงเหมาะในการนำไปไว้ในชั้นสุดท้ายของโครงข่ายคือชั้นข้อมูลผลลัพธ์ (Output Layer) ในงานจำแนกแบบหลายประเภท (Multi-Class Classification) โดยงานวิจัยนี้ใช้ฟังก์ชันนี้ในการแสดงผลลัพธ์ของแบบจำลอง แต่ก็มีข้อเสียคือเหมาะในการที่ต้องการคำตอบที่ถูกต้องที่สุดเพียงคลาสเดียว หรือเรื่องประสิทธิภาพของการทำงาน เพราะขั้นตอนสุดท้ายต้องทำการหารด้วยจำนวนคลาสทั้งหมดที่เป็นไปได้ จึงอาจเกิดปัญหาถ้ามีข้อมูลผลลัพธ์หรือคลาสที่เป็นไปได้จำนวนมาก โดยมีรูปแบบตามสมการที่ 2.13 โดยใช้ข้อมูลขาเข้าที่มีเวกเตอร์ขนาด k มิติ และสมการของสมาชิกลำดับที่ j ตามรูปที่ 2-14

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$f(x)_j = softmax(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{k} e^{x_k}} \tag{2.13}$$

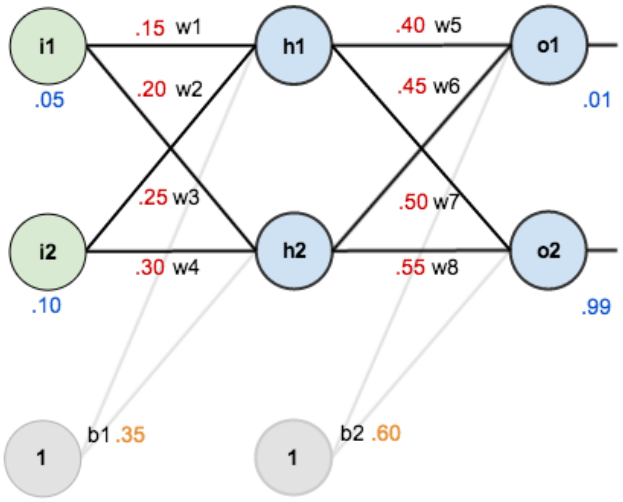


รูปที่ 2-14 แสดงซอฟต์แมกซ์ฟังก์ชันในชั้นข้อมูลส่งออก [13]

3. คำนวณค่าที่ได้จากชั้นซ่อน ในโหนดของชั้นข้อมูลออก โดยรวมค่าถ่วงน้ำหนักและค่าไบแอส จากนั้นเข้าฟังก์ชันกระตุ้น ตามสมการที่ 2.14

$$o_k(x) = f\left(w_{0k} + \sum_{i=1}^{n} w_{jk} h_i\right) \quad (2.14)$$

เมื่อ o_k เป็นสมการของหน่วยย่อยตัวที่ k ในชั้นข้อมูลผลลัพธ์ ซึ่งเป็นผลลัพธ์ที่ได้จากการคำนวณผ่านโครงข่ายและได้ผ่านฟังก์ชันก์ชันกระตุ้นแล้ว (Activation function)



รูปที่ 2-15 โครงข่าย MLP แบบมีผู้สอนและค่าพารามิเตอร์ [17]

ตามรูปที่ 2-15 เป็นตัวอย่างโครงข่ายที่ประกอบด้วยชั้นข้อมูลขาเข้า 2 โหนด คือ i_1, i_2 ชั้นข้อมูลซ่อนหนึ่งชั้นประกอบด้วย 3 โหนด ได้แก่ h_1, h_2 และชั้นข้อมูลผลลัพธ์ จำนวน 2 โหนดได้แก่ O_1, O_2 โดยกำหนดให้ค่าผลลัพธ์ (Target) ที่โหนดข้อมูลออก O_1 เป็น 0.01 และ O_2 เป็น 0.99 โดยการคำนวณค่าไปข้างหน้า เพื่อให้ได้ค่าผลลัพธ์ของแต่ละโหนดเริ่มจากในชั้นซ่อน h_1, h_2 โดยใช้สมการ 2.14 ดังนี้

$Net_{h1} = i_1*w_1+i_2*w_2+b_1*1 = 0.3775$, $Net_{h2} = i_1*w_3+i_2*w_4+b_1*1 = 0.3925$ เริ่มจากการหาผลรวมของค่าข้อมูลขาเข้าคูณกับค่าถ่วงน้ำหนักและรวมกับค่าไบแอส (Linear Combination) ของแต่ละโหนดซ่อน จากนั้นนำผลรวมที่ได้ในแต่ละโหนดเข้าฟังก์ชันกระตุ้น ในตัวอย่างนี้ใช้ฟังก์ชันซิกมอยด์ (Sigmoid Activation function) จะได้ $out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.59326$ และ $out_{h2} = \frac{1}{1+e^{-net_{h2}}} = \frac{1}{1+e^{-0.3925}} = 0.59688$ จากนั้นนำค่า Out_{h1} และ Out_{h2} เป็นค่าข้อมูลขาเข้า (Input) ของโหนดในชั้นข้อมูลออก (Output) จะได้ $net_{O1} = h_1*w_5+ h_2*w_6 + b_2*1 = 1.1059$, $net_{O2} = h_1*w_7+ h_2*w_8 + b_2*1 = 1.2249$ จากนั้น

นำผลรวมที่ได้ในแต่ละโหนดเข้าฟังก์ชันกระตุ้นซิกมอยด์ (Sigmoid Activation function) จะได้ $out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.1059}} = 0.751365$ และ $out_{o2} \frac{1}{1+e^{-net_{o2}}} = \frac{1}{1+e^{-1.2249}} = 0.77292$ จบขั้นตอนการคำนวณไปข้างหน้า (Forward Pass) เมื่อได้ค่าผลลัพธ์ในชั้นข้อมูลผลลัพธ์ out_{O1} และ out_{O2}

2.8.2 การคำนวณค่าความคลาดเคลื่อน

เป็นฟังก์ชันต้นทุน (Cost function) หรือฟังก์ชันการสูญเสีย (Loss function) เพื่อใช้สอนโครงข่ายที่เป็นการคำนวณแบบไปข้างหน้า (Feedforward) ดังนั้นกฎในการเรียนรู้จึงมีหน้าที่ในการลดค่าความคลาดเคลื่อนนี้ลงให้มากที่สุด เพื่อเพิ่มประสิทธิภาพของการทำงานโครงข่าย และการที่สามารถคำนวณค่าความผิดพลาดนี้ได้ต้องเป็นการเรียนรู้แบบมีผู้สอน (Supervise Learning) เพราะจะได้ทราบผลเฉลยหรือค่าเป้าหมายในการฝึกฝน

1. ค่าความผิดพลาด (Error) เทคนิคที่ใช้ในการคำนวณต้องเหมาะสมกับลักษณะของงาน เช่น ถ้าเป็นงานวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) นิยมใช้การคำนวณค่าผิดพลาด MSE (Mean Square Error), Mean Absolute Error (MAE) หรือ Root Mean Squared Error (RMSE) แต่ถ้าเป็นงานด้านการจำแนกประเภท (Classification) ใช้ Cross-Entropy สำหรับ Multi-class Classification หรือใช้ Binary Cross-Entropy สำหรับ Binary-class Classification

ฟังก์ชันที่คำนวณค่าความผิดพลาดมีหลายประเภท เลือกใช้ตามวัตถุประสงค์ของงาน โดยกำหนดค่าตัวแปรดังนี้ E คือค่าความผิดพลาด D คือข้อมูลฝึกสอนทั้งหมด (Training Data) d คือตัวอย่างที่นำเข้าโครงข่าย t_d คือค่าเป้าหมายหรือค่าเฉลยและ o_d คือค่าที่คำนวณได้

1) ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error : MAE) คือการคำนวณความแตกต่างระหว่างค่าเฉลยหรือเป้าหมายกับค่าที่เป็นผลลัพธ์ โดยการหาค่าเฉลี่ยที่ไม่คิดเครื่องหมายและทิศทาง ตามสมการที่ 2.15

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$E_{MAE} = \frac{1}{D}\sum_{d\epsilon D}|t_d - o_d| \quad (2.15)$$

2) ค่าเฉลี่ยความผิดพลาดยกกำลังสอง (Mean Square Error : MSE) เป็นการคำนวณความผิดพลาดที่เกิดขึ้น โดยยกกำลังสองค่าความแตกต่างระหว่างค่าที่คำนวณได้และผลเฉลย เป็นเทคนิคที่เหมาะกับการวิเคราะห์การถดถอยเชิงเส้น (Linear Regression) ตามสมการที่ 2.16 ซึ่งเป็นสมการที่มีหนึ่งโหนดข้อมูลผลลัพธ์ แต่ถ้ามีมากกว่าก็ให้หาผลรวมทั้งหมดของทุกโหนด โดย k คือจำนวนโหนดของชั้นข้อมูลผลลัพธ์ตามสมการที่ 2.17 ถ้าเป็นการใช้ Gradient descent สำหรับหาทิศการเคลื่อนที่ของกระบวนการแพร่ย้อนกลับ การใช้ MSE จะมีความเหมาะสมกว่า MAE เพราะในขั้นตอนการหาสมการเชิงอนุพันธ์ย่อย (Partial Differential Equations) สามารถทำได้ง่ายกว่า

$$E_{MSE} = \frac{1}{D}\sum_{d\in D}(t_d - o_d)^2 \quad (2.16)$$

$$E_{MSE} = \frac{1}{D}\sum_{d\in D}\sum_{k\in outputs}(t_d - o_d)^2 \quad (2.17)$$

ในสมการที่ 2.16 และ 2.17 ปกติตัวหารต้องแทนด้วยจำนวนข้อมูลสอนทั้งหมด (D) แต่จะเกิดปัญหาเวลาหาอนุพันธ์ เพราะจะทำให้ได้ $\frac{2}{D}$ ซึ่งไม่สามารถตัดกันได้หมด จึงนิยมแทน D ด้วย 2 เพื่อจะได้ตัดกันได้ เหลือ 1 จะทำให้การหาค่าอนุพันธ์ทำได้ง่ายขึ้น

3) รากที่สองของค่าเฉลี่ยความผิดพลาดยกกำลังสอง (Root Mean Squared Error : RMSE) คือ การนำ MSE มาหา Squared Root เพื่อให้ได้ค่าการสูญเสียมีหน่วยเดียวกับตัวแปรของผลลัพธ์ เพราะการยกกำลังสองค่าผิดพลาดของ MSE ทำให้หน่วยมันเปลี่ยนไปจากเดิม ตามสมการที่ 2.18

$$E_{RMSE} = \sqrt{\frac{1}{D}\sum_{d\in D}(t_d - o_d)^2} \quad (2.18)$$

4) ค่าเฉลี่ยครอส-เอนโทรปีแบบทวิภาค (Binary Cross-Entropy : BCE) หรือ Log Loss เหมาะกับปัญหาที่เป็นการจำแนกประเภทแบบสองกลุ่ม (Binary Classification) ตามสมการที่ 2.19

$$E_{BCE} = -\frac{1}{D}\sum_{d\in D}[t_d\, log(o_d) + (1 - t_d)log(1 - o_d)] \quad (2.19)$$

5) ค่าเฉลี่ยครอส-เอนโทรปี (Cross Entropy : CE) หรือค่าติดลบความเป็นไปได้ (Negative Log Like hood) เหมาะสำหรับการจำแนกงานประเภทหลายกลุ่ม (Multi classification) ตามสมการที่ 2.20 เมื่อ c คือจำนวนคลาส (Class) หรือประเภทที่เป็นไปได้ทั้งหมด และการมีเครื่องหมายด้านหน้าเพื่อให้ผลรวมมีค่าเป็นบวก เนื่องจากค่าของ Log ความน่าจะเป็นมีค่า (0,1) ทำให้ผลของ Log มีค่าติดลบ

$$\mathrm{E_{CE}} = -\frac{1}{\mathrm{D}}\sum_{\mathrm{c=1}}^{\mathrm{C}} \mathrm{t_d} * \log(\mathrm{o_d}) \quad (2.20)$$

2. ค่าผลลัพธ์ที่ได้ในชั้นข้อมูลผลลัพธ์ จากขั้นตอนการคำนวณไปข้างหน้าคือ $O_K\ (O_{k=1},\ldots,k)$ เมื่อ k คือจำนวนโหนดในชั้นผลลัพธ์ (Output Layer) เมื่อผ่านฟังก์ชันกระตุ้นแล้ว และกำหนดให้ค่าเฉลยหรือค่าเป้าหมาย (Target) ของแต่ละตัวอย่างของข้อมูลฝึกสอนคือ T_k และกำหนดค่าความคลาดเคลื่อนเป็น δ (Delta)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในขั้นตอนนี้คือเมื่อได้ค่าข้อมูลผลลัพธ์จากกการคำนวณและทราบค่าเฉลยหรือค่าเป้าหมายที่กำหนดในแต่ละโหนดของชั้นข้อมูลผลลัพธ์ ในขั้นตอนการคำนวณไปข้างหน้า (Feed Pass) แล้ว จากนั้นคำนวณหาค่าความผิดพลาดรวม โดยใช้ค่าเฉลี่ยความผิดพลาดยกกำลังสอง (MSE) ของแต่ละโหนดข้อมูลผลลัพธ์ดังนี้ เริ่มจากการคำนวณค่าความผิดพลาดของ E_{o1} และ Eo_2 จากสมการที่ 2.18 หรือคำนวณหาค่าความผิดพลาดเฉลี่ยของทั้งสองโหนดข้อมูลผลลัพธ์ ตามสมการ 2.19 แต่เพื่อให้มีความเข้าใจง่ายขึ้น ผู้วิจัยได้เลือกใช้การคำนวณหาค่าความผิดพลาดตามสมการที่ 2.20 ของทั้งสองโหนดข้อมูลผลลัพธ์ แล้วนำมาหาค่าเฉลี่ยภายหลัง ดังนี้ $E_{O1} = \frac{1}{2}(t_1 - out_{o1})^2 = \frac{1}{2}(0.01 - 0.75136)^2 = 0.274811$ และ $E_{O2} = \frac{1}{2}(t_2 - out_{o2})^2 = \frac{1}{2}(0.99 - 0.7729)^2 = 0.002356$

เมื่อ E_{oi} คือค่าความผิดพลาดของแต่ละโหนดในชั้นข้อมูลออก t_i คือผลเฉลยหรือค่าเป้าหมายของแต่โหนดในชั้นข้อมูลออก out_{oi} คือค่าผลลัพธ์ที่คำนวณได้ในแต่โหนดในชั้นข้อมูลออก ดังนั้น $E_{total} = \frac{1}{2}(t_1 - out_{o1})^2 + \frac{1}{2}(t_2 - out_{o2})^2 = 0.274811 + 0.02356 = 0.298371$

2.8.3 การส่งผ่านย้อนกลับ (Backward Pass)

เป้าหมายของกระบวนการแพร่ย้อนกลับ คือการปรับค่าถ่วงน้ำหนักและค่าถ่วงน้ำหนักพิเศษ(ไบแอส) แต่ละตัวในโครงข่าย ตามตัวอย่างในรูปที่ 2-21 เพื่อให้ผลลัพธ์สุดท้ายในชั้นข้อมูลผลลัพธ์มีค่าใกล้เคียงกับค่าเป้าหมายหรือค่าเฉลยมากที่สุด และทำให้ความคลาดเคลื่อนโดยรวมของโครงข่ายลดลง ในการคำนวณค่าความผิดพลาดเทียบกับค่าถ่วงน้ำหนักหรือค่าไบแอสตามสมการที่ 2.21 โดยพิจารณาการแพร่ย้อนหลับในสองส่วนได้แก่ ชั้นข้อมูลผลลัพธ์ (Output Layer) และชั้นซ่อน (Hidden Layer) ดังนี้

1. การแพร่ย้อนกลับในชั้นข้อมูลผลลัพธ์ (Output Layer)

ในขั้นตอนนี้เมื่อเราจะได้ค่าความผิดพลาดรวม (E_{total}) โดยการคำนวณในขั้นตอนก่อนหน้าแล้ว ในขั้นตอนนี้เป็นการคำนวณย้อนกลับ โดยเริ่มจากชั้นข้อมูลผลลัพธ์ ตามรูปที่ 2-21 โดยการหาอนุพันธ์ย่อย (Partial differential) E_{total} เทียบกับค่าถ่วงน้ำหนักและค่าไบแอสแต่ละตัว ขั้นตอนแรกคือการหาค่าความลาดชัน (Gradient descent) ของค่าผิดพลาด เพื่อกำหนดทิศทางของการเรียนรู้ โดยสามารถหาได้จากการเทียบกับค่าถ่วงน้ำหนัก (weight) และค่าไบแอส (bias) ในโครงข่าย ในตัวอย่างตามรูปที่ 2-16 คือการหาอนุพันธ์ย่อยของ E_{total} ที่โหนด O_1 ในชั้นข้อมูลผลลัพธ์ (Output Layer) ได้ดังนี้ $\frac{\partial E_{total}}{Ew_5}$ โดยเลือกพิจารณาที่ค่าถ่วงน้ำหนัก w_5 ก่อน และในชั้นนี้จะปรับค่าไบแอสของโหนดข้อมูลออกและค่าถ่วงน้ำหนักที่เชื่อมระหว่างโหนดในชั้นข้อมูลออกกับโหนดในชั้นซ่อนก่อนหน้า แต่พบว่าสมการ E_{total} ไม่มีพารามิเตอร์ของ w_5 อยู่ จึงต้องแก้ปัญหาด้วยการกฎลูกโซ่ (Chain Rule) ตามสมการที่ 2.22

$$\nabla E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}\right] \quad (2.21)$$

เมื่อ $\nabla E(\vec{w})$ คือค่าเกรเดียนของค่าความผิดพลาดเทียบกับค่าถ่วงน้ำหนักต่าง ๆ w_0 เป็นค่าถ่วงน้ำหนักพิเศษหรือเรียกว่าค่าไบแอส (bias) ในบางตำรา

$$\frac{\partial E_{total}}{\partial W_5} = \frac{\partial E_{total}}{\partial out_{O1}} * \frac{\partial out_{O1}}{\partial net_{O1}} * \frac{\partial net_{O1}}{\partial W_5} \quad (2.22)$$

แยกการหาอนุพันธ์ได้ดังนี้ จาก $E_{total} = \frac{1}{2}(t_1 - out_{o1})^2 + \frac{1}{2}(t_2 - out_{o2})^2$

จากกฎลูกโซ่หาอนุพันธ์ของค่าความผิดพลาดรวม (E_{total}) เทียบกับค่าผลลัพธ์ในชั้นข้อมูลออกโหนดที่ 1 (out_{o1})

1) หาอนุพันธ์ของค่าผิดพลาดรวม (E_{total}) เทียบกับค่าผลลัพธ์ที่ผ่านฟังก์กระตุ้นแล้ว (out_{o1}) (ในตัวอย่างใช้ฟังก์ชันซิกมอยด์ ถ้าเป็นฟังก์ชันกระตุ้นประเภทอื่นค่าหลังการหาอนุพันธ์ย่อมเปลี่ยนไป) $\frac{\partial E_{total}}{\partial out_{O1}} = 2 * \frac{1}{2}(t_1 - out_{o1}) * (-1) + 0 = (out_{o1} - t_1) = 0.751365 - 0.01 = 0.741365$

2) หาอนุพันธ์ของผลลัพธ์ที่ผ่านฟังก์กระตุ้นแล้ว (out_{o1}) เทียบกับผลรวมของค่าที่ได้จากการคำนวณรวมกับค่าไบแอสในโหนดข้อมูลส่งออก (net_{o1}) เมื่อ $out_{o1} = \frac{1}{1+e^{-net_{o1}}}$

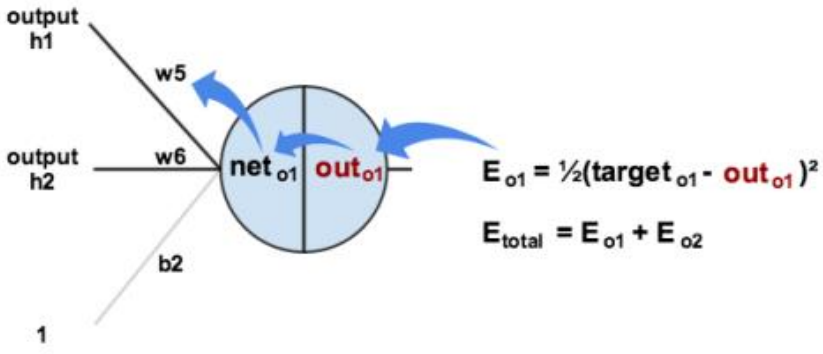
$$\frac{\partial out_{O1}}{\partial net_{O1}} = out_{o1}(1 - out_{o1}) = 0.751365(1 - 0.751365) = 0.186815$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3) หาอนุพันธ์ของผลรวมของค่าที่ได้จากการคำนวณรวมกับค่าไบแอสในโหนดข้อมูลส่งออก (net_{o1}) เทียบกับค่าถ่วงน้ำหนักที่เราสนใจ (ในตัวอย่างคือ W_5) $\frac{\partial net_{O1}}{\partial w_5} = out_{h1} = 0.59326$ ดังนั้นจากสมการ 2.22 จะได้ผลลัพธ์ที่เรียกว่า “อัตราการเปลี่ยนแปลงของ W_5” ดังนี้

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{W_5}} = \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{out_{O1}}} * \frac{\partial \mathrm{out_{O1}}}{\partial \mathrm{net_{O1}}} * \frac{\partial \mathrm{net_{O1}}}{\partial \mathrm{W_5}} = 0.741365 * 0.186815 * 0.59326 = 0.0821$$

หลังจากนั้นก็ทำการคำนวณในลักษณะเดียวกันให้ครบทุกค่าถ่วงน้ำหนักระหว่างชั้นข้อมูลส่งออก (Output Layer) กับชั้นซ่อน (Hidden Layer)



รูปที่ 2-16 การคำนวณค่าอัตราการผิดพลาดของการแพร่ย้อนกลับในชั้นข้อมูลผลลัพธ์ [17]

ในการหาค่าอนุพันธ์ย่อยของแต่โหนดสามารถใช้กฎเดลตา (Delta Rule) รวมการคำนวณได้ ดังนี้ $\frac{\partial E_{total}}{\partial out_{O_1}} * \frac{\partial out_{O_1}}{\partial Net_{O_1}} = \delta_{o1}$ ดังนั้นจะได้ว่าโหนดข้อมูลผลลัพธ์ที่1 (O_1) มีค่าของเดลตาเท่ากับ δ_{o1} ดังนั้นในการคำนวณหาค่าอนุพันธ์ย่อยของ E_{total} เทียบกับค่าถ่วงน้ำหนักที่ 5 (W_5) จึงสามารถทำการคำนวณได้ตามสมการที่ 2.23

$$\frac{\partial E_{total}}{\partial w_5} = \delta_{o1} * out_{h1} \tag{2.23}$$

จากนั้นทำการหาค่าเดลตาของทุกโหนดในชั้นข้อมูลผลลัพธ์ที่เหลือ เพื่อหาค่าการเคลื่อนลงตามความชัน (Gradient descent) สำหรับคำนวณหาค่าอนุพันธ์ย่อยเทียบกับค่าข้อมูลส่งออกจากโหนดในชั้นก่อนหน้า นอกจากการคำนวณหาค่าอนุพันธ์ค่าความผิดพลาดรวมเทียบกับค่าถ่วงน้ำหนักแต่ละตัวแล้ว ต้องหาค่าอนุพันธ์เทียบกับค่าไบแอส (Bias) ในแต่ละโหนดด้วย โดยใช้วิธีการเดียวกับค่าถ่วงน้ำหนัก แต่จะต่างกันตรงที่ไม่มีค่าที่ผ่านฟังก์ชันกระตุ้นมาเป็นข้อมูลขาเข้า ดังนั้นจึงคำนวณหาได้ตามสมการที่ 2.24 ซึ่งเป็นการคำนวณในโหนดที่ o_1 ในชั้นข้อมูลผลลัพธ์มีค่าไบแอส b_2

$$\frac{\partial E_{total}}{\partial b_2} = \frac{\partial E_{total}}{\partial out_{O1}} * \frac{\partial out_{O1}}{\partial b_2} \tag{2.24}$$

เมื่อครบทุกค่าถ่วงน้ำหนักและค่าไบแอสแล้วก็นำไปปรับค่าถ่วงน้ำหนักและค่าไบแอสใหม่ตามสมการที่ 2.25 โดยวิธีที่เป็นที่นิยมเรียกว่า Stochastic Gradient Descent (SGD) gradient descent คืออัลกอริทึมที่ใช้หาจุดต่ำสุดหรือสูงสุดของฟังก์ชันแบบสุ่ม เพราะการที่จะคำนวณ gradient แต่ละครั้งนั้น เราต้องใช้ข้อมูลทั้งหมดในการคำนวณเพื่อกำหนดทิศทาง ในการจะอัพเดตพารามิเตอร์ใหม่ ซึ่งใช้เวลามากกว่า ดังนั้นการใช้ SGD ในแต่ละการคำนวณจะทำการสุ่มข้อมูลเพียง

บางส่วน โดยการแบ่งเป็นกลุ่มข้อมูลย่อย (subsets) ที่เรียกว่า Mini-batch (1 batch คือข้อมูลทั้งหมด) คือไม่ได้ใช้ข้อมูลทั้งหมดมาคำนวณครั้งเดียว ทำให้สามารถทำงานได้ไวกว่า แต่ข้อจำกัดของ SGD คืออัตราการเรียนรู้ที่คงที่ตลอดเวลาการเรียนรู้ ตามสมการที่ 2.24 และ 2.25 โดยจากตัวอย่างเราสามารถหาค่าถ่วงน้ำหนักใหม่ของ w_5 ได้ดังนี้

$$w_5^+ = w_5 - \eta \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.082167 = 0.35891 \quad (2.25)$$

เมื่อ w_5^+ คือค่าถ่วงน้ำหนักใหม่ของ w_5 เมื่อผ่านการปรับแล้ว

η คืออัตราการเรียนรู้ (Learning rate) คือค่าที่กำหนดว่าในแต่ละรอบที่มีการปรับพารามิเตอร์(ค่าถ่วงน้ำหนัก และค่าไบแอส) จะมีการอัพเดตมากน้อยเพียงใด

เพื่อจะได้สามารถเข้าใกล้จุดต่ำสุด ดังนั้นค่าที่ใช้ในการอัพเดตหรือปรับค่าถ่วงน้ำหนักต้องเป็นค่าเกรเดียนที่ติดลบ (Negative gradient descent) เพราะในการคำนวณหาทิศทางของเกรเดียน (“Gradient” หมายถึงทิศทางที่จะมีความชันสูงขึ้น) จึงต้องกลับด้านด้วยการ ลบ (-) เพื่อให้ชี้ไปทางที่มีความชันต่ำกว่า

2. การแพร่ย้อนในชั้นซ่อน (Hidden Layer)

ขั้นตอนการคำนวณเหมือนในชั้นข้อมูลส่งออก แต่มีจุดที่แตกต่างที่ค่าเดลตา (δ) เพราะกฎลูกโซ่จะต้องเป็นการหาผลรวมของโหนดที่เกี่ยวข้องทั้งหมด และตัวแปรที่นำมาใช้ต้องมาจากชั้นถัดไป ในตัวอย่างคือชั้นข้อมูลส่งออก (Output Layer) เป็นชั้นถัดไปที่ผ่านฟังก์ชันกระตุ้นแล้วตามที่ในรูปที่ 2-17 โดยมีสมการการหาค่าเกรเดียนเทียบกับค่าถ่วงนำหนักตามสมการที่ 2.26 โดยเป็นตัวอย่างการเทียบกับค่าถ่วงน้ำหนัก w_1 และค่าผลรวมของโหนดที่เกี่ยวข้องตามสมการที่ 2.27

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1} \quad (2.26)$$

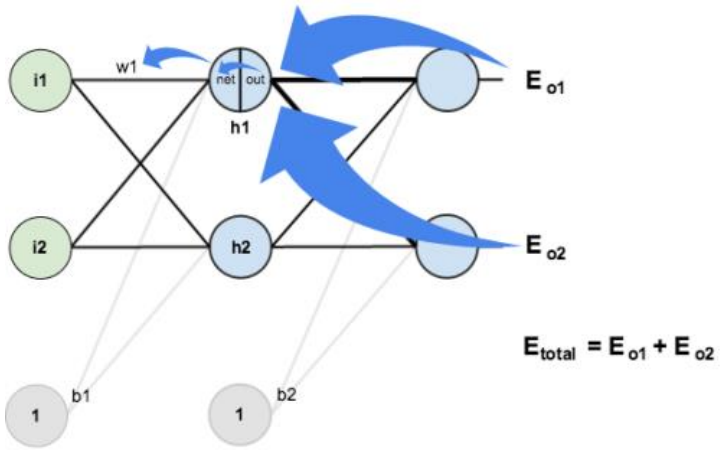
$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} \quad (2.27)$$

จะเห็นว่าในโหนด h_1 มีโหนดที่เกี่ยวข้องจำนวน 2 โหนดในชั้นข้อมูลส่งออกคือ E_{o1} , E_{o2} ตามรูปที่ 2-17 ซึ่งการทำการหาค่าอนุพันธ์เทียบกับข้อมูลส่งออกที่ผ่านฟังก์ชันกระตุ้นแล้วคือ out_{h1} ขั้นตอนต่อไปก็ทำการหาค่าอนุพันธ์เทียบกับค่า w_1 โดยสามารถสรุปขั้นตอนการคำนวณได้ตามสมการที่ 2.28 หรือ 2.29 ดังนี้

$$\frac{\partial E_{total}}{\partial w_1} = \left(\sum_o \frac{\partial E_{total}}{\partial out_o} * \frac{\partial out_o}{\partial net_o} * \frac{\partial net_o}{\partial out_{h1}}\right) * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1} \quad (2.28)$$

$$\frac{\partial E_{total}}{\partial w_1} = \delta_{h1} i_1 \quad (2.29)$$

เมื่อ δ_{h1} คือเดลตาของโหนด h_1 และ i_1 คือค่าข้อมูลขาเข้า (Input) ของโหนดที่ 1 หลังจากนั้นก็นำเข้าสมการที่ 2.28 และ 2.29 เพื่อหาค่าถ่วงน้ำหนักใหม่ของ w_1



รูปที่ 2-17 การคำนวณค่าอัตราการผิดพลาดของการแพร่ย้อนกลับในชั้นซ่อน [17]

ในงานวิจัยนี้เลือกใช้ TensorFlow เวอร์ชัน 2.0 ซึ่งเป็นไลบรารีสำหรับใช้พัฒนา machine learning ของทาง google ทำให้ไม่ต้องทำการเขียนในส่วนย้อนกลับ (Backward Pass) เพราะ TensorFlow จัดการให้ทั้งหมด ผู้วิจัยทำให้หน้าเขียนเฉพาะในส่วนของการคำนวณไปข้างหน้า (Forward Pass) เท่านั้น ในการเพิ่มประสิทธิภาพข่ายงานประสาเทียมของการเรียนรู้เชิงลึก (Optimization) ในการปรับค่าพารามิเตอร์ต่าง ๆ ใช้ Adaptive Moment Estimation (Adam) เพราะเป็น optimizer ที่สามารถปรับอัตราการเรียนรู้ สำหรับพารามิเตอร์ในแต่ละครั้งได้และยังสามารถแก้ปัญหา decaying ของ gradients แทนการใช้ SGD เพราะมีอัตราการ converge รวดเร็ว และให้ผลที่แม่นยำกว่า และที่สำคัญ TensorFlow รองรับการทำงาน

2.9 การเรียนรู้เชิงลึก (Deep Learning) [18]

ในโครงข่ายประสาทเทียม เริ่มจากเพอร์เซปตรอน (Single Layer Perceptron) ที่ใช้ในการแก้ปัญหาเชิงเส้นในลักษณะ “ใช่” หรือ “ไม่” จนพบว่าเกิดปัญหา XOR ที่ไม่สามารถแก้ปัญหาด้วยการใช้เพียงสมการเส้นตรงได้ จึงทำการเพิ่มความสามารถของโครงข่ายให้มากขึ้นด้วยการมีชั้นซ่อน

(Hidden Layer) ขึ้นมาระหว่างชั้นข้อมูลขาเข้า (Input Layer) เพื่อแก้ปัญหาที่มีความซับซ้อนและไม่เป็นเชิงเส้นได้ จนเกิดเป็นโครงข่ายที่มีมากกว่าหนึ่งชั้นเรียกว่า Multi-Layer Perceptron (MLP) ร่วมกับใช้ฟังก์ชันกระตุ้นแบบไม่เป็นเชิงเส้น (Non Linear Activation Function) เพื่อดึงความสามารถของชั้นซ่อนที่เพิ่มขึ้นมาให้ทำงานได้ มีประสิทธิภาพมากขึ้น เพราะสามารถนำชั้นซ่อนหลาย ๆ ชั้นมาต่อกันได้ และมีการใช้เมทริกซ์วัดประสิทธิภาพของแบบจำลองของโครงข่าย (model) ที่ได้ทำการเรียนรู้จากชุดข้อมูลสอน (Training set) ว่าค่าผลลัพธ์ที่ได้จากการคำนวณและค่าผลเฉลยมีค่าแตกต่างกันอย่างไร โดยฟังก์ชันการสูญเสีย (Loss Function) ทั้งปัญหาเลขจำนวนจริง (Regression) หรือการจำแนกประเภทข้อมูล(Classification) เมื่อได้ค่าความผิดพลาดจากฟังก์ชันการสูญเสียแล้ว วัตถุประสงค์ของการสร้างแบบจำลองในโครงข่ายประสาทเทียมคือลดค่าความสูญเสียลงให้อยู่ในเกณฑ์ที่ยอมรับได้ โดยการปรับค่าพารามิเตอร์ต่าง ๆ ในโครงข่าย ได้แก่ค่าถ่วงน้ำหนัก (Weight) และค่าไบแอส (Bias) เรียกกระบวนการนี้ว่าการแพร่ย้อนกลับ ด้วยอัลกอริทึมปรับปรุงประสิทธิภาพ (Optimization Algorithm) ซึ่งมีหลายชนิด ทำงานร่วมกับอัตราการเรียนรู้ (Learning Rate) แต่ใช้หลักการคล้าย ๆ กัน คือการหาจุดต่ำสุด (จุดต่ำสุด คือจุดที่มีค่าการสูญเสียจากฟังก์ชันการสูญเสียน้อยที่สุด) ด้วยการนำทางตามวิธีการหาค่าความลาดชัน Gradient descent โดยค่าความชันคืออนุพันธ์ย่อยของ ค่าที่ได้จากฟังก์ชันการสูญเสีย เช่น MSE ถือเป็นความสูง เทียบกับตำแหน่ง (ค่าถ่วงน้ำหนัก, และค่าไบแอส)

โครงข่ายประสาทเทียมแบบหลายชั้น (MLP) หรือ Deep Neural Network (DNN) ได้ถูกพัฒนาปรับปรุงเพิ่มประสิทธิภาพ ด้วยอัลกอริทึมปรับปรุงความสามารถแบบจำลองในโครงข่าย ให้สามารถทำงานได้มีประสิทธิภาพมากขึ้น และลดขั้นตอนการสกัดคุณลักษณะ (Feature Extraction) ข้อมูลขาเข้าในชั้นข้อมูลขาเข้า (Input Layer) จนเกิดเทคนิคที่เรียกว่า “การเรียนรู้เชิงลึก” ที่ความสามารถในการหาจุดเด่นของข้อมูลขาเข้า (Input) หรือพยายามเรียนรู้ที่จะหาตัวแทนของข้อมูลเพื่อให้แบบจำลองมีประสิทธิภาพมากที่สุด เช่นในงานประมวลผลภาพ ภาพหนึ่งภาพสามารถแทนได้เป็นเวกเตอร์ของความสว่างต่อจุดพิกเซล หรือการจำแนกประเภทข้อความ (Text Classification) ในงานประมวลผลภาษาธรรมชาติ (Natural Language Processing) การสร้างตัวแทนเวกเตอร์ข้อความ (Text Representation) สามารถทำได้หลายวิธี เช่น การนับคำจากคลังคำศัพท์ (Count Words) การหาน้ำหนักของคำ (TF-IDF) เพื่อหาคำที่เด่น ๆ ออกมา เป็นต้น หรือวิธีการอื่น ๆ แต่ทั้งหมดต้องอาศัยมนุษย์เป็นคนสกัดคุณลักษณะเหล่านี้ และการเรียนรู้เชิงลึกก็เป็นเทคนิคที่ถูกนำมาใช้ในการสร้างตัวแทนของข้อมูลขาเข้า (Input) ระดับคำ (Word) แทนการคัดแยกคุณลักษณะ (Feature) ด้วยมนุษย์ โดยให้อัลกอริทึมของการเรียนรู้เชิงลึก (Deep Learning) ได้เรียนรู้คุณลักษณะที่เหมาะสมแล้วสร้างเวกเตอร์ตัวแทน (Vector Representation of Words) ของคำเหล่านั้นเอง หากต้องการแบ่งระดับของการสร้างตัวแทนของคำตามหลักของภาษาศาสตร์ก็สามารถทำได้ โดยการ

เพิ่มชั้นหน่วยย่อยของโครงข่ายประสาทเทียม เพื่อให้การเรียนรู้เชิงลึกสามารถเรียนรู้ได้มากขึ้น ในการประมวลผลตั้งแต่ระดับพยางค์ คำ วลี ประโยค หรืออื่น ๆ ก็สามารถทำได้ เพียงใส่ข้อมูลเข้าไปให้แบบจำลองเรียนรู้ [19] ในงานวิจัยนี้ใช้ตัวแทนข้อความในระดับคำ (Word Representation)

2.10 ตัวแทนของข้อความ (Text Representation) [19]

ตัวแทนของข้อความคือการอธิบายความหมายของข้อความ เพื่อเป็นข้อมูลตัวแทน (Feature) โดยงานด้านการประมวลผลภาษาธรรมชาติ (NLP) ตัวแทนข้อความที่สร้างขึ้นสามารถพิจารณาได้ตั้งแต่ระดับตัวอักษร (Characters) คำ (Words) ประโยค (Sentences) ในการนำเข้าประมวลผลในโครงข่ายประสาทเทียมต่อไป แต่ในงานวิจัยนี้ใช้ระดับคำในการสร้างตัวแทนข้อความ โดยแบ่งประเภทของการสร้างตัวแทนข้อความเป็น 2 ประเภทหลัก ดังนี้

2.10.1 Lexical Semantics คือการศึกษาความหมายของคำ เพื่ออธิบายคำศัพท์นั้น ๆ ให้แตกต่างจากคำอื่น ๆ แบ่งออกเป็น 2 ประเภทได้แก่

1. Taxonomic Relations คือความสัมพันธ์เชิงอนุกรมวิธาน เพื่อใช้ในการอธิบายคำแต่ละคำด้วยการสร้างความเชื่อมโยงคำที่มีความหมายแบบ Hyponym (Subordinate) คือ sub class ของอีกคำหนึ่ง และแบบ Hypernym (Superordinate) คือ Class ใหญ่ของ Hyponym เช่น

2. One-hot Encoding การสร้างตัวแทนของแต่ละคำที่เป็นเวกเตอร์ด้วยเทคนิคการดูว่า ถ้าคำไหนใช่ก็แทนด้วยเลข “1” ถ้าไม่ใช่ก็แทนด้วยเลข “0” ซึ่งการเป็นตัวแทนของคำใดจะเป็น”1” เพียงตัวเดียว (1 มิติ) เท่านั้น ส่วนมิติที่เหลือเป็น “0” ทั้งหมด ข้อจำกัดคือไม่สามารถคำนวณความสัมพันธ์ของคำที่ใกล้เคียงกันในทางความหมายได้ นอกจากนี้ยังใช้หน่วยความจำจำนวนมากในการอธิบายแต่ละคำเพราะจะมีขนาดของเวกเตอร์เท่ากับ 1xN เมื่อ N คือจำนวนคำศัพท์ทั้งหมดในคลังคำศัพท์ ในการเก็บค่า “0” ที่มีเป็นส่วนใหญ่ เช่น กำหนดให้เอกสารข่าวจำนวน 2 ข่าว d1, d2 และคลังคำศัพท์ D (Dictionary) ดังนี้

d1= จำเลยยิงผู้ตายในทันที

d2= ผู้ตายทำร้ายจำเลยก่อนถูกยิง

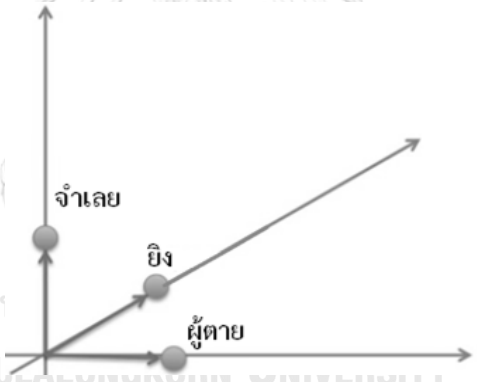
D = {'จำเลย': 1, 'ยิง': 5, 'ผู้ตาย': 4, ในทันที': 6, 'ทำร้าย': 3, 'ก่อน': 0, 'ถูก': 2}

โดยจะเรียงคำศัพท์ตามความถี่มากไปน้อย และกำหนดหมายเลขของแต่ละคำ (Word ID) จากตัวอย่างคือ 0 ถึง 6 เพราะมีคำศัพท์ที่ไม่ซ้ำกันจำนวน 7 คำ ดังนั้นในการแสดงตัวแทนของแต่ละคำจะแทนด้วยเวกเตอร์ขนาด 1x7 ตามรูปที่ 2-18 ตัวอย่าง ตัวแทนของคำว่า “จำเลย” ในคลังข้อมูลนี้ถูกแทนด้วยเวกเตอร์ [1 0 0 0 0 0 0] แต่ในงานประมวลผลภาษาธรรมชาติ ซึ่งมีคำศัพท์จำนวนมาก จึงต้องใช้หน่วยความจำมากเพื่อรองรับมิติของคำขนาด N x N นอกจากกนี้มิติของ

แต่ละคำใน Vector Space ตามรูปที่ 2-19 ทุกมิติตั้งฉากกันทำให้การค่าค่าความสัมพันธ์ระหว่างคำหรือบริบทโดยรอบไม่ได้

	One-Hot Vector							
จำเลย	[1	0	0	0	0	0	0	]
ยิง	[0	1	0	0	0	0	0	]
ผู้ตาย	[0	0	1	0	0	0	0	]
ในทันที	[0	0	0	1	0	0	0	]
ทำร้าย	[0	0	0	0	1	0	0	]
ก่อน	[0	0	0	0	0	1	0	]
ถูก	[0	0	0	0	0	0	1	]

รูปที่ 2-18 แสดง วัน-ฮอท เวกเตอร์ของคำศัพท์ในคลังข้อมูล



รูปที่ 2-19 ความสัมพันธ์ของข้อมูลแต่ละคำในเวกเตอร์สเปซ (Vector space)

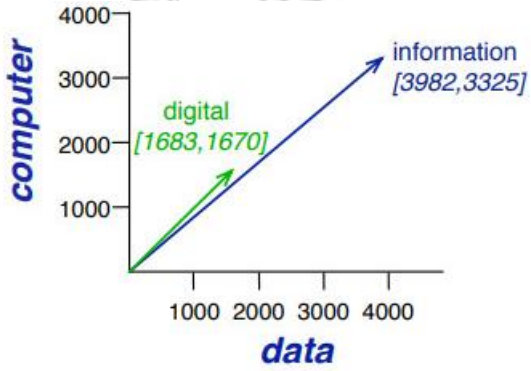
2.10.2 Vector Semantics คือการอธิบายคำหนึ่งคำด้วยบริบทโดยรอบหรือใช้ความสัมพันธ์ กับคำอื่น ๆ ที่ใกล้เคียงกัน เพื่อสร้างเป็นเวกเตอร์สำหรับคำ ๆ นั้น ดังนี้

1. Pointwise Mutual Information (PMI) คือการหาค่าความน่าจะเป็นที่คำสองคำจะปรากฏพร้อมกันในคลังคำศัพท์ (Co-occurrence Matrix) ตามสมการที่ 2.30 หรือกรณีที่ค่าที่ได้น้อยกว่าศูนย์ ถือว่าเป็นค่าที่ไม่น่าเชื่อถือเราจะไม่ให้ความสำคัญ ดังนั้นจึงเปลี่ยนค่าติดลบเป็นค่าศูนย์ตามสมการที่ 2.31 เรียกว่า PPMI (Positive Mutual Information) คือถ้าค่าของ PPMI มีค่าเข้าใกล้ 1 หมายถึงคำสองคำนั้นมีความเกี่ยวข้องกันมาก เพราะคำที่มีความหมายใกล้เคียงกันย่อมเกิดขึ้น

พร้อมกับคำอื่นในบริบทใกล้ ๆ กันนั่นเอง กรณีที่สนใจในระดับคำ (Words Level) เพื่อแก้ปัญหาของการนับเฉพาะความถี่ระดับคำ (Term Frequency) โดยตรง ตามรูปที่ 2-20

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

รูปที่ 2-20 Co-occurrence matrix ในระดับคำ [19]



รูปที่ 2-21 การหาความเหมือนของคำสองคำ (similarity) [19]

ตามรูปที่ 2-20 พบคำว่า digital เกิดพร้อมคำบริบท (Context word) computer จำนวน 1,670 ครั้ง และ data จำนวน 1,683 ครั้ง และคำว่า information เกิดพร้อมคำบริบท computer จำนวน 3,325 ครั้ง และบริบท data จำนวน 3,982 ครั้ง จึงสามารถนำเวกเตอร์ของทั้งสองคำคือ digital และ information มาคำนวณหาค่าความใกล้เคียงกันได้ ตามกราฟในรูปที่ 2-21

$$PMI_{(w,c)} = log_2 \frac{P(w,c)}{P(w)P(c)} \tag{2.30}$$

$$PPMI_{(w,c)} = max\ (log_2 \frac{P(w,c)}{P(w)P(c)}, 0) \tag{2.31}$$

เมื่อ $PMI_{(w,c)}$ คือความน่าจะเป็นที่อคำเป้าหมาย และคำบริบทเกิดขึ้นพร้อม ๆ กัน

$PPMI_{(w,c)}$ คือความน่าจะเป็นที่คำสองคำ คือคำเป้าหมายและคำบริบทเกิดขึ้นพร้อม ๆ กัน ที่ไม่สนใจค่าที่เป็นลบ เพราะถือว่าไม่มีความน่าเชื่อถือ

w คือ คำเป้าหมาย (Target Word)

c คือ คำบริบทรอบ ๆ (Context Word)

$P(w,c)$ คือ ความน่าจะเป็นที่ คำเป้าหมายและคำบริบทเกิดขึ้นพร้อมกัน

$P(w)$ คือ ความน่าจะเป็นที่เกิดคำเป้าหมาย

$P(c)$ คือ ความน่าจะเป็นที่เกิดคำบริบท

เพื่อแก้ปัญหาของ Co-occurrence ที่เกิดขึ้น เรื่องคำฟุ่มเฟือย เช่น “การ”, “ความ” ที่เกิดขึ้นบ่อยหรือมีความถี่จำนวนมาก สามารถนำเข้าคำนวณและจัดการค่าติดลบตามสมการที่ 2.31 ได้เป็นตาราง PPMI ตามรูปที่ 2-22

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

รูปที่ 2-22 แสดง PPMI Matrix ระหว่างคำที่เกิดขึ้น [19]

ตัวอย่างการคำนวณเพื่อหาค่า PPMI ของคำว่า information และ data ดังนี้

$$P_{(w=information,c=data)} = \frac{3982}{11716} = 0.3399$$

$$P_{(w=information)} = \frac{7703}{11716} = 0.6575$$

$$P_{(c=data)} = \frac{5673}{11716} = 0.4842$$

$$PPMI_{(information,data)} = max\left(log_2 \frac{0.3399}{(0.6575)*(0.4842)}, 0\right) = 0.944$$

2. TF-IDF คือวิธีการหาคำ (t) ที่มีความสำคัญในเอกสาร (d) โดยพิจารณาจากคำศัพท์ทั้งหมดในทุกเอกสาร แทนการใช้การหาความถี่ของคำที่เกิดในแต่ละเอกสาร (Term Document Matrix) อย่างเดียว ตามรูปที่ 2-23 ประกอบด้วย TF (Term Frequency) คือการนับความถี่ของคำในแต่ละเอกสาร ตามสมการที่ 2.32 หรือใช้ log ฐานสิบ ในการปรับค่าผลลัพธ์ ในกรณีข้อมูลที่จำนวนความถี่ของคำมีช่วงต่างกันมากตามสมการที่ 2.33 กำหนดให้ $tf_{(t,d)}$ คือจำนวนครั้งที่คำศัพท์ t ปรากฏในเอกสาร d

$$tf_{(t,d)} = count(t,d) \tag{2.32}$$

$$tf_{(t,d)} = log_{10}(count(t,d) + 1) \tag{2.33}$$

และ IDF (Inverse Document Frequency) คือส่วนกลับของคำศัพท์ที่ความถี่น้อยในเอกสารจะมีความสำคัญมากกว่าที่มีความถี่มาก โดยมีสูตรการคำนวณตามสมการที่ 2.34 หรือใช้

log ฐานสิบ ในการปรับค่าผลลัพธ์ให้เหมาะสม ตามสมการที่ 2.35 จากนั้นนำค่า $tf_{(t,d)}$ คูณด้วย idf_t เพื่อหาค่า $w_{t,d}$ ตามสมการที่ 2.36 โดยได้ค่าถ่วงน้ำหนักของแต่ละคำจากการคำนวณด้วย TF-IDF ตามรูปที่ 2-24 โดยกำหนดให้ $w_{t,d}$ คือ ค่าถ่วงน้ำหนักของแต่ละคำ df คือ ความถี่ของคำศัพท์แต่ละคำ และ N คือ จำนวนเอกสาร (d) ซึ่งพบว่า TF-IDF เป็นวิธีในการคัดเลือกคำที่สำคัญในคลังคำศัพท์ออกมา

$$idf_t = \frac{N}{df_t} \tag{2.34}$$

$$idf_t = log_{10}(\frac{N}{df_t}) \tag{2.35}$$

$$w_{t,d} = tf_{(t,d)} * idf_t \tag{2.36}$$

	d1	d2
จำเลย	1	1
ยิง	1	0
ผู้ตาย	1	1
ในทันที	1	0
ทำร้าย	0	1
ก่อน	0	1
ถูก	0	1

รูปที่ 2-23 ความถี่ของแต่ละคำในแต่ละเอกสาร

ดังนั้น ตัวแทนข้อความของเอกสาร d1 และ d2 ของ term-document matrix ได้แก่ d1 = [1 1 1 1 0 0 0], d2 = [1 0 1 0 1 1 1]

คำศัพท์	TF		IDF		TF-IDF	
	d1	d2			d1	d2
จำเลย	1/7	1/7	log(2/2) =	0	0	0
ยิง	1/7	1/7	log(2/2) =	0	0	0
ผู้ตาย	1/7	1/7	log(2/2) =	0	0	0
ในทันที	1/7	0/7	log(2/1) =	0.301029	0.043	0
ทำร้าย	0/7	1/7	log(2/1) =	0.301029	0	0.043
ก่อน	0/7	1/7	log(2/1) =	0.301029	0	0.043
ถูก	0/7	1/7	log(2/1) =	0.301029	0	0.043

รูปที่ 2-24 การคำนวณค่า TF-IDF จากเอกสาร 2 ชุด

ดังนั้น ตัวแทนข้อความของเอกสาร d1 และ d2 ของ TF-IDF ได้แก่

d1 = [0 0 0 0.043 0 0 0] , d2 = [0 0 0 0 0.043 0.043 0.043]

ปัญหาของการนับเฉพาะความถี่ใน term-document คือคำฟุ่มเฟือยต่าง ๆ เช่น “ความ” หรือ “การ” ที่ไม่ใช่คำสำคัญแต่กระจายอยู่ในเอกสารจำนวนมาก ทำให้การใช้คำที่สำคัญเกิดความผิดผลาดได้ และนอกจากนี้คำที่เราสนใจไม่อาจทราบความสำคัญได้ เพราะต้องทราบคำทั้งหมดในเอกสารนั้น ๆ เพื่อใช้เปรียบเทียบ

3. เอ็นแกรม (N-Grams) เป็นแบบจำลองที่ใช้คำนวณหาโอกาสในการเกิดร่วมกันของสายอักขระ (Character Sequence) ในการประกอบเป็นคำ หรือโอกาสที่แต่ละคำเรียงติดต่อกัน (Word Sequence) เพื่อประกอบเป็นประโยค โดยใช้การเปรียบเทียบจากคลังข้อมูลที่สร้างไว้ โดยข้อดีคือการเรียงตัวกันของคำหรืออักขระมีความสำคัญ ทำให้สามารถเรียนรู้ความสัมพันธ์ของคำที่ติดกันได้ โดยแกรม คือหน่วยที่ใช้ในการสร้างแบบจำลองมีค่าได้ตั้งแต่ 1 ถึง N แกรม ในการคำนวณคาความนาจะเปนของสายคำหรือสายอักขระจากคลังขอมูลสามารถทำได้ดังนี้ ไบแกรม (Bi-Gram) คำนวณจากผลรวมของค่าความน่าจะเป็นที่จะมีคำสองคำหรือสองอักขระนั้นเกิดขึ้นติดกันตามสมการที่ 2.37 แต่ข้อจำกัดของเอ็นแกรม คือต้องสร้างตารางนับตั้งแต่ยูนิแกรม (Uni-grams) ขึ้นมา จนถึงค่าประมาณ N แกรม ที่ต้องการ ทำให้ใช้เนื้อที่หน่วยความจำมาก และมีค่าที่เป็นศูนย์ในเวกเตอร์ของแต่ละคำจำนวนมาก

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}W)} \quad (2.37)$$

เมื่อ P คือ ค่าความน่าจะเป็น ซึ่งคำนวณจากคลังคำศัพท์ (Corpus)

n คือ ลำดับของคำ โดยเริ่มจาก 1

C คือ จำนวนของคำปัจจุบันและก่อนหน้าเกิดขึ้นติดกัน

W คือจำนวนคำทั้งหมด

ตามสมการที่ 2.30 เป็นสมการของการประมาณค่าความน่าจะเป็นด้วยไบแกรม (Bi-gram) คือพิจารณาคำสองคำที่เกิดขึ้นติดกัน

ข้อจำกัดของ TF-IDF และ PMI คือเวกเตอร์มีขนาดใหญ่มาก เพราะต้องใช้คำทั้งหมดในคลังคำศัพท์ เพื่ออธิบายคำหนึ่ง ๆ เป็นที่มาของ Sparse Vectors คือเต็มไปด้วยเลข 0 จำนวนมากในแต่ละเวกเตอร์ ทำให้โอกาสในการหาค่าความเหมือน (Similarity) ของแต่ละเวกเตอร์มีโอกาสผิดพลาดได้ จนนำมาสู่การใช้ Dense Vectors ที่จะกล่าวถึงในหัวข้อถัดไป

4. การฝังคำ (Word Embedding) เป็นแปลงคำให้อยู่ในรูปของเวกเตอร์ที่สามารถกำหนดจำนวนมิติหรือความยาวเวกเตอร์ของแต่ละคำได้ ทำให้สามารถลดจำนวนเวกเตอร์สเปซได้ เนื่องจากค่าที่ได้ในแต่ละมิติมาจากการสุ่มและไม่มีเลขศูนย์กระจายอยู่จำนวนมาก ทำให้มีประสิทธิภาพดีกว่าการใช้ One-Hot Vector ที่มีเลขศูนย์กระจายอยู่จำนวนมาก ดังนั้นการนำเข้าฝึกสอนในโครงข่ายประสาทเทียมจึงเหมาะสมกว่า เพราะมีพารามิเตอร์น้อยกว่าทำให้ช่วยป้องกันปัญหา Overfitting ได้ และจากแนวคิดที่คำแต่ละคำ (target word) อยู่ในบริบท (Context Words) ที่คล้ายกันจะมีความหมายของคำใกล้เคียงกัน ทำให้สามารถคำนวณหาความสัมพันธ์ของแต่ละคำได้ เทคนิคในการฝังคำมีหลายวิธี เช่น word2vec, GloVe และ FastText เป็นต้น ในงานวิจัยนี้ใช้เทคนิคการฝังคำเพื่อสร้างตัวแทนระดับคำ ในการประมวลผลของโครงข่ายประสาทเทียมเชิงลึก เมื่อฝึกสอนเรียบร้อยแล้วจะได้ 2 Embedding สำหรับหนึ่งคำศัพท์ คือ คำเป้าหมาย (Target) กับคำบริบท (Context) จากนั้นเก็บค่า Embedding ของคำเป้าหมายไว้ เพื่อใช้กำหนดเป็นเวกเตอร์ของแต่ละคำ ตามรูปที่ 2-25

คำศัพท์	Word Embedding | Skip-gram | 5-Dimensional | Windows size =1
จำเลย	[-0.0339946 0.04620537 -0.05539387 -0.02921459 0.01968751]
ยิง	[0.01247135 0.05800343 0.03198203 0.00089615 0.03117844]
ผู้ตาย	[0.04387545 0.07780102 -0.0434113 -0.08653262 -0.09533704]
ในทันที	[-0.07642165 0.01092633 -0.03444172 0.02639089 0.00934697]
ทำร้าย	[-0.05945263 -0.07573318 0.05409263 0.09808674 -0.09009952]
ก่อน	[-0.06807121 -0.02128514 -0.0806593 0.04808 -0.00908056]
ถูก	[0.06346591 0.09795233 -0.02825146 0.02294475 -0.01412445]

รูปที่ 2-25 แสดงการฝังคำด้วยเทคนิค Word2Vec

CHULALONGKORN UNIVERSITY

2.11 อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimization Algorithms) [20]

กระบวนการแพร่ย้อนกลับ (Backpropagation) คือกระบวนการในการปรับค่าพารามิเตอร์ที่ประกอบด้วยค่าถ่วงน้ำหนักและค่าไบแอส ที่เชื่อมต่อแต่ละนิวรอนในโครงข่ายประสาทเทียม เพื่อให้ได้ค่าที่ถูกต้องใกล้เคียงค่าเป้าหมายและมีความถูกต้องมากขึ้น และในการคำนวณหาค่าพารามิเตอร์นี้คือการใช้สิ่งที่เรียกว่า อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimization Algorithms) หรือเรียกว่า ออปติไมเซอร์ (Optimizer) ที่เป็นตัวที่ทำหน้าที่ในการปรับปรุงค่าต่าง ๆ ในโครงข่าย ในลักษณะตรงข้ามกับทิศทางของความชัน โดยทำการเรียนรู้ไปเรื่อย ๆ เพื่อหาจุดต่ำสุด และจะหยุดการเรียนรู้เมื่อค่าของฟังก์ชันการสูญเสียนั้นต่ำกว่าค่าขีดแบ่ง (threshold) ที่กำหนด ดังนี้

2.11.1 การแบ่งประเภทของวิธีการเคลื่อนลงตามความชัน (Gradient Descent Variants) เป็นวิธีการในการคำนวณหา จุดที่ต่ำที่สุดของฟังก์ชันการสูญเสีย ด้วยการหาความชัน (Derivative) ทีละจุดแล้วค่อยเลื่อนไปในจุดที่ต่ำกว่า ไปจนถึงจุดที่ต่ำที่สุดที่เป็นไปได้ โดยจุดที่ต่ำที่สุดจะมี 2 แบบ ได้แก่ Global Minimum (จุดที่ต่ำที่สุดจริง ๆ) และ Local Minimum (จุดที่ต่ำที่สุดในระแวกนั้น) การที่จะหา Global Minimum ยากกว่าการหา Local Minimum เพราะบางที Gradient Descent Algorithm อาจมอง Local Minimum เป็นจุดที่ต่ำที่สุดจริง ๆ

สามารถแบ่งการวิธีการหาค่าการเคลื่อนลงความชันได้ 3 ประเภทตามปริมาณของข้อมูลที่ป้อนเข้าสู่โครงข่ายให้แบบจำลองเรียนรู้ ดังนี้

1. Batch Gradient Descent เป็นวิธีการหาจุดที่ต่ำที่สุดโดยการป้อนข้อมูลฝึกสอนทั้งชุด (Entire Training Dataset) แล้วค่อยทำการปรับค่าถ่วงน้ำหนักและค่าไบแอสครั้งเดียว คือทำการอัพเดตค่าถ่วงน้ำหนักและค่าไบแอส 1 ครั้ง ต่อข้อหนึ่งรอบการเทรน (1 Iteration/Epoch) จุดเด่นของวิธีนี้คือสามารถเข้าหาจุดต่ำสุด (Global Minimum) ได้ดี เพราะได้เรียนรู้จากข้อมูลทั้งหมดแล้ว จึงทำการอัพเดตค่าพารามิเตอร์ แต่มีจุดด้อยคือถ้าขนาดของข้อมูลมีจำนวนมาก ทำให้เวลาโหลดเข้าหน่วยความจำที่มีขนาดจำกัดและถ้ามีขนาดไม่เพียงพอข้อมูลจะถูกทดเข้าสู่ฮาร์ดดิสก์ทำให้ใช้เวลานานกว่าที่จะเรียนรู้และปรับค่า พารามิเตอร์ แต่ถ้าเทียบช่วงเวลา Running Time วิธีนี้จะเร็วที่สุด สมการการปรับค่าพารามิเตอร์ของ Batch Gradient Descent แสดงในสมการที่ 2.38

$$\theta = \theta - \eta \nabla_{\theta} J(\theta) \tag{2.38}$$

2. Stochastic Gradient Descent เป็นวิธีการหาจุดที่ต่ำที่สุดโดยการป้อนข้อมูลฝึกสอนทีละตัวอย่าง (Each Training Example) และทำการปรับพารามิเตอร์ทันที คือทำการอัพเดตค่าถ่วงน้ำหนักและค่าไบแอส 1 ครั้ง ต่อข้อมูล 1 ตัว จุดเด่นของวิธีนี้คือถ้าเทียบเวลาตั้งเริ่มโหลดข้อมูลลงหน่วยความจำจนถึงการเรียนรู้และปรับค่าพารามิเตอร์จะเร็วกว่า Batch Gradient Descent ส่วนจุดด้อยของวิธีนี้คือเมื่อต้องป้อนข้อมูลที่ละตัวตัวอย่างในรอบการเรียนรู้ทำให้เกิดการรบกวน (Noise) ค่อนข้างมาก ทำให้ค่าของฟังก์ชันการสูญเสียที่ได้มีความแปรปรวนสูง การ Converge สู่จุดต่ำสุดจึงไม่เสถียร แต่ขณะเดียวกันก็ทำให้สามารถหลบจุดต่ำสุดในระแวกนั้นได้ (Local minima) และในช่วงเวลาของ Running Time เมื่อเทียบกับแบบแรกต้องใช้เวลามากกว่า โดยมีสมการการปรับค่าพารามิเตอร์ตามสมการที่ 2.39

$$\theta = \theta - \eta \nabla_{\theta} J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{2.39}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3. Mini-Batch Gradient Descent วิธีการนี้ได้นำจุดเด่นของทั้ง Batch Gradient Descent และ Stochastic Gradient Descent รวมกัน ทำให้เป็นวิธีที่ดีที่สุด แต่มีจุดด้อยคือการกำหนดค่าไฮเปอร์พารามิเตอร์ เช่น อัตราการเรียนรู้ หรือขนาดของ Batch size ที่ใช้ฝึกสอนว่าต้องมีหลักการหรือวิธีการตั้งค่าไฮเปอร์พารามิเตอร์เหล่านี้อย่างไรจึงจะให้ค่าของฟังก์ชันการสูญเสียและใช้เวลาน้อยที่สุด สามารถทำได้โดยการพลอตบนกราฟระหว่างฟังก์ชันการสูญเสีย (Loss function) กับเวลาว่าตำแหน่งใดบนกราฟที่ให้ค่าที่ดีที่สุด ซึ่งจะบอกขนาดของ Batch size นั่นเอง โดยส่วนมากนิยมกำหนดขนาดตั้งแต่ 50-256 แต่ก็ขึ้นกับข้อมูลที่นำมาทดสอบ ตามสมการที่ 2.40

$$\theta = \theta - \eta \nabla_{\theta} J\left(\theta; x^{(i:i+n)}; y^{(i:i+n)}\right) \tag{2.40}$$

2.11.2 การปรับค่าอัลกอริทึมความลาดชัน (Gradient descent optimization algorithms) ในหัวข้อที่ผ่านประเภทของเกรเดียนดีเซนต์แบ่งตามจำนวนของข้อมูลที่ป้อนเข้าสู่โครงข่าย และการการเปลี่ยนแปลงค่าของพารามิเตอร์จะขึ้นกับอัตราการเรียนรู้ (Learning rate , η) และค่าอัตราการเรียนรู้นี้จะคงที่ตลอดการเรียนรู้ในทุกรอบ (Epoch) ต่อมาได้มีผู้คิดค้นเพื่อให้การเปลี่ยนแปลงค่าพารามิเตอร์ขึ้นกับปัจจัยอื่น ๆ เพื่อให้เกรเดียนดีเซนต์ทำงานได้มีประสิทธิภาพดีขึ้น ได้แก่

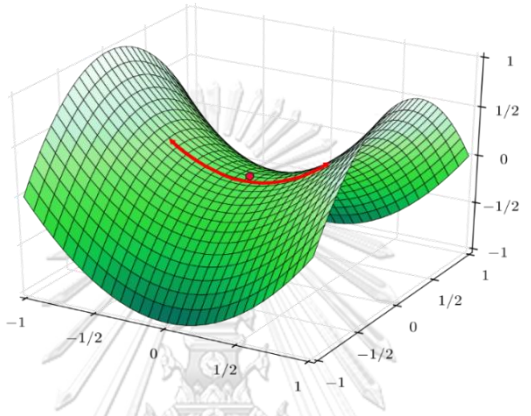
1. โมเมนตัม (Momentum) เป็นวิธีการที่ใช้ในการปรับปรุงการเคลื่อนลงตามความชัน จากเดิมที่การการปรับค่าพารามิเตอร์จะขึ้นกับอัตราการเรียนรู้และความชันเท่านั้น ในการปรับค่าในแต่ละครั้งของ SGD จะสร้างความแปรปรวนเกิดขึ้นมาก ทำให้ยากที่ลู่เข้าจุดที่ต่ำสุดได้ โดยโมเมนตัมเป็นวิธีในการปรับปรุง บริเวณที่เป็นฟังก์ชันการสูญเสีย ($\theta = \gamma v_{t-1}$) ดังที่แสดงในสมการที่ 2.41 และ 2.42 โดยนิยมกำหนดให้ค่าของ γ = 0.9 โมเมนตัม ถูกใช้ในการเร่งความเร็วเพื่อเพิ่มประสิทธิภาพของ SGD โดยการให้ความสำคัญในการพุ่งไปยังทิศทางที่ใกล้จุดกลางมากที่สุดก่อน แล้วทำให้ทิศทางที่ไม่เกี่ยวข้องมีความสำคัญลดลง ทำให้เกิดทิศทางที่ถูกต้องเกิดขึ้นโดยเป็นการเพิ่ม γ เข้ามาในการปรังปรุงทิศทางในแต่ละครั้ง โดยเฉพาะในปัญหาของฟังก์ชันอานม้า (Saddle Point) บริเวณที่เป็นพื้นราบฟังก์ชันการสูญเสียที่ตกลงบริเวณนั้นจะขยับตำแหน่งไปมาทำให้ไม่หลุดจาก Local Minima ทั้งที่มี Global Minima (จุดที่ต่ำที่สุดจริง ๆ) อยู่บริเวณขอบของอานม้าแสดงในรูปที่ 2-26 จึงต้องอาศัยเวกเตอร์ γv_{t-1} เสริมแรง เพื่อให้ไปในทิศทางที่ถูกต้องได้ ดังนั้นจึงทำให้เกรเดียนดีเซนต์สามารถออกจาก Local Minima ได้ดีขึ้น ตามที่ตามที่แสดงในรูปที่ 2-27

แต่ข้อเสียของการใช้เฉพาะโมเมนตัมคือ ขณะที่ฟังก์ชันกำลังเดินทางลงสู่จุดต่ำสุดก็จะสะสมพลังงานเพิ่มมากขึ้นเรื่อย ๆ จนถึงจุดต่ำสุด ทำให้พลังงานที่สะสมอยู่จำนวนมากมีความเร็วมากขึ้นเช่นเดียวกัน ดังนั้น ณ จุดต่ำสุด ของฟังก์ชันการสูญเสียจะทำให้มีความเร็วมากที่สุด

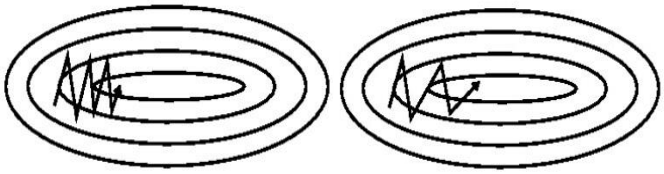
นั่นเอง อาจส่งผลให้กระดอนหรือทะลุไปอีกด้าน ทำให้ไม่อยู่ที่ตำแหน่งต่ำสุด หรือที่เรียกว่าทำให้เกิดการ Overshooting ของฟังก์ชันการสูญเสียค่อนข้างมาก

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \tag{2.41}$$

$$\theta = \theta - v_t \tag{2.42}$$



รูปที่ 2-26 กราฟของฟังก์ชันอานม้า (Saddle Point) [20]



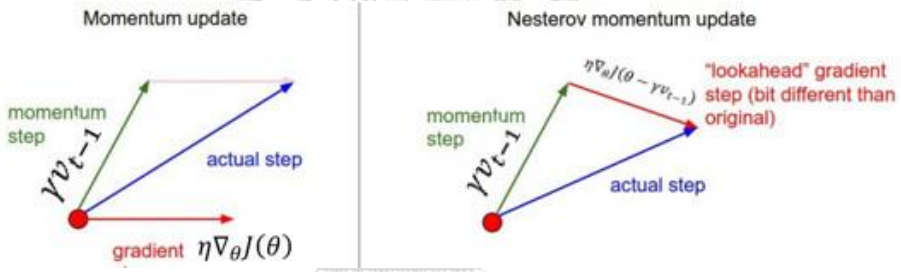
รูปที่ 2-27 Contour Plot ของ SGD ที่ไม่มีและมีโมเมนตัม [20]

2. ความชันแบบเร่งของเนสเตรอฟ (Nesterov Accelerated Gradient : NAG) จากปัญหาของการปรับปรุงค่าพารามิเตอร์โดยใช้เฉพาะโมเมนตัม จึงมีผู้คิดค้นเพิ่มเติมคือ ว่านอกจากการเพิ่มโมเมนตัมไปแล้ว ถ้ารู้ว่ากำลังจะเคลื่อนที่ไปในทิศทางใด ก็ให้ทำการเปลี่ยนค่าความชันเป็นความชันของตำแหน่งที่กำลังจะเคลื่อนที่ไปมาคูณกับอัตราการเรียนรู้ โดยทำการปรับค่าในส่วนของ Partial Derivative ของฟังก์ชันการสูญเสีย (Cost function, Loss function) จะเห็นว่าเปลี่ยนจาก θ เป็น $\theta - \gamma v_{t-1}$ ตามที่แสดงในสมการที่ 2.43 และ 2.45 ในกรณีที่ใช้เฉพาะโมเมนตัมจะทำการคำนวณค่าเกรเดียนซ์กำหนดทิศทางก่อน จากนั้นคำนวณค่าโมเมนตัมและนำทั้งสองเวกเตอร์มารวมกันตามที่แสดงด้านซ้ายของรูปที่ 2-28 แต่ถ้าโมเมนตัมที่สะสมมาชี้ไปคนละทางจากตำแหน่งต่ำที่สุด ก็จะทำให้

ค่าของทิศทางในการเข้าหาจุดต่ำสุดผิดออกไป หรือถ้าทิศทางของการคำนวณเกรเดียนดีเซนต์ถูกกำหนดมาถูกต้องแล้ว แต่ด้วยการสะสมของพลังงานย่อมทำให้เข้าหาจุดต่ำสุดยาก ด้วยเหตุนี้จึงนำ $\theta - \gamma v_{t-1}$ มาทำการคำนวณโมเมนตัมออกมาก่อน หลังจากนั้นค่อยทำการคำนวณค่าเกรเดียนดีเซนต์ด้วยฟังก์ชัน Partial derivation Nesterov ซึ่งจะเป็นตัวที่คอยปรับแก้ทิศทางการเดินของโมเมนตัม ทำให้สามารถ Converge เข้าสู่จุดต่ำสุดได้ดีขึ้น เนื่องมีฟังก์ชันของ NAG มาคอยปรับแต่งค่าตลอดเวลา เพื่อไม่ให้โมเมนตัมมีความสำคัญต่อสมการมากเกินไป ตามที่แสดงด้านขวาของรูปที่ 2-28

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1}) \tag{2.43}$$

$$\theta = \theta - v_t \tag{2.44}$$



รูปที่ 2-28 การปรับค่าพารามิเตอร์ของโมเมนตัมและโมเมนตัมร่วมกับ Nesterov [21]

NAG นิยมใช้ใน Recurrent Neural Network (RNN) เพื่อเพิ่มความแม่นยำของแบบจำลอง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3. Adagrad (Adaptive Gradient) ในการปรับแต่ง Gradient Descent สามารถกระทำได้ 3 รูปแบบ ได้แก่ 1. การปรับค่าถ่วงน้ำหนักและค่าไบแอส โดยใช้วิธีการทำงานของ Gradient Descent ทั้ง 3 แบบ คือ Batch , Stochastic และ Mini-Batch 2. การปรับแต่งภายในสมการ Partial Derivative โดยใช้โมเมนตัมและ Nesterov ทั้งสองรูปแบบการปรับแต่งได้อธิบายรายละเอียดตามหัวข้อที่ผ่านมา และการปรับแต่งแบบที่ 3 คือการปรับค่าการเรียนรู้ (Learning Rate, η) ตามพารามิเตอร์ ซึ่งเป็นออปติไมเซอร์ที่สามารถปรับค่าค่า Learning Rate ให้เหมาะสมกับพารามิเตอร์ได้ โดยจะมีการอัพเดทให้มีค่าสูงขึ้นสำหรับค่าพารามิเตอร์ที่มีจำนวนน้อย และอัพเดทค่าให้น้อยลงถ้าค่าพารามิเตอร์มีจำนวนมาก ทำให้ออปติไมเซอร์นี้ได้รับความนิยม โดยเฉพาะข้อมูลที่กระจัดกระจาย (Sparse data) ในการกำหนดค่าการเรียนรู้ซึ่งเป็นค่าไฮเปอร์พารามิเตอร์ของเกรเดียน ดีเซนต์ทั้งสามแบบ (Batch, Stochastic, Mini-Batch) โดยเป็นค่าคงที่เรากำหนดขึ้นมาเอง

ตามลักษณะของการแบ่งข้อมูลเข้าสู่โครงข่ายในหัวข้อที่ผ่านมา แต่ใน Adagrad Optimization จะทำการปรับแต่งอัตราการเรียนรู้ตามคุณลักษณะ (Features) หมายถึงจะทำการปรับแต่งอัตราการเรียนรู้เฉพาะบางพารามิเตอร์ที่มีความสัมพันธ์กับคุณลักษณะที่ต้องการในช่วงเวลานั้น ๆ เท่านั้น ตามที่แสดงในสมการที่ 2.45 ซึ่งเป็นสมการที่ใช้กำหนดพารามิเตอร์ที่ต้องการปรับปรุงพารามิเตอร์ใดพารามิเตอร์หนึ่ง เช่นค่าถ่วงน้ำหนักระหว่างโหนดใด ๆ ของแต่ละชั้น (Layer) หลังจากที่ได้กำหนดพารามิเตอร์ที่จะทำการปรับปรุงค่าแล้ว ขั้นต่อไปก็ทำการปรับปรุงค่าพารามิเตอร์นั้น ๆ ในแต่ละช่วงเวลา (time step, t) ตามสมการที่ 2.46 จากนั้นแทนที่อัตราการเรียนรู้ด้วยสมการที่ 2.40 ซึ่งจากสมการนี้ถ้าพารามิเตอร์ใดที่มีการปรับปรุง (Update) บ่อยครั้งผลรวมของ $G_{t,ii}$ หรือผลรวมของค่าเกรเดียนต์จะมีค่ามาก ส่งผลให้อัตราการเรียนรู้มีค่าน้อยลง (η) ในทางตรงกันข้ามถ้าพารามิเตอร์ใดเป็นพารามิเตอร์ที่มีการปรับปรุงค่าน้อยจะส่งผลให้ผลรวมของ $G_{t,ii}$ มีค่าน้อย ทำให้อัตราการเรียนรู้ปรับค่าสูงขึ้น

ข้อดีของวิธีการนี้คือสามารถกำหนดค่าอัตราการเรียนรู้ได้เลย (ค่า default = 0.01) หรือจะใส่ค่าเท่าใดก็ได้ เพราะหลังจากนั้นอัลกอริทึมจะทำการปรับแต่งให้เอง และข้อดีอีกอย่างคือสามารถใช้กับข้อมูลที่เป็นลักษณะกระจายได้ (Sparse Data) เช่นในงานด้านการประมวลผลภาษาธรรมชาติที่คำบางคำมีความสำคัญมากกว่าคำที่สามารถพบเจอบ่อยครั้ง

ข้อเสียของวิธีนี้คือการสะสมค่าเกรเดียนต์ไว้ที่ตัวหาร ตามสมการที่ 2.47 เพราะเมื่อทำการเรียนรู้หลายรอบการทำงาน (Iteration) จะทำให้อัตราการเรียนรู้เข้าใกล้ศูนย์เรียกว่าปัญหา Decaying Learning Rate การปรับตำแหน่งจึงแทบไม่มีการเปลี่ยนแปลงเลย

$$\theta_{t,i} = \nabla_{\theta} J(\theta_{t,i}) \tag{2.45}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot \theta_{t,i} \tag{2.46}$$

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} - \epsilon}} \cdot \theta_{t,i} \tag{2.47}$$

4. AdaDelta จากปัญหาการเพิ่มขึ้นแบบสะสมของ $G_{t,ii}$ แล้วทำให้อัตราการเรียนรู้เข้าใกล้ศูนย์ จนไม่สามารถทำการเรียนรู้ได้ต่อไป AdaDelta ถูกพัฒนาต่อยอดมาจาก AdaGrad เพื่อให้สามารถลบปัญหาที่เกิดขึ้น โดยใช้หลักการทำงานของ Running Average ดังนี้

จากชุดของข้อมูลที่เก็บรวบรวมตามระยะเวลาเป็นช่วง ๆ อย่างต่อเนื่องกัน (Time Series Data) เราสามารถหาค่าเฉลี่ยของข้อมูลที่มีการเปลี่ยนแปลงในแต่ละช่วงเวลานี้ว่า “Moving Average” ซึ่งสามารถคำนวณหาจากค่า Running Average ของข้อมูลในปัจจุบัน ($E[g^2]_t$) ได้จาก

สมการที่ 2.48 จะเห็นว่าข้อมูลที่เป็น Moving Average ของค่าเก่าจะมีความสำคัญเท่ากับ 0.9 ส่วนข้อมูลปัจจุบันให้ความสำคัญเท่ากับ 0.1 ดังนั้นเมื่อมีการเปลี่ยนแปลงข้อมูล ค่าเฉลี่ยก็จะมีการเปลี่ยนแปลงไปเรื่อย ๆ แต่ไม่มาก เพราะถูกถ่วงน้ำหนักด้วยค่าก่อนหน้า ซึ่งข้อมูลที่กล่าวมาทั้งหมดก็คือค่าเกรเดียนต์ (g^2) หรือค่าของเกรเดียนต์นั่นเอง

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \tag{2.48}$$

เพื่อแก้ไขปัญหาการสะสมของค่าเกรเดียนต์จึงทำการแก้ไขค่า G_t เป็นค่า Moving Average ตามสมการที่ 2.49 และ 2.50 โดยมีชื่อว่า RMSprop โดยกำหนดค่าเริ่มต้นให้ $\gamma = 0.9$

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1 g_t^2 \tag{2.49}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{2.50}$$

แต่จะพบว่าในสมการที่ 2.42 ค่าของ $\frac{\eta}{\sqrt{E[g^2]_t+\epsilon}}$ เป็นคนละหน่วยกับค่า θ_t ดังนั้นเมื่อนำค่าที่ได้มาลบกันอาจเป็นสาเหตุให้ค่าความแม่นยำลดลง จึงมีการคิดค้นว่าควรทำให้เป็นหน่วยเดียวกันก่อน โดยการสร้าง Running Average ของ θ_t ตามสมการที่ 2.51 คือแทนค่า g ด้วย $\Delta\theta$

$$E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1-\gamma) \Delta\theta^2{}_t \tag{2.51}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ค่าของ $\sqrt{E[g^2]_t + \epsilon}$ คือค่า RMS หรือ Mean Square Error เพราะค่า Epsilon (ϵ) ถือว่ามีค่าน้อยมาก และเมื่อปรับค่าให้เป็นหน่วยเดียวกันแล้ว สามารถเขียนความสัมพันธ์ได้ตามสมการที่ 2.52 และสรุปได้ว่าค่า $RMS[\Delta\theta]_t$ คือค่า Moving Average ของ θ_t (Iteration ปัจจุบัน) จึงทำการแก้ไขด้วยการนำ θ_{t-1} (Iteration ก่อนหน้า) มาทำเป็นตัวตั้ง ทำให้ไม่ต้องใช้ค่าอัตราการเรียนรู้อีกต่อไป และนำค่า $RMS[\Delta\theta]_{t-1}$ เป็นตัวหาร ตามสมการที่ 2.53 เรียกสมการนี้ว่า Adadelta จากนั้นทำการปรับค่าพารามิเตอร์ใหม่ตามสมการที่ 2.54 โดยข้อดีของวิธีนี้คือเมื่อใช้หลักการของ Moving Average และ RMS แล้วค่าของเกรเดียนต์สะสมจะไม่เพิ่มมากขึ้น

$$RMS[\Delta\theta]_t = \sqrt{E[\Delta\theta^2]_t + \epsilon} \tag{2.52}$$

$$\Delta\theta_t = \frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \tag{2.53}$$

$$\Delta\theta_{t+1} = \theta_t + \Delta\theta_t \tag{2.54}$$

5. Adam Adaptive Moment Estimation เป็น optimizer ที่สามารถปรับอัตราการเรียนรู้ สำหรับพารามิเตอร์ในแต่ละครั้งได้ และยังสามารถแก้ปัญหา Decaying Learning Rate ในแต่ละสเต็ปที่ผ่านมาได้เหมือนกับ AdaDelta ช่วยให้โมเดลไม่หยุดเรียนได้ อีกทั้งโมเมนตัมที่มีการสะสมค่าเอาไว้ เมื่อมีจุดที่ต้องสะดุดหรือเปลี่ยนแปลงทิศการเคลื่อนที่ ก็ยังสามารถรักษาทิศทางการเคลื่อนที่ในทิศทางเดิมได้ แต่โมเมนตัมที่สะสมใน Adam เป็นโมเมนตัมที่สะสมในแต่ละพารามิเตอร์นั้น ๆ ควบคู่กับการสะสมค่าเกรเดียนต์ของ Adagrad คือการปรับปรุงค่าอัตราการเรียนรู้ (η) โดยการหารด้วยผลรวมของ g_t^2 และขณะเดียวกันก็ทำการสะสมโมเมนตัมของแต่ละพารามิเตอร์ไปพร้อม ๆ กัน อีกทั้งยังไวกว่า Gradient Descent และลดปัญหาการแกว่งของพารามิเตอร์ได้อีกด้วย ตามสมการที่ 2.55 แสงการสะสมของเกรเดียนต์กำลังสอง โดยใช้ค่าเฉลี่ยการเคลื่อนที่ (Moving Average) เหมือนใน Adadelta และ RMSprop และนอกจากนี้ยังมีการสะสมโมเมนตัม ตามสมการที่ 2.56 ซึ่งเป็นสมการที่แสดงการสะสมของโมเมนตัมเอาไว้เฉพาะของแต่ละพารามิเตอร์เท่านั้น ไม่ได้หมายถึงทั้งสมการอย่างในโมเมนตัมตามหัวข้อที่กล่าวมาในตอนต้น

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{2.55}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{2.56}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากสมการที่ 2.55 และ 2.56 มีทั้งการสะสมของเกรเดียนต์กำลังสอง (v_t) เพื่อใช้เป็นตัวหารในการปรับปรุงอัตราการเรียนรู้ และมีการสะสมของเกรเดียนต์ (m_t) ซึ่งเป็นค่าของโมเมนตัมที่ถูกสะสมในแต่ละพารามิเตอร์ ซึ่งพบว่าเมื่อเรากำหนดค่าเริ่มต้นให้กับ v_{t-1}, m_{t-1} มีค่าเป็น 0 และกำหนดค่าให้ β_1, β_2 มีค่าเข้าใกล้ 1 ทำให้ทั้ง v_t, m_t ไม่มีการเปลี่ยนแปลงและไม่มีการปรับปรุงค่าใด ๆ เลย รวมทั้งอัตราการเรียนรู้หรือพารามิเตอร์ใด ๆ เพราะค่าการปรับปรุงวิ่งเข้าใกล้ 0 ดังนั้นแทนที่จะใช้ค่า v_t ในการปรับแต่งอย่าง RMS ก็เปลี่ยนไปใช้ค่าในสมการที่ 2.57 และ 2.58

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{2.57}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{2.58}$$

และทำการปรับปรุงตำแหน่งหรือค่าพารามิเตอร์ใหม่ตามสมการที่ 2.59 และในงานวิจัยนี้เลือกใช้ Adam Optimizer ในการปรับปรุงพารามิเตอร์ของการเรียนรู้เชิงลึก

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t \tag{2.59}$$

เมื่อกำหนดให้

$J(\theta)$ คือฟังก์ชันการสูญเสีย (Loss function , Cost function, Objective function) ที่มีพารามิเตอร์เป็น θ ประกอบด้วยค่าถ่วงน้ำหนัก (Weight) และค่าไบแอส (Bias)

∇_θ คือ ค่าความลาดชันของฟังก์ชันการสูญเสียเทียบกับพารามิเตอร์ต่าง ๆ คำนวณโดยการใช้อนุพันธ์ย่อย (Partial Derivative) ตามกฎลูกโซ่ (Chain Rule) เปรียบเหมือนเข็มทิศของระบบ Backpropagation เพื่อนำทางเข้าหาค่าความผิดพลาดที่น้อยที่สุด ในการหาตำแหน่ง (ค่าถ่วงน้ำหนัก, ค่าไบแอส) ที่ทำให้ค่าของฟังก์ชันต่ำที่สุด

η คือ อัตราการเรียนรู้ (Learning Rate) เป็นไฮเปอร์พารามิเตอร์ที่มีค่าระหว่าง 0 -1

θ คือ ค่าพารามิเตอร์ (ค่าถ่วงน้ำหนักและค่าไบแอส)

$x^{(i)}$, $y^{(i)}$ คือ ข้อมูลฝึกสอนทีละ 1 ตัวอย่างที่ถูกป้อนเข้าโครงข่าย

$x^{(i:i+n)}$, $y^{(i:i+n)}$ คือ ข้อมูลฝึกสอนทีละ n ตัวอย่างที่ถูกป้อนเข้าโครงข่าย

γ คือ ค่าของโมเมนตัม ปกติกำหนดค่าเป็น 0.9

G คือ เมทริกซ์ทแยงมุม (Diagonal Matrix) ซึ่งเป็นผลรวมกำลังสองของความชันทั้งหมดสะสมตั้งแต่เริ่มการเรียนรู้ถึง Time Step

$\theta_{t,i}$ คือ ค่าพารามิเตอร์ที่ i ซึ่งอาจเป็นค่าถ่วงน้ำหนักตัวที่เชื่อมระหว่างโหนดที่ 1 ของชั้นปัจจุบันกับ โหนดที่ 5 ของชั้นถัดไป หรือข้ามระหว่างชั้นใด ๆ ก็ได้ของเวลาที่ t (time step)

ϵ คือ ตัวเลขที่กำหนดไว้เพื่อไม่ให้ตัวหารเป็นศูนย์ (Divide by zero)

E คือ ค่า Moving Average ของข้อมูล

v_t คือ การสะสมของ Moving Average ของเกรเดียนต์กำลังสอง หรือของส่วนเบี่ยงเบนมาตรฐาน(Standard deviation : σ) ใน time step นั้น ๆ

m_t คือ ค่าโมเมนตัมที่สะสมใน time step นั้น ๆ

β_1 , β_2 คือ ค่าไฮเปอร์พารามิเตอร์

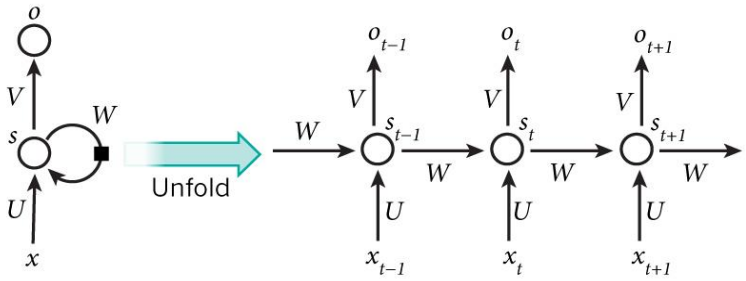
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.12 โครงข่ายประสาทเทียมแบบวกกลับ (Recurrence Neural Networks : RNN) [18]

ในโครงข่ายประสาทเทียมที่กล่าวมาในตอนต้น เป็นโครงข่ายแบบป้อนข้อมูลไปข้างหน้า (Feed forward Neural Network) ซึ่งเป็นโครงข่ายที่ไม่มีการวกกลับ ซึ่งจะเหมาะสำหรับข้อมูลที่ไม่เป็นลำดับ (Sequence) และไม่ขึ้นกับลำดับเวลา (Time Series) ส่วนโครงข่ายประสาทเทียมแบบวกกลับเป็นแบบจำลองที่นำผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลขาเข้าอีกครั้ง ดังรูปที่ 2-29 ซึ่งเหมาะกับข้อมูลที่เป็นลำดับ เช่นคำที่เรียงต่อกันเป็นลำดับเป็นประโยค หรือเป็นข้อมูลที่มีลักษณะเป็นอนุกรมเวลา (Time Series Data) การเรียงลำดับมีความสำคัญ เช่นข้อความข่าวตามที่งานวิจัยนี้ใช้เป็นข้อมูล โดยในแต่ละโหนดของ RNN จะมีข้อมูลขาเข้าสองอย่างคือ ข้อมูลขาเข้า (Input) ของโหนดนั้น ๆ และผลลัพธ์ (Output) ที่ได้จากการคำนวณของโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทางคือ ผลลัพธ์ที่ออกที่โหนดนั้น ๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าในโหนดถัดไป ดังแสดงตามรูปที่ 2-29 ข้อดีของ RNN คือมีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคต นั่นหมายถึงสิ่งที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย

แต่ข้อเสียของ RNN คือการที่สามารถดูค่าย้อนกลับได้เพียงช่วงระยะเวลาสั้น ๆ เท่านั้น ซึ่งปัญหาหลักของ RNN นั้นเกิดจากค่าความลาดชันที่เริ่มน้อยลงในข้อมูลที่มีความยาวมากขึ้น เพราะต้องรวมค่าความลาดชัน (Gradient) ทั้งหมด จนไม่สามารถเห็นความเปลี่ยนแปลงของความลาดชันได้ หรือแทบไม่เกิดการเรียนรู้ ตามที่กล่าวในหัวข้อที่ผ่านมา ตามรูปที่ 2-29 ซึ่งจะพบว่าเมื่อข้อมูลขาเข้า ณ เวลาที่ 1 ซึ่งมีสีเข้มถูกคูณด้วยค่าถ่วงน้ำหนัก W ในทุกรอบการเรียนรู้ และเมื่อเวลาผ่านไปนาน ๆ จนถึง ณ เวลาที่ 7 ทำให้ความเข้มของสีที่เป็นผลลัพธ์ค่อย ๆ จางลง หรือเกิดการลืมข้อมูลที่อยู่ไกลกันนั่นเอง ทำให้งานวิจัยนี้ไม่ได้ใช้ RNN เป็นอัลกอริทึมในการทำการจำแนกข้อความ เพราะเนื้อหาข่าวที่ใช้ในการเรียนรู้มีความยาวเทียบเท่าหนึ่งแผ่นเอกสาร

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 2-29 โครงสร้างของโครงข่ายประสาทเทียมแบบวกกลับ [18]

จากรูปที่ 2.29 กำหนดให้

x คือ ข้อมูลขาเข้า (Input) ดังนั้น x_t หมายถึงข้อมูลนำเข้า ณ เวลา t

s คือ ผลลัพธ์ของชั้นซ่อน (Hidden Layer) ดังนั้น s_t หมายถึง ผลลัพธ์ของชั้นซ่อน ณ เวลา t s_{t-1} คือ ผลลัพธ์ของชั้นซ่อนก่อนหน้าหนึ่งช่วงเวลา

o คือ ผลลัพธ์ที่คำนวณได้ (Output) ดังนั้น o_t เท่ากับข้อมูลผลลัพธ์ที่คำนวณได้ ณ เวลา t

U คือ เมทริกซ์ค่าถ่วงน้ำหนักของชั้นซ่อน ณ เวลาปัจจุบัน และมีขนาดเท่ากันทุกช่วงเวลา

V คือ เมทริกซ์ค่าถ่วงน้ำหนักของชั้นข้อมูลส่งออก ณ เวลาปัจจุบัน และมีขนาดเท่ากันทุกช่วงเวลา

W คือ เมทริกซ์ค่าถ่วงน้ำหนักของชั้นซ่อนในช่วงเวลาก่อนหน้า และมีขนาดเท่ากันทุกช่วงเวลา

b_s คือ ค่าไบแอสของชั้นซ่อน

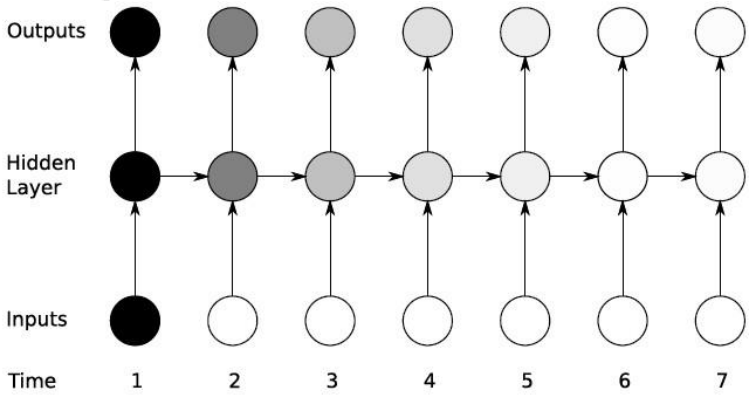
b_o คือ ค่าไบแอสของชั้นข้อมูลส่งออก

f คือฟังก์ชันกระตุ้น (Activation function) เช่น Tanh หรือ ReLU สำหรับชั้นซ่อน และเป็น Softmax สำหรับชั้นข้อมูลส่งออก สมการคำนวณของชั้นซ่อน ตามสมการที่ 2.60 และการคำนวณหาค่าผลลัพธ์ ตามสมการที่ 2.61

$$s_t = f(Ux_t + Ws_{t-1} + b_s) \tag{2.60}$$

$$o_t = f(Vs_t + b_o) \tag{2.61}$$

จุฬาลงกรณ์มหาวิทยาลัย



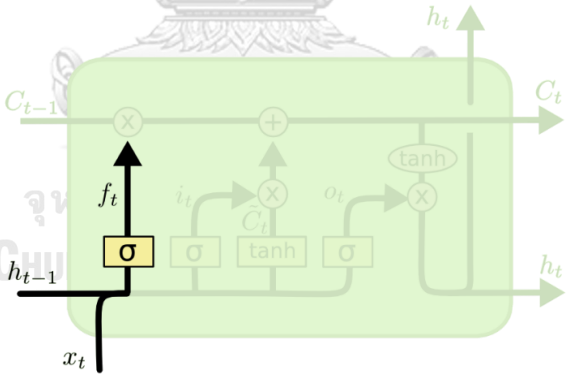
รูปที่ 2-30 ปัญหา Vanishing Gradient ของ RNN [15]

2.13 หน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory : LSTM) [15]

จากปัญหาการลืมสิ่งที่เคยเรียนรู้ของ RNN หรือปัญหา Vanishing Gradient ที่แม้จะมีโหนดของชั้นซ่อนที่ทำหน้าที่เสมือนเป็นหน่วยความจำ แต่ก็ได้เพียงระยะสั้น ๆ เท่านั้น จึงมีการนำ LSTM มาใช้งาน โดยสิ่งที่เด่นขึ้นมาคือสามารถเลือกได้ว่าข้อมูลไหนที่ควรจะจดจำหรือควรจะกำจัดทิ้งออกไป ภายใน LSTM ประกอบด้วยฟังก์ชันต่าง ๆ ที่ถูกใช้เพื่อสร้างฟังก์ชันพิเศษจำนวน 4 ฟังก์ชัน ได้แก่ การอ่านข้อมูล การเขียนข้อมูล การอัพเดตข้อมูล และการลืมข้อมูล ซึ่งทำให้การจดจำข้อมูลในแต่ละโหนดมีประสิทธิภาพมากขึ้น ซึ่งฟังก์ชันเหล่านี้เปรียบเสมือนประตูที่คอยควบคุมข้อมูลที่เข้ามาในแต่ละโหนด ให้เป็นไปตามทิศทางที่ต้องการ โดยมีฟังก์ชันแต่ละตัวทำหน้าที่ ดังนี้

2.13.1 ประตูลืม (Forget Gate Layer) คือประตูที่ใช้เป็นตัวกำหนดว่าข้อมูลที่เข้ามานั้นสมควรที่จะได้ออกไปทั้งหมด ซึ่งข้อมูลที่จำเป็นต้องใช้ในการตัดสินใจว่าจะเก็บข้อมูลนี้ไว้หรือไม่นั้นก็มาจากข้อมูลขาเข้า ณ โหนดนั้น ๆ รวมกับผลลัพธ์ที่ถูกคำนวณมาจากโหนดก่อนหน้า โดยใช้สูตรการคำนวณตามสมการที่ 2.62 รายละเอียดการทำงานตามรูปที่ 2-31 โดยผลลัพธ์ที่ได้ของประตูบานนี้จะอยู่ที่ [0,1] ซึ่งค่า 0 หมายถึงไม่มีข้อมูลใดที่จะสามารถไหลผ่านไปได้เลย ในขณะที่ 1 หมายถึงปล่อยให้ข้อมูลที่เข้ามาไหลผ่านไปได้ทั้งหมด

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{2.62}$$

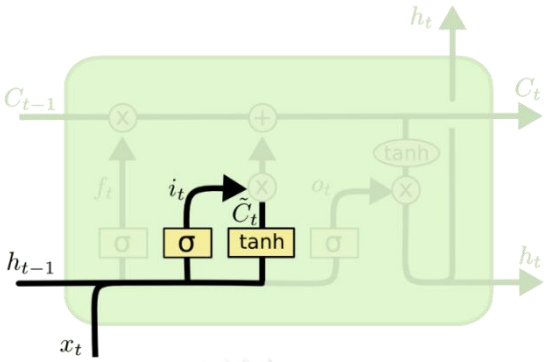


รูปที่ 2-31 แสดงโครงสร้างของประตูลืมและสูตรการคำนวณ [22]

2.13.2 ประตูข้อมูลขาเข้า (Input Gate Layer) คือประตูที่ใช้เพื่อเปิดรับข้อมูลที่เข้ามาใหม่ และทำการคำนวณค่าความสำคัญของข้อมูลเข้านั้นในการประเมินความสำคัญตามสมการที่ 2.63 หลังจากนั้นคำนวณหาค่าสถานะของเซลล์ตามสมการที่ 2.64 เพื่อเป็นค่าใหม่ในการพิจารณาว่าต้องทำการปรับปรุงต่อในขั้นตอนต่อไป โดยมีโครงสร้างการทำงานตามรูปที่ 2-32

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{2.63}$$

$$\tilde{C}_t = tanh(W_c h_{t-1} + U_c x_t + b_c) \quad (2.64)$$

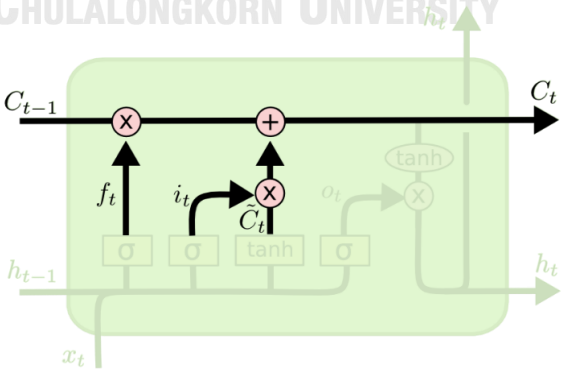


รูปที่ 2-32 แสดงประตูทางเข้าและสูตรการคำนวณ [22]

2.13.3 ประตูอัพเดตข้อมูล (Update Cell State) ระหว่างที่ประตูทางเข้าเปิดประตูเพื่อรับข้อมูลมาจดจำในโหนดตัวเอง จะมีอีกโหนดหนึ่งที่จะถูกผลักเข้าสู่กระบวนการแปลงโฉมโดยฟังก์ชัน Tanh เพื่อลดปัญหา Exploding gradient เพราะค่าที่ได้จะอยู่ระหว่าง {-1,1} และใกล้เคียงกับข้อมูลผลลัพธ์ จากโหนดนี้ตามสมการที่ 2.64 เพื่อส่งเป็นข้อมูลผลลัพธ์ต่อไป เพราะหลังจากคำนวณได้ข้อมูลที่กำลังจะเป็นข้อมูลผลลัพธ์แล้วจะถูกนำไปคำนวณกับข้อมูลที่ได้จากประตูทางเข้าและหลังจากได้ผลลัพธ์นี้ ก็นำไปรวมกับข้อมูลแรกที่ได้จากประตูลืม ซึ่งถูกคำนวณกับสถานะจากโหนดก่อนหน้าแล้วอีกครั้ง ตามสมการที่ 2.65 และเมื่อกระบวนการทั้งหมดเสร็จสิ้นค่าที่ได้ออกมาจะเป็นค่าสถานะใหม่ของโหนดนั้น ๆ ที่ผ่านการอัพเดทสถานะจากโหนดก่อนหน้าตามรูปที่ 2-33

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.65)$$

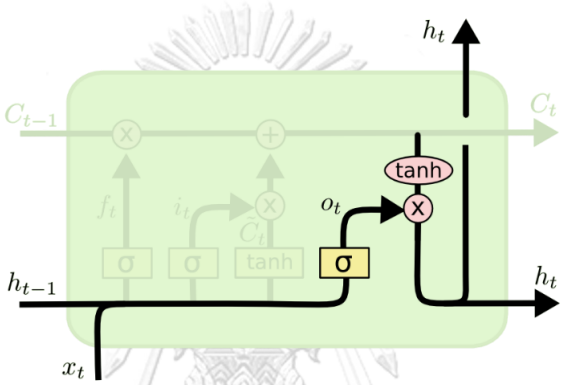


รูปที่ 2-33 โครงสร้างการทำงานของประตูอัพเดตข้อมูล [22]

2.13.4 ประตูข้อมูลส่งออก (Output Gate Layer) คือประตูที่ใช้บอกว่าข้อมูลนี้เตรียมพร้อมที่จะเป็นข้อมูลส่งออกแล้วหรือยัง ซึ่งข้อมูลผลลัพธ์ของแต่ละโหนด ใน LSTM มีด้วยกันทั้งหมด 3 แห่ง แต่

มีเพียง 2 ค่า คือค่าที่เกิดจากคำนวณโดยมีการอัพเดตสถานะแล้ว ซึ่งถูกคำนวณจากขั้นตอน Update Cell state ค่านี้จะถูกส่งต่อทันทีโดยไม่ผ่านฟังก์ชันใด ๆ อีกไปให้โหนดถัดไป ค่าของข้อมูลขาเข้าที่ถูกดัดแปลงผ่านการคำนวณในประตูทางออกนี้จะถูกส่งต่อไปเป็นข้อมูลขาเข้าของโหนดถัดไป และถูกส่งไปเป็นผลลัพธ์ของโหนดนั้น ๆ ด้วย โดยค่าที่ถูกส่งต่อไปเป็นข้อมูลขาเข้าในโหนดถัดไปนั้น ก็เช่นเดียวกับประตูบานอื่น ๆ คือการนำข้อมูลจากโหนดที่แล้วที่เข้ามารวมกับข้อมูลขาเข้าในโหนดนั้น ๆ แล้วผ่านฟังก์ชัน Sigmoid ตามสมการที่ 2.66 โดยมีโครงสร้างการทำงานตามรูปที่ 2-34

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{2.66}$$



รูปที่ 2-34 แสดงประตูทางออกและสูตรการคำนวณ [22]

หลังจากที่ผ่านการคำนวณค่า Sigmoid แล้วจะถูกนำมาคำนวณกับค่าสถานะปัจจุบันของโหนดที่ได้ถูกคำนวณมาเรียบร้อยแล้วจากประตูบานอื่นที่ผ่านมา ซึ่งค่าสถานะนั้นจะถูกนำไปเข้าฟังก์ชัน Tanh ก่อน เพื่อลดปัญหา Exploding Gradient มีค่าระหว่าง {-1,1} จากนั้นนำผลลัพธ์ที่ได้หลังจากเข้าฟังก์ชันนั้นแล้วมาคำนวณตามสมการที่ 2.67 ซึ่งผลลัพธ์ที่ได้จากการกระทำนี้จะถูกแบ่งออกไปเป็นสองข้อมูลคือผลลัพธ์ของโหนดนั้น ๆ และข้อมูลที่จะถูกส่งต่อไปเป็นข้อมูลขาเข้าในโหนดต่อไป

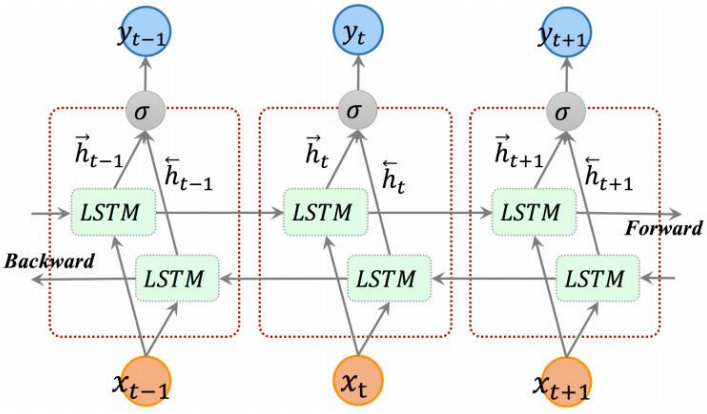
$$h_t = o_t \cdot \tanh(C_t) \tag{2.67}$$

จากความสามารถของ LSTM ที่สามาถแก้ปัญหาการเลือนหายของข้อมูลได้ โดยอาศัยเซลล์หน่วยความจำที่มีประตูสัญญาณต่าง ๆ รวมทั้งการเลือกที่จะลืมข้อมูลบางส่วนที่ไม่สำคัญออกไปได้นั้น ทำให้งานวิจัยนี้เลือกใช้อัลกอริทึมนี้ในการฝึกแบบจำลองการทำงานในการจำแนกประเภทของข้อความ ในการเรียนรู้ของโครงประสาทเทียมทั้งแบบวกกลับและหน่วยความจำระยะสั้นแบบยาวที่มีประสิทธิภาพมากที่สุดคือเทคนิคการแพร่ย้อนกลับ (Backpropagation) โดยทำการคำนวณไป

ข้างหน้าและย้อนกลับตามที่อธิบายในส่วนของการเรียนรู้ของโครงข่ายประสาทเทียมหลายชั้น (MLP) แต่ด้วยจำนวนลำดับของข้อมูล (Sequence) ที่ยาวมาก ทำให้ต้องใช้ขนาดหน่วยความจำที่มากและอาจทำให้โครงข่ายไม่สามารถเรียนรู้ได้ต่อไป ดังนั้นเพื่อแก้ไขปัญหานี้จะทำการแบ่งจำนวนข้อมูลในมิติของเวลาสำหรับการเรียนรู้ โดยให้คำนวณไปข้างหน้า k steps (k คือจำนวน time step) จากนั้นทำการแพร่ย้อนกลับจนครบ k step ตอนต้น แล้วทำการตัด Computation Graph ส่วนที่ดำเนินการเสร็จแล้ว จากนั้นค่อยทำส่วนที่เหลือไปเรื่อย ๆ จนครบข้อมูลทั้งหมด เราเรียกกระบวนการนี้ว่า “Backpropagation Through Time”

2.14 หน่วยความจำระยะสั้นแบบยาวสองทิศทาง (Bidirectional LSTM) [23]

LSTM เป็นโครงข่ายประสาทเทียมที่เพิ่มความสามารถของ RNN โดยการใช้เซลล์หน่วยความจำ (memory cell) เพื่อควบคุมการทำงานของประตูต่าง ๆ เช่นการจดจำข้อมูลสำคัญหรือลืมข้อมูลบางอย่าง เพื่อแก้ปัญหาการเลือนหายของข้อมูล เมื่อมีความยาวของลำดับข้อมูลขาเข้ามาก (Vanishing gradient Problem) หรือปัญหา Exploding gradient ที่ขนาดของค่าถ่วงน้ำหนักมีปริมาณที่เพิ่มมากขึ้นกว่าเซลล์หน่วยความจำรองรับได้ แต่พบว่า LSTM เป็นการเรียนรู้ทิศทางเดียวทำให้ได้บริบท (Context) ของข้อมูลจากซ้ายไปขวาเท่านั้น แต่ในงานประมวลผลภาษาธรรมชาติต้องการบริบทโดยรอบคือทั้งจากซ้ายไปขาวและขาวไปซ้ายพร้อม ๆ กัน เช่นในงานวิจัยนี้คำหรือข้อความที่ส่งผลต่อการทำนายในบางเนื้อข่าวอยู่ตอนต้นของประโยคหรือท้ายของประโยคในชุดข้อมูลการเรียนของแบบจำลองและในเนื้อหาข่าวมีความซับซ้อน ดังนั้นการใช้ LSTM แบบทิศทางเดียวอาจให้ผลลัพธ์ที่มีประสิทธิภาพไม่ดีได้ จึงมีการใช้ LSTM แบบสองทิศทางในโครงข่ายให้แบบจำลองเรียนรู้ข้อมูลจากทั้งสองด้าน เพื่อให้ได้บริบทโดยรอบและครอบคลุม เรียกว่า หน่วยความจำระยะสั้นแบบยาวสองทิศทาง (Bi-direction LSTM) โครงสร้างการทำงานตามรูปที่ 2-35



รูปที่ 2-35 โครงสร้างการทำงานของ Bi-directional LSTM [23]

ตามรูปที่ 2-35 มี LSTM ตัวที่อ่านข้อมูลไปข้างหน้า (Forward) ขณะเดียวกับก็มี LSTM อีกตัวอ่านข้อมูลจากข้างหลังมาข้างหน้า (Backward) แล้วนำข้อมูลจากทั้งสองตัวมารวมกันก่อนส่งไปที่ชัดถัดไป ทำให้โครงข่ายสามารถตัดสินใจ ณ ช่วงเวลาใด โดยใช้บริบทจากทั้งประโยคได้ แต่ LSTM แบบสองทิศทางมีข้อเสียสำหรับงานที่ต้องตอบสนองแบบทันเวลา (Real Time) เพราะการที่โครงข่ายจะอ่านข้อมูลย้อนหลังจะต้องมีข้อมูลในอนาคต ทำให้งานลักษณะที่ต้องการตอบสนองแบบ Real Time ไม่เหมาะสม แต่งานวิจัยนี้เลือกใช้โครงข่ายของ LSTM แบบสองทิศทางได้เพราะสามารถอ่านข้อมูลจนจบประโยคในเนื้อหาข่าวก่อนทำการประมวลผล แต่จะต้องใช้เวลาในการทำงานมากกว่าแบบทิศทางเดียว (Uni directional LSTM)

2.15 การประมวลผลข้อความ

2.15.1 การรวบรวมข้อมูล (Data Scraping) ข้อมูลที่ใช้ในการฝึกสอนแบบจำลอง (Dataset) ที่อยู่ในรูปแบบของ html จากเว็บไซต์เผยแพร่คำพิพากษาของศาลฎีกา [5] และมีเนื้อหาส่วนในคำพิพากษาที่ไม่มีผลต่อการพิจารณาตัดสินคดี ดังนั้นในขั้นตอนการเตรียมข้อมูลเพื่อฝึกสอน จึงใช้ BeautifulSoup[24] ซึ่งเป็น Library ของภาษาไพธอน ที่ใช้จัดการหน้าเว็บไซต์ เพื่อดึงข้อมูลเฉพาะแท็ก (Tag) และคลาส (Class) ที่มีเนื้อหาเกี่ยวข้องที่ใช้ในงานวิจัยเพื่อใช้ในการประมวลผลเท่านั้น

2.15.2 การทำความสะอาดข้อมูล (Data Cleansing) เพื่อให้ข้อมูลที่จะถูกประมวลผลในโครงข่ายประสาทเทียม เป็นข้อมูลที่ส่งผลต่อการพิจารณาคดีเท่านั้น จึงทำการตัดคำที่ไม่สำคัญออก โดยใช้นิพจน์ปกติ (Regular Expression) [25] ซึ่งเป็นรูปแบบหรือกลุ่มคำ (pattern) ที่กำหนดขึ้นเพื่อใช้ค้นหาข้อความหรือตัวอักษรต่างๆ ว่าตรงตามเงื่อนไขที่กำหนดขึ้นหรือไม่ ร่วมกับการสร้างคลังคำศัพท์ที่ไม่จำเป็น (Stop words) เพื่อให้คำที่อยู่ในข้อมูลถูกลบออก ได้แก่ คำฟุ่มเฟือย คำนำหน้าชื่อ ยศ ตำแหน่ง หมายจับ ชื่อสถานที่ ชื่อหน่วยงานหรือองค์กร เพื่อคัดกรองและค้นหาคำหรือตัวเลขที่ไม่ต้องการออกจากข้อความ ตามที่แสดงในรูปที่ 2-36

Original Data	Cleaned Data
~~เมื่อวันที่ 5 มี.ค. พ.ต.อ.สัญญา พาตินธุ ผกก.สภ.ปากน้ำหลังสวน จ.ชุมพร~~ พร้อมด้วย ~~ร.ต.อ.วิฑูรย์ เผือกวารี รอง สว.(สอบสวน)~~ เจ้าของคดี นำกำลังคุมตัว ~~นายชู หรือนายตะวัน~~ แรงงานต่างด้าวชาวมอญ ไปทำแผนประกอบคำรับสารภาพภายในสวนปาล์มพื้นที่ ~~หมู่ 1 ต.บางน้ำจืด อ.หลังสวน~~ หลัง~~นายชู~~ก่อเหตุข่มขืนแล้วฆ่ายาย~~วัย 72 ปี~~ โดยมีบรรดาทางญาติผู้ตาย เดินทางมาดูหน้า~~นายชู~~เป็นจำนวนมาก	พร้อมด้วยเจ้าของคดีนำกำลังคุมตัวแรงงานต่างด้าวชาวมอญไปทำแผนประกอบคำรับสารภาพภายในสวนปาล์มพื้นที่หลังก่อเหตุข่มขืนแล้วฆ่ายาย โดยมีบรรดาทางญาติผู้ตายเดินทางมาดูหน้าเป็นจำนวนมาก

รูปที่ 2-36 การตัดคำที่ไม่มีผลต่อการพิจารณาคดีออกจากข้อมูลสำหรับฝึกฝน

2.15.3 การตัดคำ (Word Tokenization) ภาษาไทยเป็นภาษาที่ไม่ได้ทำการแบ่งคำเหมือนในภาษาอังกฤษ ดังนั้นในการนำเข้าประมวลผลต้องทำการตัดคำก่อน ซึ่งเทคนิคในการตัดคำได้แบ่งเป็น 3 ประเภท ได้แก่

1) การตัดคำโดยใช้กฎ (Rule based approach) ซึ่งเป็นวิธีการพิจารณาการตัดคำ เพื่อกำหนดขอบเขตของการตัดคำ จากพยัญชนะ สระ วรรณยุกต์ ตัดสะกด การันต์ เช่น เช่น ถ้าพบ “โ” จะตัดหลังสระนี้ไม่ได้ เพราะต้องมีพยัญชนะตามมาเสมอ [26] วิธีนี้เป็นวิธีที่ง่ายและประมวลผลได้ไวที่สุด แต่ไม่สามารถแก้ปัญหาความกำกวมของกฎได้ เช่น พยัญชนะบางตัวสามารถเป็นได้ทั้งพยัญชนะต้นและตัวสะกด

2) การตัดคำโดยอาศัยคำในพจนานุกรม (Dictionary approach) เป็นการนำคำในพจนานุกรมไปเทียบกับข้อมูลหากพบรูปตรงก็จะตัดได้ ซึ่งเป็นการตัดคำแบบ ตัดคำให้ยาวที่สุดก่อน (longest matching) [27] ต่อมามีการเสนอแนวคิดการตัดคำจากความเหมือนมากที่สุด (Maximal matching) คือตัดคำให้ได้จำนวนน้อยที่สุด เพราะภาษาไทยมีการสร้างคำจากการประสมคำจำนวนมาก เช่นเมื่อเจอคำว่า “ลูกเสือ” โอกาสที่จะเป็นคำเดียวคือ “ลูกเสือ” ย่อมมีมากกว่าสองคำคือ “ลูก” “เสือ” ดังนั้นการเลือกผลที่มีจำนวนคำน้อยสุดจึงมีโอกาสถูกต้องมากกว่า แต่อย่างไรก็ตามแม้ว่าการตัดคำด้วยพจนานุกรมนี้มีความถูกต้องมากกว่าการตัดคำโดยใช้กฎ แต่ถ้าพบรูปแบบของคำที่ไม่รู้จัก (Out-of-Vocabulary) ก็ไม่สามารถตัดคำได้ อาจเพราะคลังคำศัพท์มีไม่เพียงพอ เป็นคำศัพท์ใหม่ ๆ หรือการสะกดผิด เป็นต้น

3) การตัดคำโดยใช้ค่าสถิติจากคลังเอกสาร (Corpus based approach) เป็นวิธีการที่ใช้ค่าสถิติการเกิดคำและหน้าที่ของคำเข้ามาช่วยในการคำนวณหาความน่าจะเป็น เพื่อเลือกแบบที่มีโอกาสการเกิดมากที่สุด วิธีการนี้มีความถูกต้องมากกว่า 2 วิธีการแรก แต่มีข้อจำกัดคือต้องมีฐานข้อมูลที่มีการตัดคำไว้อย่างถูกต้อง ร่วมกับการใช้เทคนิคการเรียนรู้ต่าง ๆ เช่น Support Vector Machine (SVM) การเรียนรู้เชิงลึก (Deep Learning)

ในงานวิจัยนี้ใช้การตัดคำโดยใช้ค่าสถิติจากคลังเอกสาร โดยการสร้างคลังคำศัพท์ด้านกฎหมายอาญา (Criminal Dictionary) และใช้เทคนิค Longest Matching ร่วมกับตัวตัดคำ Deepcut [28] ซึ่งเป็นเทคนิคการตัดคำ โดยใช้การเรียนรู้เชิงลึกด้วยโครงข่ายประสาทเทียม Convolutional Neural network (CNN) แบบ 1 มิติ จากชุดข้อมูล BEST 2009 ของ NECTEC ซึ่งเป็นคลังข้อมูลภาษาไทยที่มีการตัดคำขนาดใหญ่แล้ว เพื่อจัดการคำศัพท์ที่ไม่มีในคลังคำศัพท์ด้านกฎหมายอาญา ตามรูปที่ 2-37

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Original Text	Criminal dictionary on Deepcut
พร้อมด้วยเจ้าของคดีนำกำลังคุมตัวแรงงานต่างด้าวชาวมอญไปทำแผนประกอบคำรับสารภาพภายในสวนปาล์มพื้นที่หลังก่อเหตุข่มขืนแล้วฆ่ายายโดยมีบรรดาทางญาติผู้ตายเดินทางมาดูหน้าเป็นจำนวนมาก	พร้อม|ด้วย|เจ้าของ|คดี|นำกำลัง|คุมตัว|แรงงาน|ต่างด้าว|ชาวมอญ|ไป|ทำแผน|ประกอบคำรับสารภาพ|ภายใน|สวนปาล์ม|พื้นที่|หลังก่อเหตุ|ข่มขืน|แล้วฆ่า|ยาย|โดย|มี|บรรดา|ทาง|ญาติ|ผู้ตาย|เดินทาง|มา|ดู|หน้า|เป็น|จำนวนมาก

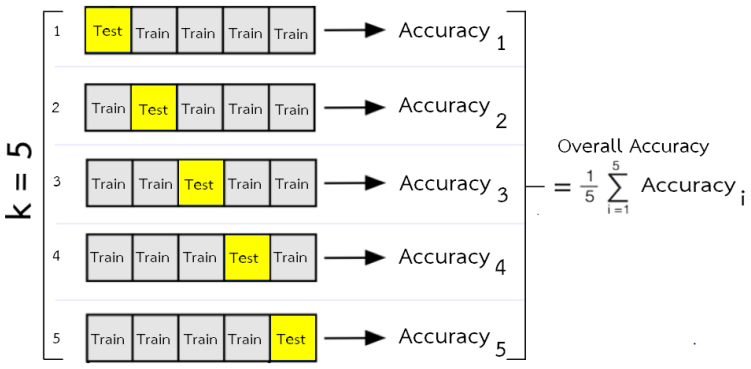
รูปที่ 2-37 การตัดคำโดยอาศัยคลังข้อมูลด้านกฎหมายอาญาร่วมกับการเรียนรู้เชิงลึก Deepcut

2.16 การวัดประสิทธิภาพ

2.16.1 Cross Validation การตรวจสอบแบบไขว้ [29] เป็นเทคนิคในการวัดประสิทธิภาพการทำงานของแบบจำลองในการเรียนรู้ของเครื่อง (Machine Learning) โดยมีหลักการทำงานเริ่มจากการสุ่มข้อมูลออกเป็นส่วนเท่า ๆ กัน เช่น 5 ส่วน หรือ 10 ส่วน เป็นต้น ตัวเลขจำนวนส่วนเหล่านี้เรียกว่าค่า “k” จึงเป็นที่มาของชื่อ k-fold cross validation ถ้ากำหนดให้ k เท่ากับ 5 หมายถึงการแบ่งข้อมูลเป็น 5 ส่วน เพื่อสร้างและทดสอบแบบจำลอง เพราะการแยกข้อมูลทดสอบเพียงครั้งเดียวอาจไม่สามารถแสดงประสิทธิภาพของแบบจำลองได้ ถ้าข้อมูลที่ถูกนำมาทดสอบมีการกระจายตัวไม่ดีพอ

ดังนั้นถ้าแบ่งข้อมูลเป็น 5 ส่วน ตามที่แสดงในรูปที่ 2-38 โดยรอบแรก กำหนดให้ข้อมูลชุดที่ 1 เป็นข้อมูลตรวจสอบ (Testing Data) ข้อมูลชุดที่เหลืออีก 4 ส่วนใช้เป็นข้อมูลฝึกฝน (Training Data) (2,3,4,5) รอบที่ 2 กำหนดให้ข้อมูลชุดที่ 2 เป็นข้อมูลตรวจสอบ ข้อมูลชุดที่เหลือเป็นข้อมูลฝึกฝน (1,3,4,5) วนไปจนชุดข้อมูลทั้งหมดถูกนำมาใช้ในการทดสอบ จากนั้นนำประสิทธิภาพที่ได้จากการฝึก k ครั้งมาหาเฉลี่ย เพื่อแสดงถึงประสิทธิภาพของแบบจำลอง ตามรูปที่ 2-38

ในงานวิจัยนี้ใช้ k-fold cross-validation โดยกำหนดค่า k=5 ในการวัดค่าความแม่นยำของต้นไม้ตัดสินใจ ก่อนนำมาสร้างเป็นกฎในการพิจารณาการทำนายแนวทางการตัดสินของคดีอาญา



รูปที่ 2-38 5-fold cross-validation และการหาค่าเฉลี่ยความถูกต้อง

2.16.2 Confusion Matrix [30] เมทริกซ์ที่ใช้วัดประสิทธิภาพของแบบจำลองประเภทการจำแนกข้อมูล ทั้งแบบสองกลุ่ม (Binary Classification) และหลายกลุ่ม (Multiclass Classification) โดยใช้ผลรวมของจำนวนของกลุ่มข้อมูลในชุดข้อมูลทดสอบ ที่ทำนายได้ถูกต้องและไม่ถูกต้อง นำมาเขียนเป็นตาราง โดยแนวตั้งกำหนดเป็นค่าที่ได้จากการทำนายของแบบจำลอง (Predicted Class) และแนวนอนเป็นค่าที่ความจริง (Actual Class) ตามที่แสดงในตารางที่ 2-1

ตารางที่ 2-1 ตารางคอนฟิวชันเมทริกต์

	Actually Positive	Actually Negative
Predicted Positive	**TP**	**FP**
Predicted Negative	**FN**	**TN**

ตารางที่ 2-1 แสดง Confusion Matrix ขนาด 2x2 ใช้ในการประเมินประสิทธิภาพแบบจำลองการจำแนกแบบสองกลุ่ม (Binary Classification) โดยกำหนดให้

TP (True Positive) คือ แบบจำลองทำนายว่า “ใช่” และ ค่าที่แท้จริงมีค่าเป็น “ใช่”

FP (False Positive) คือ แบบจำลองทำนายว่า “ใช่” และ ค่าที่แท้จริงมีค่าเป็น “ไม่ใช่”

FN (False Negative) คือ แบบจำลองทำนายว่า “ไม่ใช่” และ ค่าที่แท้จริงมีค่าเป็น “ใช่”

TN (True Negative) คือ แบบจำลองทำนายว่า “ไม่ใช่” และ ค่าที่แท้จริงมีค่าเป็น “ไม่ใช่”

โดยมีวิธีวัดประสิทธิภาพแบบจำลองในงานจำแนกประเภท ดังนี้

1) ค่าความถูกต้อง (Accuracy) เป็นการวัดความถูกต้องหรือความแม่นยำโดยรวมของแบบจำลอง โดยคำนวณรวมทุกคลาส ตามสมการที่ 2.68

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{2.68}$$

2) ค่าความแม่นยำ (Precision) เป็นการวัดความแม่นยำของข้อมูล โดยคำนวณแยกทีละคลาส และสนใจเฉพาะส่วนของการทำนายเท่านั้น ตามสมการที่ 2.69

$$Precision = \frac{TP}{TP+FP} \tag{2.69}$$

3) ค่าความระลึก (Recall) เป็นการวัดความถูกต้องของแบบจำลอง โดยพิจารณาแยกทีละคลาส และสนใจเฉพาะส่วนของความเป็นจริงเท่านั้น ตามสมการที่ 2.70

$$Recall = \frac{TP}{TP+FN} \tag{2.70}$$

4) Specificity สร้างขึ้นมาเพื่อวัดประสิทธิภาพของแบบจำลอง โดยใช้ค่าเฉลี่ยของการวัดระหว่าง Precision และ Recall ตามสมการที่ 2.71

$$Specificity = \frac{TN}{TN+FP} \tag{2.71}$$

5) F1-Score สร้างขึ้นมาเพื่อวัดประสิทธิภาพของแบบจำลอง โดยใช้ค่าเฉลี่ยของการวัดระหว่าง Precision และ Recall ตามสมการที่ 2.72 แต่ในงานวิจัยนี้ใช้ค่าเฉลี่ยมหภาคเอฟวัน (Macro-average F1 score) ซึ่งเกิดจากการนำค่าตัววัดประสิทธิภาพแต่ละตัวมาหาค่าเฉลี่ย กำหนดให้ค่า $F1_1$ แทนค่าเอฟวันที่สนใจคำตอบที่เป็น 1

$F1_0$ แทนค่าเอฟวันที่สนใจคำตอบที่เป็น 0

โดยการหาค่าเฉลี่ยมหภาคของเอฟวัน ตามที่แสดงในสมการที่ *2.73*

$$F1 - Score = \frac{2(Precision*Recall)}{Precision+Recall} \tag{2.72}$$

$$F1_{macro} = \frac{2(Precision*Recall)}{Precision+Recall} \tag{2.73}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

งานวิจัยนี้ในส่วนที่เป็นการเรียนรู้เชิงลึกของการจำแนกเนื้อหาข่าวแบบไม่มีโครงสร้างที่มีลักษณะเป็นทั้งคำตอบสองกลุ่มและหลายกลุ่ม การเลือกใช้ค่าความถูกต้อง (Accuracy) เพียงอย่างเดียวอาจไม่เหมาะสมครอบคลุมการจำแนกแบบหลายกลุ่ม จึงใช้ค่าค่าเฉลี่ยมหภาคของเอฟวัน ($F1_{macro}$) ร่วมร่วมกับค่าความถูกต้อง (Accuracy) ในการวัดประสิทธิภาพของแบบจำลอง

2.17 เทคนิคการสุ่มเพิ่มตัวอย่างข้างน้อยสังเคราะห์ (Synthetic Minority Over-sampling Technique : SMOTE) [31]

ในงานจำแนกประเภทของข้อมูล (Classification) ประสิทธิภาพของแบบจำลอง ส่วนหนึ่งขึ้นอยู่กับปริมาณของกลุ่มตัวอย่างข้อมูล (Dataset) หากชุดข้อมูลแต่ละชุดมีปริมาณที่แตกต่างกันมาก (Imbalanced Class) การประมวลผลข้อมูลย่อมทำให้ประสิทธิภาพลดลง เพราะจะทำให้มีการเรียนรู้

แต่ข้อมูลกลุ่มมาก (Majority Class) ผลที่ได้ก็จะจำแนกไปในข้อมูลกลุ่มมาก ส่วนข้อมูลกลุ่มน้อย (Minority Class) ไม่ได้รับการเรียนรู้และถูกจำแนกทั้งที่เป็นข้อมูลสำคัญ

การแก้ปัญหาการแบ่งกลุ่มข้อมูลที่ไม่สมดุลแบ่งออกเป็น 2 วิธี ได้แก่ วิธีการแก้ปัญหาด้วยอัลกอริทึมการเรียนรู้ (Algorithmic Level Approach) โดยทำการปรับปรุงกระบวนการและเทคนิคสำหรับการเรียนรู้ เพื่อให้ตัวจำแนกมีประสิทธิภาพมากที่สุด เช่น การใช้แบบจำลองหลายตัว เพื่อช่วยในการหาคำตอบ หรือ การสุ่มข้อมูลฝึกสอนเป็นหลายชุด แต่สร้างโมเดลด้วยเทคนิคเดียวกันทั้งหมด เป็นต้น และอีกวิธีและงานวิจัยนี้เลือกใช้คือ วีธีแก้ปัญหาระดับข้อมูล (Data Level Approach)

Data Level Approach สำหรับการจัดการปัญหาในระดับข้อมูลนั้น สามารถแก้ปัญหาได้โดยใช้วิธีการสุ่มตัวอย่างซ้ำ (Resampling Techniques) ซึ่งมีวิธีการเลือกใช้ที่หลากหลาย เช่น Over-Sampling เป็นวิธีการที่ใช้ในการสุ่มเพิ่มจำานวนข้อมูลในกลุ่มข้างน้อยให้มีจำนวนใกล้เคียงกับข้อมูลในกลุ่มข้างมากและ Under-Sampling เป็นวิธีในการสุ่มเลือกข้อมูลจากกลุ่มข้างมากให้ได้จำนวนที่ใกล้เคียงกับกลุ่มข้างน้อย ในงานวิจัยนี้ได้แก้ปัญหาความไม่สมดุลของคลาสในระดับข้อมูล ด้วยเทคนิคการสุ่มเพิ่มตัวอย่างข้างน้อยสังเคราะห์ (Synthetic Minority Over-sampling Technique : SMOTE) เพื่อให้ตัวอย่างข้างน้อยมีจำนวนเท่ากับตัวอย่างข้างมาก ตามสูตรการคำนวณเพื่อหาตัวอย่างข้อมูลข้างน้อยตามสมการที่ 2.73 เนื่องจากข้อมูลตัวอย่างมีจำนวนน้อยและเพื่อป้องกันการสูญหายของข้อมูลสำคัญในชั้นข้อมูลจำนวนกลุ่มมาก ตามที่แสดงในรูปที่ 2-39 และ 2-40

รูปที่ 2-39 แสดงการทำงานของเทคนิค SMOTE ในการเพิ่มข้อมูลตัวอย่างข้างน้อย

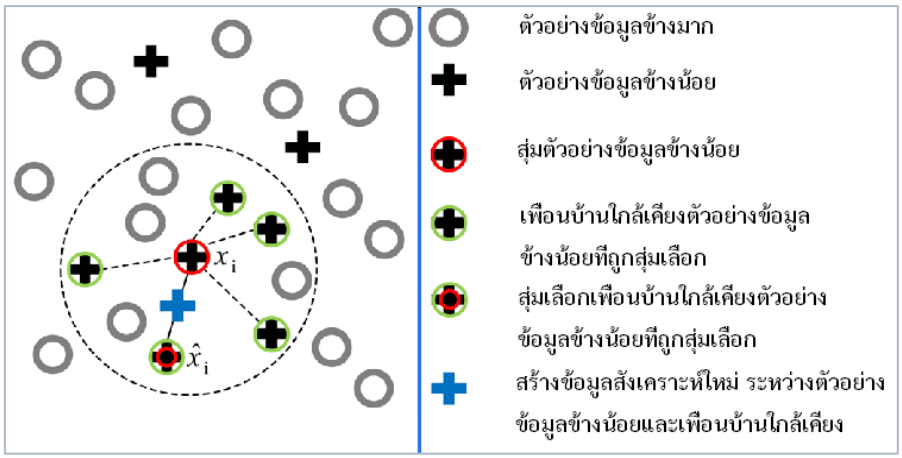
$$x_{new} = x_i + rand[0,1] * (\hat{x_i} - x_i) \tag{2.74}$$

เมื่อ x_{new} คือ ข้อมูลสังเคราะห์ใหม่ของข้อมูลข้างน้อย

x_i คือ ข้อมูลข้างน้อยที่สุ่มครั้งที่ i

$rand[0,1]$ คือ ค่าสุ่มตั้งแต่ 0-1

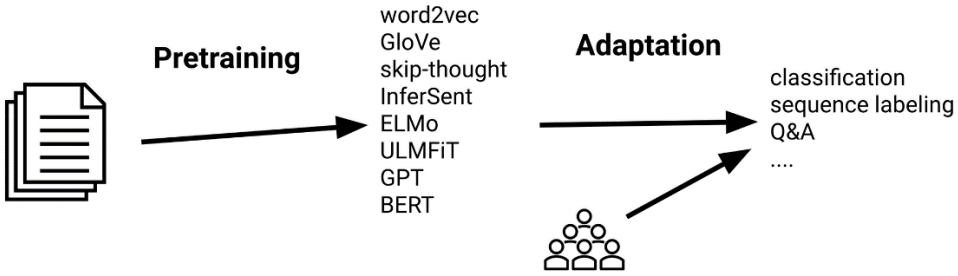
$\hat{x_i}$ คือ ข้อมูลเพื่อนบ้านที่ใกล้ที่สุดแบบสุ่มของ x_i



รูปที่ 2-40 การสังเคราะห์เพิ่มตัวอย่างข้างน้อย (SMOTE) [31]

2.18 การถ่ายโอนการเรียนรู้ (Transfer Learning) [32]

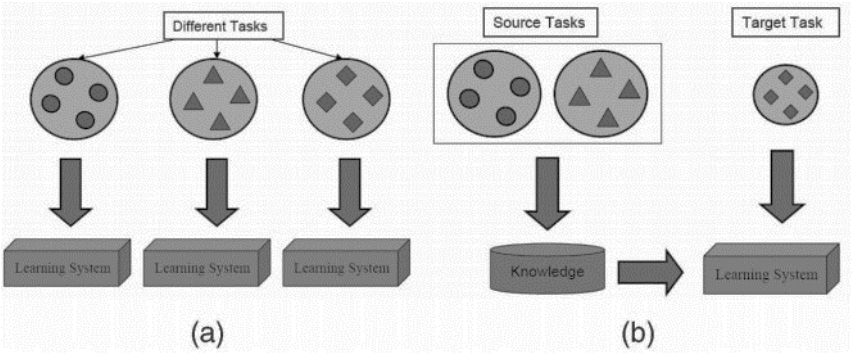
ประสิทธิภาพของแบบจำลองในการเรียนรู้เชิงลึก (Deep Learning) ที่มีชั้นซ่อน (Hidden Layer) หลาย ๆ ชั้น ปัจจัยหนึ่งขึ้นอยู่กับปริมาณข้อมูลการฝึกฝน (Dataset) ดังนั้นจึงมีความจำเป็นที่ต้องใช้ข้อมูลจำนวนมาก ส่งผลให้พารามิเตอร์มีจำนวนมหาศาล การเริ่มต้นฝึกฝนแบบจำลองที่เริ่มจากศูนย์ (From Scratch) โดยใช้ค่าถ่วงน้ำหนักเริ่มต้นแบบสุ่ม (Weight Initialization) จึงต้องใช้ทั้งข้อมูลฝึกฝนจำนวนมมาก และฮาร์ดแวร์ที่มีความสามารถในการประมวลผลสูง จึงจะได้ประสิทธิภาพของแบบจำลองที่ดี



รูปที่ 2-41 การถ่ายโอนการเรียนรู้ [32]

การถ่ายโอนความรู้หรือ Transfer Learning คือการนำองค์ความรู้ที่ได้จากฝึกฝนด้วยข้อมูลจำนวนมากด้วยฮาร์ดแวร์ประสิทธิภาพสูงในงานด้านอื่นที่เปรียบเสมือนเป็น (General Model) มาปรับใช้กับงานที่ต้องการ เช่นงานจำแนกข้อความ เป็นต้น โดยนำค่าพารามิเตอร์ที่ได้จากแบบจำลองที่ผ่านการฝึกฝนแล้ว กำหนดเป็นค่าเริ่มต้นให้กับงานใหม่ ซึ่งจะช่วยลดเวลาการการฝึกฝนลงแบบจำลอง และเพิ่มประสิทธิภาพของแบบจำลองใหม่ที่เป็นงานเฉพาะทางได้ [33] ตามที่แสดงใน

รูปที่ 2-41 และแสดงการเปรียบเปรียบระหว่างการฝึกฝนแบบจำลองด้วยข้อมูลตนเอง และการใช้คลังข้อมูลอื่น ตามที่แสดงในรูปที่ 2.42



รูปที่ 2-42 เปรียบเทียบการเรียนรู้แบบดั้งเดิม (a) และการใช้ Transfer Learning (b) [33]

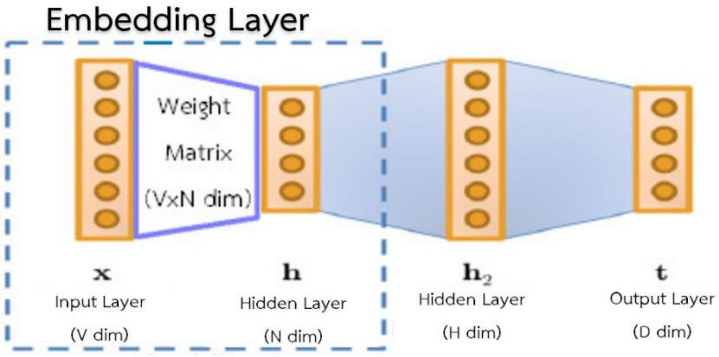
ในงานวิจัยนี้ได้ใช้เทคนิคการถ่ายโอนข้อมูลมาปรับใช้ในขั้นตอนการฝึกฝนแบบจำลอง ด้วยการใช้ค่าถ่วงน้ำหนักที่ผ่านการฝึกฝนแบบจำลองทางภาษาจากคลังข้อมูลขนาดใหญ่ มาเป็นค่าถ่วงน้ำหนักเริ่มต้น ดังนี้

1) FastText [34] คือ Pretrained Word Embedding ของ Facebook ที่ผ่านการฝึกฝนด้วยข้อมูลจาก Thai Wikipedia จำนวน 2,000,000 คำ ขนาด 300 มิติ

2) Thai2Vec [35] คือ Pretrained Word Embedding ของกลุ่ม PyThaiNLP ที่เทรนด้วยข้อมูลจาก Thai Wikipedia จำนวน 60,000 คำ ขนาด 300 มิติ

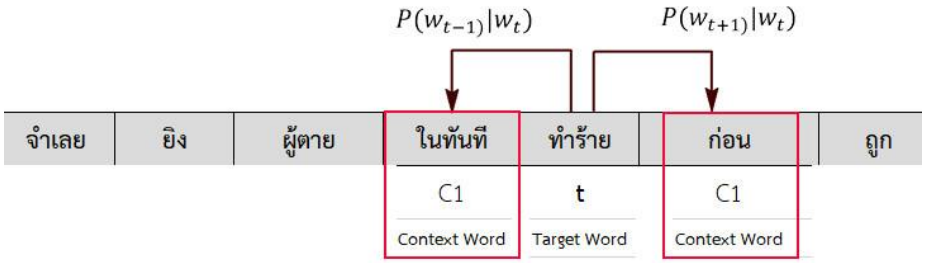
3) Word2Vec [36] เป็นอัลกอริทึมหนึ่งในการสร้างการฝังคำ (Word Embedding) โดยเรียนรู้การสร้างเวกเตอร์จากคำบริบทโดยรอบ เพื่อสร้างตัวจำแนก (Classifier) จากแบบจำลองการเรียนรู้โครงข่ายประสาทเทียม ในการหาความน่าจะเป็นของคำที่อยู่โดยรอบเกิดใกล้กับคำที่เป็นเป้าหมาย จากนั้นนำค่าถ่วงน้ำหนักและค่าไบแอสที่ได้ในชั้นซ่อน (Hidden Layer) ไปเป็นค่าการฝังคำ ซึ่งเป็นคอนเซปต์หนึ่งของ Prediction Base Method ตามที่แสดงในรูปที่ 2-43 ข้อดีของการสร้างเวกเตอร์ด้วย Word2Vec คือได้เวกเตอร์ขนาดที่สามารถกำหนดความยาวหรือมิติของเวกเตอร์ได้แทนที่ต้องสร้างเวกเตอร์ที่มีความยาวเท่าขนาดของคำศัพท์แต่มีเลข 0 เยอะ (Sparse Vectors) และเมื่อเป็นเวกเตอร์สั้นลงแล้วทำให้ค่าพารามิเตอร์ลดลงด้วย จึงเป็นการป้องกันปัญหาการยึดติดข้อความที่ใช้สอนมากเกินไป (overfitting) โดยแบ่งเป็น 2 ประเภท ได้แก่ (Continuous Bag of Word : CBOW) คือการใช้คำที่อยู่รอบ ๆ (Context Words) เป็นข้อมูลขาเข้า เพื่อทำนายคำเป้าหมาย (Target Word) และ Skip-gram คือการใช้คำเป้าหมายในการทำนายความน่าจะเป็นของคำที่อยู่โดยรอบ การใช้ในงานวิจัยนี้เลือกใช้เทคนิคของ Skip-gram ขนาดเวกเตอร์ของแต่ละคำเท่ากับ 300 มิติ เนื่องจากข้อมูลที่ใช้ในการเรียนรู้มีจำนวนน้อย ในการฝึกฝนแบบจำลองแบบไม่มีผู้สอน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 2-43 ค่าถ่วงน้ำหนักที่ได้จากการฝึกฝนแบบจำลองถูกใช้ในการกำหนดค่าเริ่มต้นชั้นซ่อน

Embedding Layer คือ ชั้นที่ใช้เป็นตัวแทนของแต่ละคำ คือค่าถ่วงน้ำหนัก (Weight) จากชั้นข้อมูลขาเข้า (Input Layer) ไปยังชั้นซ่อน (Hidden Layer) เป็นค่า Word Embedding



รูปที่ 2-44 Skip-gram Model

ตามรูปที่ 2-44 กำหนดให้คำเป้าหมาย (target word : t) คือ “ทำร้าย” และกำหนดขนาด Windows size = 1 (Context = $\pm$1) ดังนั้นคำบริบทรอบข้าง ข้างละ 1 คำ คือ “ ในทันที” และ “ก่อน” ดังนั้นในการหาเวกเตอร์ของคำศัพท์แต่ละคำ ถ้าคำนั้นเป็น Positive Class จะต้องมีความใกล้เคียงกับคำบริบทรอบข้าง ซึ่งกำหนดให้เป็นคลาส 1 ในขณะเดียวกันคำนั้น ซึ่งเป็น Negative Class ต้องแตกต่างจากคำที่ไม่ใช่คำบริบทที่อยู่รอบข้าง เพราะคำเหล่านั้นกำหนดเป็นคลาส 0 ดังแสดงในตารางที่ 2-2

ตารางที่ 2-2 แยกคำที่เป็น Positive และ Negative

Positive Class (1)		Negative Class (0)	
target	Context	target	Context
ทำร้าย	ในทันที	ทำร้าย	จำเลย
ทำร้าย	ก่อน	ทำร้าย	ผู้ตาย

การที่กำหนดให้จำนวน Negative Class มีค่าเท่ากับสองคำเช่นเดียวกับ Positive Class เพื่อแก้ปัญหาคลาสไม่สมดุลในการฝึกการเรียนรู้ (Train) แต่คำของคลาส 0 เป็นค่าใดก็ได้ เพราะได้จากการสุ่ม เรียกเทคนิคนี้ว่า Negative Sampling ค่าที่ถูกสุ่มมาเป็นคำให้กับคลาส 0 นั้นเรียกว่า Noise Word ซึ่งเพื่อให้คำฟุ่มเฟือยที่อาจถูกเลือกกว่ามีค่าความน่าจะเป็นลดลง และช่วยให้คำที่พบไม่บ่อยในเอกสาร (Rare Word) มีค่าความน่าจะเป็นสูงขึ้น ตามสมการที่ 2.74 โดยกำหนดให้ α = 0.75

$$P_\alpha(w) = \frac{count(W)^\alpha}{\sum_{w'} count(W')^\alpha} \tag{2.75}$$

ในคำที่เจอบ่อยเช่นคำฟุ่มเฟือย โอกาสที่จะสุ่มเจอมีค่า 0.99 ส่วนคำที่เป็นคำที่ไม่พบบ่อยอาจมีโอกาสเพียง 0.01 แต่เมื่อผ่านสมการ จะทำให้โอกาสในการสุ่มได้เป็น 0.97 และ 0.3 ตามลำดับ

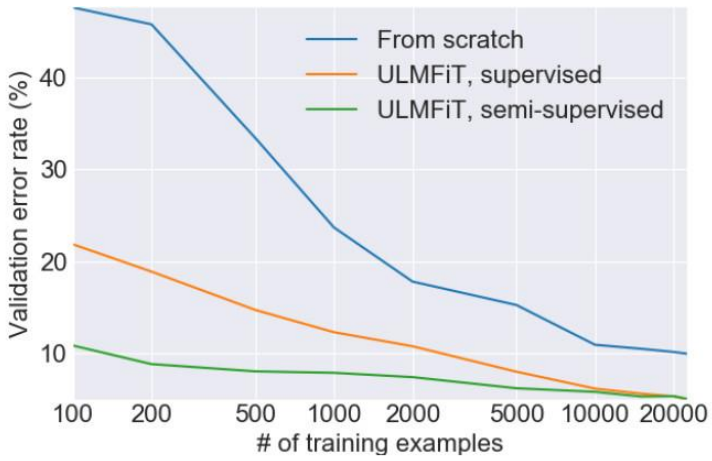
วัตถุประสงค์ในการเรียนรู้คือต้องการให้ได้ Embedding ที่ทำให้ได้ค่าความเหมือน (Similarity) ของคำเป้าหมายและคำบริบทใน Positive Class ให้มากที่สุด ขณะเดียวกันก็ลดความคล้ายของคำเป้าหมายและคำบริบทใน Negative Class ให้น้อยที่สุด นั้นก็จะใช้ SGD เพื่อฝึกสอนแบบจำลองตามชุดข้อมูลฝึกสอน (Training Set) ดังนี้สมการที่ 2.75

$$L(\theta) = \sum_{(t,c)\epsilon+} logP(+|t,c) + \sum_{(t,c)\epsilon-} logP(-|t,c) \tag{2.76}$$

เมื่อ $L(\theta)$ คือ Learning Object ของค่าถ่วงน้ำหนักที่ได้จากการเรียนรู้

แบบจำลองภาษา (Language Model : LM) คือการใช้เทคนิคทางสถิติและความน่าจะเป็นต่าง ๆ เพื่อกำหนดความน่าจะเป็นของลำดับคำที่กำหนดที่เกิดขึ้นในประโยค ในปัจจุบันแบบจำลองส่วนใหญ่มักจะเป็นการใช้โครงข่ายประสาทเทียมเข้ามาช่วยคำนวน เนื่องจากสามารถจับข้อมูลบริบทรอบ ๆ ได้ เช่น RNN, LSTM หรือ Word2Vec เป็นต้น

การถ่ายโอนการเรียนรู้ทำให้ลดขนาดของข้อมูลการฝึกฝนลดลง ขณะที่ประสิทธิภาพของแบบจำลองดีขึ้น [37] โดยใช้เวลาในการฝึกฝนน้อยลง ตามที่แสดงรูปที่ 2-45



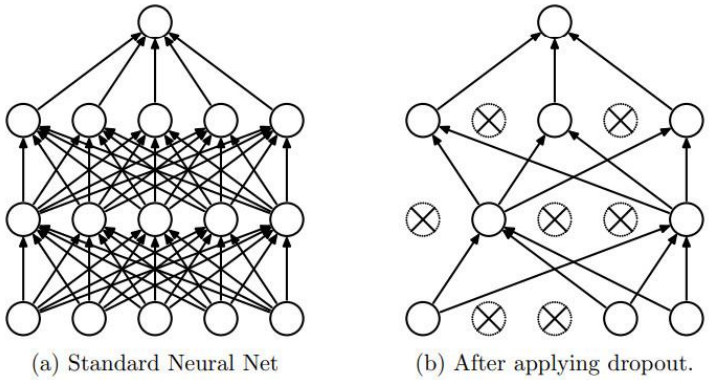
รูปที่ 2-45 เปรียบเทียบการเรียนรู้แบบตั้งแต่เริ่มต้นเทียบเรียนรู้โดยใช้ ULMFiT [37]

2.19 การหยุดเรียนรู้กลางคัน (Dropout) [38]

Overfitting หรือปัญหาการยึดติดข้อความที่ใช้ฝึกสอน (Training set) มากเกินไป ซึ่งเป็นปัญหาหนึ่งของการเรียนรู้เชิงลึก เพราะความซับซ้อนของแบบจำลองที่มีปริมาณโหนดซ้อนกันหลายชั้นทำให้มีพารามิเตอร์ในการฝึกฝนมาก ดังนั้นการทดสอบกับข้อมูลฝึกฝน (Training set) จึงให้ค่าความถูกต้องที่สูงหรือมีค่าไบแอสต่ำ (Low-Bias) แต่เมื่อทำการทดสอบด้วยข้อมูลทดสอบ (Test set) ทำให้เกิดค่าความแปรปรวนสูง (High-Variance) ซึ่งสามารถแก้ไขปัญหานี้ด้วยการทำ Regularization ได้แก่ L1, L2, DropConnect และ Dropout

Dropout เป็นหนึ่งในวิธี Regularization ของการเรียนรู้เชิงลึกที่ได้รับความนิยมและใช้ในงานวิจัยนี้ ซึ่งเป็นวิธีการลดขนาดของพารามิเตอร์ลงบางส่วนหรือบางชั้นที่ต้องการได้ โดยการสุ่มเลือกโหนดที่จะหยุดการเรียนรู้ตามจำนวนที่กำหนดในแต่ละรอบการฝึกฝน เพื่อปรับปรุงค่าพารามิเตอร์ของแบบจำลอง ตามรูปที่ 2-46

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 2-46 เปรียบเทียบโครงข่ายทั่วไป และการเพิ่มการหยุดการเรียนรู้บางโหนด [38]

2.20 ต้นไม้ตัดสินใจ (Decision Tree) [39]

ต้นไม้ตัดสินใจ เป็นแบบจำลองประเภท Rule-Based คือ สร้างกฎ If-Else จากค่าของแต่ละคุณลักษณะ (Attribute) ในการสร้างต้นไม้ตัดสินใจคือสิ่งสำคัญคือการแบ่ง (Split) ค่าคุณลักษณะแต่ละครั้ง จะต้องทำการลด (Minimize) ค่าของ Cost Function ให้น้อยที่สุด ซึ่งได้กลายเป็นอัลกอริทึมพื้นฐานที่ใช้สำหรับเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมอื่น ๆ ต้นไม้ตัดสินใจแบ่งออกเป็น 2 ประเภท คือ Regression Tree สำหรับทำ Regression และ Classification Tree สำหรับงานประเภทการจำแนกหรือบางครั้งก็เรียก Decision Tree ทั้ง 2 ประเภท รวมกันว่า Classification And Regression Tree (CART) อัลกอริทึมการสร้างต้นไม้ตัดสินใจที่สำคัญคือ ID3 (Iterative Dichotomize) และได้รับการปรับปรุงเป็น C4.5 โดยต้องการ 3 ข้อมูลขาเข้าประกอบด้วย ชุดข้อมูลที่จะทำการพิจารณา เซ็ตของรายการคุณลักษณะและวิธีการในการคัดเลือกคุณลักษณะที่มีความสัมพันธ์กับคลาสมากที่สุดขึ้นมาเป็นโหนดราก หลังจากนั้นก็จะค้นหาคุณลักษณะถัดไปเรื่อย ๆ โดยตัวชี้วัดการเลือกคุณลักษณะที่ใช้ในงานวิจัยนี้คือ ดัชนีจีนี่ (Gini Index) ซึ่งเป็นตัวชี้วัดที่แบ่งชุดข้อมูลจะแบ่งออกเป็น 2 ชุดย่อยเท่านั้นหรือเป็นตัวชี้วัดที่จะสามารถทำงานได้กับต้นไม้แบบไบนารี่ สำหรับค่าแบบไม่ต่อเนื่องและแตกต่างกัน โดยที่ทำการพิจารณาความไม่บริสุทธิ์ของชุดข้อมูล D ซึ่งจะมีเซตของแถวที่มีหมวดหมู่ของข้อมูลไม่เหมือนกันอยู่ โดยเริ่มจากคำนวณหาค่าความไม่บริสุทธิ์ของชุดข้อมูล D ก่อน ตามสมการที่ 2.76

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{2.77}$$

เมื่อ D คือ ชุดข้อมูลทั้งหมด ตั้งแต่เรคคอร์ดที่ 1 ถึง m

p_i คือ ความน่าจะเป็นที่เรคคอร์ดหนึ่ง ๆ จะมีหมวดหมู่ของข้อมูลเป็นหมวดหมู่ C_i

จากนั้นทำการพิจารณาทุกสับเซตที่เป็นไปได้ที่เกิดขึ้นในคุณลักษณะที่เราสนใจ ที่ถูกจัดไว้ใน 2 เซตย่อยด้วยกัน เช่น คุณลักษณะผลจากการกระทำผิด (Causation) ที่มีค่า 3 ค่าที่เป็นไปได้คือ {nothing, injury, die} ซึ่งจากทั้ง 3 ค่าที่เกิดขึ้น สามารถแบ่งค่าทั้ง 3 ออกเป็นเซตย่อย 2 เซตได้ดังนี้ {<{nothing, injury}, {die}>, <{nothing, die}, {injury}>,<{injury, die}, {nothing}>, <{nothing}, {injury, die}>, <{injury}, {nothing, die}>, <{die}, {nothing, injury}>} เมื่อแบ่งค่าที่เกิดขึ้นออกเป็น 2 เซตย่อยแล้ว ก็นำเซตย่อยทั้ง 2 ที่แบ่งไว้มาทำการคำนวณค่าน้ำหนักรวมของค่าความไม่บริสุทธิ์ โดยการพิจารณาคุณลักษณะ A ใด ๆ โดยทำการแบ่งชุดข้อมูล D ออกเป็น 2 ชุดข้อมูลย่อยคือ D_1 และ D_2 ดังนั้นเราสามารถหาค่าดัชนีจีนี่เมื่อทำการพิจารณาคุณลักษณะ A ภายใต้การพิจารณา D_1 และ D_2 ได้ตามสมการที่ 2.77

$$Gini_A(D) = \left|\frac{D_1}{D}\right| Gini(D_1) + \left|\frac{D_2}{D}\right| Gini(D_2) \tag{2.78}$$

และเมื่อคำนวณค่าดัชนีจี่นี่สำหรับ 2 เซตย่อยใด ๆ ภายใต้คุณลักษณะ A 2 เซต D_1 และ D_2 ใด ที่ให้ค่า $Gini_A(D)$ น้อยที่สุดจะถูกเลือกเป็นตัวแทนของคุณลักษณะ A จากนั้นการแบ่งชุดข้อมูลคือทำการคำนวณค่าความไม่บริสุทธิ์ นั่นคือ $\Delta Gini(A)$ โดยเมื่อทำการคำนวณค่า $\Delta Gini(A)$ สำหรับทุก ๆ คุณลักษณะแล้ว คุณลักษณะใดที่มีค่า $\Delta Gini(A)$ มากที่สุดจะถูกเลือกเพื่อใช้แบ่งชุดข้อมูลย่อย ตามสมการที่ 2.78

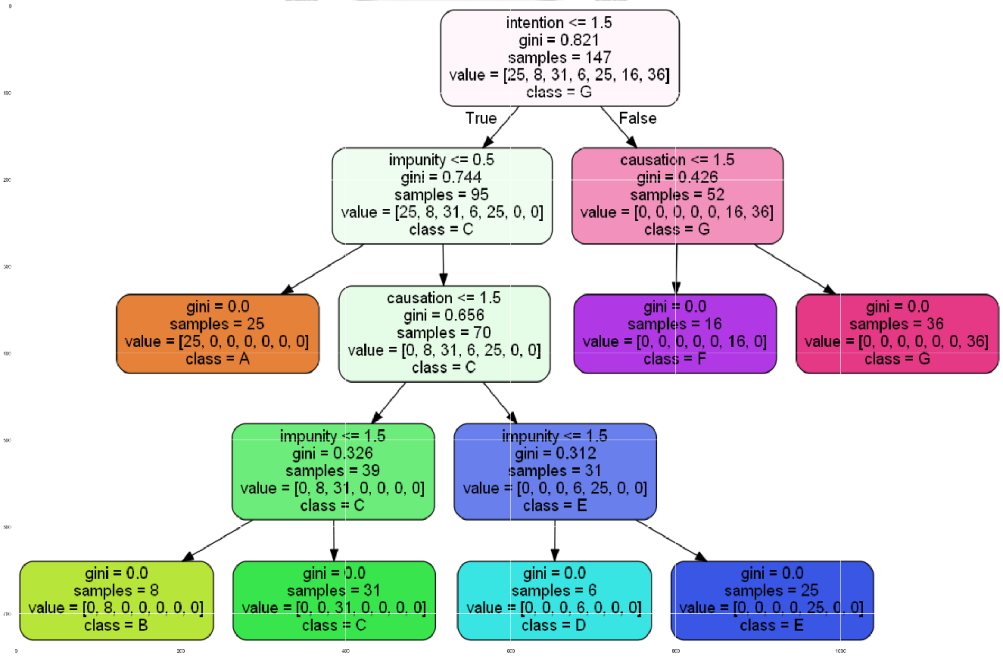
$$\Delta Gini(A) = Gini(D) - Gini_A(A) \tag{2.79}$$

ในงานวิจัยนี้ได้เลือกใช้ตัวชี้วัดการแบ่งข้อมูลด้วยค่าดัชนีจีนี่ (Gini Index) เพราะเป็นงานจำแนก ที่แยกเป็นค่าจริงกับเท็จในแต่ละคุณลักษณะ โดยค่าจีนี่สูงสุดคือคุณลักษณะเจตนา (Intention) ซึ่งถูกเลือกเป็นโหนดรากของการสร้างต้นไม้ตัดสินใจในงานวิจัยนี้ รวมทั้งการเลือกคุณลักษณะสำคัญ (Feature Importance) ตามสมการที่ 2.79 ที่ส่งผลต่อการกำหนดบทลงโทษเหลือเพียง 4 คุณลักษณะ ตามรูปที่ 2-49

$$fi = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in allnodes} ni_k} \tag{2.80}$$

เมื่อ fi_i คือ ค่าความสำคัญของ Feature i

ni_j คือ ค่าความสำคัญของโหนด j



รูปที่ 2-47 คุณลักษณะสำคัญของต้นไม้ตัดสินใจ

การสกัดกฎจากต้นไม้ตัดสินใจ ต้นไม้ตัดสินใจอาจทำให้เกิดความยุ่งยากในการตีความหากมีขนาดใหญ่ ดังนั้นการสกัดกฎจากต้นไม้ให้อยู่ในรูปแบบ “If-Then” จะทำให้ง่ายต่อการตีความ [40] โดยงานวิจัยนี้ได้แบ่งกลุ่มของการลงโทษไว้ 7 กลุ่ม ดังรูปที่ 2-47 ดังนั้นสามารถเขียนเป็นกฎได้ 7 กฎ ดังนี้

R1 : IF intention <= 1.5 AND justification <= 0.5 THEN punishment = A

R2 : IF intention <= 1.5 AND justification > 0.5 AND causation <= 1.5 AND impunity <= 1.5 THEN punishment = B

R3 : IF intention <= 1.5 AND justification > 0.5 AND causation <= 1.5 AND Impunity > 1.5 THEN punishment = D

R4 : IF intention <= 1.5 AND justification > 0.5 AND causation > 1.5 AND impunity <= 1.5 THEN punishment = C

R5 : IF intention <= 1.5 AND justification > 0.5 AND causation > 1.5 AND impunity > 1.5 THEN punishment = E

R6 : IF intention > 1.5 AND causation <= 1.5 THEN punishment = F

R7 : IF intention > 1.5 AND causation > 1.5 THEN punishment = G

2.21 นาอีฟเบย์ส (Naïve Bayes) [41]

เป็นแบบจำลองที่ได้รับความนิยมในการจำแนกข้อความ (Text Classification) เพราะใช้งานง่าอาศัยหลักการคือการคำนวณความน่าจะเป็นที่จะเกิดเหตุการณ์บางอย่าง โดยพิจารณาเหตุการณ์ก่อนหน้า ข้อดีคือความรวดเร็วในการสอนการเรียนรู้และการทดสอบจึงเหมาะกับงานในลักษณะ Real Time เพราะแบบจำลองมีขนาดเล็กเหมาะกับข้อมูลไม่มากได้ สำหรับรูปแบบการคำนวณความน่าจะเป็นของนาอีฟเบย์สามารถคำนวณได้ตามสมการที่ 2.80 แต่แบบจำลองนี้มีข้อเสียคือไม่ได้พิจารณาลำดับของคำที่มีผลต่อการพิจารณาคดี ยกเว้นการเลือกใช้ N-Gram ในการสร้างคุณลักษณะ ก่อนนำเข้าประมวลผล และที่สำคัญกรณีข้อความยาว ๆ จะไม่สามารถแบ่งได้ว่ามีได้หลายคลาสในข้อความเดียวกัน

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.81}$$

เมื่อ c คือ คลาสหรือประเภท (Class) เช่นในงานวิจัย มีคลาส บาดเจ็บ (Injury) คลาส ตาย (Death) และไม่เป็นอะไร (Nothing) ในคุณลักษณะ (Attribute) ที่ชื่อว่า ผลการกระทำ (Causation)

x คือ คุณลักษณะ เช่น ผลการกระทำ (Causation) หรือ Intention (เจตนา)

$P(c|x)$ คือ ความน่าจะเป็นที่จะเป็นคลาส c เมื่อเป็นคุณลักษณะ x

$P(x|c)$ คือ ความน่าจะเป็นที่จะเป็นคุณลักษณะ x เมื่อเป็นคลาส c

$P(c)$ คือ ความน่าจะเป็นที่จะเป็นคลาส c จากคลาสทั้งหมด

$P(x)$ คือ ความน่าจะเป็นที่เป็นคุณสมบัติ x จากคุณสมบัติทั้งหมด

2.22 Soft-Voting [42]

เป็นเทคนิคหนึ่งของ Ensemble คือการนำแบบจำลองที่ได้จากการฝึกฝนด้วยชุดข้อมูลเดียวกันหลาย ๆ แต่ใช้อัลกอริทึมในการฝึกฝนแตกต่างกัน เพื่อดึงเอาข้อดีของแต่ละตัวมาทดแทนจุดอ่อนของอีกตัว ทำให้ผลการทำนายรวมมักจะดีกว่าโมเดลที่ฝึกฝนแบบเดียว โดยการโหวตเลือกมีสองประเภทหลัก ได้แก่ Majority Vote คือการเอาผลลัพท์สุดท้ายที่แต่ละแบบจำลองทำนายมาเลือกคำตอบที่มากที่สุด เช่น ถ้าเป็นการจำแนกแบบมัลติคลาส ผลลัพท์ที่ได้ของทั้ง 3 แบบจำลองคือ 0, 0, 1 ก็จะโหวตผลสุดท้ายเป็น 0 เพราะทำนาย 0 มากกว่า 1 และอีกเทคนิคคือการโหวตแบบคือ Soft ที่ใช้ในงานวิจัยนี้ คือการนำค่าความน่าจะเป็นที่แต่ละแบบจำลองคำนวณได้มาหาค่าเฉลี่ย ตามสมการที่ 2.81

$$\gamma = argmax\left(\frac{1}{n}\sum_{Classifier}(p_1, p_2, p_3, \dots p_n)\right) \quad (2.82)$$

เมื่อ γ คือ ผลลัพธ์ที่ได้จากฟังก์ชัน Soft Voting

p_n คือ ค่าความน่าจะเป็นของตัวจำแนกที่ n

n คือ จำนวนตัวจำแนก (Classifier)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

วิทยานิพนธ์นี้ได้อ้างอิงงานวิจัยอื่นที่เกี่ยวข้อง เพื่อนำมาใช้ประโยชน์ในขั้นตอนการวิจัย ซึ่งประกอบด้วย งานวิจัยที่เกี่ยวข้องกับคดีอาญาและการเรียนรู้เชิงลึก ดังนี้

3.1 งานวิจัย "Research on paten text classification based on Word2Vec and LSTM"

งานวิจัยโดย Lizhong Xiao, Guangzhong Wang และ Yang Zuo (2018) **[43]** ได้นำเสนอวิธีการจำแนกประเภทข้อความในสิทธิบัตรด้านความปลอดภัย โดยใช้อัลกอริทึม LSTM ในการเป็นตัวจำแนก (Classifier) และปรับปรุงเพิ่มประสิทธิภาพด้วยการใช้งานร่วมกับ Pre-Trained Weight Embedding ด้วยเทคนิค Word2Vec ทำให้มีค่าความถูกต้องเพิ่มมากขึ้นจาก 85.76% เป็น 93.48% จากจำนวน 10,000 แบบทดสอบ ซึ่งเมื่อเทียบกับแบบจำลองอื่น ๆ เช่น CNN, KNN และ CNN+Word2Vec แล้ว LSTM ร่วมกับ Word2Vec มีความถูกต้องสูงกว่า

งานวิจัยดังกล่าวปรับปรุงประสิทธิภาพด้วยการใช้ Pre-Trained เพียงอย่างเดียวทำให้ไม่เห็นการเปรียบเทียบว่าหากใช้ตัวอื่นหรือเพิ่มมิติของ Embedding จาก 128 เป็นขนาดอื่นจะมีความถูกต้องมากขึ้นหรือไม่ ดังนั้นงานวิจัยนี้จึงได้ใช้ Pre-Trained 2 ตัว เพื่อเปรียบเทียบประสิทธิภาพ คือ FastText [34] ของ Facebook และ Thai2Vec [35] ของกลุ่ม Pythainlp และเลือกขนาดของเวกเตอร์หรือมิติของการฝังคำขนาด 300 ที่ถูกฝึกฝนมาจากคลังข้อมูลภาษาไทยอย่าง Thai Wikipedia และนอกจากนี้งานวิจัยที่ผู้วิจัยพัฒนาขึ้นได้นำการถ่ายโอนการเรียนรู้ (Transfer Learning) [32] มาปรับใช้เพื่อเพิ่มประสิทธิภาพให้แบบจำลอง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 งานวิจัย "A Scheme of Criminal Law Knowledge Acquisition Using Ontology"

งานวิจัยโดย Pongpanut Osathitporn, Nuanwan Soonthornphisaj และ Wiwat Vatanawood (2017) [44] ได้นำเสนอวิธีการสร้างองค์ความรู้กฎหมายอาญาโดยใช้ออนโทโลยีในการวิจัย โดยองค์ความรู้ที่สร้างขึ้นมานั้นจะแสดงองค์ประกอบต่าง ๆ รวมถึงความสัมพันธ์ระหว่างองค์ประกอบในกฎหมายอาญา โดยนำบทบัญญัติกฎหมายอาญา ภาค 1 บทบัญญัติทั่วไป ซึ่งมีทั้งหมด 106 มาตรา และความรู้จากผู้เชี่ยวชาญด้านกฎหมาย มาทำการสกัดเพื่อสร้างตัวแทนองค์ความรู้ขึ้นมา ในรูปแบบของโครงสร้างออนโทโลยี ซึ่งถูกพัฒนาโดยใช้ภาษาอาร์ดีเอฟ/อาวล์ และสร้างกฎเอสดับบลิวอาร์แอล (SWRL) จากประมวลกฎหมายอาญา ในภาค 2 ลักษณะ 10 หมวดความผิดต่อชีวิตและความผิดต่อร่างกาย เพื่อใช้สำหรับการพิจารณาเจตนาผู้กระทำผิด เหตุยกเว้นความผิด เหตุยกเว้นโทษ รวมถึงการพิจารณาฐานความผิดอีกด้วย ซึ่งจะทำการส่งกลับผลลัพธ์จากการ

ประมวลผลทางเว็ปไซต์ที่ผู้วิจัยได้สร้างขึ้นเพื่ออำนวยความสะดวกต่อผู้ใช้งานระบบ จากการทดสอบนั้น ผู้วิจัยได้ใช้คำพิพากษาศาลฎีกาและข้อสอบทางด้านกฎหมายมาเป็นเหตุการณ์จำลองใช้สำหรับการทดสอบระบบ และระบบตอบกลับผู้ใช้งานได้ถูกต้องตามที่คาดหวัง

งานวิจัยดังกล่าวให้ผลลัพธ์เป็นโครงสร้างออนโทโลยีและกฎความสัมพันธ์ แต่ไม่ครอบคลุมกรณีมีผู้ใช้ ตัวการร่วม หรือผู้สนับสนุน และในการใช้งานระบบเป็นในลักษณะเลือกจากกลุ่มคำที่จัดเตรียมไว้ ทำให้มีข้อจำกัดหากมีข้อเท็จจริงหรือคดีที่ไม่ได้อยู่ในระบบ ดังนั้นงานวิจัยนี้จึงออกแบบให้สามารถครอบคลุมถึงการมีผู้ใช้ ตัวการร่วม หรือผู้สนับสนุน รวมทั้งการนำเข้าวิเคราะห์สามารถใช้ภาษาธรรมชาติตามเนื้อข่าวอาชญกรรมในขอบเขตงานวิจัยได้ แต่ก็อาศัยองค์ความรู้ทางกฎหมายอาญาจากงานวิจัยนี้เป็นหลักในการสร้างกฎการพิจารณาข้อเท็จจริงที่เกิดขึ้น

3.3 งานวิจัย “A Framework of Multi-Stage Classifier for Identifying Criminal Law Sentences”

เป็นงานวิจัยโดย Sotarat Thammaboosadee, Bunthit Watanapa และ Nipon Charoenkitkarn (2012) [45] ได้นำเสนอเฟรมเวิร์กในการแยกข้อเท็จจริงจากคดีที่สนใจแล้วนำมาสกัดเป็นข้อเท็จจริงโดยใช้โดยใช้โครงสร้างประสาทเทียม (Artificial Neural Network: ANN) เป็นตัวจำแนกจาก 99 คุณลักษณะที่เป็นข้อเท็จจริง จำแนกออกมาเป็นกลุ่มจำนวน 25 องค์ประกอบการกระทำผิด จากนั้นใช้ โครงสร้างต้นไม้ตัดสินใจ (Decision tree) เข้ามาทำนายว่าจากองค์ประกอบการกระทำผิดเหล่านั้น สามารถระบุความผิด โดยเปรียบเทียบกับตัวบทกฎหมายอาญาอย่างไร โดยระบบนั้นจะถูกสร้างอยู่ในรูปแบบของแบบจำลอง ซึ่งถูกสร้างขึ้นมาในรูปแบบสามขั้นตอนด้วยกันโดยแต่ละขั้นตอนนั้นถูกสร้างจากพื้นฐานของตัวบทกฎหมายอาญา ขั้นตอนแรกนั้นเป็นการจำแนกชนิดของการกระทำโดยเป็นการจำแนกว่าเป็นการกระทำที่มีความผิดตามที่ถูกระบุไว้ในกระบวนกฎหมายหรือไม่ ในขั้นตอนต่อมานั้นเป็นการระบุถึงองค์ประกอบเพิ่มเติมของกฎหมาย ซึ่งเป็นส่วนสำคัญซึ่งอาจส่งผลถึงการระบุอัตรารับโทษ และขั้นตอนสุดท้ายคือขั้นตอนการจำแนก ซึ่งถูกออกแบบให้อยู่ในลักษณะที่เป็นอิสระต่อกัน

จากงานวิจัยดังกล่าวใช้แบบจำลองของตัวจำแนก (Classifier Model) 2 ตัวคือ mANN และต้นไม้ตัดสินใจ ในการจัดหมู่ข้อเท็จจริงจากคดีอาญา 150 คดีตัวอย่าง ในคดีความผิดเกี่ยวกับชีวิตและร่างกาย ตามมาตรา 288 ถึงมาตรา 297 แห่งประมวลกฎหมายอาญา และจำแนกให้เป็นองค์ประกอบความผิด ซึ่งเมื่อพิจารณาค่าความถูกต้องของ mANN ที่อยู่ระหว่าง 65-82% แล้วนำผลลัพธ์มาประมวลผลต่อที่ต้นไม้ตัดสินใจที่ให้ความถูกต้องต่ำลงไปอีก อาจทำให้ผลลัพธ์ไม่น่าเชื่อถือ เพราะเป็นเรื่องสำคัญที่เกี่ยวกับชีวิตและร่างกายมนุษย์ ดังนั้น งานวิจัยนี้จึงแก้ไขโดยให้ส่วนที่เป็นกฎ

ที่สร้างจากต้นไม้ตัดสินใจมีความถูกต้อง 100% ก่อน เมื่อรับข้อมูลนำเข้ามาผ่านการตัดสินใจจะยังคงความถูกต้องตามที่ผ่านการจำแนกมาในเบื้องต้น

อย่างไรก็ตามงานวิจัยดังกล่าวใช้การสร้างข้อมูลตัวแทนด้วยเทคนิค SVM ของการเรียนรู้ของเครื่อง เพื่อสกัดคุณลักษณะ และการทำนายใช้อัลกอริทึม C4.5 ของต้นไม้ตัดสินใจ ในการทำนายผลการตัดสิน แต่งานวิจัยนี้ไม่สามารถนำเข้าข้อมูลเพื่อประมวลผลข้อเท็จจริงใหม่ได้ เพราะข้อมูลถูกเข้ารหัสในลักษณะฐานข้อมูลไว้ จึงเป็นการใช้ข้อมูลที่มีอยู่ทดสอบประสิทธิภาพแบบจำลอง แต่งานวิจัยนี้สามารถนำเข้าข้อมูล ในชั้นข้อมูลขาเข้าของโครงข่ายประสาทเทียมได้ ทั้งการสร้างข้อมูลตัวแทนและการทำนายผลคำตัดสินก็สามารถทำในโครงข่ายแบบจำลองการเรียนรู้เชิงลึกของการจำแนกข้อความโครงข่ายเดียวได้เลย

3.4 งานวิจัย “Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism”

งานวิจัยโดย Kankawin Kowsrihawat, Peerapon Vateekul และ Prachya Boonkwan (2018) [46] ได้นำเสนอแบบจำลองสำหรับทำนายผลการตัดสินคดีอาญา โดยใช้ข้อมูลจากแนวคำพิพากษาศาลฎีกาในโครงข่ายการเรียนรู้เชิงลึก (Deep Learning) ซึ่งเป็นโครงข่ายประสาทเทียมแบบหลายชั้น แบบจำลองนี้สร้างตัวแทนข้อความด้วยโครงข่ายประสาทเทียมแบบประตูวกกลับสองทิศทาง (Bidirectional GRU) ร่วมด้วยกลไกจุดสนใจ (Attention mechanism) ก่อนนำตัวแทนข้อความนั้นไปทำนายผลคำตัดสินในคดีอาญา โดยพบว่าแบบจำลองนี้ให้ประสิทธิภาพที่ดีกว่าเทคนิค Naïve Bayes และ SVM ที่เป็นการสร้างตัวแทนด้วยแบบจำลองดั้งเดิมที่ใช้การสร้างตัวแทนข้อความในลักษณะแบบถุงคำ (Bag of Words) เมื่อพิจารณาจากค่าเฉลี่ยมหภาคของเอฟวัน

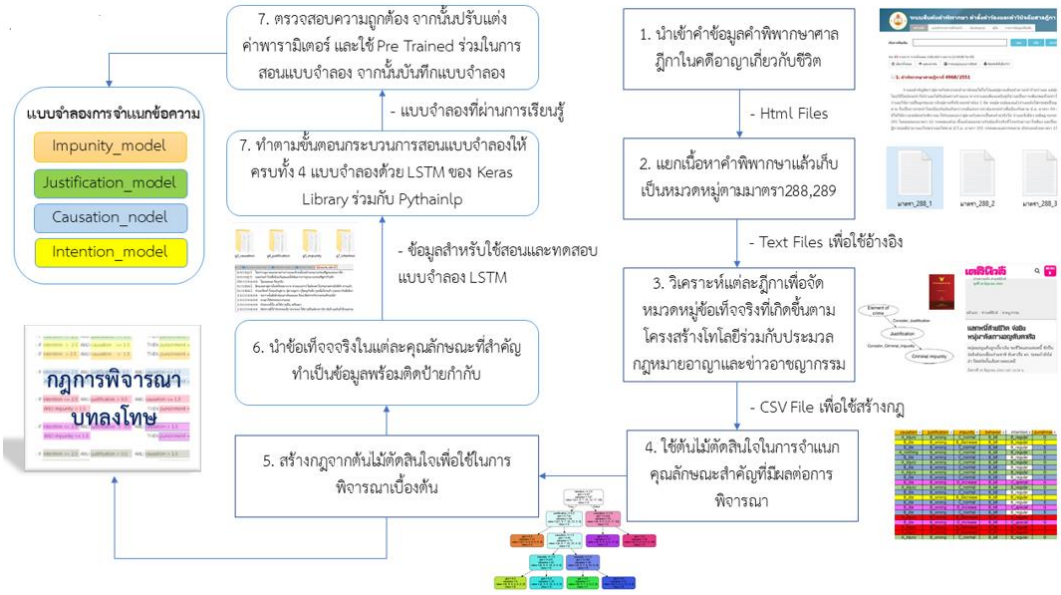
งานวิจัยนี้ใช้เทคนิคการเรียนรู้เชิงลึกคล้ายกับงานวิจัยที่ผู้วิจัยดำเนินการแต่ใช้อัลกอริทึมในการฝึกฝนแตกต่างกันโดยผู้วิจัยใช้หน่วยความจำระยะสั้นแบบยาวทั้งแบบทิศทางเดียว (LSTM) และแบบสองทิศทาง (BiLSTM) นอกจากนี้ ข้อมูลนำเข้าของงานวิจัยนี้ต้องใช้ข้อเท็จจริงและข้อกฎหมายที่เกี่ยวข้องในประเด็นคดีอาญาที่สนใจจำนวน 8 ประเด็น เพื่อทำนายว่าข้อเท็จจริงนั้นเข้าองค์ประกอบในแต่ละประเด็นหรือไม่ และสุดท้ายนำเข้าสู่การทำนายผลการตัดสินที่จะทำนายว่าผู้กระทำได้กระทำความผิดหรือไม่เท่านั้น ไม่ได้ทำนายว่าต้องรับโทษในลักษณะใด ซึ่งต่างจากงานวิจัยของผู้วิจัยที่นำเข้าเพียงข้อเท็จจริงในคดีอาญาที่เกิดขึ้น เพื่อให้แบบจำลองแยกประเมินทั้ง 4 คุณลักษณะ และนำเข้ากฎของต้นไม้ตัดสินใจ เพื่อให้มนุษย์สามารถเข้าใจลักษณะการทำนายของแต่ละคุณลักษณะ จนสามารถทำนายออกมาเป็นแนวทางการลงโทษที่เข้าใจได้ แต่งานวิจัยที่ผู้วิจัยพัฒนาขึ้นครอบคลุมในคดีความผิดเกี่ยวกับชีวิตที่ผู้กระทำมีเจตนากระทำผิดเท่านั้นคือมาตรา 288 และมาตรา 289 แห่ง

ประมวลกฎหมายอาญา ซึ่งต่างจากงานวิจัยที่นี้ที่ครอบคลุมคดีอาญา ทั้งในฐานความผิดเกี่ยวกับชีวิตและร่างกาย ความผิดเกี่ยวกับชื่อเสียง และความผิดเกี่ยวกับทรัพย์ ที่ครอบคลุมฐานความผิดมากกว่า

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
แนวคิดและวิธีการดำเนินการ

วิทยานิพนธ์นี้มีจุดประสงค์เพื่อนำเสนอกระบวนการวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญา เพื่อให้ผู้มีส่วนเกี่ยวข้องทั้งผู้บังคับใช้กฎหมายหรือบุคคลที่สนใจ สามารถเข้าใจแนวทางการตัดสินคดีได้อย่างถูกต้องจากคดีอาญาที่เกิดขึ้นว่ามีแนวทางการกำหนดบทลงโทษในเบื้องต้นไว้อย่างไร โดยใช้ข้อมูลทดสอบจากข่าวอาชญากรรมออนไลน์ ที่พนักงานสอบสวนได้แจ้งข้อกล่าวหาแก่ผู้กระทำผิดในเบื้องต้นหรือได้ขออนุมัติหมายจับเรียบร้อยแล้ว รวมทั้งข้อคิดเห็นจากนักกฎหมาย เพื่อเป็นข้อมูลในการทดสอบความถูกต้องของงานวิจัยนี้ และกฎที่ใช้เป็นตัวทดสอบผ่านการรวบรวมจากคำพิพากษาศาลฎีกา ตามตัวบทของประมวลกฎหมายอาญา แล้วจำแนกประเภทของหมวดหมู่บทลงโทษไว้ด้วยต้นไม้ตัดสินใจ [39] เพื่อให้มนุษย์สามารถเข้าใจแนวทางการวิเคราะห์ดังรูปที่ 4-1 ในการใช้งานผู้ใช้สามารถนำข้อมูลจากเว็บข่าวออนไลน์หรือไฟล์ข้อความ (Text Files) เข้าสู่ระบบ และเมื่อผ่านการวิเคราะห์แล้วจะให้ผลลัพธ์ว่าคดีความดังกล่าวได้กำหนดลงโทษเบื้องต้นไว้อย่างไรตามขอบเขตที่งานวิจัยนี้ได้กำหนดไว้ ดังรูปที่ 4-1



รูปที่ 4-1 ภาพรวมการสร้างแบบจำลองการจำแนกข้อความและกฎที่ใช้พิจารณา

จากรูปที่ 4-1 แสดงภาพรวมกระบวนการสร้างแบบจำลองทั้งการจำแนกข้อความภาษาธรรมชาติด้วยเทคนิคการเรียนรู้เชิงลึก (Deep Learning) [18] ด้วยอัลกอริทึมความจำระยะสั้นแบบยาว (LSTM) [15] และการสร้างแบบจำลองด้วยเทคนิคการเรียนรู้ของเครื่อง (Machine

Learning) โดยใช้ต้นไม้ตัดสินใจ (Decision Tree) [39] จากนั้นในการทดสอบจากข้อมูลที่ไม่เคยเห็นของแบบจำลอง โดยการนำเข้าข้อมูลทดสอบด้วยเนื้อหาข่าวอาชญากรรมออนไลน์ [6] เพื่อทำนายว่าจากเนื้อหาข่าวที่นำเข้าเป็นความรับผิดในกลุ่มใด ตามขอบเขตของงานวิจัย ตามรูปที่ 4-2

Group	ฐานความผิด(New)
A	ไม่มีความผิด
B	พยายามฆ่าผู้อื่นโดยเจตนามีเหตุลดโทษ
C	พยายามฆ่าผู้อื่น
D	ฆ่าผู้อื่นโดยเจตนามีเหตุลดโทษ
E	ร่วมกันฆ่าผู้อื่นโดยเจตนา
F	ร่วมกันพยายามฆ่าผู้อื่นโดยไตร่ตรอง
G	ร่วมกันฆ่าผู้อื่นโดยไตร่ตรอง

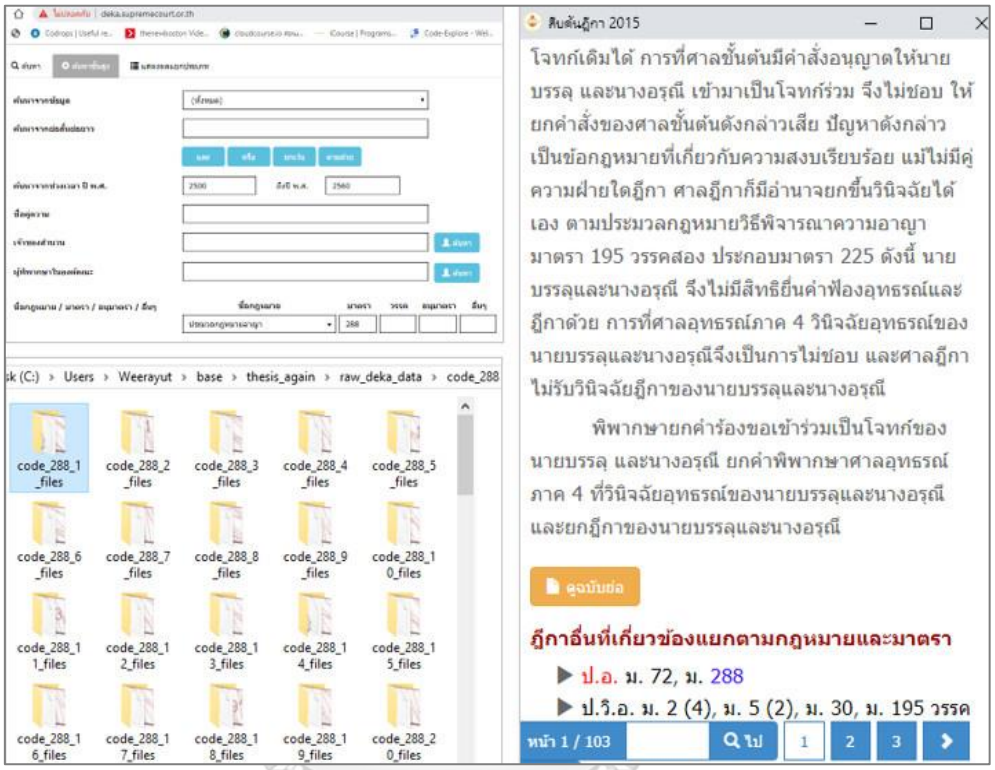
รูปที่ 4-2 การใช้งานแบบจำลอง

จากภาพรวมการสร้างและทดสอบการทำงานของแบบจำลองที่ผ่านมา พบว่างานวิจัยนี้แบ่งเป็นสองส่วนหลักได้ ส่วนที่เป็นกฎในการตัดสินคดี ในส่วนนี้ข้อมูลที่นำมาใช้ในการฝึกฝนคือคำพิพากษาศาลฎีกา ในหมวดความผิดเกี่ยวกับชีวิต ในมาตรา 288 และมาตรา 289 และส่วนที่สองคือส่วนของการจำแนกข้อความภาษาธรรมชาติ โดยข้อมูลที่นำมาใช้การในการฝึกสอนได้แก่ข้อมูลจากคำพิพากษาศาลฎีกา [5] และข้อมูลจากเว็บไซต์ข่าวอาชญากรรมออนไลน์ [6] โดยมีขั้นตอนดังนี้

4.1 การเตรียมข้อมูลสำหรับงานวิจัย

4.1.1 ข้อมูลสำหรับสร้างกฎที่ใช้ในการตัดสิน เป็นส่วนที่ใช้จำแนกกลุ่มการลงโทษตามขอบเขตของงานวิจัย เพื่อให้มนุษย์สามารถเข้าใจได้ในแนวทางการตัดสิน ในส่วนนี้ได้ใช้ข้อมูลโดยผู้วิจัยได้สืบค้นคำพิพากษาศาลฎีกาจากเว็บไซต์เผยแพร่คำพิพากษาศาลฎีกา [5] โดยกรองเลือกหมวดหมู่มาตรา 288 และ 289 โดยประมวลกฎหมายอาญา มาตรา 288 บัญญัติว่า “ผู้ใดฆ่าผู้อื่นต้องระวางโทษประหารชีวิต จำคุก ตลอดชีวิต หรือจำคุกตั้งแต่สิบห้าปีถึงยี่สิบปี” และมาตรา 289 บัญญัติว่า “ผู้ใด (1) ฆ่าบุพการี (2) ฆ่าเจ้าพนักงาน ซึ่งกระทำการตามหน้าที่ หรือเพราะเหตุที่จะกระทำ หรือได้กระทำการตามหน้าที่ (3) ฆ่าผู้ช่วยเหลือเจ้าพนักงาน ในการที่เจ้าพนักงานนั้นกระทำตามหน้าที่ หรือเพราะเหตุที่บุคคลนั้นจะช่วยหรือได้ช่วยเจ้าพนักงานดังกล่าวแล้ว (4) ฆ่าผู้อื่นโดยไตร่ตรองไว้ก่อน (5) ฆ่าผู้อื่นโดยทรมานหรือโดยกระทำทารุณโหดร้าย (6) ฆ่าผู้อื่นเพื่อตระเตรียมการหรือเพื่อความสะดวกในการที่จะกระทำความผิดอย่างอื่น หรือ (7) ฆ่าผู้อื่นเพื่อจะเอา หรือเอาไว้ซึ่งผลประโยชน์อันเกิดแต่การที่ตนได้กระทำความผิดอื่น เพื่อปกปิดความผิดอื่นของตน หรือเพื่อหลีกเลี่ยงให้พ้นอาญาในความผิดอื่นที่ตนได้กระทำไว้ ต้องระวางโทษประหารชีวิต” โดยสืบค้นตั้งแต่

พ.ศ.2500 - พ.ศ.2560 และบันทึกไฟล์ Html ที่กรองผ่านหน้าเว็บไซต์มาเก็บไว้ทั้งหมด 2,698 ฎีกา แบ่งเป็นคดีที่เกี่ยวข้องกับมาตรา 288 จำนวน 2,078 ฎีกา และมาตรา 289 จำนวน 620 ฎีกา เพื่อใช้เป็นชุดข้อมูลการฝึกฝนแบบจำลอง ตามรูปที่ 4-3

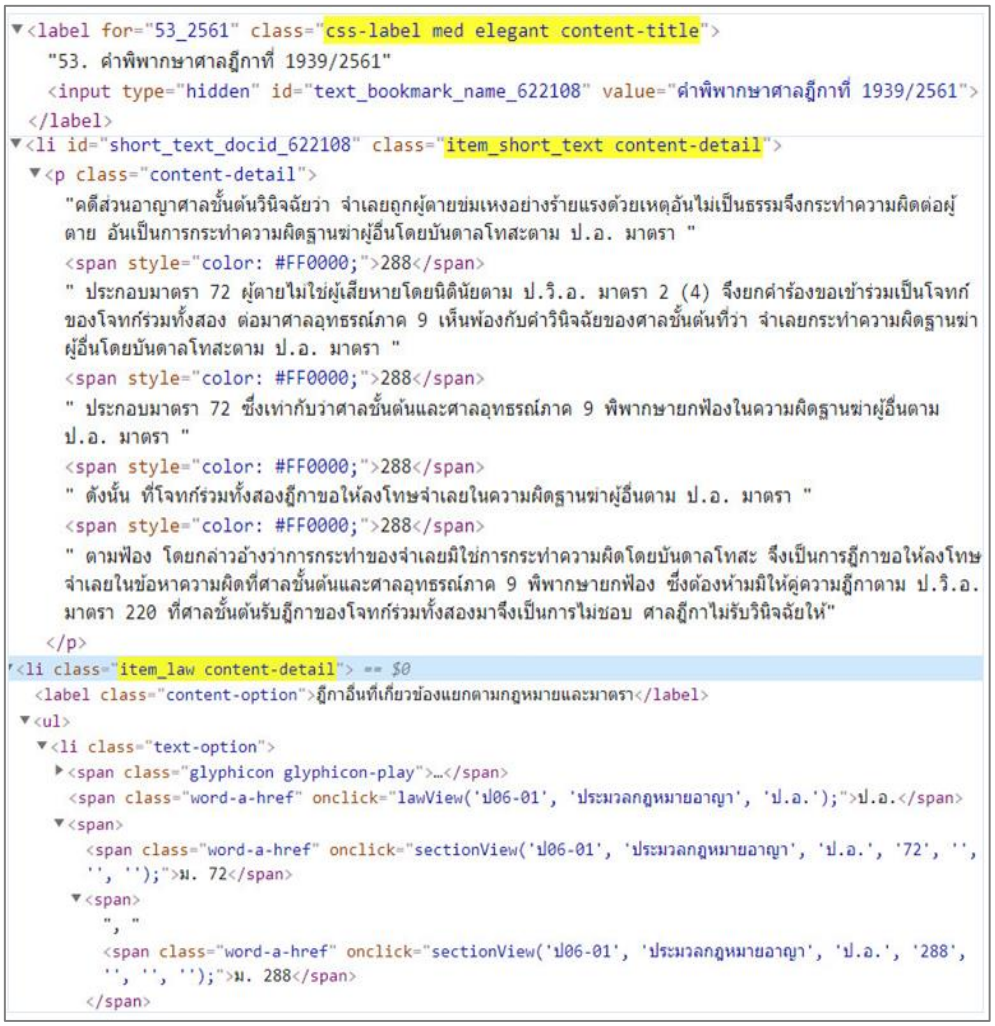


รูปที่ 4-3 แสดงกระบวนการนำข้อมูลคำพิพากษาศาลฎีกาเข้ามาในระบบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ทำการแยกเนื้อหาคำพิพากษาแล้วเก็บเป็นหมวดหมู่ โดยใช้ BeautifulSoup [24] ซึ่งเป็น Library ของภาษาไพธอน เพื่อใช้จัดการหน้าเว็บไซต์ ในการดึงข้อมูลเฉพาะแท็ก (Tag) และคลาส (Class) ที่มีเนื้อหาเกี่ยวข้องที่ใช้ในงานวิจัย เพื่อใช้ในการประมวลผลเท่านั้น โดยทำงานร่วมกับนิพจน์ปกติ (Regular Expression) [25] โดยในคำพิพากษาของศาลฎีกาจะประกอบด้วย เลขที่คำพิพากษา ชื่อคู่ความ (โจทก์และจำเลย) มาตราที่เกี่ยวข้อง หมายเลขคดีแดงศาลชั้นต้น รายชื่อผู้พิพากษาทั้ง 3 ศาล คำพิพากษาศาลชั้นต้น คำพิพากษาศาลอุทธรณ์ คำพิพากษาศาลฎีกา แหล่งที่มาของคำพิพากษา เป็นต้น แต่ในงานวิจัยนี้เก็บข้อมูลจาก 3 ส่วนสำคัญได้แต่ เลขที่คำพิพากษา จากแท็ก label ในคลาส css-label med elegant content-title คำพิพากษาของศาลฎีกา จากแท็ก li ในคลาส item_short_text content-detail และ มาตราที่เกี่ยวข้อง จากแท็ก li ในคลาส item_law content-detail เก็บเป็นลิสต์และบันทึกลงไฟล์เอกสารเรียงตามปีที่ทำการพิพากษา พ.ศ.2561 – 2464 ในคำ

พิพากษาที่เกี่ยวข้องในงานวิจัยนี้คือ มาตรา 288 จำนวน 2078 ฎีกา และมาตรา 289 จำนวน 620 ฎีกา ตามที่แสดงในรูปที่ 4-4



รูปที่ 4-4 การดึงข้อมูลจากเว็บไซต์เผยแพร่คำพิพากษาศาลฎีกา

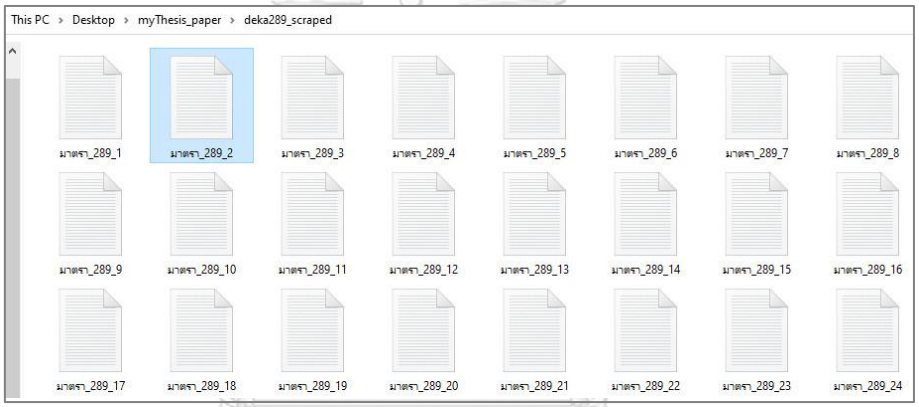
เพื่อให้ง่ายต่อการวิเคราะห์ข้อมูล โดยเลือกแต่ละฎีกาไว้เพียงเลขที่ฎีกา เนื้อหาการตัดสินของศาลฎีกา และมาตราที่ใช้ปรับบทลงโทษ โดยทำการตัดส่วนที่ไม่สำคัญในเนื้อหาออกเช่น รายชื่อคณะตุลาการ รายชื่อโจทก์จำเลย หมายเลขคดี รวมทั้งการตัดสินของศาลชั้นต้นและศาลอุทธรณ์ออก ตามที่แสดงในรูปที่ 4-5 และเขียนเป็นไฟล์ข้อความ บันทึกไว้เพื่อใช้เป็นหลักฐานในการอ้างอิง ตามที่แสดงตัวอย่างในดังรูปที่ 4-6

2. คำพิพากษาศาลฎีกาที่ 1646/2559

ก่อนเกิดเหตุผู้ตายกับจำเลยต่างเมาสุราโต้เถียงทะเลาะกัน จำเลยกลับบ้านเอามีดดาบของกลางมาขู่ผู้ตาย ผู้ตายกับจำเลยโต้เถียงกันอีก แล้วจำเลยถือมีดดาบวิ่งไล่ผู้ตายไปจนถึงบริเวณที่ผู้ตายกับพวกนั่งดื่มสุรากัน อ.เข้าไปห้ามจำเลย จำเลยจึงกลับบ้านโดยไม่ได้ฟันผู้ตาย ผู้ตายกลับมานั่งดื่มสุรากับพวกต่อ สักครู่หนึ่งผู้ตายเดินออกไปทางท้ายซอยห่างจากวงสุราประมาณ 10 เมตร และยืนชี้มาทางบ้านจำเลยลักษณะท้าทาย จำเลยโมโหจึงได้ขับรถจักรยานยนต์เข้าไปหาผู้ตายแล้วใช้มีดดาบฟันผู้ตาย ดังนี้ พฤติการณ์ของจำเลยหลังจากที่จำเลยใช้มีดดาบของกลางวิ่งไล่ผู้ตายไปจนถึงที่เกิดเหตุและใช้มีดดาบของกลางเล่มเดิมฟันผู้ตาย เป็นเหตุการณ์ที่ต่อเนื่องกันไม่ขาดตอน ถือไม่ได้ว่าจำเลยกระทำการโดยไตร่ตรองไว้ก่อน

ฎีกาอื่นที่เกี่ยวข้องแยกตามกฎหมายและมาตรา ป.อ. ม. 288, ม. 289, ม. 80

รูปที่ 4- 5 การเลือกส่วนของเนื้อหาที่สำคัญจาก Html files



รูปที่ 4-6 ไฟล์ข้อมูลที่ผ่านการคัดเลือกส่วนที่ใช้ในงานวิจัยแล้ว

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากนั้นรวบรวมข้อมูลที่ได้จากแนวคำพิพากษาจากจำนวน 210 คดีตัวอย่าง เพื่อกำหนดโครงสร้างความรับผิดอาญา [2] ในหมวดความผิดเกี่ยวกับชีวิต ที่อาศัยการตีความกฎหมายอย่างเคร่งครัด [3] และพิจารณาเหตุยกเว้นความผิดหรือเหตุลดโทษร่วมด้วย [4] โดยงานวิจัยนี้พิจารณาการกำหนดโครงสร้างความรับผิด ตามการองค์ความรู้กฎหมายอาญา [44] และปรับเข้ากับขอบเขตงานวิจัย โดยกำหนดขอบเขตความรับผิดเป็น 7 กลุ่มการรับโทษตามระดับความร้ายแรงของความประเภทผิด โดยมีรายละเอียดของแต่ละกลุ่มฐานความผิดและกฎหมายที่เกี่ยวข้องตามที่แสดงในตารางที่ 4-1

ตารางที่ 4-1 รายละเอียดของกลุ่มความรับผิดตามหลักกฎหมาย

กลุ่ม	การรับผิด	หลักกฎหมายหลัก	หลักกฎหมายย่อย
A	ไม่มีความผิดและไม่ต้องรับโทษ	288,289	68, หลักความยินยอม
B	พยายามฆ่าผู้อื่นเจตนามีเหตุลดโทษ	288	60,65,66,67,69,70,72,73,80,81,82,86
C	พยายามฆ่าผู้อื่น	288	60,80,83
D	ฆ่าผู้อื่นโดยเจตนามีเหตุลดโทษ	288	60,65,66,67,69,70,72,73,81,82,86
E	ฆ่าผู้อื่นโดยเจตนา	288	60,83,84
F	พยายามฆ่าผู้อื่นโดยไตร่ตรอง	289	80,83,84
G	ฆ่าผู้อื่นโดยไตร่ตรอง	289	83,84

ตามประมวลกฎหมายอาญา [1] มีมาตราที่เกี่ยวข้องกับการกำหนดความรับผิดในงานวิจัยนี้ ได้แก่

1. มาตรา 288 “ผู้ใดฆ่าผู้อื่นต้องระวางโทษประหารชีวิต จำคุกตลอดชีวิต หรือจำคุกตั้งแต่สิบห้าปีถึงยี่สิบปี”

2. มาตรา 289 ผู้ใด

(1) ฆ่าบุพการี

(2) ฆ่าเจ้าพนักงาน ซึ่งกระทำการตามหน้าที่ หรือเพราะเหตุที่จะกระทำ หรือได้กระทำการตามหน้าที่

(3) ฆ่าผู้ช่วยเหลือเจ้าพนักงาน ในการที่เจ้าพนักงานนั้นกระทำตามหน้าที่ หรือเพราะเหตุที่บุคคลนั้นจะช่วยหรือได้ช่วยเจ้าพนักงานดังกล่าวแล้ว

(4) ฆ่าผู้อื่นโดยไตร่ตรองไว้ก่อน

(5) ฆ่าผู้อื่นโดยทรมานหรือโดยกระทำทารุณโหดร้าย

(6) ฆ่าผู้อื่นเพื่อตระเตรียมการ หรือเพื่อความสะดวกในการที่จะกระทำความผิดอย่างอื่นหรือ

(7) ฆ่าผู้อื่นเพื่อจะเอาหรือเอาไว้ซึ่งผลประโยชน์อันเกิด แต่การที่ตนได้กระทำความผิดอื่นเพื่อปกปิดความผิดอื่นของตน หรือเพื่อหลีกเลี่ยงให้พ้นอาญาในความผิดอื่นที่ตนได้กระทำไว้ต้องระวางโทษประหารชีวิต

3. มาตรา 60 “ผู้ใดเจตนาที่จะกระทำต่อบุคคลหนึ่ง แต่ผลของการกระทำเกิดแก่อีกบุคคลหนึ่งโดยพลาดไป ให้ถือว่าผู้นั้นกระทำโดยเจตนาแก่บุคคลซึ่งได้รับผลร้ายจากการกระทำนั้น แต่ใน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กรณีที่กฎหมายบัญญัติให้ลงโทษหนักขึ้นเพราะฐานะของบุคคลหรือเพราะความสัมพันธ์ระหว่างผู้กระทำกับบุคคลที่ได้รับผลร้าย มิให้นำกฎหมายนั้นมาใช้บังคับเพื่อลงโทษผู้กระทำให้หนักขึ้น”

4. มาตรา 65 “ผู้ใดกระทำความผิดในขณะไม่สามารถรู้ผิดชอบหรือไม่สามารถบังคับตนเองได้เพราะมีจิตบกพร่อง โรคจิตหรือจิตฟั่นเฟือน ผู้นั้นไม่ต้องรับโทษสำหรับความผิดนั้น แต่ถ้าผู้กระทำความผิดยังสามารถรู้ผิดชอบอยู่บ้าง หรือยังสามารถบังคับตนเองได้บ้าง ผู้นั้นต้องรับโทษสำหรับความผิดนั้น แต่ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้”

5. มาตรา 67 “ผู้ใดกระทำความผิดด้วยความจำเป็น

(1) เพราะอยู่ในที่บังคับ หรือภายใต้อำนาจซึ่งไม่สามารถหลีกเลี่ยงหรือขัดขืนได้ หรือ

(2) เพราะเพื่อให้ตนเองหรือผู้อื่นพ้นจากภยันตรายที่ใกล้จะถึงและไม่สามารถหลีกเลี่ยงให้พ้น

โดยวิธีอื่นใดได้ เมื่อภยันตรายนั้นตนมิได้ก่อให้เกิดขึ้นเพราะความผิดของตน ถ้าการกระทำนั้นไม่เป็นการเกินสมควรแก่เหตุแล้ว ผู้นั้นไม่ต้องรับโทษ”

6. มาตรา 68 “ผู้ใดจำต้องกระทำการใดเพื่อป้องกันสิทธิของตนหรือของผู้อื่น ให้พ้นภยันตรายซึ่งเกิดจากการประทุษร้ายอันละเมิดต่อกฎหมายและเป็นภยันตรายที่ใกล้จะถึง ถ้าได้กระทำพอสมควรแก่เหตุ การกระทำนั้นเป็นการป้องกันโดยชอบด้วยกฎหมาย ผู้นั้นไม่มีความผิด”

7. มาตรา 69 “ในกรณีที่บัญญัติไว้ในมาตรา 67 และมาตรา 68 นั้น ถ้าผู้กระทำได้กระทำไปเกินสมควรแก่เหตุ หรือเกินกว่ากรณีแห่งความจำเป็น หรือเกินกว่ากรณีแห่งการจำต้องกระทำเพื่อป้องกัน ศาลจะลงโทษน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้ แต่ถ้าการกระทำนั้นเกิดขึ้นจากความตื่นเต้น ความตกใจหรือความกลัว ศาลจะไม่ลงโทษผู้กระทำก็ได้”

8. มาตรา 70 “ผู้ใดกระทำตามคำสั่งของเจ้าพนักงาน แม้คำสั่งนั้นจะมิชอบด้วยกฎหมาย ถ้าผู้กระทำมีหน้าที่หรือเชื่อโดยสุจริตว่ามีหน้าที่ต้องปฏิบัติตาม ผู้นั้นไม่ต้องรับโทษ เว้นแต่จะรู้ว่าคำสั่งนั้นเป็นคำสั่ง ซึ่งมิชอบด้วยกฎหมาย”

9. มาตรา 72 “ผู้ใดบันดาลโทสะโดยถูกข่มเหงอย่างร้ายแรงด้วยเหตุอันไม่เป็นธรรม จึงกระทำความผิดต่อผู้ข่มเหงในขณะนั้น ศาลจะลงโทษผู้นั้นน้อยกว่าที่กฎหมายกำหนดไว้สำหรับความผิดนั้นเพียงใดก็ได้”

10. มาตรา 73 “เด็กอายุยังไม่เกินเจ็ดปี กระทำการอันกฎหมายบัญญัติเป็นความผิด เด็กนั้นไม่ต้องรับโทษ”

11. มาตรา 80 “ผู้ใดลงมือกระทำความผิดแต่กระทำไปไม่ตลอด หรือกระทำไปตลอดแล้วแต่การกระทำนั้นไม่บรรลุผล ผู้นั้นพยายามกระทำความผิด ผู้ใดพยายามกระทำความผิด ผู้นั้นต้องระวางโทษสองในสามส่วนของโทษที่กฎหมายกำหนดไว้สำหรับความผิดนั้น”

12. มาตรา 81 “ผู้ใดกระทำการโดยมุ่งต่อผลซึ่งกฎหมายบัญญัติเป็นความผิด แต่การกระทำนั้นไม่สามารถจะบรรลุผลได้อย่างแน่แท้ เพราะเหตุปัจจัยซึ่งใช้ในการกระทำหรือเหตุแห่งวัตถุที่มุ่งหมายกระทำต่อให้ถือว่าผู้นั้นพยายามกระทำความผิด แต่ให้ลงโทษไม่เกินกึ่งหนึ่งของโทษที่กฎหมายกำหนดไว้สำหรับความผิดนั้น ถ้าการกระทำดังกล่าวในวรรคแรกได้กระทำไปโดยความเชื่ออย่างงมงาย ศาลจะไม่ลงโทษก็ได้”

13. มาตรา 82 “ผู้ใดพยายามกระทำความผิด หากยับยั้งเสียเองไม่กระทำการให้ตลอดหรือกลับใจแก้ไขไม่ให้การกระทำนั้นบรรลุผล ผู้นั้นไม่ต้องรับโทษสำหรับการพยายามกระทำความผิดนั้น แต่ถ้าการที่ได้กระทำไปแล้วต้องตามบทกฎหมายที่บัญญัติเป็นความผิด ผู้นั้นต้องรับโทษสำหรับความผิดนั้นๆ”

14. มาตรา 84 “ผู้ใดก่อให้ผู้อื่นกระทำความผิดไม่ว่าด้วยการใช้บังคับ ขู่เข็ญ จ้าง วานหรือยุยงส่งเสริม หรือด้วยวิธีอื่นใด ผู้นั้นเป็นผู้ใช้ให้กระทำความผิด ถ้าผู้ถูกใช้ได้กระทำความผิดนั้น ผู้ใช้ต้องรับโทษเสมือนเป็นตัวการ ถ้าความผิดมิได้กระทำลง ไม่ว่าจะเป็นเพราะผู้ถูกใช้ไม่ยอมกระทำ ยังไม่ได้กระทำหรือเหตุอื่นใด ผู้ใช้ต้องระวางโทษเพียงหนึ่งในสามของโทษที่กำหนดไว้สำหรับความผิดนั้น”

โครงสร้างความรับผิดอาญาที่ต้องพิจารณา 7 องค์ประกอบ ตามที่แสดงในตารางที่ 4-2 ได้แก่

1) ผู้กระทำผิด (offender) ประกอบด้วย เป็นผู้ที่อยู่ในสภาพไม่ปกติ (innocent) เป็นผู้สนับสนุน (support) เป็นผู้กระทำผิดด้วยตนเอง (self) เป็นตัวการร่วม (joint) และเป็นผู้ใช้ให้ผู้อื่นกระทำผิด (hire)

2) ผู้ถูกกระทำ (victim) ประกอบด้วย บุคคลทั่วไป (people) บุพการี (parent) และเจ้าหน้าที่ที่ปฏิบัติงานตามหน้าที่ (officer)

3) ผลการกระทำ (Causation) ประกอบด้วย ไม่เป็นอะไร (Nothing) บาดเจ็บ (Injury) และตาย (Death)

4) พิจารณาความผิด (Justification) ประกอบด้วย ผิด (Wrong) และ ไม่เป็นความผิด (Notwrong)

5) พิจารณาโทษ (Impunity) ประกอบด้วย ยกเว้นความผิด (Except) ลดโทษ (Decrease) ปกติ (Normal) เพิ่มโทษ (Increase)

6) เจตนา (Intention) ประกอบด้วย ไม่มีเจตนา (None) เจตนาปกติ (Regular) และเจตนาพิเศษ (Special)

7) พฤติกรรม (Behavior) ประกอบด้วย ทำร้าย (Attack) และ ฆ่า (Kill)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-2 ข้อเท็จจริงตามโครงสร้างความรับผิดอาญา

ฎีกา	ผู้กระทำ	ผู้ถูกกระทำ	ผลการกระทำ	พิจารณาความผิด	พิจารณาโทษ	พฤติกรรม	เจตนา	ความรับผิด
615/03	Self	People	Death	Notwrong	Except	Kill	None	A
12384/56	Support	People	Injury	Wrong	Decrease	Kill	Regular	B
1939/61	Self	People	Death	Wrong	Decrease	Kill	Regular	C
2107/61	Joint	People	Injury	Wrong	Normal	Kill	Regular	D
753/61	Joint	People	Death	Wrong	Normal	Kill	Regular	E
513/53	Self	Officer	Nothing	Wrong	Increase	Kill	Special	F
12613/57	Joint	People	Death	Wrong	Increase	Kill	Special	G

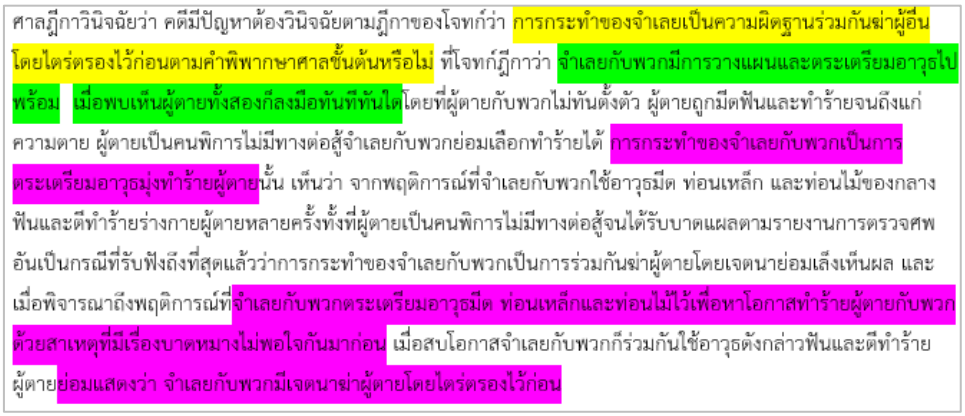
จากตารางที่ 4-2 พบว่าองค์ประกอบความรับผิด เพื่อจำแนกออกเป็นกลุ่มการลงโทษนั้น ประกอบด้วย 7 องค์ประกอบหลัก ได้แก่ ผู้กระทำผิด ผู้ถูกกระทำ ผลการกระทำ การพิจารณาความผิด การพิจารณาโทษ พฤติกรรม และเจตนา ส่วนเลขที่ฎีกาไม่ได้ถูกนำมาประมวลผล เพราะใช้เพื่อการอ้างอิงเท่านั้น

ต้นไม้ตัดสินใจ [39] มีฟังก์ชันการในคัดเลือกการคัดเลือกคุณสมบัติสำคัญ (Feature importance) ตามโครงสร้างความรับผิดอาญา เพื่อดูความสำคัญของแต่ละคุณลักษณะที่ส่งผลต่อการจำแนกกลุ่มความรับผิด ดังนั้นงานวิจัยนี้จึงเลือกใช้ฟังก์ชันในการคัดเลือกคุณสมบัติสำคัญ เพื่อลดจำนวนคุณลักษณะที่ต้องถูกใช้เป็นแบบจำลองในกาวิเคราะห์ภาษาธรรมชาติต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.1.2 ข้อมูลสำหรับการสร้างแบบจำลองจำแนกข้อความภาษาธรรมชาติ

การสร้างแบบจำลอง เพื่อทำการวิเคราะห์ข้อความภาษาธรรมชาตินั้น ผู้วิจัยใช้ข้อมูลจากแนวคำพิพากษาของศาลฎีกาในแต่ละคดีว่ามีข้อเท็จจริงที่เป็นองค์ประกอบตามโครงสร้างการรับผิดอาญาที่ออกแบบไว้ตามออนโทโลยีกฎหมายอาญาร่วมกับการใช้ประมวลกฎหมายอาญาประกอบการตัดสินใจอย่างไร ซึ่งในคำพิพากษาท่านได้กล่าวไว้โดยละเอียดว่าการกระทำเช่นนี้ฟังได้ว่ามีเจตนาหรือประสงค์อย่างไร ทำให้ผู้วิจัยได้ข้อเท็จจริง อันเป็นข้อมูลสำคัญในการว่าเข้าคุณสมบัติข้อใด จากทั้งหมดกว่า 60 คลาสที่ออกแบบไว้ในออนโทโลยีองค์ความรู้ความรับผิดทางอาญา[39] เช่น ในคำพิพากษาศาลฎีกาที่ 13360/2556 ระบุว่า “การที่จำเลยกับพวกตระเตรียมอาวุธ และเข้าทำร้ายผู้ตายทันทีเมื่อพบเพราะเคยมีเรื่องบาดหมางกันมาก่อน ถือว่าจะเลยฆ่าผู้อื่นโดยไตร่ตรองไว้ก่อน” แสดงดังรูปที่ 4-7


ศาลฎีกาวินิจฉัยว่า คดีมีปัญหาต้องวินิจฉัยตามฎีกาของโจทก์ว่า การกระทำของจำเลยเป็นความผิดฐานร่วมกันฆ่าผู้อื่นโดยไตร่ตรองไว้ก่อนตามคำพิพากษาศาลชั้นต้นหรือไม่ ที่โจทก์ฎีกาว่า จำเลยกับพวกมีการวางแผนและตระเตรียมอาวุธไปพร้อม เมื่อพบเห็นผู้ตายทั้งสองก็ลงมือทันทีทันใดโดยที่ผู้ตายกับพวกไม่ทันตั้งตัว ผู้ตายถูกมีดฟันและทำร้ายจนถึงแก่ความตาย ผู้ตายเป็นคนพิการไม่มีทางต่อสู้จำเลยกับพวกย่อมเลือกทำร้ายได้ การกระทำของจำเลยกับพวกเป็นการตระเตรียมอาวุธมุ่งทำร้ายผู้ตายนั้น เห็นว่า จากพฤติการณ์ที่จำเลยกับพวกใช้อาวุธมีด ท่อนเหล็ก และท่อนไม้ของกลางฟันและตีทำร้ายร่างกายผู้ตายหลายครั้งทั้งที่ผู้ตายเป็นคนพิการไม่มีทางต่อสู้จนได้รับบาดแผลตามรายงานการตรวจศพอันเป็นกรณีที่รับฟังถึงที่สุดแล้วว่าการกระทำของจำเลยกับพวกเป็นการร่วมกันฆ่าผู้ตายโดยเจตนาย่อมเล็งเห็นผล และเมื่อพิจารณาถึงพฤติการณ์ที่จำเลยกับพวกตระเตรียมอาวุธมีด ท่อนเหล็กและท่อนไม้ไว้เพื่อหาโอกาสทำร้ายผู้ตายกับพวกด้วยสาเหตุที่มีเรื่องบาดหมางไม่พอใจกันมาก่อน เมื่อสบโอกาสจำเลยกับพวกก็ร่วมกันใช้อาวุธดังกล่าวฟันและตีทำร้ายผู้ตายย่อมแสดงว่า จำเลยกับพวกมีเจตนาฆ่าผู้ตายโดยไตร่ตรองไว้ก่อน

รูปที่ 4-7 แสดงการสกัดข้อเท็จจริงจากฎีกา เพื่อจัดให้อยู่ในโครงสร้างความรับผิดอาญา

จากรูปที่ 4-7 ข้อเท็จจริงจากการกระทำของผู้กระทำผิดคือการที่เขาเคยมีเรื่องบาดหมางกับผู้ตาย และเมื่อพบเจอก็เข้าทำร้ายทันทีพร้อมอาวุธนั้น แสดงว่าเขาได้เตรียมการไว้ก่อนหน้า จึงได้ข้อเท็จจริงนี้ เป็นส่วนหนึ่งขององค์ประกอบของการกระทำโดยมีการวางแผนไว้ล่วงหน้า ซึ่งเป็นเจตนาพิเศษ (Special) ในคุณสมบัติของเจตนา (Intention) และส่งผลให้ผู้ถูกกระทำถึงแก่ความตาย (Death) เพราะมีข้อความว่า ผู้ตายเป็นคนพิการไม่มีทางสู้จนได้รับบาดแผลตามรายงานการตรวจศพในคุณสมบัติผลจากการกระทำ (Causation) และไม่มีคุณสมบัติของการลดโทษหรือเพิ่มโทษ (Impunity) ก็ได้ข้อเท็จจริงว่ารับโทษแบบปกติ (Normal) และพบว่าองค์ประกอบการพิจารณาความผิด (Justification) ระบุว่ามีความผิด (Wrong) จากการที่จำเลยกับพวกร่วมกันใช้อาวุธฟันและตีทำร้ายผู้ตาย เมื่อข้อเท็จจริงที่ได้รับครบองค์ประกอบความผิด ทำให้ผู้กระทำต้องรับผิดในกลุ่ม G คือร่วมกันฆ่าผู้อื่นโดยไตร่ตรองไว้ก่อน

เมื่อทำการสกัดข้อความจากคำพิพากษาศาลฎีกาในคดีที่รวบรวมมา ให้อยู่ในองค์ประกอบความรับผิดหรือคุณลักษณะทั้ง 4 อย่าง และพิจารณาในแต่ละเหตุการณ์หรือประโยคว่าจัดอยู่ในกลุ่มหรือคลาสใด โดยขั้นตอนนี้เป็นขั้นตอนสำคัญในการสร้างข้อมูลสำหรับการฝึกสอนแบบจำลอง (Data set) ในงานวิจัยที่อาศัยแนวการตัดสินของศาลฎีกาที่ผ่านมากำหนด เพื่อข้อเท็จจริงในแต่ละองค์ประกอบความผิด

4.2 การทำความสะอาดข้อมูล (Data Cleansing)

การทำความสะอาดข้อมูลในขั้นตอนนี้รวมถึงการตัดคำไม่สำคัญ (Stop words) หรือคำฟุ่มเฟือยต่าง ๆ ได้แก่ ชื่อที่เป็นคำเฉพาะเช่น ชื่อจังหวัด เขต อำเภอ ชื่อที่ใช้ในการพิจาณาของศาลเช่น โจทก์ จำเลย หรือยศ ตำแหน่งเจ้าหน้าที่ หมายเลขที่คำสั่งต่างๆ ชื่อบุคคล ชื่อหน่วยงานหรือองค์กร วันที่เกิดเหตุ รวมทั้งคำที่เจอบ่อยในทุกๆ เอกสาร โดยผู้วิจัยได้ได้เรียกใช้ Thai_stopwords API [47]

ร่วมกับคำศัพท์ที่ผู้วิจัยเพิ่มเติมเข้าไปจำนวน 2,506 คำ ในการตัดชื่อเฉพาะรวมทั้งเพิ่มคำศัพท์ที่เป็นศัพท์เฉพาะในการเขียนข่าว เช่น ‘ซัลโว’ ที่หมายถึงการกระหน่ำยิง โดยสร้างไว้ในคลังคำศัพท์ของคำฟุ่มเฟือย ตามตัวอย่างในรูปที่ 4-8

ฯ	ๆ	ฯล	ๆ	ค่ะ	ครั้ง	จ้ะ	ของ	จ๊ะ
จ๋า	จึง	น่ะ	ครับ	คือ	หัวร้อน	เมื่อวันที่	รับแจ้ง	เปิดเผย
ว่า	สน.	ผกก.	ผู้กำกับการ	อ.	อาจารย์	ดร.	ด๊อกเตอร์	ผศ.
ด.ช.	เด็กชาย	ด.ญ.	เด็กหญิง	อ.ตร.	อธิบดีกรมตำรวจ	นศท.	นักศึกษาวิชาทหาร	
โจ๋	ซัลโว	ซิว	เป๋า	บ.ก.	บรรณาธิการ	ท.บ.บก	น.น.ร.	นักเรียนนายร้อย
พลฯ	พลทหาร	ส.ต.	สิบตรี	ผู้อำนวยการ	สวญ.	สารวัตรใหญ่	ปธน.	ประธานาธิบดี
ผช.	ผู้ช่วย	ผญบ.	ผู้ใหญ่บ้าน	อก.	อัยการ	พ.ต.	อายุ	ปี
พันตรี	พ.ต.ต.	พันตำรวจตรี	กทม.	กรุงเทพมหานคร	ธ.	ธนาคารธอส.	ธนาคารอาคารสงเคราะห์	
คสช.	คณะรักษาความสงบแห่งชาติ		กกต.	คณะกรรมการการเลือกตั้ง		กกท.	การกีฬาแห่งประเทศไทย	
กคช.	การเคหะแห่งชาติ		ก.จ.	คณะกรรมการข้าราชการส่วนจังหวัด		ก.ต.	คณะกรรมการตุลาการ	
กทท.	การท่าเรือแห่งประเทศไทย		มข.	มหาวิทยาลัยขอนแก่น		กรกฎาคม	เดือน	ก.ย.
ก้นยายน	วันพฤหัสบดี	วัน	พุทธศักราช	สภ.	สถานีตำรวจภูธร	ธันยนันท์	กิรณา	ศุภรักษ์
รัตนา	พัชรมัย	เคนยา	แคนาดา	แคเมอรูน	โครเอเชีย	โคลอมเบีย	จอร์เจีย	จอร์แดน
เซนติเมตร	กม.	กิโลเมตร	มม.	มิลลิเมตร	โตโยต้า	อีซูซุ	ดีแมคซ์	เบนซ์

รูปที่ 4-8 ตัวอย่างคำฟุ่มเฟือยที่ถูกตัดออกในการทำความสะอาดข้อมูล

4.3 การประมวลผลข้อมูลเบื้องต้น

4.3.1 การตัดคำ (Word Segmentation)

ในการฝึกฝนแบบจำลองในโครงข่ายการเรียนรู้เชิงลึกข้อมูลที่ใช้ในงานวิจัย (Dataset) เป็นข้อความภาษาไทยที่เป็นรูปแบบการเขียนติดกัน ซึ่งต่างจากภาษาอังกฤษที่มีช่องว่างแบ่งคำชัดเจน ในงานวิจัยนี้ข้อมูลนำเข้า เพื่อใช้ในการประมวลผลโครงข่ายประสาทเทียมเชิงลึกเป็นระดับคำ (Word Level) จึงต้องมีการตัดคำ (Word Segmentation) ด้วยเทคนิคการเรียนรู้เชิงลึก โดยใช้เครื่องมือตัดคำ Deepcut [28] ของ PyThaiNLP ที่ใช้คลังข้อมูลคำศัพท์ Best 2009 ของศูนย์ศูนย์วิจัยเทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ร่วมกับเทคนิคการตัดคำแบบยาวที่สุด (Longest Matching) ของคลังคำศัพท์ด้านกฎหมายอาญา (Criminal Dictionary) ที่ผู้วิจัยสร้างขึ้น ดังที่แสดงในตาราง 4-3 โดยทำการตัดคำในคลังข้อมูล (Dataset) ที่ใช้ในงานวิจัยทั้ง 4 แบบจำลอง

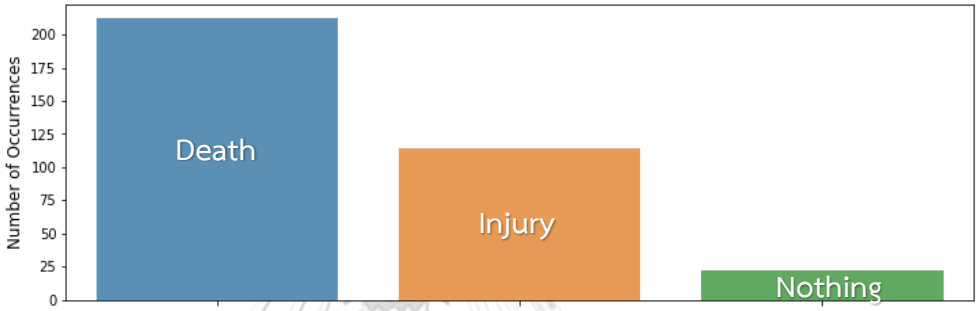
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-3 คลังข้อมูลคำศัพท์ที่ใช้ในงานวิจัย

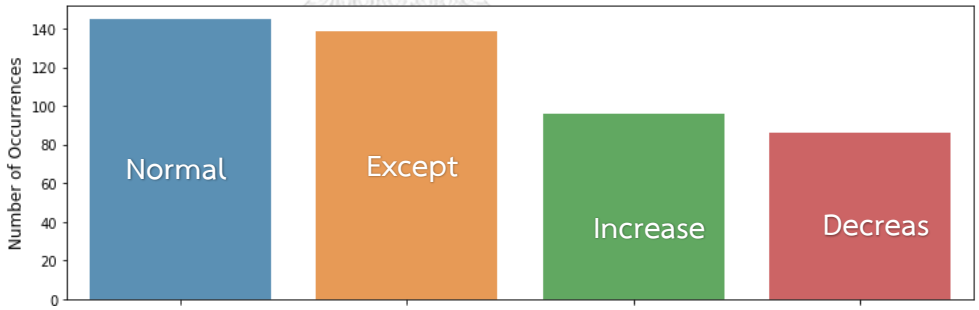
คลังข้อมูลการตัดคำ	จำนวนคำศัพท์ (คำ)
Best 2009	5,000,000
Criminal Dictionary	2,156
รวม	5,000,2156

4.3.2 การแก้ปัญหาความไม่สมดุลของคลาส

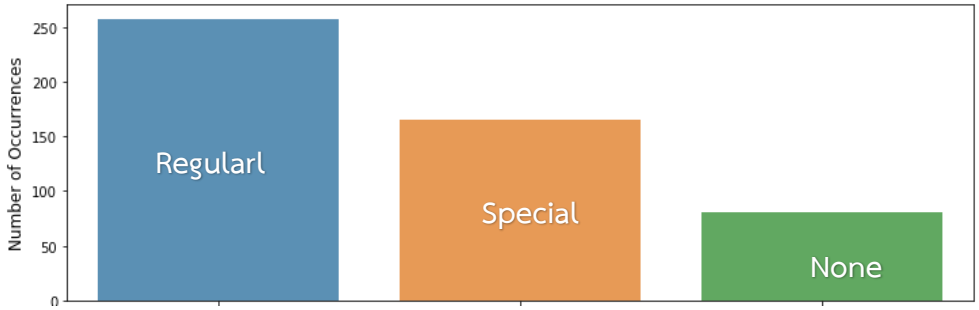
จากชุดข้อมูลที่ใช้ในงานวิจัย (Dataset) พบว่าทั้ง 4 คุณลักษณะที่เป็นองค์ประกอบความรับผิดมีจำนวนตัวอย่างข้อมูลที่แตกต่างกันค่อนข้างมาก ซึ่งจะส่งผลต่อประสิทธิภาพของแบบจำลอง (Models) เพราะเมื่อข้อมูลในคลาสใดมีจำนวนมากกว่าโอกาสที่แบบจำลองจะทำนายเป็นไปในคลาสนั้นย่อมมีมากกว่าคลาสที่มีตัวอย่างน้อย เพราะมีการเรียนรู้แต่ข้อมูลกลุ่มมาก (Majority Class) ผลที่ได้ก็จะจำแนกไปในข้อมูลกลุ่มมาก ส่วนข้อมูลกลุ่มน้อย (Minority Class) ไม่ได้รับการเรียนรู้ทั้งที่เป็นข้อมูลที่มีความสำคัญ ตามที่แสดงในรูปที่ 4-9 ถึงรูปที่ 4-12 ที่แสดงถึงจำนวนข้อมูลในแต่ละคลาสของทั้ง 4 คุณลักษณะ



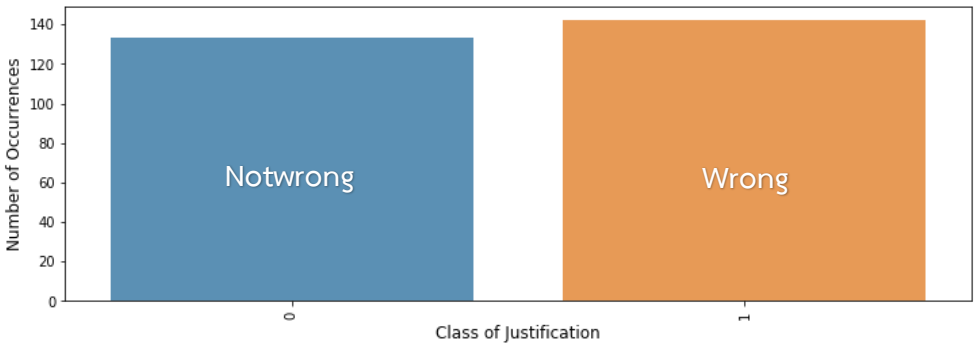
รูปที่ 4-9 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะผลของการกระทำ (Causation)



รูปที่ 4-10 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะการพิจารณาโทษ (Impunity)



รูปที่ 4-11 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะเจตนา (Intention)



รูปที่ 4- 12 จำนวนข้อมูลในแต่ละคลาสของคุณลักษณะการพิจารณาความผิด (Justification)

ในการแก้ปัญหาความไม่สมดุลของคลาสนั้น สามารถทำได้หลายวิธี แต่ในงานวิจัยนี้เลือกใช้การแก้ปัญหาในระดับข้อมูล (Data Level) โดยใช้วิธีการสุ่มตัวอย่างซ้ำ (Resampling Techniques) ชนิด Over-Sampling ซึ่งเป็นวิธีการที่ใช้ในการสุ่มเพิ่มจำนวนข้อมูลในกลุ่มข้างน้อยให้มีจำนวนใกล้เคียงกับข้อมูลในกลุ่มข้างมาก เพราะข้อมูลที่ใช้ในการฝึกฝนแบบจำลองในงานวิจัยนี้มีจำนวนน้อย จึงเลือกใช้การสุ่มเพิ่มตัวอย่างแทนการใช้ตัดลดตัวอย่างออก (Under-Sampling)

เทคนิคการสุ่มเพิ่มตัวอย่างข้างน้อยสังเคราะห์ (Synthetic Minority Over-sampling Technique : SMOTE) [31] ถูกนำมาใช้ในงานวิจัยนี้ เพื่อให้ตัวอย่างข้างน้อยมีจำนวนเท่ากับตัวอย่างข้างมาก โดยใช้ขั้นตอนวิธีการเพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbor Algorithm) ในการเพิ่มข้อมูลของคลาสกลุ่มน้อยให้มีจำนวนเท่ากับคลาสกลุ่มมาก และเมื่อดำเนินการเรียบร้อยพบว่าข้อมูลในแต่ละคลาสจะเท่ากันทั้งหมดและมีจำนวนเท่ากับจำนวนคลาสที่มากที่สุดเป็นชุดข้อมูลในเพื่อใช้ในงานวิจัย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4 การคัดเลือกแบบจำลองในงานวิจัย

ก่อนการดำเนินการจัดทำวิทยานิพนธ์นี้ ต้องทำการคัดเลือกแบบจำลองที่เหมาะสมกับงานวิจัย โดยใช้ชุดข้อมูลเรียนรู้ (Training set) และชุดข้อมูลทดสอบ (Test set) ชุดเดียวกัน โดยใช้ข้อมูลจากการรวบคำพิพากษาศาลฎีกา [5] ในหมวดความผิดเกี่ยวกับชีวิต โดยการกระทำที่เจตนา คือ มาตรา 288 และ 289 ผ่านการกระบวนการประมวลผลเบื้องต้น

เพื่อประเมินประสิทธิภาพของแบบจำลองโดยวัดจากค่าความแม่นยำ (Accuracy) ในการจำแนกภาษาธรรมชาติของแต่ละแบบจำลอง ได้แก่ แบบจำลองนาอีฟเบย์ (Naïve Bayes) ที่ในขั้นตอนการสร้างตัวแทน (Text Representation) เพื่อกำหนดคุณลักษณะ (Feature) แต่ละคำ เพื่อใช้ในการประมวลผลกับแบบจำลองที่แตกต่างกัน ได้แก่ การนับความถี่และคำนวณค่าน้ำหนักของคำ (TF-IDF) ใน 3 ประเภทได้แก่ ค่าน้ำหนักของแต่ละคำ (Word Level TF-IDF) ค่าน้ำหนักของคำด้วยเทคนิค N-Gram (N-Gram Level TF-IDF) โดยงานวิจัยเลือกใช้แกรมขนาด 2-แกรม และ 3-แกรม และค่าน้ำหนักของคำในระดับตัวอักษร (Character Level TF-IDF) และใช้โครงข่ายประสาทเทียม

หน่วยความจำระยะสั้นขนาดยาว (Long Short Term Memory networks : LSTM) ที่ใช้เทคนิคการฝังคำ (Word Embedding) จากชุดข้อมูลฝึกสอน (Training set) เป็นตัวแทนข้อความ ในการประมวลผลข้อมูล

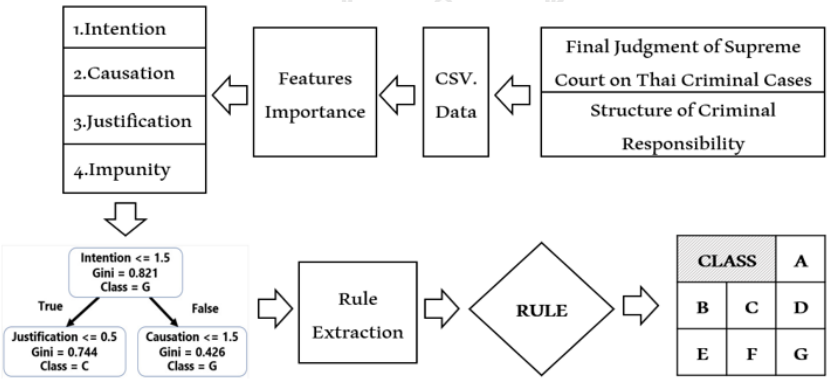
เพื่อเปรียบเทียบแบบประสิทธิภาพของจำลองการเรียนรู้ของเครื่องนาอีฟเบย์ และการเรียนรู้เชิงลึกด้วยแบบจำลอง LSTM ด้วยค่าความแม่นยำ (Accuracy) ในการคัดเลือกแบบจำลองที่เหมาะสมในงานวิจัย

4.5 การออกแบบโครงสร้างและฝึกฝนแบบจำลอง

4.5.1 การออกแบบและสร้างกฎด้วยต้นไม้ตัดสินใจ

งานวิจัยนี้ได้นำข้อมูลที่ได้จากแนวคำพิพากษาศาลฎีกาประกอบกับโครงสร้างความรับผิดทางอาญา จำนวน 210 คดีตัวอย่าง ที่ผ่านการคัดเลือกคุณสมบัติสำคัญเหลือ 4 อย่าง ได้แก่ Intention, Justification, Causation และ Impunity นำมาสร้างกฎในการตัดสิน เพื่อวิเคราะห์ข่าวอาชญากรรมในขอบเขตที่กำหนด โดยมีโครงสร้างการทำงานตามรูปที่ 4-13

ต้นไม้ตัดสินใจซึ่งเป็นแบบจำลองที่ใช้ในจำแนกเป็นความรับผิดที่สามารถอธิบายให้มนุษย์สามารถเข้าใจในกระบวนการวิเคราะห์แต่ละประเด็นข้อเท็จจริงได้ โดยแต่ละโหนดคือองค์ประกอบความผิดที่ใช้จำแนกข้อมูลที่นำเข้าว่าเป็นจริงหรือเท็จ (ด้านจริงจะอยู่ด้านซ้ายของโหนด ส่วนเท็จจะอยู่ด้านขวา) โดยใช้อัลกอริทึม CART และใช้ดัชนีจีนี่ (Gini Index) เป็นตัวชี้วัดการเลือกคุณลักษณะในการแบ่งชุดข้อมูลจะแบ่งออกเป็น 2 ชุดย่อย หรือเป็นตัวชี้วัดที่จะสามารถทำงานได้กับต้นไม้แบบไบนารี่ ที่เหมาะสมกับการพิจารณา ในแต่ละองค์ประกอบความผิดว่าในข้อเท็จจริงที่พิจารณา โดยใช้ข้อมูลในตารางที่ 4-2 จำนวน 210 คดีตัวอย่าง



รูปที่ 4-13 กระบวนการสร้างกฎเพื่อตัดสินข่าวอาชญากรรม

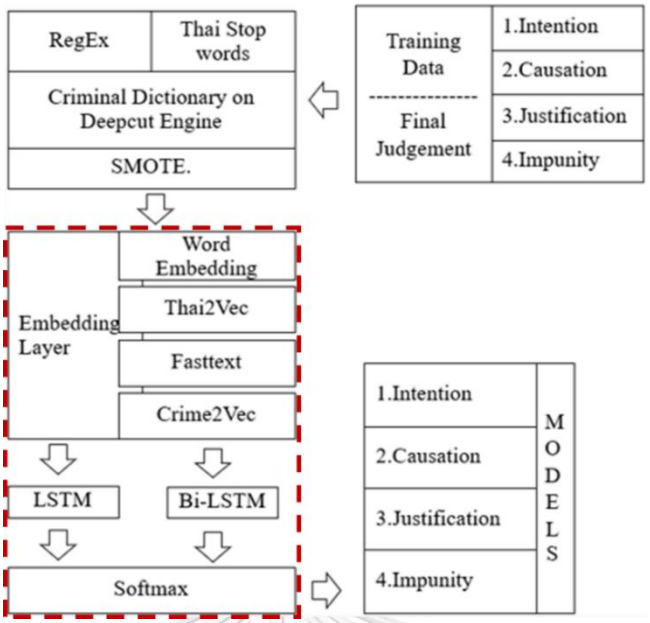
4.5.2 แบบจำลองในการประมวลผลภาษาธรรมชาติ

ในการวิเคราะห์ภาษาธรรมชาติของแต่ละคุณลักษณะสำคัญ เพื่อจำแนกข้อความที่นำเข้าว่ามีข้อเท็จจริงของแต่ละคุณลักษณะตรงตามกลุ่มใด โดยในงานวิจัยนี้เลือกใช้แบบจำลองของการเรียนรู้เชิงลึกที่เหมาะกับข้อความหรือประโยคในภาษาธรรมชาติที่เป็นการเรียงต่อกันของคำ (Sequence of word) เพราะลำดับของแต่ละคำมีความสำคัญ ส่งผลต่อความหมายของประโยค ดังนั้นจึงเลือกใช้ LSTM [15] ซึ่งเป็นแบบจำลองที่เหมาะกับข้อมูลที่เป็นลำดับ (Sequence Data) โดยการใช้ข้อมูลส่งออก (Output) ของสถานะก่อนหน้ากลับมาเป็นข้อมูลนำเข้า (Input) ของสถานะปัจจุบันในชั้นข้อมูลซ่อน (Hidden Layer) ร่วมกับข้อมูลขาเข้าในสถานะปัจจุบัน ทำให้เข้าใจความหมายของประโยคได้ดีกว่าเพราะได้อ่านข้อมูลจากคำก่อนหน้ามาแล้ว และนอกจากนี้ LSTM ยังสามารถแก้ปัญหา Vanishing Gradient ในข้อมูลที่มีความยาวมาก ๆ ที่ทำให้ไม่สามารถเห็นความเปลี่ยนแปลงในการปรับค่าถ่วงน้ำหนักได้ ซึ่งเป็นปัญหาหลักของแบบจำลองที่เป็นลำดับอย่าง Recurrent Neural Network (RNN) [18]

LSTM [15] เป็นแบบจำลองโครงข่ายประสาทเทียมแบบไปข้างหน้าในทิศทางเดียว ทำให้ได้บริบทของข้อมูลจากซ้ายไปขวาเท่านั้น แต่งานวิจัยชุดข้อมูลที่ใช้ในการฝึกฝนแบบจำลอง (Training Data) คำสำคัญที่ส่งผลต่อการจำแนกข้อความในแต่ละชุดข้อมูลไม่ได้อยู่แต่ในส่วนท้ายของประโยค แต่อยู่ตอนต้นหรือท้ายของประโยค ดังนั้นเพื่อเพิ่มประสิทธิภาพการทำงานของแบบจำลอง จึงมีการใช้ LSTM แบบสองทิศทาง เพื่อให้ได้บริบทโดยรอบทั้งการอ่านจากซ้ายไปขวาและอ่านจากขวาไปซ้าย เรียกว่าหน่วยความจำระยะสั้นแบบยาวสองทิศทาง (Bidirection LSTM) [23]

การถ่ายโอนความรู้ (Transfer Learning) [32] ถูกนำมาใช้ เพื่อเพิ่มประสิทธิภาพให้กับแบบจำลอง โดยใช้ Pretrained Word Embedding ของ Facebook ที่ผ่านการฝึกสอนจากคลังข้อมูลภาษาไทยขนาดใหญ่จาก Thai Wikipedia มาแล้ว และใช้ Thai2Vec [35] ซึ่งเป็น Pretrained Word Embedding ของกลุ่ม PyThaiNLP มาแทนการกำหนดค่าถ่วงน้ำหนักของแต่ละคำในชั้นการฝังคำ (Embedding Layer) ด้วยการสุ่มค่าในการฝึกฝนจากคลังข้อมูลที่ใช้ในงานวิจัย

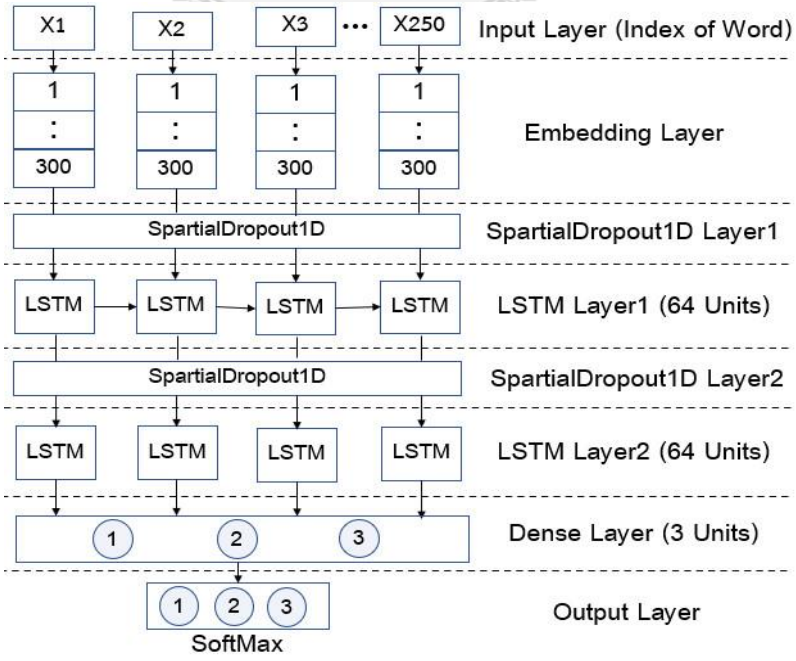
จากการเพิ่มประสิทธิภาพด้วยการใช้ Pretrained word embedding ที่ได้รับการฝึกฝนจากคลังข้อมูลขนาดใหญ่แล้ว นำมากำหนดเป็นค่าเริ่มต้นแทนการสุ่มค่าถ่วงน้ำหนัก ให้กับการฝึกฝนโครงข่ายนั้นเช่น FastText [34] หรือ Thai2Vec [35] แต่เป็นการใช้บริบทของแบบจำลองทางภาษาในโดเมนอื่น ดังนั้นผู้วิจัยจึงใช้ Gensim ซึ่งเป็นไลบรารีของ Word2Vec โดยการฝึกฝนแบบจำลองทางภาษา (Language Model) กับคลังข้อมูลข่าวอาชญากรรมที่ผู้วิจัยรวบรวมจำนวน 504 ข่าว เพื่อนำไปฝึกฝนกับตัวจำแนกข้อความต่อไป โดยมีภาพรวามของการออกแบบโครงสร้าง ตามที่แสดงในรูป 4-14



รูปที่ 4-14 โครงสร้างการทำงานของแบบจำลองภาษาธรรมชาติ

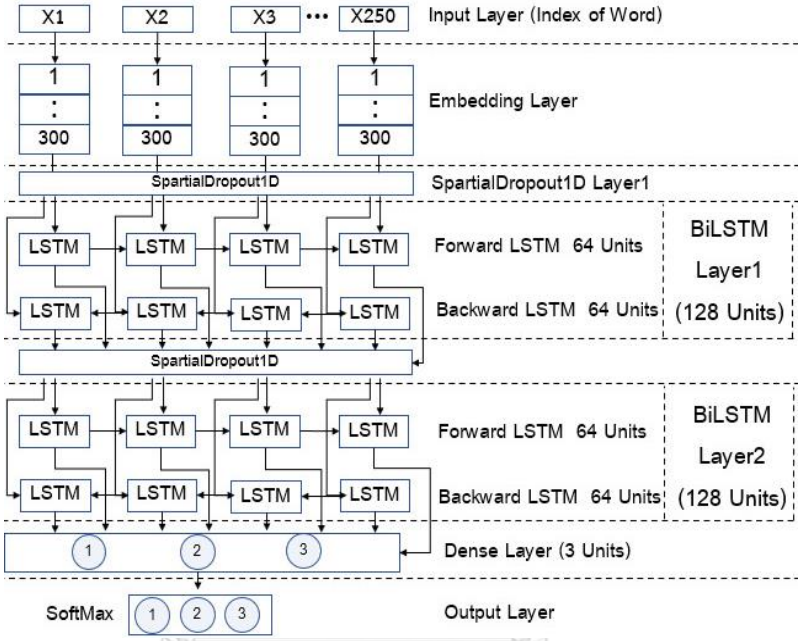
1) การออกแบบและฝึกฝนแบบจำลองด้วย LSTM และ Bi-LSTM

ผู้วิจัยใช้ข้อมูลฝึกสอนที่ผ่านขั้นตอนการตัดคำและเทคนิคการสุ่มเพิ่มตัวอย่างข้างน้อย สังเคราะห์เพื่อเป็นข้อมูลนำเข้าสำหรับการฝึกฝน โดยใช้อัลกอริทึม LSTM ทั้ง 4 คุณลักษณะตามที่แสดงในรูป 4-15



รูปที่ 4- 15 โครงสร้างภายในของแบบจำลอง LSTM

จากนั้นผู้วิจัยใช้ข้อมูลฝึกสอน (Training set) และชุดข้อมูลทดสอบ (Test set) ชุดเดียวกับที่ใช้ในโครงสร้างแบบจำลองของหน่วยความจำระยะสั้นแบบยาวทิศทางเดียว (LSTM) ตามที่แสดงในรูปที่ 4-15 ในการออกแบบโครงสร้างการฝึกฝนแบบจำลองของหน่วยความจำระยะสั้นแบบยาวสองทิศทาง (Bi-direction LSTM) ทั้ง 4 คุณลักษณะตามที่แสดงในรูป 4-16 และกำหนดค่าไฮเปอร์พารามิเตอร์ที่ใช้ในการฝึกฝนแบบจำลองแสดงตามตารางที่ 4-4



รูปที่ 4-16 โครงสร้างภายในของแบบจำลอง BiLSTM

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

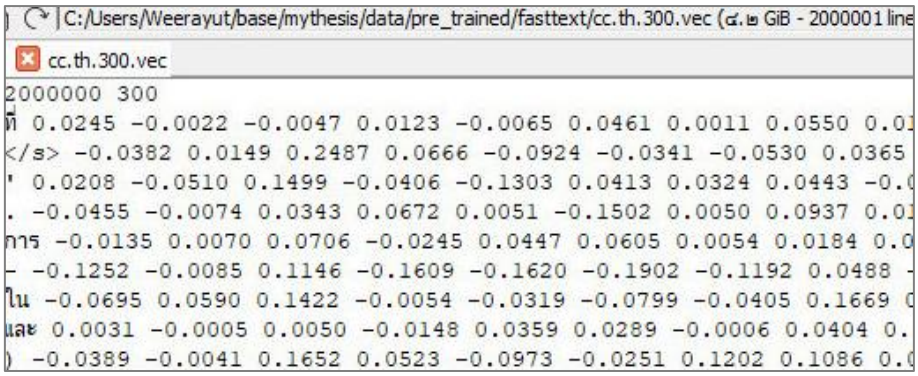
หลังจากการออกแบบโครงสร้างแบบจำลองแล้ว ขั้นตอนต่อไปคือการฝึกฝนแบบจำลองให้ครบทั้ง 4 คุณลักษณะหรือแบบจำลองที่ต้องการฝึกฝนได้แก่ แบบจำลองผลการกระทำ (Causation), แบบจำลองการพิจารณาโทษ (Impunity), แบบจำลองของเจตนา (Intention) และแบบจำลองการพิจารณาความผิดหรือละเว้นความผิด (Justification) โดยกำหนดค่าไฮเปอร์พารามิเตอร์ ตามที่แสดงในตารางที่ 4-4

ตารางที่ 4-4 การกำหนดค่าไฮเปอร์พารามิเตอร์ในการฝึกฝนแบบจำลอง

Models / Hyperparameter	Justification	Causation	Impunity	Intention
Optimization Algorithms	Adam	Adam	Adam	Adam
Loss Function	Binary Crossentropy	Categorical Crossentropy	Categorical Crossentropy	Categorical Crossentropy
Activation Function Hidden layer	Tanh	Tanh	Tanh	Tanh
Activation Function Output layer	Sigmoid	Softmax	Softmax	Softmax
Learning Rate	0.01	0.01	0.01	0.01
Max Words	1000	1000	1000	1000
Max length	100	100	100	100
Epoch	10	10	10	10
Batch size	32	64	32	32

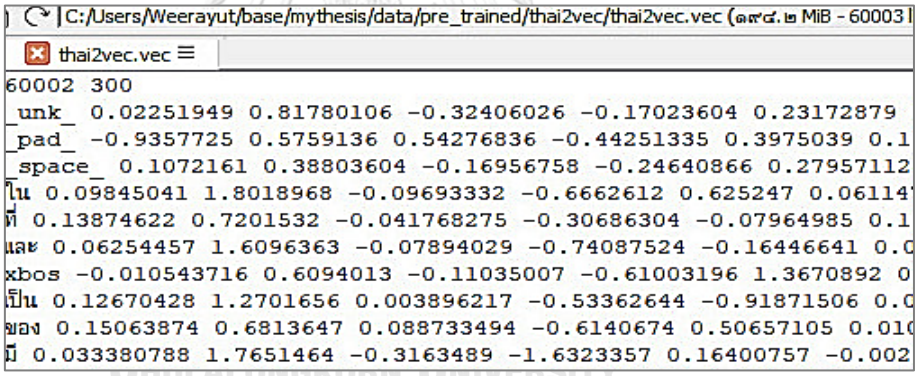
2) การเพิ่มประสิทธิภาพของแบบจำลองด้วยการใช้ Pre-trained Word Embedding

ค่าเริ่มต้นของค่าถ่วงน้ำหนักเกิดจากการสุ่ม (*Initial random weight*) ในกรณีที่ฝึกฝนโมเดลตั้งแต่ต้นด้วยชุดข้อมูลที่มี นอกจากต้องใช้เวลาในการฝึกฝนแบบจำลองเป็นเวลานาน และถ้าข้อมูลที่ใช้มีจำนวนไม่มากพอ ความแม่นยำของแบบจำลองอาจลดลงตามไปด้วย ผู้วิจัยจึงใช้ Pre-trained Word Embedding ที่ได้ผ่านการฝึกฝนจากคลังข้อมูล (Corpus) ขนาดใหญ่อย่าง FastText [34] ของเฟสบุคที่ใช้ข้อมูลฝึกฝนจาก Wikipedia ขนาด 2,000,000 embedding 300 มิติ ตามที่แสดงในรูป 4-17 ช่วยในการลู่เข้าหาคำตอบที่เหมาะสม (Converge) ของแบบจำลองทำได้เร็วและมีประสิทธิภาพมากขึ้น เพราะไม่ต้องสุ่มค่าถ่วงน้ำหนักเริ่มต้นในการฝึกฝน แต่อาศัยแบบจำลองที่เรียนรู้ลักษณะของรูปแบบทางภาษา (Language Model) มาแล้ว โดยทำการดาวน์โหลด FastText แล้วนำมาใช้เป็นค่าถ่วงน้ำหนักในชั้นระหว่างข้อมูลขาเข้า (Input) ไปยังชั้นซ่อนแรก (First Hidden Layer) ซึ่งเป็นเลเยอร์ของการฝังคำ จากนั้นก็ทำการตั้งค่าตามตารางที่ 4-4 และฝึกฝนทั้ง 4 แบบจำลอง



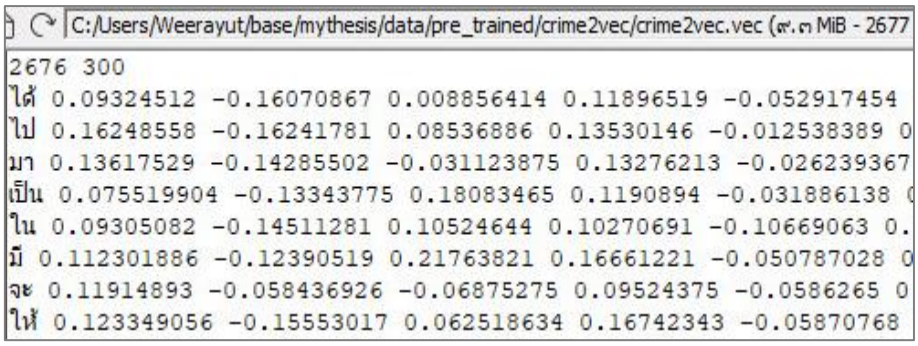
รูปที่ 4-17 FastText Pretrained Word Embedding

นอกจากนี้ผู้วิจัยได้ใช้ Pretrained Word Embedding อย่าง Thai2Vec [35] ของกลุ่ม PyThaiNLP ที่ฝึกฝนด้วยข้อมูลจาก Thai Wikipedia จำนวน 60,000 embedding ขนาด 300 มิติ เป็นค่าถ่วงน้ำหนักเริ่มต้นในชั้นซ่อนแรก (First Hidden Layer) ของแบบจำลอง ตามที่แสดงในรูป 4-18



รูปที่ 4-18 Thai2Vec Pretrained Word Embedding

และเพื่อให้แบบจำลองในงานวิจัยนี้ เข้าใจความหมายของบริบททางภาษาที่เกี่ยวกับคดีอาญา ผู้วิจัยจึงใช้ข้อมูลที่รวบรวมจากข่าวอาชญากรรมที่ไม่มีการกำหนดป้ายกำกับหรือข้อมูลที่ทำการฝึกฝนแบบไม่มีผู้สอน เพื่อให้แบบจำลองได้เรียนรู้ลักษณะทางภาษาที่เกี่ยวข้องกับงานวิจัยนี้จำนวน 504 ข่าว 2,676 embeddings ขนาด 300 มิติ โดยใช้เทคนิคการฝึกฝนด้วย Skip-gram ด้วย Gensim ซึ่งเป็นไลบรารีของ Word2Vec [36] โดยกำหนดขนาดของ window หรือจำนวนคำบริบทรอบข้าง 5 คำ (ซ้าย 5 ขวา 5) เมื่อฝึกฝนแบบจำลองเรียบร้อยก็บันทึกเป็นไฟล์ .Vec เพื่อนำไปเป็น Pretrained Word Embedding กำหนดค่าถ่วงน้ำหนักเริ่มต้นให้แบบจำลองทั้ง 4 ต่อไป



รูปที่ 4-19 Crime2Vec Pretrained Word Embedding

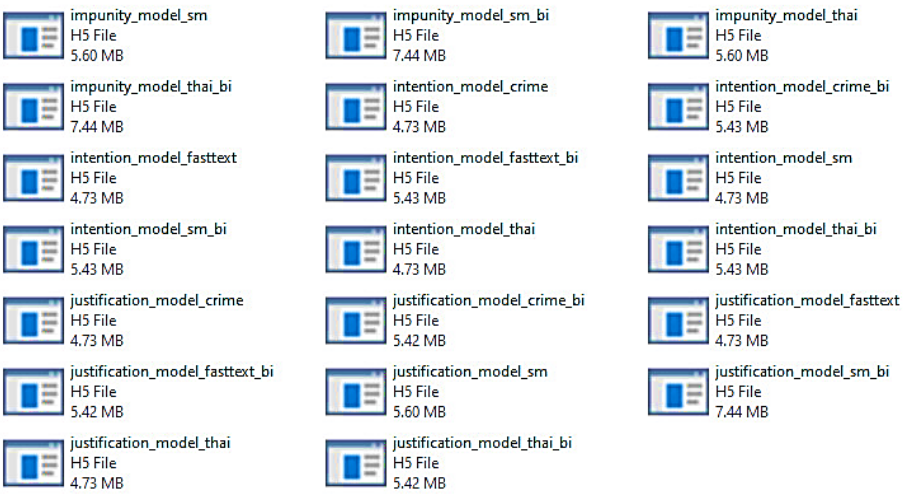
4.6 การบันทึกแบบจำลองและค่าถ่วงนำหนักเริ่มต้น

งานวิจัยนี้ทำการโหลดค่าถ่วงน้ำหนักเริ่มต้นจาก FatsText และ Thai2Vec ขนาด 300 มิติ (Dimension) มาเพื่อใช้กำหนดเป็นค่าเริ่มต้นให้การฝึกฝนแบบจำลอง และทำการฝึกฝนแบบจำลองภาษาด้วยข้อมูลฝึกสอนจากข่าวอาชญากรรมออนไลน์ ด้วยเทคนิค Skip-gram และ windows size ขนาด 5 ที่ทำนายคำรอบข้างด้านซ้ายจำนวน 5 คำ และด้านขวาจำนวน 5 คำ และบันทึกค่าถ่วงน้ำหนักเริ่มต้นเป็นไฟล์ .Vec ขนาด 300 มิติ คือ Crime2Vec.vec ทั้งสาม Pretrained Word Embedding ถูกบันทึกไว้ตามที่แสดงในรูปที่ 4-20 โดยพบว่า cc.th.300 คือไฟล์ของ FastText ซึ่งมีขนาด 4.21 GB ส่วน crim2vec และ thai2vec ขนาด 9.26 MB และ 194 MB ตามลำดับ

รูปที่ 4-20 ไฟล์ที่ใช้ในการกำหนดค่าถ่วงน้ำหนักเริ่มต้นในแบบจำลอง

และเมื่อทำการฝึกฝนแบบจำลองครบทั้ง 4 แบบหลัก ได้แก่ Intention, Causation, Justification และ Impunity โดยในแต่ละแบบจำลองถูกฝึกฝนด้วยอัลกอริทึมแบบเป็นลำดับ (Sequence of Vector) ทั้ง LSTM ทิศทางเดียว และแบบสองทิศทาง (Bidirectional LSTM) ด้วยคุณลักษณะของการสร้างคุณลักษณะ (Feature) ของการนำเข้ามูล 4 ลักษณะได้แก่ การสร้างการฝังคำจากข้อมูลหรือเรียนรู้ตั้งแต่เริ่มต้นจากคลังข้อมูลในงานวิจัยเอง การใช้ Pretrained Word Embedding จากภายนอกแทนการสุ่มค่าเริ่มต้นของค่าถ่วงน้ำหนักคือ FastText และ Thai2Vec และสร้าง Pretrained Word embedding ด้วยคลังข้อมูลจากข่าวอาชญากรรมออนไลน์ เรียกว่า Crime2Vec

จากนั้นทำการบันทึกแบบจำลองเป็นชนิดไฟล์ .h5 เพื่อเรียกใช้งานในการประเมินประสิทธิภาพ และทดสอบกับข้อมูลจริงที่ไม่เคยเห็นมาก่อน (Unseen Data) โดยการบันทึกข้อมูลผู้วิจัยบันทึกเป็นไฟล์ตามที่แสดงในรูปที่ 4-21



รูปที่ 4 21 แบบจำลองในงานวิจัยที่ทำการบันทึกเป็นไฟล์ .h5

4.7 การประเมินประสิทธิภาพของแบบจำลอง

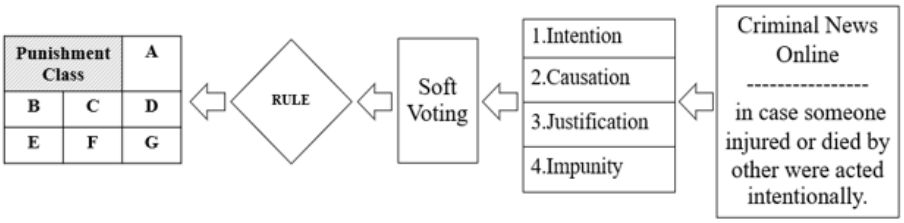
4.7.1 แบบจำลองที่ฝึกฝนด้วยต้นไม้ตัดสินใจ โดยใช้ข้อมูลที่เป็นโครงสร้างในรูปของตาราง ผู้วิจัยใช้การประเมินประสิทธิภาพของแบบจำลองด้วย การตรวจสอบแบบไขว้ (k-Fold) [29] โดยใช้ 5-fold cross-validation, 10-fold cross-validation และจากข้อมูลแบบทดสอบ (Test set) เพื่อดูค่าความแม่นยำของแบบจำลอง (Accuracy)

4.7.2 แบบจำลองที่ฝึกฝนด้วยภาษาธรรมชาติ โดยใช้ข้มูลแบบไม่เป็นโครงสร้าง ผู้วิจัยใช้การประเมินประสิทธิภาพของแบบจำลองทั้ง 4 แบบด้วยตารางคอนฟิวชันเมตริกซ์ (Confusion Metrics) [30] และวัดประสิทธิภาพแต่ละแบบจำลองด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score)

3) ในการเลือกใช้แบบจำลองในงานวิจัย ผู้วิจัยเลือกใช้ตัววัดประสิทธิภาพแบบจำลองด้วยค่าความแม่นยำ (Accuracy) ทั้งแบบจำลองนาอีฟเบย์ และแบบจำลอง LSTM

4) การประเมินประเมินความแม่นยำของแบบจำลองโดยรวม เมื่อได้แบบจำลองทั้งด้านการประมวลผลภาษาธรรมชาติและด้านการทำนายจากกฎของต้นไม้ตัดสินใจ ผู้วิจัยเลือกใช้ข่าวอาชญากรรมที่อยู่ในขอบเขตของงานวิจัย จำนวน 100 ข่าว และใช้การวิเคราะห์ผลการตัดสินในแต่ละข่าว โดยใช้ซอฟต์โหวต (Soft-Voting [42] คือการหาค่าเฉลี่ยของข้อเท็จจริงในแต่ละองค์ประกอบ

ความผิด โดยเทียบคำตอบที่ถูกต้องจากความเห็นของนักกฎหมาย เพื่อคิดเป็นร้อยละความถูกต้อง ตามที่แสดงในรูปที่ 4-22



รูปที่ 4- 22 การวัดประเมินความแม่นยำของแบบจำลองโดยภาพรวม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 5

การทดลองและผลการทดลอง

จากแนวคิดและวิธีการวิจัยในบทที่ผ่านมา ในบทนี้จะกล่าวถึงกระบวนการต่าง ๆ ในการทดลอง และผลการทดลองในแต่ละขั้นตอน ได้แก่ การเตรียมข้อมูลสำหรับงานวิจัย การประมวลผลข้อมูลเบื้องต้น การคัดเลือกแบบจำลองที่ใช้ในงานวิจัย การออกแบบโครงสร้างและฝึกฝนแบบจำลอง การปรับปรุงประสิทธิภาพของแบบจำลอง การบันทึกแบบจำลอง และการประเมินประสิทธิภาพของแบบจำลอง มีรายละเอียดดังนี้

5.1 สมมติฐานการทดลอง

การทดลองในขั้นตอนนี้ มีวัตถุประสงค์เพื่อวัดประสิทธิภาพของแบบจำลองทั้ง 4 แบบจำลอง ซึ่งเป็นองค์ประกอบความรับผิดอาญา เพื่อใช้เป็นข้อมูลนำเข้าให้แก่แบบจำลองต้นไม้ตัดสินใจ โดยแบบจำลองทั้งหมดจะถูกฝึกสอนด้วยชุดข้อมูลเดียวกัน แต่มีกลไกการเพิ่มประสิทธิภาพต่าง ๆ เรียกแบบจำลองนี้ว่า แบบจำลองที่นำเสนอโดยจะมีการทดสอบสมมติฐาน ดังนี้

1) แบบจำลองจากการเรียนรู้ของเครื่องด้วยนาอีฟเบย์ ซึ่งเป็นการเรียนรู้แบบเดิม มีประสิทธิภาพต่ำกว่าแบบจำลองที่นำเสนอ เมื่อวัดประสิทธิภาพด้วยค่าเฉลี่ยของความถูกต้อง (Accuracy)

2) แบบจำลองที่นำเสนอด้วยเทคนิคการฝึกฝนแบบสองทิศทางจะมีประสิทธิภาพสูงกว่าการฝึกฝนแบบทิศทางเดียว เมื่อวัดประสิทธิภาพด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3) แบบจำลองที่เพิ่มประสิทธิภาพด้วยการถ่ายโอนการเรียนรู้จากคลังข้อมูลอื่น (Transfer Learning) ที่มีข้อมูลสำหรับฝึกสอน (Training set) ขนาดใหญ่ให้ประสิทธิภาพสูงกว่าแบบจำลองที่เรียนรู้ด้วยชุดข้อมูลของตนเองที่มีขนาดของข้อมูลเล็กกว่า และชุดข้อมูลการเรียนรู้ (Corpus) จากงานอื่น ถ้ามีรูปแบบลักษณะทางภาษาคล้ายกับงานที่ต้องการทำการจำแนกข้อความ จะทำให้ประสิทธิภาพสูงกว่า เมื่อวัดประสิทธิภาพด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score)

4) การใช้ค่าเฉลี่ยของความน่าจะเป็นในชั้นข้อมูลส่งออก (Output Layer) ที่มีฟังก์ชันกระตุ้น SoftMax ส่งค่าความน่าจะเป็นของแต่ละคลาสออกมา ผ่านเทคนิคการหาค่าเฉลี่ยด้วย Soft-Voting ก่อนนำข้อมูลเข้าสู่ต้นไม้ตัดสินใจ มีประสิทธิภาพสูงกว่าการเลือกใช้แบบจำลองใดแบบจำลองหนึ่ง เมื่อวัดประสิทธิภาพด้วยค่าความถูกต้องคิดเป็นเปอร์เซ็นต์ของข้อมูลที่นำเข้าทดสอบ

5.2 การออกแบบการทดลอง

5.2.1 การสร้างชุดข้อมูลสำหรับวิจัย

ขั้นตอนนี้เป็นการกำหนดป้าย (Labels) ให้แต่ละข้อความในประโยคหรือคำที่เรียงต่อกันว่าเข้าข้อเท็จจริงในคลาสใด โดยทำการวิเคราะห์จากคำพิพากษาศาลฎีกาจนครบทั้ง 210 คดีตัวอย่าง ในทั้ง 4 คุณสมบัติหลักที่ส่งผลต่อการพิจารณาคดีในการกำหนดกลุ่มความรับผิดที่ได้กำหนดลักษณะของการกระทำหรือเหตุการณ์ที่เกิดขึ้นว่าตรงกับข้อเท็จจริงใด ตามที่แสดงตัวอย่างข้อมูลในตารางที่ 5-3 ถึง 5-6

ตารางที่ 5-1 ตัวอย่างข้อมูลคุณลักษณะผลการกระทำ (Causation)

ข้อเท็จจริง	ตัวอย่างข้อมูลตามแนวทางพิจารณาของศาลฎีกา
Nothing	กระสุนพลาดทำให้ไม่ถูกผู้ใด
Nothing	ผู้เสียหายหลบทันรอดตายหวุดหวิด
Injury	บาดเจ็บเพียงมีแผลถลอกและรอยแดงตามผิวหนัง
Injury	แพทย์รักษาไว้ทันเมื่อถึงโรงพยาบาล
Death	ฟันผู้ตายจนทรุดลงกับพื้นแล้วยังตามไปฟันซ้ำอีกหลายครั้งจนสิ้นใจ
Death	ถูกพบกลายเป็นศพทิ้งไว้ท้ายหมู่บ้าน

ตารางที่ 5-2 ตัวอย่างข้อมูลคุณลักษณะเจตนา (Intention)

ข้อเท็จจริง	ตัวอย่างข้อมูลตามแนวทางพิจารณาของศาลฎีกา
None	เมื่อจำเลยชกผู้ตายล้มลง จำเลยก็ไม่ได้ชกต่อยผู้ตายซ้ำอีก
None	จำเลยใช้มีดฟันร่างผู้ตาย ขณะผู้ตายถึงแก่ความตายแล้ว
Regular	จำเลยได้ใช้อาวุธปืนของกลางยิงผู้เสียหาย
Regular	จำเลยใช้มีดแทงผู้ตายในส่วนที่เป็นอวัยวะสำคัญ
Special	ได้ตระเตรียมอาวุธปืนและอาวุธมีดเพื่อใช้กระทำความผิดมาล่วงหน้า
Special	เมื่อพบเห็นผู้ตายก็ลงมือทันที โดยที่ผู้ตายกับพวกไม่ทันตั้งตัว

ตารางที่ 5-3 ตัวอย่างข้อมูลของคุณลักษณะการพิจารณาโทษ (Impunity)

ข้อเท็จจริง	ตัวอย่างข้อมูลตามแนวทางพิจารณาของศาลฎีกา
Except	ผู้เสียหายเป็นฝ่ายก่อเหตุขึ้นก่อน
Except	ไม่ได้ตีให้ตาย แต่ตีเพื่อให้หยุดทำร้ายและปล่อยตัวเท่านั้น
Decrease	ผู้ตายถือหินและไม้จะเข้าทำร้าย จำเลยพูดห้ามปรามไม่ยอมหยุดจึงถูกยิง
Decrease	ยิงผู้เสียหาย ในเวลาต่อเนื่องกระชั้นชิดกับที่จำเลยยังมีโทสะอยู่
Normal	ผู้เสียหายกับจำเลยรู้จักกันมานานแล้ว ไม่เคยมีสาเหตุโกรธเคืองกันมาก่อน
Normal	แทงอย่างแรงในขณะผู้เสียหายหันหลังให้
Increase	ก่อนเกิดเหตุผู้ตายและจำเลยทั้งสองมีความขัดแย้งกัน
Increase	ผู้เสียหายขัดขวาง จำเลยยิงผู้เสียหายทันที และยิงซ้ำอีก

ตารางที่ 5-4 ตัวอย่างข้อมูลของคุณลักษณะ พิจารณาความผิด (Justification)

ข้อเท็จจริง	ตัวอย่างข้อมูลตามแนวทางพิจารณาของศาลฎีกา
Wrong	ใช้อาวุธปืนยิงไปที่ประตูบ้านถูกผู้ตาย
Wrong	พวกของจำเลยยิงมาทางผู้เสียหายแต่กระสุนไม่ลั่น
Wrong	จำเลยยิงในขณะที่ผู้เสียหายขับรถสวนมา
Notwrong	ผู้ตายวิ่งไล่จำเลยพร้อมกับใช้ปืนพร้อมจะยิง จำเลยจึงยิงสวนผู้ตายเพียงนัดเดียว
Notwrong	จำเลยบอกว่าพาพ่อไปหาหมอ ผู้ตายพูดว่าสงสัยใกล้ตายแล้วข้องใจหรือไง
Notwrong	จำเลยใช้อาวุธปืนยิงผู้ตายในทันที ยับยั้งการกระทำของผู้ตาย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.2.2 การทำความสะอาดข้อมูล

ในการทำความสะอาดข้อมูลและตัดคำฟุ่มเฟือยใช้เทคนิคของนิพจน์ปกติ [25]ร่วมกับการตัดคำ โดยขั้นตอนการจัดการกับคำฟุ่มเฟือยก็โดยการเปรียบเทียบคำที่เราตัดมาแล้วว่ามีในคลังคำศัพท์ฟุ่มเฟือยหรือไม่ ถ้ามีก็ตัดออก เช่น การใช้ฟังก์ชัน sub() ในการลบคำที่ขึ้นต้นด้วยคำว่า “แยก” และมีคำอื่นตามหลัง เช่น ถ้าเจอคำว่า “แยกหลักสี่” ก็ให้ตัดทิ้ง ด้วยการแทนที่ว่าง “ ” เป็นต้น โดยแสดงตัวอย่างฟังก์ชันในการลบคำที่ไม่สำคัญออกตามเงื่อนไขที่กำหนด ดังแสดงในรูปที่ 5-1 และคำสั่งการลบคำฟุ่มเฟือยที่เก็บในคลังคำศัพท์ ตามที่แสดงในรูปที่ 5-2

```python
def clean_data(text):
# #ตัดคำที่เป็นที่อยู่ เพราะมีการเว้นวรรคอยู่แล้ว
# #1. สถานที่
    text = re.sub(r'\s+แยก+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+ซอย+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+ถ\.+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+เขต+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+แขวง+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+บ\.+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+บ้าน+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+ต\.+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+ตำบล+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+อ\.+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+อำเภอ+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+จ\.+[ก-๙]+' ,'',text)
    text = re.sub(r'\s+จังหวัด[ก-๙]+' ,'',text)
    text = re.sub(r'(?<=รพ\.)[ก-๙]+' ,'',text) # ตัดชื่อ รพ. ออก
    text = re.sub(r'(?<=สน\.)[ก-๙]+' ,'',text) # ตัดชื่อ สน.ออก
# #2. ตำแหน่งและหน่วยงาน
    text = re.sub(r'\s+ผกก\.สภ\.+[ก-๙]+','',text)
    text = re.sub(r'สภ+\.+[ก-๙]+','',text)
    text = re.sub(r'\s+รองสว\.','',text)
```

รูปที่ 5-1 ตัวอย่างการกำหนดรูปแบบในการตัดคำออกโดยการใช้นิพจน์ปกติ

```python
with open(r'C:\Users\Weerayut\base\mythesis\data\thai_stop_words.txt',encoding="utf8") as f:
    thai_stop_words = f.read().split('\n')

with open(r'C:\Users\Weerayut\base\mythesis\data\Province_district_sub.txt',encoding="utf8") as f:
    location_words = f.read().split('\n')
```

```python
final_text = []
for doc in pre_data:
    if not doc in thai_stop_words:
        final_text.append(doc)
```

รูปที่ 5-2 ตัวอย่างการใช้คำที่อยู่ในคลังคำฟุ่มเฟือยเป็นตัวเปรียบเทียบคำที่มีอยู่เพื่อลบออก

เมื่อผ่านกระบวนการทำความสะอาดและจัดการคำฟุ่มเฟือยแล้ว ข้อมูลที่ใช้สำหรับงานวิจัยทั้งที่มาจากคำพิพากษาศาลฎีกา เพื่อใช้ในการสร้างแบบจำลอง และข้อมูลที่มาจากข่าวอาชญากรรมออนไลน์ เพื่อใช้ในการสร้าง Pretrained Word Embedding และทดสอบแบบจำลองเพื่อใช้ในการทำนายข้อมูลจริง แสดงตัวอย่างในตารางที่ 5-5 และ 5-6

ตารางที่ 5-5 ตัวอย่างการทำความสะอาดข้อมูลจากคำพิพากษาศาลฎีกา

ข้อเท็จจริง	ตัวอย่างข้อมูลตามแนวทางพิจารณาของศาลฎีกา
Wrong	พวกของ~~จำเลย~~ยิงมาทางผู้เสียหายแต่กระสุนไม่ลั่น
Notwrong	~~จำเลย~~ใช้อาวุธปืนยิงผู้ตายในทันที ยับยั้งการกระทำของผู้ตาย

ตารางที่ 5-6 ตัวอย่างการทำความสะอาดข้อมูลจากเนื้อหาข่าวอาชญากรรม

ข้อมูลจากข่าวอาชญากรรม	ข้อมูลที่ผ่านกระบวนการทำความสะอาด
~~เมื่อวันที่ 5 มี.ค. พ.ต.อ.สัญญา พาดินธุ ผกก.สภ.ปากน้ำหลังสวน จ.ชุมพร~~ พร้อมด้วย ~~ร.ต.อ.วิฑูรย์ เผือกวารี รอง สว.(สอบสวน)~~ เจ้าของคดี นำกำลังคุมตัว~~นายชูหรือนายตะวัน~~ แรงงานต่างด้าวชาวมอญ ไปทำแผนประกอบคำรับสารภาพ ภายในสวนปาล์มพื้นที่หมู่ ~~1 ต.บางน้ำจืด อ.หลังสวน~~ หลังนายชูก่อเหตุข่มขืนแล้วฆ่ายาย~~วัย 72 ปี~~ โดยมีบรรดาญาติผู้ตาย เดินทางมาดูหน้า~~นายชู~~เป็นจำนวนมาก	พร้อมด้วยเจ้าของคดีนำกำลังควบคุมตัวแรงงานต่างด้าวชาวมอญไปทำแผนประกอบคำรับสารภาพภายในสวนปาล์มพื้นที่หลังก่อเหตุข่มขืนแล้วฆ่ายายโดยมีบรรดาทางญาติผู้ตายเดินทางมาดูหน้าเป็นจำนวนมาก

5.2.3 การประมวลผลข้อมูลเบื้องต้น

เพื่อให้แบบจำลองสามารถประมวลผลภาษาธรรมชาติได้ ขั้นตอนแรกหลังจากการทำความสะอาด ข้อมูลในแต่ละแบบจำลองคือการตัดคำโดยใช้ไลบรารีของ PyThaiNLP ด้วยตัวตัดคำ Deepcut [28] ร่วมกับการใช้คลังคำศัพท์ด้านกฎหมายที่ผู้วิจัยสร้างขึ้น ในการเลือกตัดคำแบบยาวที่สุดก่อน ตามที่แสดงในรูปที่ 5-3 และผลจากการตัดคำแสดงในตารางที่ 5-7

```
all_key_causation= []
for index, row in trainDF_causation.iterrows():
    text_str_action = row['action']
    text_str_status = row['status']
    text_str_action = str(text_str_action)
    text_str_status = str(text_str_status)
    text_clear_action = clean_data(text_str_action)
    all_text = [text_str_status,text_clear_action]
    all_text = ' '.join(all_text)
    text_token = myTokenonDeepcut(all_text)
    text_str_token = ' '.join(text_token)
    all_key_causation.append(text_str_token)
#  Tokenize word
all_text_str_causation = '\n'.join(all_key_causation)
causation_status_key = open("data/criminal_keyword/g3_causation/myOutCausationKey.txt","w",encoding='utf8')
causation_status_key.writelines(all_text_str_causation)
causation_status_key.close()
```

รูปที่ 5-3 การเรียกใช้ฟังก์ชันในการทำความสะอาดข้อมูลและการตัดคำ

ตารางที่ 5-7 การตัดคำด้วยเทคนิค Longest Matching ร่วมกับตัวตัดคำ Deepcut [28]

ข้อมูลที่ผ่านกระบวนการทำความสะอาดแล้ว	ข้อมูลที่ผ่านกระบวนการตัดคำ
พร้อมด้วยเจ้าของคดีนำกำลังควบคุมตัวแรงงานต่างด้าวชาวมอญไปทำแผนประกอบคำรับสารภาพภายในสวนปาล์มพื้นที่หลังก่อเหตุข่มขืนแล้วฆ่ายายโดยมีบรรดาทางญาติผู้ตายเดินทางมาดูหน้าเป็นจำนวนมาก	พร้อม|ด้วย|เจ้าของ|คดี|นำกำลัง|ควบคุมตัว|แรงงาน|ต่างด้าว|ชาวมอญ|ไป|ทำแผน|ประกอบคำรับสารภาพ|ภายใน|สวนปาล์ม|พื้นที่|หลังก่อเหตุ|ข่มขืน|แล้วฆ่า|ยาย|โดย|มีบรรดา|ทาง|ญาติ|ผู้ตาย|เดินทาง|มา|ดู|หน้า|เป็นจำนวนมาก

เมื่อข้อมูลที่ใช้ในงานวิจัยผ่านกระบวนการเตรียมข้อมูลและการตัดคำเรียบร้อยแล้ว ชุดข้อมูลก็ถูกเขียนบันทึกไว้ในรูปแบบเอกสารข้อความ (Text File) จนครบทั้ง 4 คุณลักษณะหลัก ตามที่แสดงตัวอย่าง ในรูปที่ 5-4 ถึงรูปที่ 5-7

action	status
สภาพ|ศพ	death
คว้า|ขวดปากฉลาม|แทงคอ|ทหาร|ดับ|งาน|คอนเสิร์ต|ส...	death
พบ|อีก|ที|ถูก|ฆ่า|หมก|ยัด|ท่อ|เน่าเป็นซาก	death
ผู้รับเหมา|เดือดทวงเงิน|ไม่ได้|ยิง|นายจ้าง	injury
ตามรวบ|หนุ่มใหญ่|หนีคดี|มีดฟัน|สายลับตำรวจ|ปางตาย	injury

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

รูปที่ 5-4 ตัวอย่างข้อมูลของคุณลักษณะผลการกระทำที่ผ่านกระบวนการตัดคำ

action	status
แม้|ยับยั้ง|ไม่|ยิง|ต่อไป|จนตาย	normal
ฆ่าผู้อื่นโดยทรมาน|หรือ|โดย|กระทำทารุณโหดร้าย|...	increase
คนร้าย|ชกต่อย|และ|ถูก|คนร้าย|ใช้|เหล็กขูดชาฟท์...	normal
ผู้ตาย|ซึ่ง|มี|อาวุธ|ปืนพกลูกซองสั้น|ติดตัว|ได...	except
ถูก|ทำร้าย	decrease

รูปที่ 5-5 ตัวอย่างข้อมูลของคุณลักษณะการพิจารณาโทษที่ผ่านกระบวนการตัดคำ

action	status
ผู้ตาย|ล้มลง|ไม่ได้|ซ้ำอีก	none
เดิน|ออก|จาก|ร้าน|ไป|ผู้ตาย|เดิน|ตาม|ผู้...	none
ตี|และ|ฟัน|ผู้ตาย|กับ|พวก	regular
ไม่|พบ|ความผิดปกติ|ทาง|สมอง	regular
รถ|ผู้เสียหาย|ล้มลง|มา|จับ|รถ|ของ|ผู้เสี...	special
เชื่อ|ว่า|ผู้ตาย|จะ|ตาม|มา|เอา|เรื่อง	special

รูปที่ 5-6 ตัวอย่างข้อมูลของคุณลักษณะเจตนาที่ผ่านกระบวนการตัดคำ

action	status
ฟัน|ผู้ตาย|จนทรุด|ลง|กับ|พื้น	wrong
ถูกฟัน|เป็น|บาดแผล|ที่|บริเวณ|ใบหน้า|ด้า...	wrong
ใช้|มือ|ผลักออก|ผู้ตาย|แต่|ผู้ตาย|กลับ	notwrong
ตาม|ขึ้น|ไป|แล้ว|ใช้|อาวุธ|ปืน|ของกลาง|ยิง|ผู้...	wrong
จึง|ไม่มีทางเลือก|นอก|จาก|แย่ง|อาวุธ|ปืน|จาก	notwrong

รูปที่ 5-7 ตัวอย่างข้อมูลของคุณลักษณะพิจารณาความผิดที่ผ่านกระบวนการตัดคำ

หลังจากทำการตัดคำในแต่ละตัวอย่างของทั้ง 4 แบบจำลองแล้ว ขั้นตอนต่อไปคือการสุ่มเพิ่มข้อมูลของคลาสกลุ่มน้อย (Minority Class) และคลาสกลุ่มอื่นที่มีจำนวนตัวอย่างน้อยกว่าให้มีจำนวนเท่ากับคลาสกลุ่มที่มากที่สุด โดยงานวิจัยนี้จะทำการสุ่มเพิ่มข้อมูลในทุกคลาสยกเว้นคลาสของกลุ่มข้อมูลมาก (Majority Class) ให้มีจำนวนเท่ากับคลาสของกลุ่มข้อมูลมากตามตัวอย่างคำสั่งในรูปที่ 5-8 และแสดงจำนวนข้อมูลในแต่ละคลาสตามที่แสดงในรูปที่ 5-9 และเมื่อดำเนินการเรียบร้อยพบว่าข้อมูลในแต่ละคลาสจะเท่ากันทั้งหมดและมีจำนวนเท่ากับจำนวนคลาสที่มากที่สุด ตามที่แสดงในตารางที่ 5-8

```
21
22 sm = SMOTE('not majority')
23 X_res_caus, Y_res_caus = sm.fit_resample(X_new_caus, Y_new_caus)
24 X_res_intt, Y_res_intt = sm.fit_resample(X_new_intt, Y_new_intt)
25 X_res_imp, Y_res_imp = sm.fit_resample(X_new_imp, Y_new_imp)
26 X_res_just, Y_res_just = sm.fit_resample(X_new_just, Y_new_just)
27
```

รูปที่ 5-8 การสุ่มเพิ่มตัวอย่างของคลาสทุกคลาสยกเว้นคลาสกลุ่มมาก

```
df_class_injury_over=df_class_injury.sample(count_class_die, replace=True)
df_class_nothing_over=df_class_nothing.sample(count_class_die, replace=True)
df_test_over_caus=pd.concat([df_class_die, df_class_injury_over,df_class_nothing_over],axis=0)
print(df_test_over_caus.shape)
print(df_test_over_caus['status'].value_counts())
```

```
(645, 2)
injury     215
nothing    215
death      215
```

รูปที่ 5-9 การดูจำนวนข้อมูลของแต่ละคลาสในแบบจำลอง Causation

จากรูปที่ 5-9 เมื่อข้อมูลในแบบจำลองผลการกระทำ Causation ผ่านเทคนิค SMOTE แล้ว พบว่าจำนวนข้อมูลของ injury Nothing และ Death มีจำนวนเท่ากัน

ตารางที่ 5-8 เปรียบเทียบข้อมูลในแต่ละคลาสก่อนและหลังการใช้เทคนิค SMOTE

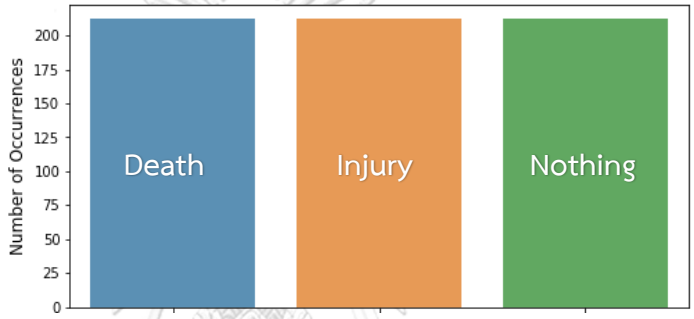
คุณลักษณะ	**คลาส**	**จำนวนข้อมูลในแต่ละคลาสเดิม**	**จำนวนข้อมูลหลัง SMOTE**
Intention	Regular	261	261
	Special	168	261
	None	134	261
Causation	Death	215	215
	Injury	114	215
	Nothing	22	215
Impunity	Except	139	215
	Decrease	84	145
	Normal	145	145
	Increase	97	145
Justification	Wrong	142	142
	Notwrong	134	142

จากตารางแสดงจำนวนข้อมูลที่ผ่านมา แสดงให้เห็นว่าในแต่ละแบบจำลองหรือแต่ละคุณลักษณะ จะมีจำนวนข้อมูลในแต่ละคลาสไม่เท่ากัน ส่งผลให้การทำนายการจำแนกข้อความเอนเอียงกับคลาสที่เป็นกลุ่มมาก ทั้งที่ข้อมูลกลุ่มที่น้อยกว่ามีความสำคัญ จึงแก้ปัญหาโดยการให้แต่ละกลุ่มข้อมูลของทุกคลาสมีจำนวนเท่ากัน และเมื่อผ่านกระบวนการ SMOTE แล้ว ทุกคลาสจะมีจำนวนข้อมูลเท่ากันและมีขนาดเท่ากับจำนวนของคลาสที่มีจำนวนมากที่สุด (Majority Class) ตามที่แสดงใน

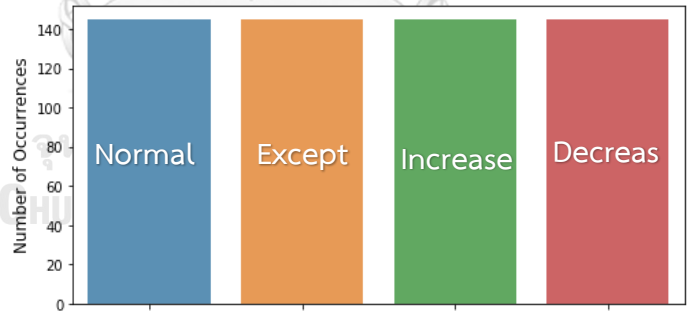
รูปที่ 5-10 เพื่อแสดงกราฟในแต่ข้อมูลของทุกคลาสแต่ละแบบจำลอง ตามที่แสดงในรูปที่ 5-11 ถึงรูปที่ 5-14 ที่แสดงถึงจำนวนข้อมูลในแต่ละคลาสของทั้ง 4 แบบจำลอง

```
cnt_imp_over = df_test_over_imp['status'].value_counts()
plt.figure(figsize=(8,4))
sns.barplot(cnt_imp_over.index, cnt_imp_over.values, alpha=0.8)
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel( 'Class of Causation', fontsize=12)
plt.xticks(rotation=90)
plt.show();
```

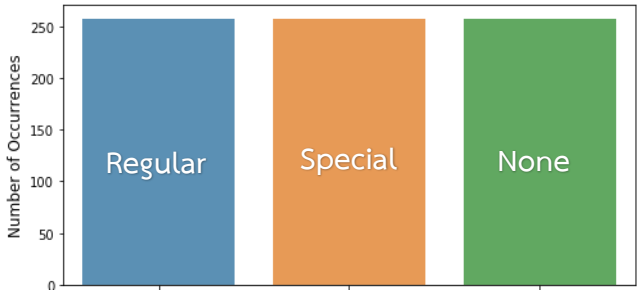
รูปที่ 5-10 การแสดงกราฟจำนวนข้อมูลในแต่ละคลาสของแบบจำลอง



รูปที่ 5-11 จำนวนข้อมูลในแต่ละคลาสของ Causation เมื่อผ่านเทคนิค SMOTE



รูปที่ 5-12 จำนวนข้อมูลในแต่ละคลาสของ Impunity เมื่อผ่านเทคนิค SMOTE



รูปที่ 5-13 จำนวนข้อมูลในแต่ละคลาสของ Intention เมื่อผ่านเทคนิค SMOTE

รูปที่ 5-14 จำนวนข้อมูลในแต่ละคลาสของ Justification เมื่อผ่านเทคนิค SMOTE

5.3 การคัดเลือกแบบจำลอง

ในงานวิจัยนี้ในส่วนของการเลือกแบบจำลองสำหรับการจำแนกข้อความภาษาธรรมชาติเพื่อใช้ในงานวิจัยที่เกี่ยวกับการประมวลผลข้อความที่เกี่ยวข้องกับคดีอาญา โดยผู้วิจัยเปรียบเทียบระหว่างแบบจำลองการเรียนรู้ของเครื่องด้วยนาอีฟเบย์ ที่มีการคัดเลือกคุณลักษณะ (Features) ด้วยเทคนิคต่าง ๆ เปรียบเทียบกับแบบจำลองของการเรียนรู้เชิงลึก LSTM ซึ่งเป็นอัลกอริทึมหลักที่งานวิจัยนี้นำเสนอ โดยผู้วิจัยใช้การประเมินประสิทธิภาพของแบบจำลองด้วยค่าความแม่นยำ ตามตัวอย่างที่แสดงในรูปที่ 5-15 และ 5-16 และมีผลการประเมินความแม่นยำ ตามตารางที่ 5-9

```
accuracy = train_model(naive_bayes.MultinomialNB(), xtrain_count, train_y, xvalid_count)
print("Naive Bayes Just Model : Count Vectors = {:.2f}%".format(accuracy*100))

# Naive Bayes on Word Level TF IDF Vectors
accuracy = train_model(naive_bayes.MultinomialNB(), xtrain_tfidf, train_y, xvalid_tfidf)
print("Naive Bayes Just Model : TF-IDF Vectors = {:.2f}%".format(accuracy*100))

# Naive Bayes on Ngram Level TF IDF Vectors
accuracy = train_model(naive_bayes.MultinomialNB(), xtrain_tfidf_ngram, train_y,
                       xvalid_tfidf_ngram)
print("Naive Bayes Just Model : N-Gram Vectors = {:.2f}%".format(accuracy*100))

# Naive Bayes on Character Level TF IDF Vectors
accuracy = train_model(naive_bayes.MultinomialNB(), xtrain_tfidf_ngram_chars, train_y,
                       xvalid_tfidf_ngram_chars)
print("Naive Bayes Just Model : Character Level Vectors = {:.2f}%".format(accuracy*100))
```

```
Naive Bayes Just Model : Count Vectors = 74.32%
Naive Bayes Just Model : TF-IDF Vectors = 68.24%
Naive Bayes Just Model : N-Gram Vectors = 74.32%
Naive Bayes Just Model : Character Level Vectors = 70.27%
```

รูปที่ 5-15 การวัดความแม่นยำแบบจำลอง Justification ด้วยแบบจำลองนาอีฟเบย์

```
# Accuracy Score
score_lstm_just, accJust_lstm = model_justification_sm_lstm.evaluate( X_test_sm_justification, Y_test_sm_justification ,verbose
score_bi_lstm_just, accJust_bi_lstm = model_justification_sm_bi_lstm.evaluate( X_test_sm_justification, Y_test_sm_justification
score_lstm_caus, accCaus_lstm = model_causation_sm_lstm.evaluate( X_test_sm_causation, Y_test_sm_causation ,verbose=2)
score_bi_lstm_caus, accCaus_bi_lstm = model_causation_sm_bi_lstm.evaluate( X_test_sm_causation, Y_test_sm_causation ,ver
score_lstm, accIntt_lstm = model_intention_sm_lstm.evaluate( X_test_sm_intention, Y_test_sm_intention, verbose=2)
score_bi_lstm, accIntt_bi_lstm = model_intention_sm_bi_lstm.evaluate( X_test_sm_intention, Y_test_sm_intention, verbose=2)
score_lstm, accImp_lstm = model_impunity_sm_lstm.evaluate( X_test_sm_impunity, Y_test_sm_impunity,verbose=2)
score_bi_lstm, accImp_bi_lstm = model_impunity_sm_bi_lstm.evaluate( X_test_sm_impunity, Y_test_sm_impunity,verbose=2)
print('Test accuracy justification lstm model :', accJust_lstm)
print('Test accuracy causation lstm model :', accCaus_lstm)
print('Test accuracy intention lstm model :', accIntt_lstm)
print('Test accuracy impunity lstm model :', accImp_lstm)
```

```
Test accuracy justification lstm model : 0.8070175323569984
Test accuracy causation lstm model : 0.6201550387596899
Test accuracy intention lstm model : 0.7898089187160419
Test accuracy impunity lstm model : 0.7586206937658375
```

รูปที่ 5-16 การวัดความแม่นยำแบบจำลอง Justification ด้วยแบบจำลอง LSTM

ตารางที่ 5-9 เปรียบเทียบค่าความถูกต้องของแต่ละแบบจำลอง

Model	Accuracy(%)				ค่าเฉลี่ย Acc. (%)
	Causation	Impunity	Justification	Intention	
Naïve Bayes: Count Vectors	72.58	58.97	74.32	72.30	69.54
Naïve Bayes: Word Level TF-IDF	64.52	49.57	68.24	64.86	61.79
Naïve Bayes: N-Gram Level TF-IDF	66.13	67.52	74.32	79.05	71.75
Naïve Bayes: Character Level TF-IDF	74.19	53.85	70.27	66.22	66.13
Long-Short Term Memory (LSTM)	62.02	75.86	80.70	78.98	74.39

จากตารางที่ 5-3 แสดงให้เห็นการเปรียบเทียบการใช้อัลกอริทึมในการจำแนกข้อความ (Text Classification) ได้แก่ นาอีฟเบย์ (Naïve Bayes) ที่ใช้การสร้างตัวแทนข้อความด้วยเทคนิคที่แตกต่างกัน ซึ่งเป็นการเรียนรู้แบบเดิม และแบบจำลองที่นำเสนอคือ LSTM ใช้เทคนิคการฝังคำ (Word Embedding) ผลการประเมินด้วยข้อมูลทดสอบ (Test set) ชุดเดียวกัน พบว่าแบบจำลองโครงข่ายการเรียนรู้เชิงลึกด้วยแบบจำลอง LSTM ให้ค่าความถูกต้อง (Accuracy) เฉลี่ยจากทั้ง 4 แบบจำลองมากที่สุด

จุฬาลงกรณ์มหาวิทยาลัย

และในส่วนการสร้างแบบจำลองเพื่อจำแนกความรับผิดออกเป็นกลุ่มตามที่กำหนดในขอบเขตงานวิจัยคือกลุ่ม A-G เพื่อให้การการทำงานของแบบจำลองสามารถอธิบายให้มนุษย์เข้าใจได้ถึงเหตุผลว่าทำไมถึงตัดสินเป็นกลุ่มนั้น ๆ งานวิจัยนี้จึงเลือกใช้แบบจำลองต้นไม้ตัดสินใจในการสร้างกฎและจำแนกผลการตัดสิน

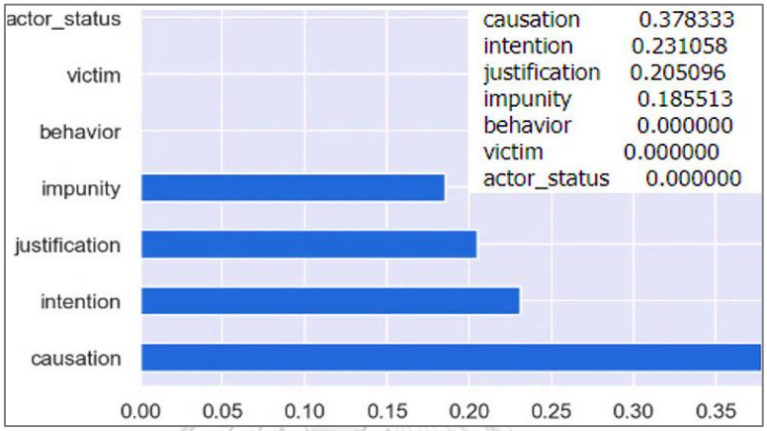
5.4 การคัดเลือกคุณลักษณะสำคัญ

ต้นไม้ตัดสินใจ [39] มีฟังก์ชันการในคัดเลือกการคัดเลือกคุณสมบัติสำคัญ (Feature importance) ตามโครงสร้างความรับผิดอาญาที่สร้างไว้ในตารางที่ 4-1 เมื่อผ่านการคัดเลือกคุณสมบัติสำคัญที่ส่งผลต่อการจำแนกประเภทความรับผิดแล้ว ทำให้มีคุณสมบัติสำคัญเหลือ 4 อย่าง ได้แก่ เจตนา (Intention) ความรับผิด (Justification) การพิจารณาโทษ (Impunity) และผลการ

กระทำ (Causation) โดยใช้ตัวอย่างตามรูปที่ 5-17 และแสดงค่าความสำคัญของแต่ละคุณลักษณะตามรูปที่ 5-18

```
1 fim = pd.Series(dtree.feature_importances_, index=X_train.columns).sort_values(ascending=False)
```

รูปที่ 5-17 การเรียกใช้ฟังก์ชันการคัดเลือกคุณสมบัติสำคัญในต้นไม้ตัดสินใจ



รูปที่ 5-18 แสดงการคัดเลือกคุณลักษณะสำคัญที่ส่งผลต่อการจำแนกกลุ่มความรับผิด

เมื่อได้คุณลักษณะสำคัญที่ส่งผลต่อการจำแนกประเภทของต้นไม้ตัดสินใจแล้ว ก็สร้างตารางข้อเท็จจริงและการจำแนกกลุ่มความรับผิดใหม่ เพื่อใช้ในการสร้างแบบจำลองภาษาธรรมชาติ ได้แก่ แบบจำลอง Intention, Causation, Justification และ Impunity ตามที่แสดงในตารางที่ 5-10

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 5-10 ข้อเท็จจริงตามโครงสร้างความรับผิดอาญาเมื่อผ่านการคัดเลือกคุณสมบัติที่สำคัญ

ผลการกระทำ (Causation)	พิจารณาความผิด (Justification)	พิจารณาโทษ (Impunity)	เจตนา (Intention)	กลุ่มความรับผิด (Class)
Death	Notwrong	Except	None	A
Injury	Wrong	Decrease	Regular	B
Death	Wrong	Decrease	Regular	C
Injury	Wrong	Normal	Regular	D
Death	Wrong	Normal	Regular	E
Nothing	Wrong	Increase	Special	F
Death	Wrong	Increase	Special	G

5.5 การออกแบบและสร้างกฎการตัดสินใจ

ในการฝึกฝนแบบจำลองต้นไม้ตัดสินใจ โดยใช้ข้อมูลในตารางที่ 5-2 จำนวน 210 ตัวอย่าง แบ่งข้อมูลในการฝึกฝน/การทดสอบ ด้วยอัตราส่วน 70/30 โดยมีข้อมูลฝึกฝน (Training set) 147 ตัวอย่าง และข้อมูลทดสอบ (Test set) 63 ตัวอย่าง ประเมินประสิทธิภาพของแบบจำลองด้วยค่าความแม่นยำ (Accuracy) ด้วยตัวอย่างชุดคำสั่งตามรูปที่ 5-19 ถึง 5-22 โดยใช้ชุดข้อมูลทดสอบ (Test set), 5-fold cross-validation และ 10-fold cross-validation [29] และได้ค่าความถูกต้องตามที่แสดงในตารางที่ 5-11

```
# split data randomly into 70% training and 30% test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,stratify=y,)
dtree = DecisionTreeClassifier(criterion='gini')
dtree.fit(X_train, y_train)
```

รูปที่ 5-19 การแบ่งข้อมูลและฝึกฝนแบบจำลองต้นไม้ตัดสินใจ

```
count_misclassified = (y_test != y_pred).sum()
print('Misclassified samples: {}'.format(count_misclassified))
accuracy = metrics.accuracy_score(y_test, y_pred)
accr = accuracy*100
print('Accuracy: {:.2f}%'.format(accr))
```

```
Misclassified samples: 0
Accuracy: 100.00%
```

รูปที่ 5-20 ค่าความแม่นยำชุดข้อมูลทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
from sklearn.model_selection import cross_val_score
clf = DecisionTreeClassifier(random_state=9)
score_cv = cross_val_score(clf, X, y, cv=5)
print("Accuracy of 10-fold cross validation is :" ,score_cv.mean())
```

```
Accuracy of 10-fold cross validation is : 0.9948717948717949
```

รูปที่ 5-21 ค่าความแม่นยำเฉลี่ยของ 5-Fold Cross Validation

```
from sklearn.model_selection import cross_val_score
clf = DecisionTreeClassifier(random_state=9)
score_cv = cross_val_score(clf, X, y, cv=10)
print("Accuracy of 10-fold cross validation is :" ,score_cv.mean())
```

```
Accuracy of 10-fold cross validation is : 0.9944444444444445
```

รูปที่ 5-22 ค่าความแม่นยำเฉลี่ยของ 10-Fold Cross Validation

ตารางที่ 5-11 เปรียบเทียบประสิทธิภาพแบบจำลองต้นไม้ตัดสินใจด้วยค่าความแม่นยำ

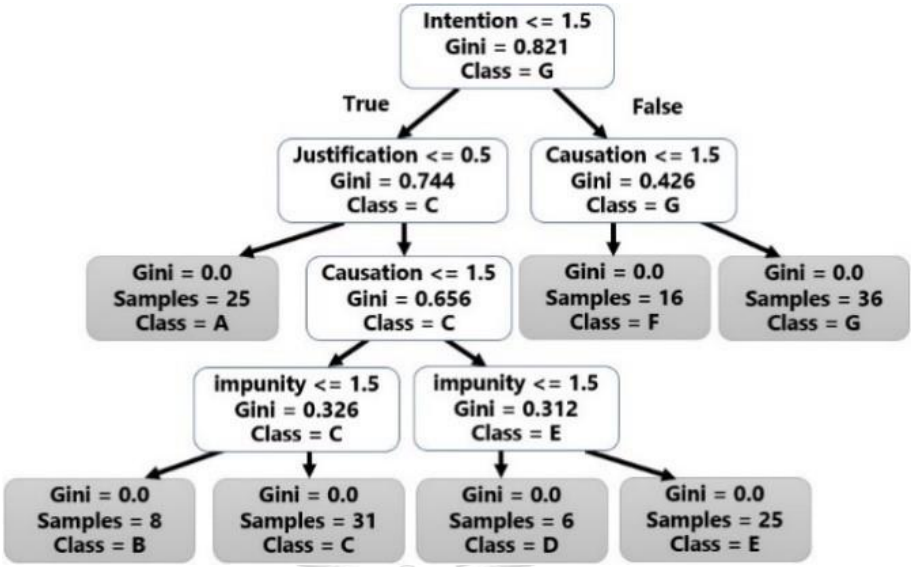
รูปแบบการประเมินค่าความแม่นยำ	**เปอร์เซ็นต์ (100%)**
Test set (30%)	100
5-Fold Cross Validation	99.48
10-Fold Cross Validation	99.44

จากตารางที่ 5-11 เพื่อป้องกันปัญหาการยึดติดข้อความที่ใช้สอนมากเกินไป (Overfitting) ในการทำงานของแบบจำลองต้นไม้ตัดสินใจ ผู้วิจัยจึงเลือกใช้การประเมินจากการการตรวจสอบไขว้กันแบบ 5-Fold Cross Validation และ 10-Fold Fold Cross Validation ได้ผลค่าความถูกต้อง 99.48% และ 99.44% ตามลำดับ ขณะที่ข้อมูลที่ใช้ในการทดสอบเป็นข้อมูลที่แบ่งไว้จำนวน 30% คิดเป็น 63 ตัวอย่าง ให้ค่าความถูกต้อง 100% ซึ่งถือว่าแบบจำลองต้นไม้สามารถให้ค่าความถูกต้องสูง สามารถจำแนกประเภทกลุ่มความรับผิดได้แม่นยำและยังสามารถอธิบายให้เข้าใจได้ถึงแนวทางการตัดสินตามหลักเกณฑ์ทางกฎหมาย และแสดงไดอะแกรมของต้นไม้ตัดสินใจ (Decision Tree Diagram) จากจำนวนตัวอย่างที่ใช้ฝึกสอนแบบจำลองจำนวน 70% หรือ 147 ตัวอย่าง ตามตารางที่ 5-10 จากนั้นจึงนำแบบจำลองที่ผ่านการฝึกฝนแล้วไปสร้างกฎการตัดสินใจ [39] ตามที่แสดงในรูปที่ 5-23 และแสดงผลลัพธ์ของไดอะแกรม ตามที่แสดงในรูปที่ 5-24

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
dot_data = tree.export_graphviz(new_dtree, out_file=None, filled=True, rounded=True,
                                feature_names=new_features,
                                class_names=class_names)
filename = "tree_graph/new_gini_graph .png"
pydotplus.graph_from_dot_data(dot_data).write_png(filename)
plt.figure(figsize=(40,40))
img = mpimg.imread(filename)
imgplot = plt.imshow(img)
plt.show();
```

รูปที่ 5-23 คำสั่งที่ใช้แสดงไดอะแกรมของกราฟต้นไม้ตัดสินใจ



รูปที่ 5-24 ไดอะแกรมต้นไม้ตัดสินใจด้วยชุดข้อมูลฝึกสอน (Training set)

สร้างกฎการคำนวณการจำแนกข้อมูลจากแบบจำลองต้นไม้ตัดสินใจออกเป็น 7 กฎ ตามจำนวนกลุ่มความรับผิดของแบบจำลองที่ออกแบบไว้ โดยใช้คำสั่งตามที่แสดงในรูปที่ 5-25 และได้ฟังก์ชันแบบ If_else เพื่อใช้เป็นฟังก์ชันการคำนวณการตัดสินคดี ตามที่แสดงในรูปที่ 5-26

```
1  tree_to_code(dtree,old_features)

def rule_punishment(actor_status, victim, causation, justification, impunity, behavior, intention):
  if intention <= 1.5:
    if justification <= 0.5:
      return [[25.  0.  0.  0.  0.  0.  0.]]
    else:  # if justification > 0.5
      if causation <= 1.5:
        if impunity <= 1.5:
          return [[0. 8. 0. 0. 0. 0. 0.]]
        else:  # if impunity > 1.5
          return [[ 0.  0. 31.  0.  0.  0.  0.]]
      else:  # if causation > 1.5
        if impunity <= 1.5:
          return [[0. 0. 0. 6. 0. 0. 0.]]
        else:  # if impunity > 1.5
          return [[ 0.  0.  0.  0. 25.  0.  0.]]
  else:  # if intention > 1.5
    if causation <= 1.5:
      return [[ 0.  0.  0.  0.  0. 16.  0.]]
    else:  # if causation > 1.5
      return [[ 0.  0.  0.  0.  0.  0. 36.]]
```

รูปที่ 5-25 แสดงตัวอย่างคำสั่งในการสร้างฟังก์ชันการตัดสินของต้นไม้ตัดสินใจ

Rule 1: IF intention <=1.5 AND justification <=0.5 THEN Class = A
Rule 2: IF intention <=1.5 AND justification > 0.5 AND causation <= 1.5 AND impunity <= 1.5 THEN Class = B
Rule 3: IF intention <=1.5 AND justification > 0.5 AND causation > 1.5 AND impunity <= 1.5 THEN Class = C
Rule 4: IF intention <=1.5 AND justification > 0.5 AND causation <= 1.5 AND impunity > 1.5 THEN Class = D
Rule 5: IF intention <=1.5 AND justification > 0.5 AND causation > 1.5 AND impunity > 1.5 THEN Class = E
Rule 6: IF intention >1.5 AND causation <= 1.5 THEN Class = F
Rule 7: IF intention >1.5 AND causation > 1.5 THEN Class = G

รูปที่ 5-26 กฎการพิจารณาที่สกัดจากต้นไม้ตัดสินใจ

5.6 การออกแบบและฝึกฝนแบบจำลองการจำแนกข้อความภาษาธรรมชาติ

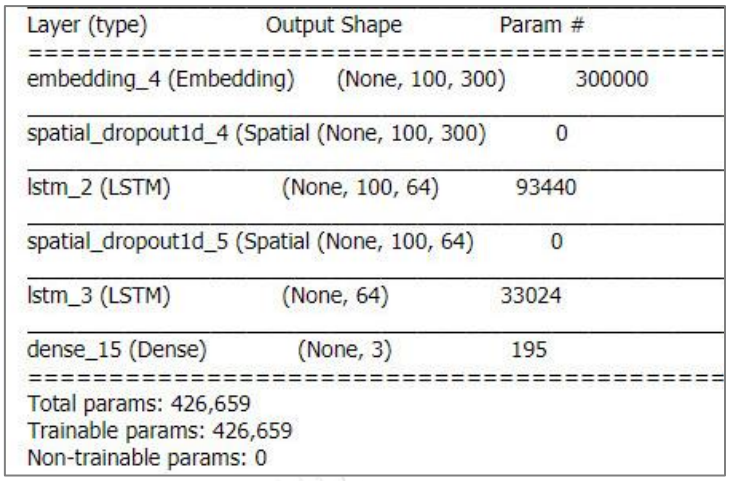
1) หน่วยความจำระยะสั้นแบบยาว (Long-Short Term Memory: LSTM) [15]

ในการฝึกฝนแบบจำลองการจำแนกข้อความภาษาธรรมชาติ ผู้วิจัยแบ่งข้อมูลโดยการสุ่มเป็นข้อมูลสำหรับการฝึกสอน (Training set) 80% ข้อมูลสำหรับทดสอบ (Test set) 20% และใน 80% ของข้อมูลการฝึกสอน ถูกแบ่งเป็นชุดข้อมูลการตรวจสอบระหว่างการฝึกฝนแบบจำลอง 10% โดยในชั้นข้อมูลขาเข้ามีจำนวนนิวรอนเท่ากับ 100 นิวรอน เท่ากับขนาดของความยาว Max_length และมีชั้นซ่อนเป็น LSTM จำนวนสองชั้น ชั้นละ 64 นิวรอน ในแต่ละชั้นมีการป้องกัน Overfitting ด้วยการกำหนดค่า Dropout 20% และกำหนดชั้นซ่อนเพิ่มด้วย SpatialDropout1D จำนวน 2 ชั้น และชั้นข้อมูลส่งออก (Output Layer) มีจำนวนตามคลาสในแต่ละแบบจำลอง ตามรูปที่ 5-27 ซึ่งเป็นแบบจำลอง Causation ที่มี 3 คลาส ก็ใช้จำนวน 3 นิวรอน และใช้ฟังก์ชันกระตุ้นเป็นซอฟต์แมกซ์เพื่อกำหนดค่าความน่าจะเป็นของแต่ละคลาส และแสดงการตั้งค่าไฮเปอร์พารามิเตอร์ ตามรูปที่ 5-28

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
16  def create_model_causation():
17      model_causation_sm = Sequential()
18      model_causation_sm.add(Embedding(MAX_WORDS, EMBEDDING_DIM_CAUSATION, input_length=X_causation.shape[1]))
19      model_causation_sm.add(SpatialDropout1D(0.2))
20      model_causation_sm.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2,return_sequences=True,activation='tanh'))
21      model_causation_sm.add(SpatialDropout1D(0.2))
22      model_causation_sm.add(LSTM(64 , recurrent_dropout=0.2))
23      model_causation_sm.add(Dense(3, activation='softmax'))
24      model_causation_sm.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
25      print('model_causation_sm.summary: ', model_causation_sm.summary())
26      return model_causation_sm
```

รูปที่ 5-27 การออกแบบโครงสร้างและค่าพารามิเตอร์สำหรับแบบจำลอง Causation

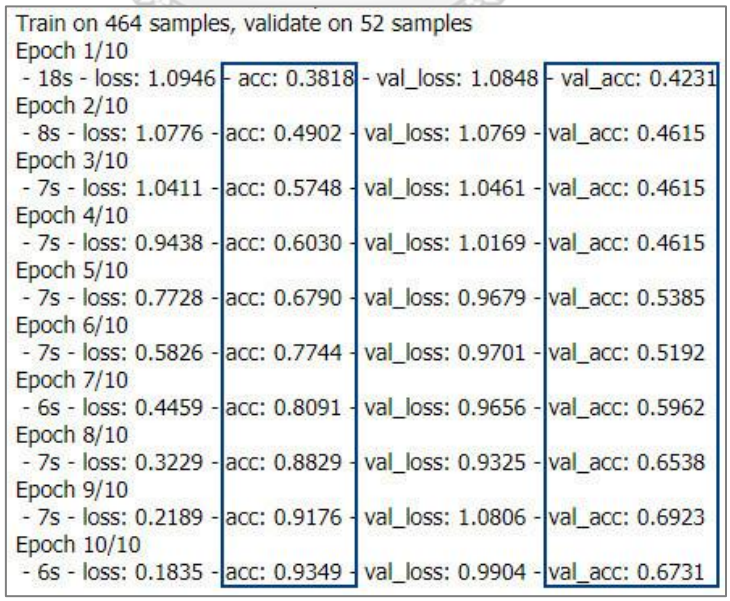


Layer (type)	Output Shape	Param #
embedding_4 (Embedding)	(None, 100, 300)	300000
spatial_dropout1d_4 (Spatial	(None, 100, 300)	0
lstm_2 (LSTM)	(None, 100, 64)	93440
spatial_dropout1d_5 (Spatial	(None, 100, 64)	0
lstm_3 (LSTM)	(None, 64)	33024
dense_15 (Dense)	(None, 3)	195

Total params: 426,659
Trainable params: 426,659
Non-trainable params: 0

รูปที่ 5-28 ค่าพารามิเตอร์ของโครงสร้างการฝึกฝนแบบจำลองด้วย LSTM

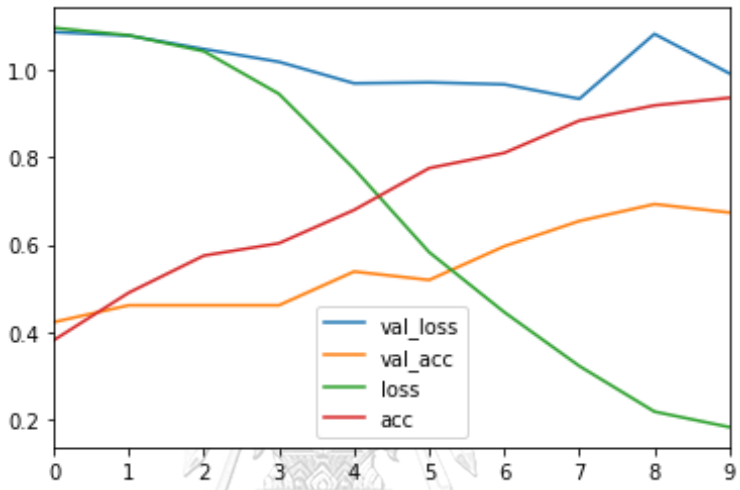
จากรูปที่ 5-28 แสดงค่าพารามิเตอร์ของโครงสร้างแบบจำลอง Causation โดยแบ่งข้อมูลสำหรับใช้ฝึกสอน (Training set) 80% จำนวน 516 ตัวอย่าง ข้อมูลสำหรับทดสอบ (Test set) 20% จำนวน 129 ตัวอย่าง และในระหว่างการฝึกสอนได้แบ่งชุดข้อมูล 10% จากชุด Training set จำนวน 52 ตัวอย่าง ใช้เป็นข้อมูลตรวจสอบ (Validation set) เพื่อวัดค่าความแม่นยำของแบบจำลองในแต่ละรอบการฝึกฝน ทำให้มีข้อมูลฝึกสอนเหลือจำนวน 464 ตัวอย่าง โดยมีค่าความแม่นยำถูกต้องในแต่ละรอบที่เพิ่มขึ้นเมื่อวัดด้วย Training set และ Validation set ตามที่แสดงในรูปที่ 5-29



Train on 464 samples, validate on 52 samples
Epoch 1/10
- 18s - loss: 1.0946 - acc: 0.3818 - val_loss: 1.0848 - val_acc: 0.4231
Epoch 2/10
- 8s - loss: 1.0776 - acc: 0.4902 - val_loss: 1.0769 - val_acc: 0.4615
Epoch 3/10
- 7s - loss: 1.0411 - acc: 0.5748 - val_loss: 1.0461 - val_acc: 0.4615
Epoch 4/10
- 7s - loss: 0.9438 - acc: 0.6030 - val_loss: 1.0169 - val_acc: 0.4615
Epoch 5/10
- 7s - loss: 0.7728 - acc: 0.6790 - val_loss: 0.9679 - val_acc: 0.5385
Epoch 6/10
- 7s - loss: 0.5826 - acc: 0.7744 - val_loss: 0.9701 - val_acc: 0.5192
Epoch 7/10
- 6s - loss: 0.4459 - acc: 0.8091 - val_loss: 0.9656 - val_acc: 0.5962
Epoch 8/10
- 7s - loss: 0.3229 - acc: 0.8829 - val_loss: 0.9325 - val_acc: 0.6538
Epoch 9/10
- 7s - loss: 0.2189 - acc: 0.9176 - val_loss: 1.0806 - val_acc: 0.6923
Epoch 10/10
- 6s - loss: 0.1835 - acc: 0.9349 - val_loss: 0.9904 - val_acc: 0.6731

รูปที่ 5-29 ความแม่นยำของแบบจำลองระหว่างการฝึกฝนในแต่ละรอบของ BiLSTM

เมื่อแบบจำลอง LSTM ผ่านการฝึกฝนกับชุดข้อมูลฝึกสอน (Training set) แล้วแสดงด้วยกราฟเส้น เพื่อดูแนวโน้มค่าความแม่นยำ (Accuracy) และค่าความผิดพลาด (Loss) ในแต่ละรอบการฝึกฝน โดยค่าความแม่นยำ (val_acc) และค่าและความผิดพลาด (val_loss) ที่วัดจากชุดข้อมูลตรวจสอบ (Validation set) ส่วน acc และ loss คือค่าความแม่นยำและความผิดพลาดที่วัดจากชุดข้อมูลฝึกฝน (Training set) ตามที่แสดงในรูปที่ 5-30



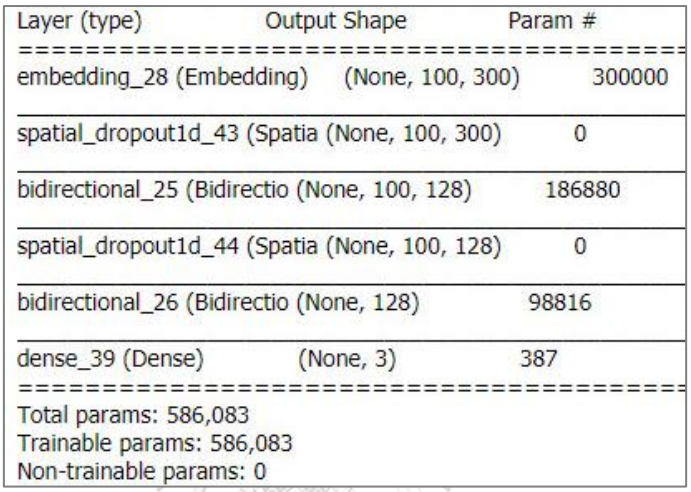
รูปที่ 5-30 แนวโน้มค่าความแม่นยำและค่าความผิดพลาดระหว่างการฝึกฝนของ LSTM

2) หน่วยความจำระยะสั้นขนาดยาวแบบสองทิศทาง (Bi-direction LSTM) [23]

เพื่อปรับปรุงให้แบบจำลองมีความแม่นยำเพิ่มขึ้น โดยการให้การเรียนรู้ไปสองทิศทาง คือจากซ้ายไปขวา และขวาไปซ้ายพร้อม ๆ กัน จึงทำการออกแบบโครงสร้างแบบจำลอง Bi-direction LSTM หรือ BiLSTM โดยการใช้ BiLSTM Layer ในชั้นซ่อน โดยในหนึ่งชั้นของ BiLSTM ประกอบด้วย LSTM แบบไปข้างหน้า (Forward LSTM) จำนวน 64 นิวรอน และ LSTM แบบแบบย้อนกลับ (Backward LSTM) จำนวน 64 นิวรอน โดยออกแบบไว้ใช้สองชั้น และมีค่าพารามิเตอร์ที่สำคัญเช่นเดียวกับใน LSTM ทิศทางเดียว ตามที่แสดงในรูปที่ 5-31 และการตั้งค่าไฮเปอร์พารามิเตอร์แสดงในรูปที่ 5-32 โดยจะพบว่าค่าพารามิเตอร์ที่อยู่ในโครงข่ายของ LSTM แบบสองทิศทางจะมีปริมาณมากกว่า LSTM แบบทิศทางเดียว

```
def create_model_causation_bi():
    model_causation_sm = Sequential()
    model_causation_sm.add(Embedding(MAX_WORDS, EMBEDDING_DIM_CAUSATION, input_length=X_causation.shape[1]))
    model_causation_sm.add(SpatialDropout1D(0.2))
    model_causation_sm.add(Bidirectional(LSTM(100 , dropout=0.2, recurrent_dropout=0.2,return_sequences=True,activation='tanh')))
    model_causation_sm.add(SpatialDropout1D(0.2))
    model_causation_sm.add(Bidirectional(LSTM(100 , recurrent_dropout=0.2)))
    model_causation_sm.add(Dense(3, activation='softmax'))
    model_causation_sm.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print('model_causation_sm.summary: ', model_causation_sm.summary())
    return model_causation_sm
```

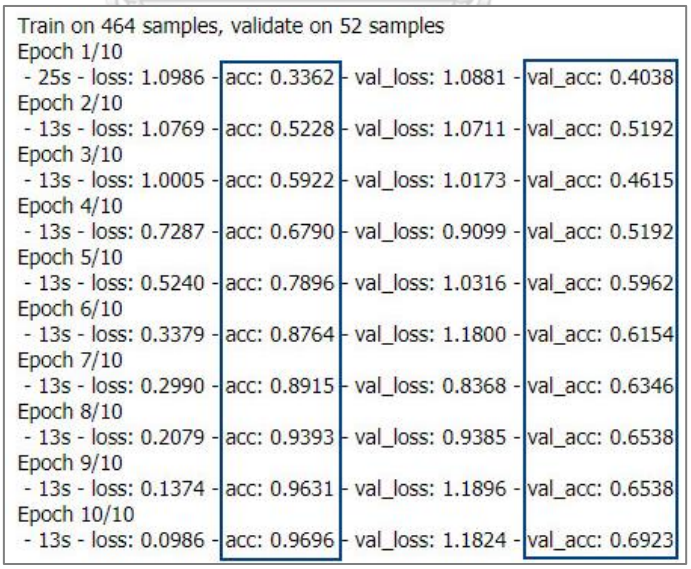
รูปที่ 5-31 การออกแบบโครงสร้างแบบจำลอง BiLSTM



Layer (type)	Output Shape	Param #
embedding_28 (Embedding)	(None, 100, 300)	300000
spatial_dropout1d_43 (Spatia	(None, 100, 300)	0
bidirectional_25 (Bidirectio	(None, 100, 128)	186880
spatial_dropout1d_44 (Spatia	(None, 100, 128)	0
bidirectional_26 (Bidirectio	(None, 128)	98816
dense_39 (Dense)	(None, 3)	387

Total params: 586,083
Trainable params: 586,083
Non-trainable params: 0

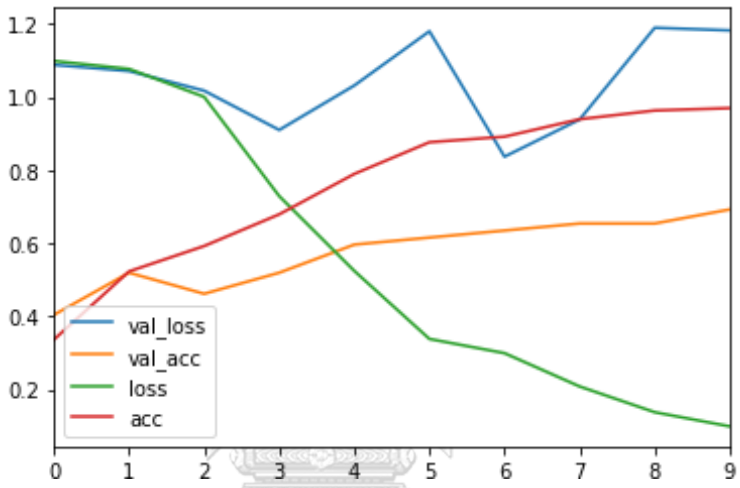
รูปที่ 5-32 ค่าพารามิเตอร์ของโครงสร้างการฝึกฝนแบบจำลองด้วย BiLSTM



```
Train on 464 samples, validate on 52 samples
Epoch 1/10
 - 25s - loss: 1.0986 - acc: 0.3362 - val_loss: 1.0881 - val_acc: 0.4038
Epoch 2/10
 - 13s - loss: 1.0769 - acc: 0.5228 - val_loss: 1.0711 - val_acc: 0.5192
Epoch 3/10
 - 13s - loss: 1.0005 - acc: 0.5922 - val_loss: 1.0173 - val_acc: 0.4615
Epoch 4/10
 - 13s - loss: 0.7287 - acc: 0.6790 - val_loss: 0.9099 - val_acc: 0.5192
Epoch 5/10
 - 13s - loss: 0.5240 - acc: 0.7896 - val_loss: 1.0316 - val_acc: 0.5962
Epoch 6/10
 - 13s - loss: 0.3379 - acc: 0.8764 - val_loss: 1.1800 - val_acc: 0.6154
Epoch 7/10
 - 13s - loss: 0.2990 - acc: 0.8915 - val_loss: 0.8368 - val_acc: 0.6346
Epoch 8/10
 - 13s - loss: 0.2079 - acc: 0.9393 - val_loss: 0.9385 - val_acc: 0.6538
Epoch 9/10
 - 13s - loss: 0.1374 - acc: 0.9631 - val_loss: 1.1896 - val_acc: 0.6538
Epoch 10/10
 - 13s - loss: 0.0986 - acc: 0.9696 - val_loss: 1.1824 - val_acc: 0.6923
```

รูปที่ 5-33 ความแม่นยำของแบบจำลองระหว่างการฝึกฝนในแต่ละรอบของ BiLSTM

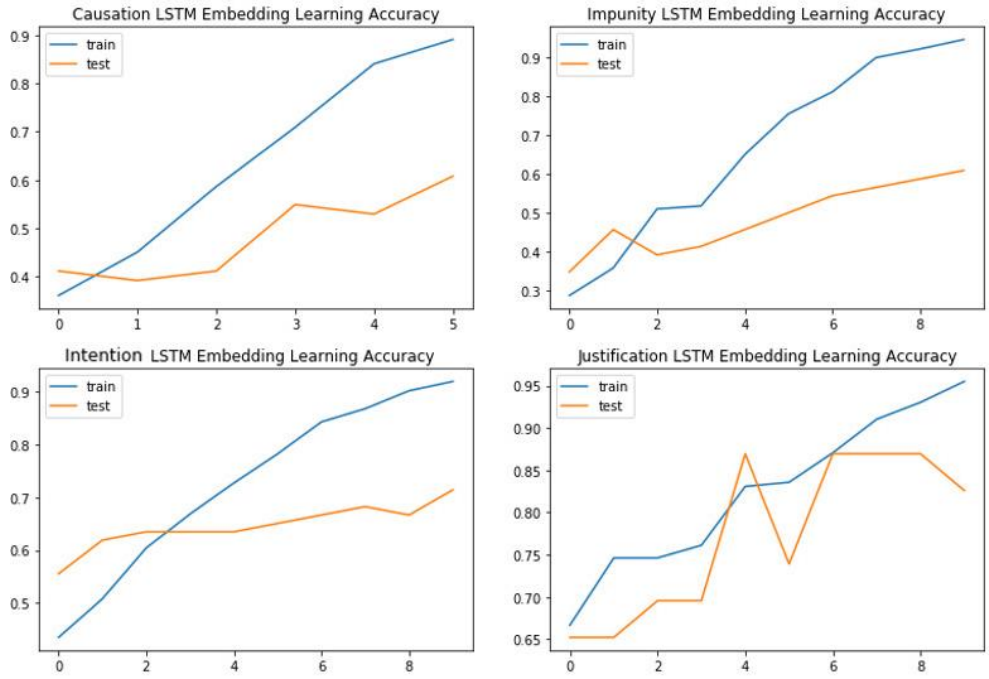
เมื่อแบบจำลอง BiLSTM ผ่านการฝึกฝนกับชุดข้อมูลฝึกสอน Training set แล้วแสดงด้วยกราฟเส้น เพื่อดูแนวโน้มค่าความแม่นยำ (Accuracy) และค่าความผิดพลาด (Loss) ในแต่ละรอบการฝึกฝน โดย val_acc และ val_loss คือค่าความแม่นยำและความผิดพลาดที่วัดจากชุดข้อมูลทดสอบ (Validation set) ส่วน acc และ loss คือค่าความแม่นยำและความผิดพลาดที่วัดจากชุดข้อมูลฝึกฝน (Training set) ตามที่แสดงในรูป ที่ 5-34 และพบว่าแนวโน้มของ BiLSTM มีค่าความแม่นยำเพิ่มมากขึ้นเมื่อเทียบกับ LSTM



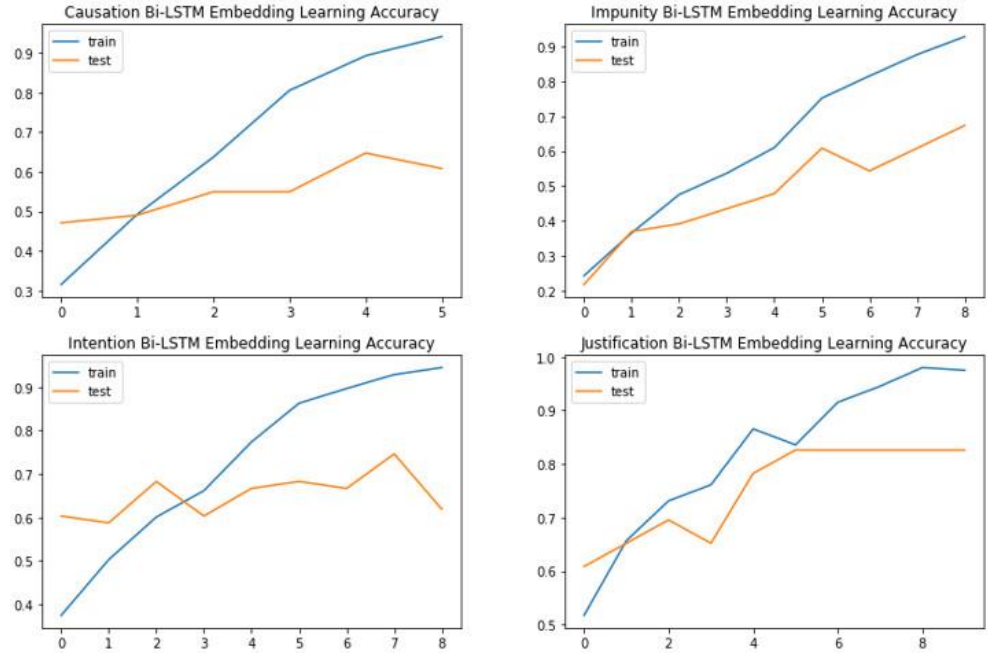
รูปที่ 5-34 แนวโน้มค่าความแม่นยำและค่าความผิดพลาดระหว่างการฝึกฝนของ BiLSTM

และผู้วิจัยก็ทำการฝึกฝนครบทั้ง 4 แบบจำลอง ด้วยการตั้งค่าไฮเปอร์พารามิเตอร์ต่าง ๆ ในลักษณะเดียวกัน ทั้ง LSTM แบบทิศทางเดียวและสองทิศทาง (BiLSTM) ในระหว่างการฝึกฝนก็ทำการแบ่งชุดข้อมูลตรวจสอบ (Validation set) ออกจากชุดข้อมูลฝึกสอน (Training set) จำนวน 10% ในทุกแบบจำลอง และแสดงกราฟแนวโน้มค่าความแม่นยำของทั้ง 4 แบบจำลอง เปรียบเทียบระหว่าง LSTM และ BiLSTM ตามที่แสดงในรูปที่ 5-35 และ 5-36

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 5-35 นวโน้มค่าความแม่นยำระหว่างการฝึกฝนแบบจำลอง LSTM ทั้ง 4 แบบจำลอง



รูปที่ 5-36 แนวโน้มค่าความแม่นยำระหว่างการฝึกฝนแบบจำลอง BiLSTM ทั้ง 4 แบบจำลอง

5.7 การเปรียบเทียบประสิทธิภาพแบบจำลอง LSTM ทิศทางเดียว และสองทิศทาง (BiLSTM)

แบบจำลองทั้ง 4 ประกอบด้วย แบบจำลองที่เป็นไบนารีคลาส 1 แบบจำลองคือ Justification เพื่อจำแนกว่าผิดหรือไม่ และอีก 3 แบบจำลอง ประกอบด้วย Intention, Impunity และ Causation เป็นแบบจำลองแบบมัลติคลาส ดังนั้นในการเปรียบเทียบประสิทธิภาพด้วยค่าความแม่นยำ (Accuracy) ไม่เพียงพอ งานวิจัยนี้ใช้เปรียบเทียบด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) ตามที่แสดงในรูปที่ 5-37 และรูปที่ 5-38

Classification Causation Lstm model Report:

	precision	recall	f1-score	support
0	0.76	0.56	0.64	36
1	0.64	0.55	0.59	49
2	0.55	0.77	0.64	43
accuracy			0.62	129
macro avg	0.65	0.63	0.62	129
weighted avg	0.65	0.63	0.62	129

Classification Impunity_LSTM model Report:

	precision	recall	f1-score	support
0	0.80	0.62	0.70	32
1	0.69	0.71	0.70	27
2	0.73	0.70	0.71	27
3	0.70	0.70	0.70	28
accuracy			0.70	114
macro avg	0.73	0.68	0.70	114
weighted avg	0.73	0.68	0.70	114

Classification Intention LSTM Report:

	precision	recall	f1-score	support
0	0.86	0.76	0.81	58
1	0.70	0.91	0.79	46
2	0.70	0.69	0.69	52
accuracy			0.76	156
macro avg	0.75	0.79	0.76	156
weighted avg	0.75	0.78	0.76	156

Classification Justification LSTM Report:

	precision	recall	f1-score	support
0	0.81	0.80	0.81	29
1	0.80	0.80	0.80	27
accuracy			0.80	56
macro avg	0.81	0.80	0.81	56
weighted avg	0.81	0.80	0.81	56

รูปที่ 5-37 ค่าเฉลี่ย Macro-average F1 Score แบบจำลอง LSTM

Classification Causation Bi-Lstm model Report:

	precision	recall	f1-score	support
0	0.91	0.83	0.87	36
1	0.72	0.73	0.72	49
2	0.66	0.72	0.69	43
accuracy			0.76	129
macro avg	0.76	0.76	0.76	129
weighted avg	0.76	0.76	0.76	129

Classification Impunity_Bi-LSTM model Report:

	precision	recall	f1-score	support
0	0.73	0.62	0.67	32
1	0.76	0.89	0.82	27
2	0.72	0.70	0.70	27
3	0.80	0.86	0.83	28
accuracy			0.76	114
macro avg	0.75	0.77	0.76	114
weighted avg	0.75	0.77	0.76	114

Classification Intention Bi-LSTM Report:

	precision	recall	f1-score	support
0	0.88	0.79	0.84	58
1	0.76	0.91	0.83	46
2	0.80	0.75	0.77	52
accuracy			0.81	156
macro avg	0.81	0.82	0.81	156
weighted avg	0.82	0.81	0.81	156

Classification Justification LSTM Report:

	Precision	recall	f1-score	suport
0	0.91	0.90	0.90	29
1	0.88	0.96	0.92	27
accuracy			0.91	56
macro avg	0.90	0.93	0.91	56
weighted avg	0.90	0.93	0.91	56

รูปที่ 5-38 ค่าเฉลี่ย Macro-average F1 Score แบบจำลอง BiLSTM

จากการเปรียบเทียบประสิทธิภาพของแบบจำลอง LSTM และ BiLSTM ตามรูปที่ 5-37 และ 5-38 ด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) ในแบบจำลองทั้ง 4 พบว่า แบบจำลองที่ฝึกฝนด้วย BiLSTM มีค่ามากกว่า LSTM แบบทิศทางเดียว

5.8 การปรับปรุงประสิทธิภาพของแบบจำลองด้วย Pretrained Word Embedding

งานวิจัยนี้มีชุดข้อมูลสำหรับการทำวิจัย (Data set) จำนวนไม่มากพอ สำหรับการเรียนรู้เชิงลึก (Deep Learning) ที่ประสิทธิภาพของแบบจำลองปัจจัยหนึ่ง คือปริมาณข้อมูลที่ใช้ในการฝึกฝนแบบจำลอง และการใช้แบบจำลองทำการฝึกฝนจากข้อมูลที่มีในงานวิจัย อาจทำให้ประสิทธิภาพของแบบจำลองมีค่าต่ำกว่าเมื่อเทียบกับข้อมูลที่มีจำนวนมาก การถ่ายโอนการเรียนรู้ (Transfer Learning) [32] เป็นเทคนิคที่ถูกนำมาใช้ในงานวิจัยนี้ โดยการใช้ Pretrained Word Embedding ที่ผ่านการเรียนรู้ลักษณะโครงสร้างทางภาษา (Language Model) จากคลังข้อมูลอื่น (Corpus) ที่มีขนาดใหญ่มาแล้ว มาเป็นค่าถ่วงน้ำหนัก (Weight) แก่แบบจำลองในชั้นซ่อนแรกของโครงสร้างประสาทแทนการสุ่มค่าเริ่มต้น ส่งผลให้ลดระยะเวลาในการฝึกฝน และทำให้ประสิทธิภาพของแบบจำลองมีค่าสูงขึ้น ดังนี้

1) การใช้ FastText

FastText [34] เป็นของห้องวิจัยเฟสบุค (Facebook) ที่ใช้ข้อมูลฝึกฝนจาก Wikipedia ขนาด 2,000,000 embedding 300 มิติ โดยสามารถทำการโหลดค่าถ่วงน้ำหนักมากำหนดค่าเริ่มต้นให้แบบจำลองได้ ตามตัวอย่างคำสั่งในรูปที่ 5-39 โดยมีตัวอย่างคำสั่งเริ่มต้นการทำงานคล้ายการฝึกฝนจากชุดข้อมูล แต่ต่างกันที่การกำหนดค่าถ่วงน้ำหนักในชั้นซ่อนแรกจะถูกกำหนดให้เป็นค่าจากค่าถ่วงน้ำหนักของ FastText ตามที่แสดงในรูปที่ 5-40

Chulalongkorn University

```
def load_fasttext_causation(word_index=word_index_causation , max_words=MAX_WORDS, embed_size=300):
    EMBEDDING_FILE = 'data/pre_trained/fasttext/cc.th.300.vec'
    emb_mean, emb_std = -0.0033470048, 0.109855264
    embedding_matrix_fast = np.random.normal(emb_mean, emb_std, (max_words, embed_size))
    with open(EMBEDDING_FILE, 'r', encoding="utf8" ) as f:
        for line in f:
            if len(line) <= 100:
                continue
            word, vec = line.split(' ', 1)
            if word not in word_index:
                continue
            i = word_index[word]
            if i >= max_words:
                continue
            embedding_vector = np.asarray(vec.split(' '), dtype='float32')[:300]
            if len(embedding_vector) == 300:
                embedding_matrix_fast[i] = embedding_vector
    return embedding_matrix_fast
```

รูปที่ 5-39 การโหลดค่าถ่วงน้ำหนักของ FastText

```
def create_model_fasttext_causation():
    model_causation = Sequential()
    model_causation.add(Embedding(MAX_WORDS, EMBEDDING_DIM_CAUSATION, weights=[embedding_matrix_causation_fast],input_length=X_cau
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ,return_sequences=True, activation='tanh' ))
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ))
    model_causation.add(Dense(3, activation='softmax'))
    model_causation.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print('model_Causation.summary: ', model_causation.summary())
    return model_causation
```

รูปที่ 5-40 การกำหนดโครงสร้างแบบจำลองที่ใช้ FastText

2) การใช้ Thai2Vec

Thai2Vec [35] ของกลุ่ม PyThaiNLP ที่ฝึกฝนด้วยข้อมูลจาก Thai Wikipedia จำนวน 60,000 embedding ขนาด 300 มิติ โดยสามารถทำการโหลดค่าถ่วงน้ำหนักมากำหนดค่าเริ่มต้นให้แบบจำลองได้ ตามตัวอย่างคำสั่งในรูปที่ 5-41 โดยมีตัวอย่างคำสั่งเริ่มต้นการทำงานคล้ายการฝึกฝนจากชุดข้อมูลในงานวิจัย (Data set) แต่ต่างกันที่การกำหนดค่าถ่วงน้ำหนักในชั้นซ่อนแรกจะถูกกำหนดให้เป็นค่าจากค่าถ่วงน้ำหนักของ Thai2Vec ตามที่แสดงในรูปที่ 5-42

```
def load_thai2vec_causation(word_index=word_index_causation , max_words=MAX_WORDS, embed_size=300):
    EMBEDDING_FILE = 'data/pre_trained/thai2vec/thai2vec.vec'
    emb_mean, emb_std = -0.0033470048, 0.109855264
    embedding_matrix_thai = np.random.normal(emb_mean, emb_std, (max_words, embed_size))
    with open(EMBEDDING_FILE, 'r', encoding="utf8" ) as f:
        for line in f:
            if len(line) <= 100:
                continue
            word, vec = line.split(' ', 1)
            if word not in word_index:
                continue
            i = word_index[word]
            if i >= max_words:
                continue
            embedding_vector = np.asarray(vec.split(' '), dtype='float32')[:300]
            if len(embedding_vector) == 300:
                embedding_matrix_thai[i] = embedding_vector
    return embedding_matrix_thai
```

รูปที่ 5-41 การโหลดค่าถ่วงน้ำหนักของ Thai2Vec

```
def create_model_thai_causation():
    model_causation = Sequential()
    model_causation.add(Embedding(MAX_WORDS, EMBEDDING_DIM_CAUSATION,weights=[embedding_matrix_causation_thai],input_length=X_cau
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ,return_sequences=True, activation='tanh' ))
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ))
    model_causation.add(Dense(3, activation='softmax'))
    model_causation.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print('model_Causation.summary: ', model_causation.summary())
    return model_causation
```

รูปที่ 5-42 การกำหนดโครงสร้างแบบจำลองที่ใช้ Thai2Vec

3) การใช้ Crime2Vec

ค่าถ่วงน้ำหนักเริ่มต้นที่ได้จาก FastText และ Thai2Vec ถูกฝึกสอนด้วยบริบทที่ต่างจากงานวิจัยนี้ ดังนั้นเพื่อให้การเข้าใจลักษณะโครงสร้างทางภาษาไปในทางเดียวกัน ผู้วิจัยจึงได้รวบรวมข่าวอาชญากรรม จากแหล่งข่าวออนไลน์จำนวน 504 ข่าว จำนวน 2,676 คำ เพื่อฝึกสอนแก่แบบจำลองภาษา (Language Model) ด้วย Gensim ที่เป็นไลบรารีของ Word2Vec [36] และสร้างค่าถ่วงน้ำหนักเริ่มต้นบันทึกเป็นไฟล์ .Vec และการใช้งานก็สามารถดาวน์โหลดค่าถ่วงน้ำหนัก เพื่อกำหนดค่าเริ่มต้นให้แบบจำลองในงานวิจัยได้ โดยใช้คำสั่งตามที่แสดงในรูปที่ 5-43

```
from gensim.models import Word2Vec
model = Word2Vec(content_lines, min_count=3,size= 300,workers=4, window =5, sg=1)
#Vocab size
words = list(model.wv.vocab)
print('Vocab size : ', len(words))
```

```
Vocab size :  2676
```

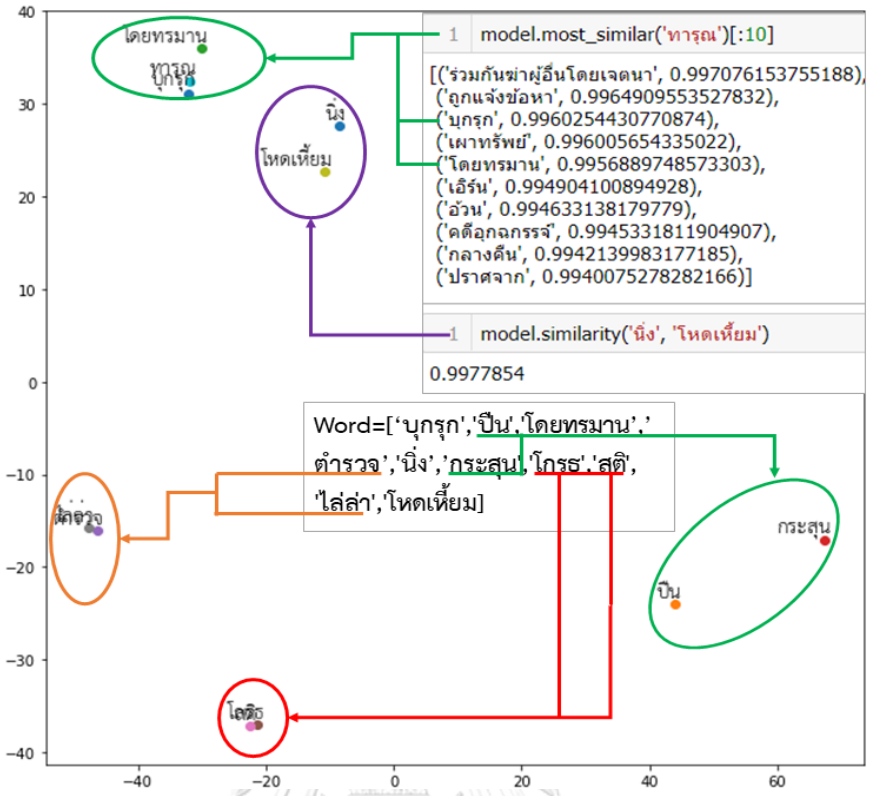
รูปที่ 5-43 คำสั่งการฝึกฝนแบบจำลองภาษา (Language Model)

ความสัมพันธ์ของแต่ละคำสามารถดูได้จากการลดมิติข้อมูลจาก 300 มิติ เป็น 2 มิติ เพื่อแสดงผลความสัมพันธ์ของแต่ละคำในแนวระนาบแกน X และ Y ด้วย t-SNE ตามที่แสดงในรูปที่ 5-44 และพบว่าแบบจำลองที่บันทึกไว้จากการฝึกฝนชุดข้อมูลที่เกี่ยวกับคดีอาชญากรรม (Word2Vec.model) สามารถเข้าใจโครงสร้างของคำ เมื่อทำการทดสอบคำแต่ละคำที่มีความเกี่ยวข้องกัน ตามที่แสดงในรูปที่ 5-45

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
sample_words = ['บุกรุก','ปืน','โดยทรมาน','กระสุน','ตำรวจ','โกรธ','สติ','ไล่ล่า','โหดเหี้ยม','ทารุณ','นิ่ง']
sample_idx = []
for word in sample_words:
    sample_idx.append(labels.index(word))
sample_plot = crime2plot[sample_idx]
plot_with_labels(sample_plot,sample_words,f'{MISC_PATH}word_arithematic1.png')
plt.show()
```

รูปที่ 5-44 คำสั่งการแสดงแบบจำลองทางภาษา



รูปที่ 5-45 ความสัมพันธ์ทางความหมายของแต่ละคำในแบบจำลองทางภาษา

จากรูปที่ 5-45 แสดงให้เห็นว่าคำที่มีความหมายใกล้เคียงกันจะอยู่ในตำแหน่งที่ใกล้เคียงกัน เช่น คำว่า “ตำรวจ” อยู่ใกล้คำว่า “ไล่ล่า” คำว่า “ปืน” อยู่ใกล้คำว่า ”กระสุน” หรือถ้าคำนวณค่าความใกล้เคียงของคำเช่น คำว่า “ทารุณ” จะมีค่าความคล้ายกับคำว่า “ร่วมกันฆ่าผู้อื่นโดยเจตนา” 99.7% เป็นต้น

ทำการโหลดค่าถ่วงน้ำหนัก (Weight) ที่ได้จากการฝึกฝนข้อมูลข่าวอาชญากรรม เพื่อกำหนดเป็นค่าเริ่มต้นให้กับแบบจำลองการจำแนกข้อความตามที่แสดงในรูปที่ 5-46 และ 5-47

```
def load_crime2vec_causation(word_index=word_index_causation , max_words=MAX_WORDS, embed_size=300):
    EMBEDDING_FILE = 'data/pre_trained/crime2vec/crime2vec.vec'
    emb_mean, emb_std = -0.0033470048, 0.109855264
    embedding_matrix_thai = np.random.normal(emb_mean, emb_std, (max_words, embed_size))
    with open(EMBEDDING_FILE, 'r', encoding="utf8" ) as f:
        for line in f:
            if len(line) <= 100:
                continue
            word, vec = line.split(' ', 1)
            if word not in word_index:
                continue
            i = word_index[word]
            if i >= max_words:
                continue
            embedding_vector = np.asarray(vec.split(' '), dtype='float32')[:300]
            if len(embedding_vector) == 300:
                embedding_matrix_thai[i] = embedding_vector
    return embedding_matrix_thai
```

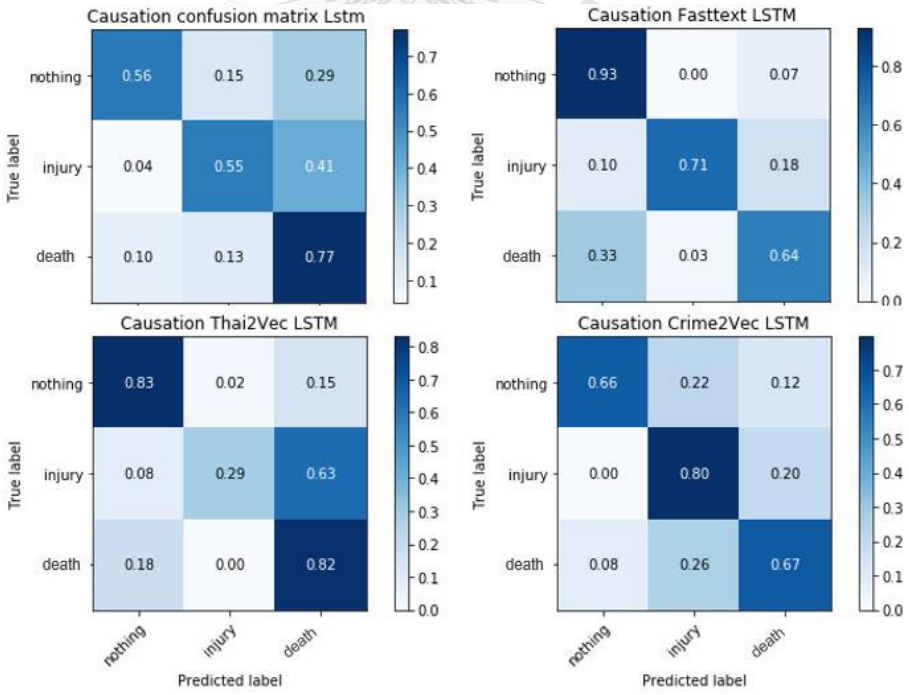
รูปที่ 5-46 การโหลดค่าถ่วงน้ำหนักของ Crime2Vec

```
def create_model_crime_causation():
    model_causation = Sequential()
    model_causation.add(Embedding(MAX_WORDS, EMBEDDING_DIM_CAUSATION,weights=[embedding_matrix_causation_crime],input_length=X_ca
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ,return_sequences=True, activation='tanh' ))
    model_causation.add(SpatialDropout1D(0.2))
    model_causation.add(LSTM(64 , dropout=0.2, recurrent_dropout=0.2 ))
    model_causation.add(Dense(3, activation='softmax'))
    model_causation.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print('model_Causation.summary: ', model_causation.summary())
    return model_causation
```

รูปที่ 5-47 การกำหนดโครงสร้างแบบจำลองที่ใช้ Crime2Vec

5.9 การวัดประสิทธิภาพของแบบจำลองที่ใช้เทคนิคการถ่ายโอนความรู้

ในการวัดประสิทธิภาพของแบบจำลองที่ผ่านการฝึกฝน โดยใช้เทคนิคการถ่ายโอนความรู้จากคลังข้อมูลขนาดใหญ่อื่น ๆ เพื่อให้แบบจำลองเข้าใจรูปแบบลักษณะทางภาษา ที่ผ่านฝึกฝนด้วยชุดข้อมูลจำนวนมากแบบไม่มีผู้สอน (Unsupervised Learning) แล้วนำมาเป็นค่าถ่วงน้ำหนักเริ่มต้นให้กับแบบจำลองที่นำเสนอ โดยเป็นการเปรียบเทียบจากสอง Pretrained Word embedding ที่ผ่านผ่านการเทรนด้วยข้อมูลจาก Thai Wikipedia และหนึ่ง Pretrained Word Embedding ที่ผ่านการเทรนจากแบบจำลองภาษาด้วยชุดข้อมูลด้านข่าวอาชญากรรมออนไลน์ โดยประเมินประสิทธิภาพของแบบจำลองทั้ง 4 ด้วยตารางคอนฟิวชันเมตริกซ์ (Confusion Metrics) [33] ที่ปรับให้อยู่ในรูปเปอร์เซ็นต์ที่ทำนายได้ถูกต้องในแต่ละคลาส และวัดประสิทธิภาพแต่ละแบบจำลองด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) ดังแสดงในรูปที่ 5-48



รูปที่ 5-48 ประสิทธิภาพแบบจำลอง Causation ที่ฝึกฝนด้วย LSTM

จากรูปที่ 5-48 เป็นการวัดประสิทธิภาพของแบบจำลองด้วยตารางคอนฟิวชันเมทริกซ์ ในแบบจำลอง Causation ที่ฝึกฝนด้วย LSTM และทำการประเมินด้วยชุดข้อมูลทดสอบ พบว่าแบบจำลองที่ใช้ Pretrain Word Embedding สามารถทำนายได้ถูกต้องคิดเป็นเปอร์เซ็นต์ได้แม่นยำว่าแบบจำลองที่ฝึกฝนจากการสุ่มค่าถ่วงน้ำหนักเริ่มต้น โดยทั้ง 3 เทคนิคสามารถทำนายได้แม่นยำมากกว่า ดังนี้ FastText และ Crim2Vec ทำนายได้แม่นยำกว่าในคลาส nothing และ injury ส่วน Thai2Vec ทำนายได้แม่นยำกว่าในคลาส nothing และ death และค่าฉลี่ยของ F1 ของแบบจำลองทั้ง 4 ที่ผ่านการฝึกฝนด้วย LSTM แบบทิศทางเดียวเปรียบเทียบการใช้ Pretrained Word Embedding ทั้ง 3 แบบ กับการสุ่มค่าเริ่มต้น (Word Embedding) สามารถสรุปได้ตามตารางที่ 5-12

ตารางที่ 5-12 เปรียบเทียบประสิทธิภาพแบบจำลอง LSTM ด้วยค่าเฉลี่ยมหภาคเอฟวัน

Models	Word Embedding	FastText	Thai2Vec	Crime2Vec
Causation	62.40	75.50	60.69	71.32
Impunity	70.25	72.63	33.40	68.79
Intention	76.33	78.47	68.75	75.01
Justification	80.68	94.71	78.94	92.87
Average	**72.41**	**80.33**	**69.45**	**76.99**

จากตางรางเปรียบเทียบที่ 5-12 พบว่าแบบจำลองมัลติคลาส ได้แก่ Causation, Impunity และ Intention ของ FastText มีค่าเฉลี่ย F1 มากที่สุด คิดเป็นเปอร์เซ็นต์เท่ากับ 75.5% , 72.63% และ 78.47% ตามลำดับ ส่วนในแบบจำลองไบนารีคลาส Justification ค่าเฉลี่ย F1 มากที่สุด คือ Crim2Vec 98.24% และทำการเปรียบเทียบด้วยการใช้ BiLSTM ตามตารางที่ 5-13

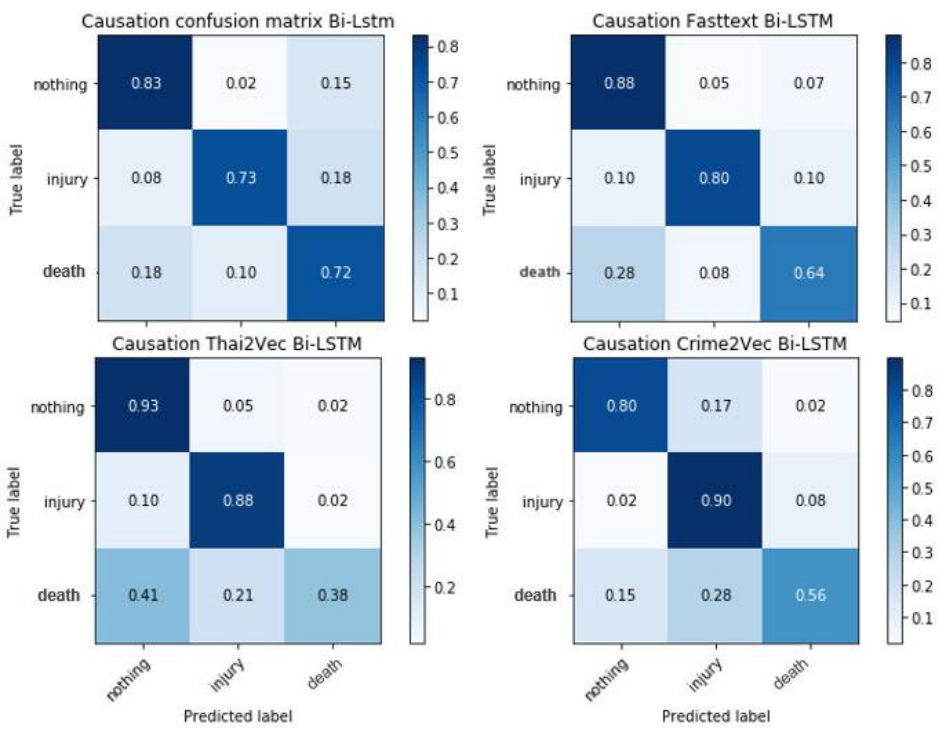
ตารางที่ 5-13 เปรียบเทียบประสิทธิภาพแบบจำลอง BiLSTM ด้วยค่าเฉลี่ยมหภาคเอฟวัน

Models	Word Embedding	FastText	Thai2Vec	Crime2Vec
Causation	75.79	78.46	71.30	75.81
Impunity	76.48	79.11	52.01	76.33
Intention	81.09	82.08	67.36	75.79
Justification	91.13	98.24	87.58	98.24
Average	**81.12**	**84.47**	**69.56**	**81.54**

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตางรางเปรียบเทียบที่ 5-13 พบว่าแบบจำลองมัลติคลาส (Multiclass Classification) ได้แก่ Causation, Impunity และ Intention ของ FastText มีค่าเฉลี่ย F1 มากที่สุด คิดเป็นเปอร์เซ็นต์เท่ากับ 78.46% , 79.11% และ 82.08% ตามลำดับ ส่วนในแบบจำลองไบนารีคลาส Justification ค่าเฉลี่ย F1 มากที่สุดมีค่าเท่ากันสองเทคนิคคือ Crim2Vec และ FastText คิดเป็น 98.24%

นอกจากนี้เมื่อฝึกฝนด้วยแบบจำลอง BiLSTM โดยใช้แบบจำลอง Causation ผลการทดลองของทั้ง 3 Pretrained Word Embedding เทียบกับการด้วยค่าสุ่ม ตามที่แสดงในรูปที่ 5-49 โดยทั้ง 3 เทคนิคสามารถทำนายได้แม่นยำมากกว่า ดังนี้ FastText และ Thai2Vec ทำนายได้แม่นยำกว่าในคลาส nothing และ injury แต่ในคลาส death น้อยกว่า ส่วน Crime2Vec ทำนายได้แม่นยำกว่าในคลาส injury เท่านั้น แต่ในคลาส nothing และ death น้อยกว่า แต่อย่างไรก็ตามค่าเฉลี่ย F1 ของ Crime2Vec ก็ยังมีค่ามากกว่า BiLSTM ที่ฝึกฝนจากการสุ่มค่าเริ่มต้นด้วยชุดข้อมูลที่ใช้ในงานวิจัย



รูปที่ 5-49 ประสิทธิภาพแบบจำลอง Causation ที่ฝึกฝนด้วย BiLSTM

และสามารถสรุปค่าความแม่นยำของแบบจำลองทั้ง 4 ที่ฝึกฝนด้วย LSTM และ BiLSTM ด้วยการเรียนรู้ผ่านการฝังคำ (Word Embedding) และการใช้ Pretrained Word Embedding ได้แก่ FastText, Thai2Vec และ Crime2Vec ด้วยการวัดค่าความแม่นยำ (Accuracy) และค่าเฉลี่ยมหภาคของเอฟวัน

ตามที่แสดงในตารางที่ 5-14 และตารางคอนฟิวชันเมตริกซ์ของทุกแบบจำลอง ในแต่ละเทคนิคการฝึกฝนทั้งหมด รายละเอียดตามที่แสดงในภาคผนวก ก.

ตารางที่ 5-14 การเปรียบเทียบค่าความแม่นยำและค่าเฉลี่ยมหภาคเอฟวันของทุกแบบจำลอง

Features		Models							
		Intention		Causation		Impunity		Justification	
		Acc.	F-1	Acc.	F-1	Acc.	F-1	Acc.	F-1
LSTM	Word Embedding	76.48	76.33	55.55	62.40	55.55	70.25	80.70	80.68
	FastText	78.98	78.47	75.97	75.50	73.28	72.63	94.74	94.71
	Thai2Vec	68.79	68.75	62.02	60.69	33.62	33.40	78.95	78.94
	Crime2Vec	75.16	75.01	71.32	71.32	68.97	68.79	92.98	92.87
BiLSTM	Word Embedding	81.53	81.09	75.97	75.79	76.72	76.48	91.23	91.13
	FastText	82.17	82.08	77.52	78.46	79.45	79.11	98.25	98.24
	Thai2Vec	66.88	67.36	74.42	71.30	54.31	52.01	87.72	87.58
	Crime2Vec	75.80	75.79	76.74	75.81	76.72	76.33	98.25	98.24

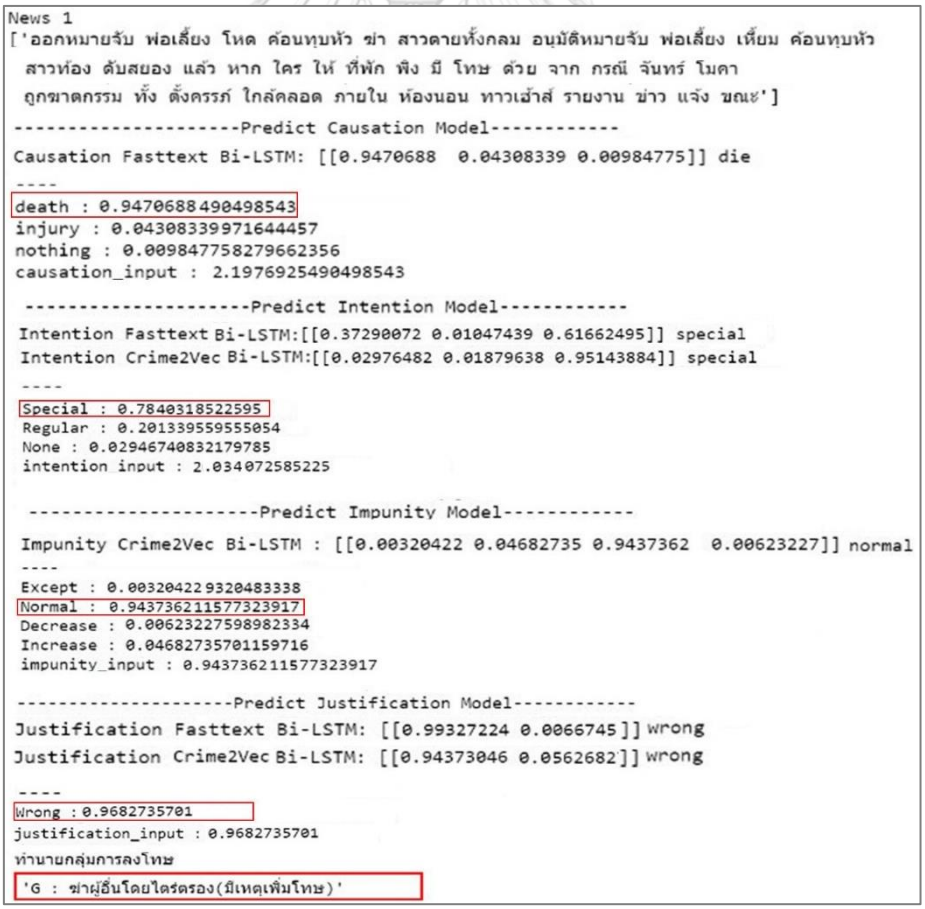
จากตารางสรุปผลการทดลองที่ 5-14 พบว่าเมื่อทดสอบกับชุดข้อมูลทดสอบ (Test set) พบว่าค่าเฉลี่ยความแม่นยำ (Accuracy) ที่มีค่ามากที่สุด 2 อันดับแรกคือการฝึกฝนด้วย LSTM แบบสองทิศทาง (BiLSTM) ได้แก่ FastText 84.35% และ Crime2Vec 81.88% และค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) ที่มีค่ามากที่สุด 2 อันดับแรกก็เช่นเดียวกัน คือแบบจำลองที่ผ่านการฝึกฝนด้วย LSTM แบบสองทิศทาง (BiLSTM) ได้แก่ FastText 84.47% และ Crime2Vec 81.54% ตามลำดับ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.10 การวัดประสิทธิภาพแบบจำลองเมื่อใช้ทำนายข่าวอาชญากรรม

แม้จะพบว่าแบบจำลองที่ให้ค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) แบบที่ใช้ Pretrained Word Embedding ด้วย BiLSTM ให้ค่าที่สูงกว่า แต่ในบางคลาสที่สำคัญแบบจำลองอื่นให้ค่าที่สูงกว่าได้เช่นกันเช่น แบบจำลอง Causation ในรูปที่ 5-49 ที่ฝึกฝนด้วย LSTM พบว่าแบบจำลองที่ใช้ Word Embedding ทำนายว่าตาย (death) ได้ถูกต้องตามค่าจริง 77% ขณะที่แบบจำลองที่ใช้ FastText ถูกต้อง 64% หรือในรูปที่ 5-14 ที่ฝึกฝนด้วย BiLSTM พบว่าแบบจำลอง Causation แบบจำลองที่ใช้ Word Embedding ทำนายว่าตาย (death) ได้ถูกต้องตามค่าจริง 72% ขณะที่แบบจำลองที่ใช้ FastText ถูกต้อง 64%

ดังนั้นเพื่อให้การทำนายมีความถูกต้องแม่นยำ งานวิจัยนี้จึงใช้เทคนิคซอฟต์โหวด Soft-Voting [42] ซึ่งเป็นการหาค่าเฉลี่ยของการทำนายแบบจำลอง จากนั้นกำหนดเป็นข้อมูลขาเข้าต้นไม้ตัดสินใจ โดยใช้ความเห็นของนักกฎหมายเป็นค่าเฉลยคำตอบในแต่ละข่าวที่นำมาใช้ทดสอบ โดยใช้จำนวน 100 ข่าวทดสอบ ดังที่แสดงตัวอย่างในรูปที่ 5-50 โดยใช้เนื้อหาข่าวจากเว็บไซต์ข่าวออนไลน์ [46] วันที่ 12 เม.ย.62 ดังนี้ “ออกหมายจับพ่อเลี้ยงโหด ค้อนทุบหัวฆ่าสาวตายทั้งกลม ศาลอนุมัติหมายจับพ่อเลี้ยงเหี้ยม ค้อนทุบหัวสาวท้อง 7 เดือนดับสยองแล้ว และหากใครให้ที่พักพิงมีโทษด้วย จากกรณี น.ส.นันทชยา จันทร์โมคา อายุ 29 ปี ถูกฆาตกรรมทั้งที่ตั้งครรภ์ใกล้คลอด ภายในห้องนอนทาวเฮ้าส์เลขที่ 668/64 ซอยรามอินทรา109 แขวงบางชัน เขตคลองสามวา กรุงเทพมหานคร โดยจากการตรวจสอบพบว่า บริเวณศีรษะถูกทุบด้วยค้อน และยังถูกทุบตีบริเวณลำตัวทำให้เด็กเสียชีวิตในครรภ์ หลังเกิดเหตุ นายภูมใจ เหลืองทอง อายุ 57 ปี พ่อเลี้ยงของผู้ตายที่พักอาศัยอยู่ด้วยกันหายตัวไป เหตุเกิดเมื่อวันที่ 11 เม.ย.62” และข่าวนี้ตำรวจได้ขออนุมัติหมายจับจากศาลจังหวัดมีนบุรี ที่ 279/2562 ลงวันที่ 12 เม.ย.62 ในข้อหา "ฆ่าผู้อื่นโดยทรมานหรือโดยกระทำทารุณโหดร้าย" ตามประมวลกฎหมายอาญา มาตรา 289(5) ต้องระวางโทษประหารชีวิต



รูปที่ 5-50 การคำนวณข้อมูลนำเข้าของแต่ละแบบจำลองและทำนายกลุ่มการลงโทษ

งานวิจัยนี้ผู้วิจัยได้เลือกแบบจำลองที่ให้ค่าความแม่นยำมากที่สุดในแต่ละแบบจำลองร่วมกับการใช้เทคนิคซอฟต์โหวต (Soft Voting) เพื่อใช้ในการทำนายข่าวอาชญากรรมจำนวน 100 ข่าว ดังนี้ แบบจำลองที่ใช้ทำนายเจตนาของผู้กระทำผิด (Intention) และแบบจำลองที่ใช้ทำนายเหตุละเว้นความผิด (Justification) เลือกใช้ FastText และ Crime2Vec เพราะเป็นสองเทคนิคที่ให้ค่าความแม่นยำสูงที่สุด ผ่านการหาค่าข้อมูลนำเข้าต้นไม้ตัดสินใจด้วยเทคนิคซอฟต์โหวต ส่วนแบบจำลองที่ใช้พิจารณาความสัมพันธ์ของผลการกระทำ (Causation) และแบบจำลองที่ใช้พิจารณาเหตุลดโทษ (Impunity) เลือกใช้เทคนิค FastText BiLSTM และ Crime2Vec BiLSTM ที่ให้ค่าความแม่นยำสูงที่สุด ตามลำดับ และให้ค่าความแม่นยำเมื่อเทียบกับความเห็นของนักกฎหมายตามที่แสดงในตารางที่ 5-15

ตารางที่ 5-15 การวัดประสิทธิภาพแบบจำลองด้วยข่าวอาชญากรรม

แบบจำลอง		**ข้อมูลทดสอบ 100 ข่าวอาชญากรรม**		
		ความเห็นนักกฎหมาย	**การทำนายของแบบจำลอง**	**ความแม่นยำ(%)**
Intention	None	2	0	0
	Regular	56	51	91
	Special	42	22	52
Causation	Nothing	1	0	0
	Injury	10	6	60
	death	89	69	78
Impunity	Except	3	0	0
	Decrease	5	2	40
	Normal	61	56	92
	Increase	31	24	77
Justification	NotWrong	2	0	0
	Wrong	98	96	98

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 5-15 แสดงให้เห็นว่าในแบบจำลองที่มีจำนวนตัวอย่างทดสอบน้อย ความเห็นของนักกฎหมายและของแบบจำลองจะมีความแตกต่างกันมาก อาจไม่ได้ให้ค่านัยสำคัญในการวิเคราะห์ค่าความแม่นยำได้ และจะพบว่าในแบบจำลองเจตนาของผู้กระทำ (Intention) คือเจตนา

ปกติ (Regular) และเจตนาร้ายแรง (Special) ให้ค่าที่ใกล้เคียงกัน เพราะมีปัจจัยหลายอย่างในการจำแนกข้อเท็จจริงนี้ ส่วนแบบจำลองความสัมพันธ์ของการกระทำ (Causation) ระหว่างบาดเจ็บ (Injury) และตาย (Death) แบบจำลองทำนายได้แม่นยำใกล้เคียงกับความเห็นของนักกฎหมาย ส่วนแบบจำลองการพิจาณาละเว้นโทษ (Impunity) การละเว้นโทษ (Except) และการลดโทษ (Decrease) แบบจำลองยังให้ค่าความแม่นยำต่ำ เพราะด้วยจำนวนข่าวที่นำมาทดสอบมีจำนวนน้อย ส่วนการลงโทษปกติ (Normal) และการยืนยันโทษที่สูงขึ้น (Increase) ให้ค่าใกล้เคียงกันมาก และท้ายสุดแบบจำลองพิจารณการละเว้นความผิด (Justification) ให้ค่าความแม่นยำสูงสำหรับการกระทำผิด และให้ค่าที่ความแม่นยำที่ต่ำมากกับการละเว้นความผิด

สรุปการทดสอบ 100 ข่าวอาชญากรรมออนไลน์ ค่าความแม่นยำของแบบจำลองในการทำนายกลุ่มการลงโทษเทียบกับความเห็นของนักกฎหมาย คิดเป็นร้อยละ 59 แต่พบว่ามีอีก 11 คดีที่มีแนวโน้มที่อาจแม่นยำได้ นั่นคือกลุ่มการลงโทษ E และ G ที่ศาลอาจพิพากษาในทิศทางใดก็ได้ระหว่างสองกลุ่มนี้ นั่นอาจทำให้แบบจำลองมีความแม่นยำสูงถึงร้อยละ 70 ได้ รายละเอียดการทดสอบข่าวตามที่แสดงในภาคผนวก ข.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์ฉบับนี้ นำเสนอการวิเคราะห์ข้อความภาษาธรรมชาติตามประมวลกฎหมายอาญา โดยใช้ข้อมูลการฝึกฝนแบบจำลองจากคำพิพากษาศาลฎีกา ในคดีที่เกี่ยวข้องกับการเจตนากระทำผิดด้วยประสงค์ต่อชีวิตผู้อื่น ในขอบเขตการรับผิดมาตรา 288 และ 289 ด้วยเทคนิคการเรียนรู้เชิงลึกรูปแบบต่างๆ ด้วยการเพิ่มประสิทธิภาพของแบบจำลอง เช่นการสังเคราะห์ชุดข้อมูลฝึกสอนให้มีจำนวนเท่ากัน (SMOTE) การถ่ายโอนความรู้ (Transfer Learning) จากคลังข้อมูลอื่น ในการให้แบบจำลองเข้าใจรูปแบบภาษา (Language) รวมทั้งการหาค่าเฉลี่ยของข้อมูลนำเข้าแต่ละแบบจำลองด้วยเทคนิคซอฟต์โหวต (Soft Voting) ก่อนทำนายด้วยต้นไม้ตัดสินใจ จากการทดลองสามารถสรุปผลการวิจัย ข้อจำกัดงานวิจัย และแนวทางในการพัฒนาต่อได้ดังนี้

6.1 สรุปผลการวิจัย

1) แบบจำลองจากการเรียนรู้ของเครื่องด้วยนาอีฟเบย์ ซึ่งเป็นการเรียนรู้แบบเดิม มีประสิทธิภาพต่ำกว่าแบบจำลองที่นำเสนอ เมื่อวัดประสิทธิภาพด้วยค่าเฉลี่ยของความถูกต้อง (Accuracy) ด้วยชุดข้อมูลการฝึกสอนและทดสอบเดียวกัน พบว่า นาอีฟเบย์ที่ใช้ N-Gram Level TF-IDF ให้ค่าความแม่นยำสูงที่สุดเมื่อเทียบกับเทคนิค Count Vector , Word Level TF-IDF หรือ Character Level TF-IDF คือ ร้อยละ 71.75 ขณะที่งานวิจัยนี้ใช้แบบจำลอง LSTM ระดับคำ ค่าความแม่นยำคือ ร้อยละ 74.39

2) แบบจำลองที่ผ่านการฝึกฝนแบบสองทิศทาง (BiLSTM) จะมีประสิทธิภาพสูงกว่าการฝึกฝนแบบทิศทางเดียว (LSTM) เมื่อวัดประสิทธิภาพด้วยค่าเฉลี่ยมหภาคของเอฟวัน (Macro-average F1 Score) ด้วยชุดข้อมูลทดสอบ และเทคนิคการฝังคำเช่นเดียวกัน พบว่าแบบจำลอง LSTM แบบสองทิศทาง (BiLSTM) ให้ค่าเฉลี่ย F1 ร้อยละ 81.12 ขณะที่ LSTM แบบทิศทางเดียว ร้อยละ 72.41

3) แบบจำลองที่เพิ่มประสิทธิภาพด้วยการถ่ายโอนการเรียนรู้จากคลังข้อมูลอื่นที่มีขนาดใหญ่กว่าจะให้ประสิทธิภาพที่สูงกว่า เมื่อวัดด้วยค่าเฉลี่ย F1 เช่น FastText ที่มีจำนวน 4 ล้าน Embedding ในการเรียนรู้รูปแบบภาษาก่อนการฝึกฝนแบบจำลอง โดยให้ค่าเฉลี่ย F1 ของทั้ง 4 แบบจำลองคิดเป็นร้อยละ 84.47 สำหรับ BILSTM และ 80.33 สำหรับ LSTM แต่น่าสังเกต Crime2Vec ที่แม้ค่าเฉลี่ย F1 จะน้อยกว่า FastText แต่มีค่าที่สูงกว่า Thai2Vec ซึ่งเมื่อเทียบจำนวนคำแล้ว Crime2Vec มีคำเพียง 2,676 คำ ขณะที่ Thai2Vec มี 60,000 คำ เพราะการใช้เนื้อหา

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ฝึกฝนแบบจำลองทางภาษา (Language Model) ทำให้ Crime2Vec มีค่าเฉลี่ย F1 ร้อยละ 81.54 สำหรับ BILSTM และ 76.99 สำหรับ LSTM

4) การใช้ค่าเฉลี่ยของความน่าจะเป็นในชั้นข้อมูลส่งออก (Output) ของแต่ละแบบจำลอง แทนค่าข้อมูลนำเข้า (Input) ของต้นไม้ตัดสินใจ ให้ค่าความแม่นยำใกล้ตรงกับความคิดเห็นของนักกฎหมาย ร้อยละ 59 จากข่าวที่นำมาทดสอบจำนวน 100 ข่าวอาชญากรรมออนไลน์

6.2 ข้อจำกัดงานวิจัย

1) แบบจำลองที่ใช้ในการวิเคราะห์ข้อมูลจากข่าวอาชญากรรมออนไลน์นี้ เป็นการเลือกทำนายกลุ่มความรับผิด A-G เท่านั้น ทำให้ข้อเท็จจริงที่อยู่ในลักษณะความผิดกลุ่มอื่นหรือที่เกินขอบเขตไม่สามารถให้คำตอบที่ถูกต้องได้

2) การนำเข้าเพื่อทดสอบประสิทธิภาพแบบจำลองใช้การส่งผ่านข้อมูลผ่านไฟล์ข้อความ (Text File) ทำให้การประมวลผล ถ้ามีจำนวนข่าวปริมาณมากทำให้ใช้เวลาในการประมวลผลนาน

3) ในบางคดีหรือข้อเท็จจริงที่เกิดขึ้น อาจมีคำตอบได้หลายกลุ่ม เช่น มีทั้งผู้บาดเจ็บ และเสียชีวิต ซึ่งผู้กระทำผิดอาจต้องรับผิดเรียงทุกฐานความผิดไป แต่งานวิจัยนี้จะใช้การรับผิดที่สูงสุด นั่นคือดูที่การเสียชีวิต ดังนั้นผลลัพธ์จากงานวิจัยนี้ จะให้คำแนะนำเบื้องต้นเท่านั้น ไม่ได้ครอบคลุมทั้งหมด

6.3 แนวทางการวิจัยในขั้นถัดไป

1) ข้อมูลที่ใช้ในการฝึกฝนแบบจำลองในงานวิจัยนี้มีปริมาณน้อย โดยได้จากการรวบคำพิพากษาของศาลฎีกาที่เผยแพร่ ทำให้ประสิทธิภาพของแบบจำลองเมื่อนำไปใช้งานกับข้อมูลทดสอบจริง ประสิทธิภาพไม่ดี ดังนั้นเพื่อให้แบบจำลองเพิ่มประสิทธิภาพความแม่นยำ โดยการเพิ่มจำนวนข้อมูลการฝึกสอน (Training Set) ให้มากขึ้น

2) การใช้เทคนิคการถ่ายโอนความรู้ด้วย Pretrained Words Embedding ช่วยให้ประสิทธิภาพของแบบจำลองดีขึ้น คือการใช้ค่าถ่วงน้ำหนักที่ผ่านการฝึกฝนมาแทนการสุ่มค่าเริ่มต้นในการฝึกฝนแบบจำลอง โดยประสิทธิภาพจะสูงขึ้นถ้าข้อมูลภายนอกที่ทำการฝึกฝนมีรูปแบบลักษณะทางภาษาคล้ายคลึงกับงานวิจัย ดังนั้นในอนาคต ผู้วิจัยก็เก็บสะสมแหล่งข่าวอาชญากรรมให้มีปริมาณที่มากขึ้น เพื่อใช้ฝึกฝนแบบจำลองทางภาษา ก่อนเข้าฝึกฝนแบบจำลองของงานวิจัย

3) การใช้เทคนิคการถ่ายโอนความรู้ตามข้อ 2 เป็นเพียงการแทนคำ (Word Representation) ในชั้นแรกสุดของโครงข่ายประสาทเทียมเท่านั้น แต่รูปแบบของภาษาในงานที่เรา

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ต้องการฝึกฝนยังไม่มีการปรับค่าให้เหมาะสมกับบริบทรอบๆ ดังนั้นในอนาคตจึงมีการใช้เทคนิคที่เรียกว่า ULMFIT (Universal Language Model Fine-tuning for Text Classification) ที่ต้องมีการทำ fine-tune พารามิเตอร์ ในชั้นต่างๆ ให้เข้ากับงานที่เราต้องการจำแนก เพราะให้แบบจำลองเข้าใจลักษณะทางภาษาหรือบริบทของงานด้านคดีอาญาได้ดีขึ้น

4) ในงานวิจัยนี้ใช้การฝังคำระดับคำ (Word Level) ทำให้เกิดปัญหาการไม่รู้จักคำ (Out - of-Vocabulary) ในคลังข้อมูล ซึ่งอาจส่งผลต่อประสิทธิภาพความแม่นยำของแบบจำลอง ดังนั้นเพื่อแก้ปัญหานี้จะใช้การตัดคำระดับตัวอักษร (Character Level) รวมทั้งการปรับค่าไฮเปอร์พารามิเตอร์ต่างๆ ให้ได้ประสิทธิภาพที่ดีขึ้น

5) เพื่อสะดวกต่อผู้ใช้งาน งานวิจัยจะถูกพัฒนาขึ้นให้สามารถเข้าใช้งานผ่านหน้าเว็บไซต์ โดยมีกล่องรับข้อความจากผู้ใช้ และแสดงผลลัพธ์แยกเป็นร้อยละความน่าจะเป็นของแต่ละองค์ประกอบความผิด รวมทั้งแนวทางการตัดสินสุดท้าย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บรรณานุกรม

1. มีนะกนิษฐ, ท., ประมวลกฎหมายอาญา ฉบับอ้างอิง. 43 ed. 2020, ประเทศไทย: วิญญูชน. 776.
2. วัจนะสวัสดิ์, เ., คำอธิบายกฎหมายอาญาภาค *1* บทบัญญัติทั่วไป. 2008, ประเทศไทย: สำนักพิมพ์ พลสยาม พริ้นติ้ง
3. เนตสุภา, ก.เ.แ.อ., การตีความกฎหมายอาญา. 2009, ประเทศไทย: กองทุนศาสตราจารย์ จิตติ ติงศภัทิย์ คณะนิติศาสตร์ มหาวิทยาลัยธรรมศาสตร์.
4. แสงอุทัย, ห., กฎหมายอาญา ภาค *1*. 2008, ประเทศไทย: สำนักพิมพ์ มหาวิทยาลัยธรรมศาสตร์.
5. ระบบสืบค้นคำพิพากษา. 2015 [cited 2018 1 Oct]; Available from: http://deka.supremecourt.or.th.
6. สำนักข่าวเดลินิวส์. ข่าวอาชญากรรม. 1998 [cited 2018 1 Dec]; Available from: https://www.dailynews.co.th/crime.
7. McNeill, D.A.a.G., *ARTIFICIAL NEURAL NETWORKS TECHNOLOGY*. 1992.
8. M, T.M.M., *Machine Learning*. 1997: McGraw-Hill.
9. Chester, M. *Neural networks - a tutorial*. 1993.
10. Elman, J.L., *Finding structure in time.* COGNITIVE SCIENCE, 1990. **14**: p. 179-211.
11. McCullock, J. *Neural Network* 2011 [cited 2019 5 Dec]; Available from: http://mnemstudio.org/neural-networks-elman.
12. Negnevitsky, M., *Artificial Intelligence A Guide to Intelligent Systems*. 2 ed. 2005: Addison-Wesley.
13. Chuangsuwanich, P.V.a.E., *2110595:Natural Language Processing*. 2018, ประเทศไทย: Chulalongkorn University.
14. Yosi Taguri, S.E., and Rahav Lussato. *7 Types of Neural Network Activation Function*. 2016 [cited 2019 10 Dec]; Available from: https://missinglink.ai.
15. Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. in *Studies in Computational Intelligence*. 2008.
16. Carter, D.S.a.S. *Activation Function* 2016 [cited 2019 20 Dec]; Available from: https://playground.tensorflow.org
17. Mazur, M. *A Step by Step Backpropagation Example*. 2008 [cited 2020 20 Jan]; Available from: https://mattmazur.com.
18. LeCun, Y.a.B., Y. and Hinton, Geoffrey, *Deep Learning.* Nature, 2015. **521**.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

19. Jurafsky, D.a.M., James, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Vol. 2. 2008.
20. Ruder, S., *An overview of gradient descent optimization algorithms.* 2016.
21. Science., S.C. *CS231n: Convolutional Neural Networks for Visual Recognition*. 2015 [cited 2019 12 Mar]; Available from: https://cs231n.github.io.
22. Olah, C. *Understanding LSTM Networks*. 2014 [cited 2019 15 Apr]; Available from: https://colah.github.io.
23. Wang, Z.C.a.R.K.a.Z.P.a.Y., *Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction*. 2018.
24. Pratiksha Ashiwal, S.R.T., Priyanka Tripathi, Rohit Miri, *Web Information Retrieval Using Python and BeautifulSoup*, in *International Journal for Research in Applied Science and Engineering Technology (INRASET)*. 2016.
25. Fabian Beck, S.G., Benjamin Biegel, Sebastian Baltes, and Daniel Weiskopf. *RegViz: visual debugging of regular expressions*. in *In Companion Proceedings of the 36th International Conference on Software Engineering*. 2014. New York, USA.
26. Thairatananond, Y., *Towards the Design of a Thai Text Syllable Analyzer*. 1981, Asian Institute of Technology.
27. รัตติกร วรากูลศิริพันธ์, จ.ง., สมศักดิ์จันวัน, สุธาทิพย์จิวัธยากูล และศักดิ์ชัย ทิพย์จักษุรัตน์, การตัดคำจากประโยคภาษาไทยด้วยวิธีการเทียบคำที่ยาวที่สุด. 1995.
28. Oparad, R.K.a.K.C.a.T.A.a.K.S.a.P.C.a.C.K.a.T.R.a.K. *DeepCut: A Thai word tokenization library using Deep Neural Network*. 2019 [cited 2020 4 Mar]; Available from: https://doi.org/10.5281/zenodo.3457707.
29. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, in *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. 1995. p. 1137–1143, .
30. K.M., T., *Confusion Matrix In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining Second Edition*. 2017, Boston, MA, USA: Springer.
31. Ma, H.a.H., Wenjiang and Jing, Yuanshu and Yang, Chenghai and Han, Liangxiu

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

and Dong, Yingying and Ye, Huichun and Shi, Yue and Zheng, Qiong and Liu, Linyi and Ruan, Chao, *Integrating Growth and Environmental Parameters to Discriminate Powdery Mildew and Aphid of Winter Wheat Using Bi-Temporal Landsat-8 Imagery.* Remote Sensing, 2019. **11**: p. 846.

32. Ruder, S. *The State of Transfer Learning in NLP*. 2017 [cited 2019 1 Aug]; Available from: https://ruder.io.
33. Pan, S.a.Y., Qiang, *A Survey on Transfer Learning.* Knowledge and Data Engineering, IEEE Transactions on, 2010. **22**: p. 1345-1359.
34. Joulin, A.a.G., Edouard and Bojanowski, Piotr and Mikolov, Tomas, *Bag of Tricks for Efficient Text Classification*. 2017.
35. Polpanumas, W.P.a.C. *State-of-the-Art language modeling and text classification in Thai language*. 2016 [cited 2019 12 Oct]; Available from: https://github.com/cstorm125/thai2fit
36. Mikolov, T.a.S., Ilya and Chen, Kai and Corrado, G.s and Dean, Jeffrey, *Distributed Representations of Words and Phrases and their Compositionality.* Advances in Neural Information Processing Systems, 2013. **26**.
37. Sebastian, H.J.a.R. *Universal Language Model Fine-tuning for Text Classification*. 2018. Melbourne, Australia: Association for Computational Linguistics.
38. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* J. Mach. Learn. Res., 2014. **15**: p. 1929-1958.
39. Amphawan, K. เอกสารประกอบการสอนวิชา *Data Mining*. 2016; Available from: https://staff.informatics.buu.ac.th/~komate/.
40. ศิริเนาวกุลม, บ. การแสดงความรู้ด้วยกฎ *Rule-Based Representation,* เอกสารประกอบการสอนวิชา *AI*. 2016 Available from: http://teacher.en.rmutt.ac.th/ktw/Resources
41. Zhang, H. and D. Li, *Naïve Bayes Text Classifier.* 2007 IEEE International Conference on Granular Computing (GRC 2007), 2007: p. 708-708.
42. Shahjalal, R.I.a.M.A. *Soft voting-ased ensemble approach to predict early stage DRC violations*. in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2019. Dallas, TX, USA.
43. L. Xiao, G.W.a.Y.Z., *Research on Patent Text Classification Based on Word2Vec and LSTM*, in *2018 11th International Symposium on Computational*

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Intelligence and Design (ISCID). 2018: Hangzhou, China. p. 71-74.

44. P. Osathitporn, N.S., and W. Vatanawood. *A scheme of criminal law knowledge acquisition using ontology*. in *in Proceedings - 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2017*. 2017. Kanazawa, Japan.

45. Thammaboosadee, S.a.W., Bunthit and Charoenkitkarn, Nipon, *A Framework of Multi-Stage Classifier for Identifying Criminal Law Sentences.* Procedia Computer Science, 2012. **13**: p. 53-59.

46. Kowsrihawat, K.a.V., Peerapon and Boonkwan, Prachya. *Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism*. 2018.

47. Wannaphong Phatthiyaphaibun, K.C., Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul and Pattarawat Chormai. *Stopword* ภาษาไทย. [cited 2020 2 July]; Available from: https://pythainlp.readthedocs.io/en/latest/pythainlp-1-4-thai/#stopword.

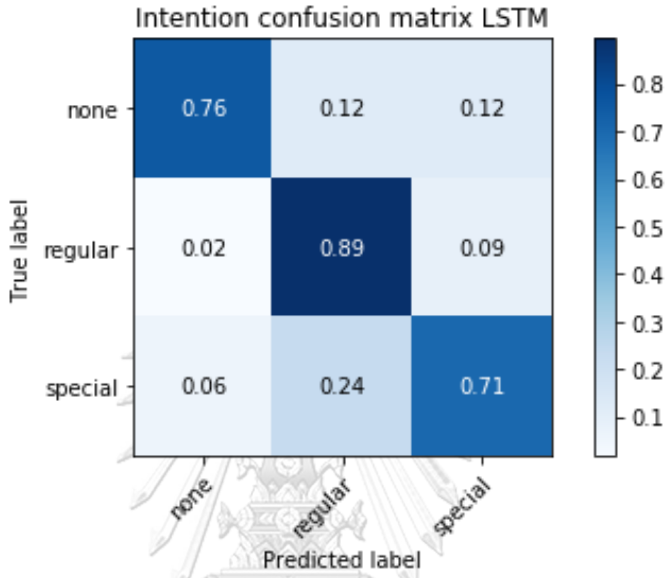
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

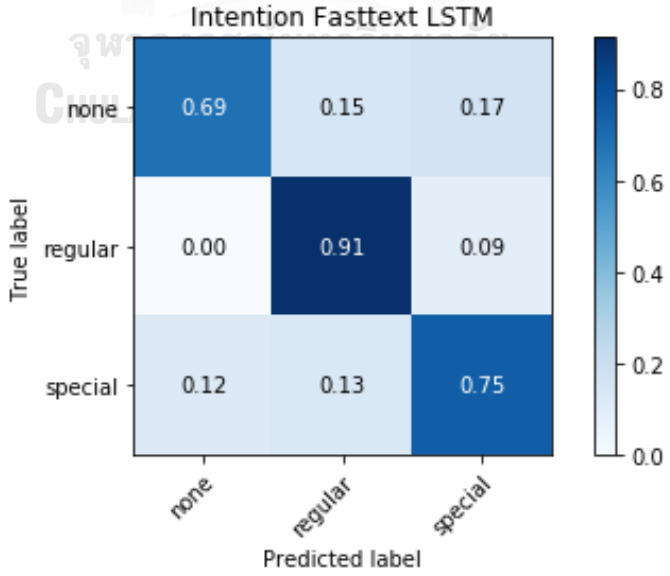
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก

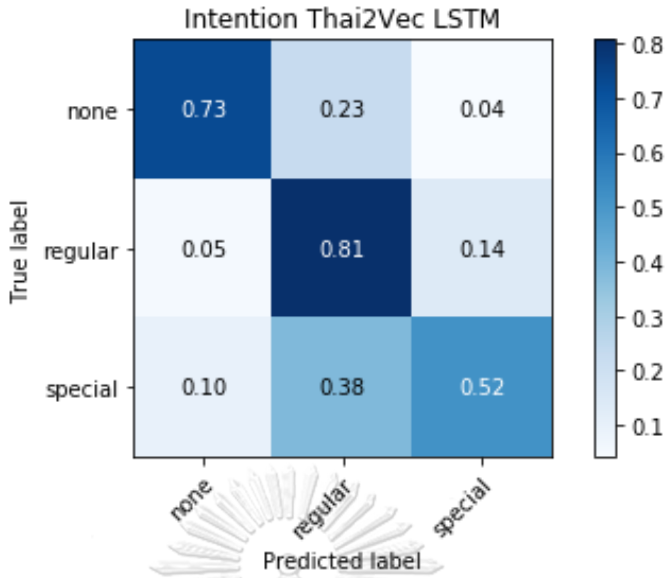
ประสิทธิภาพของแบบจำลอง



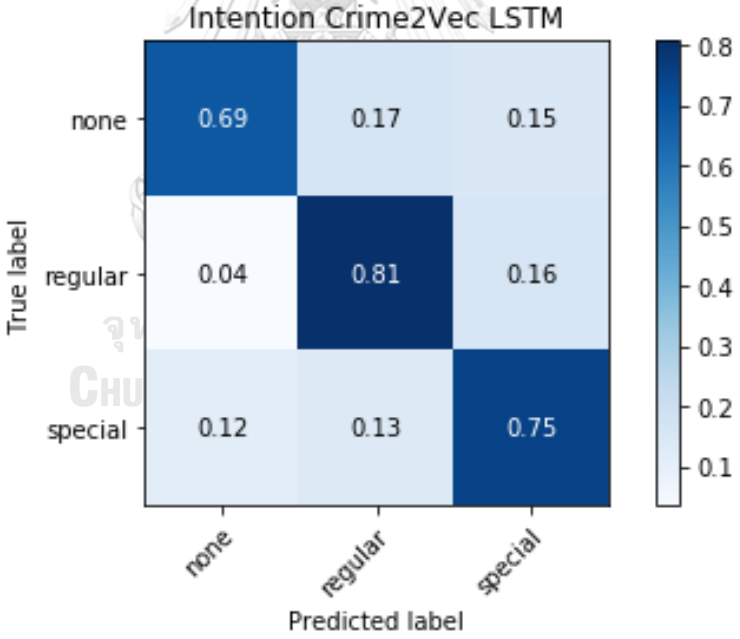
รูปที่ ก-1 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention LSTM



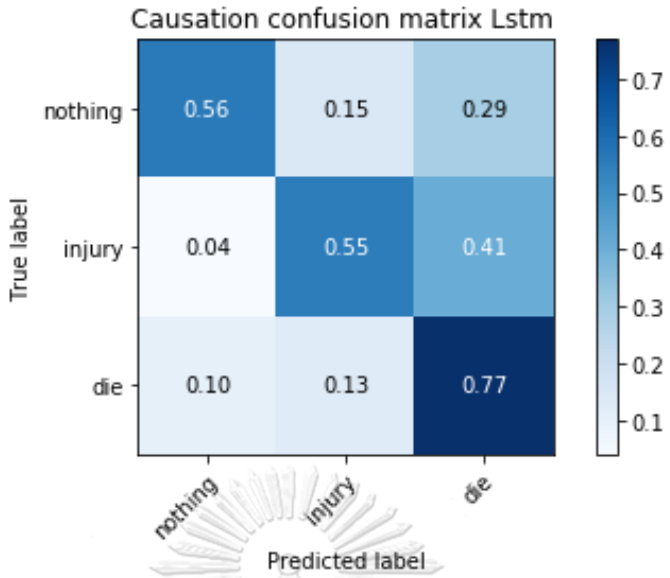
รูปที่ ก-2 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention FastText LSTM



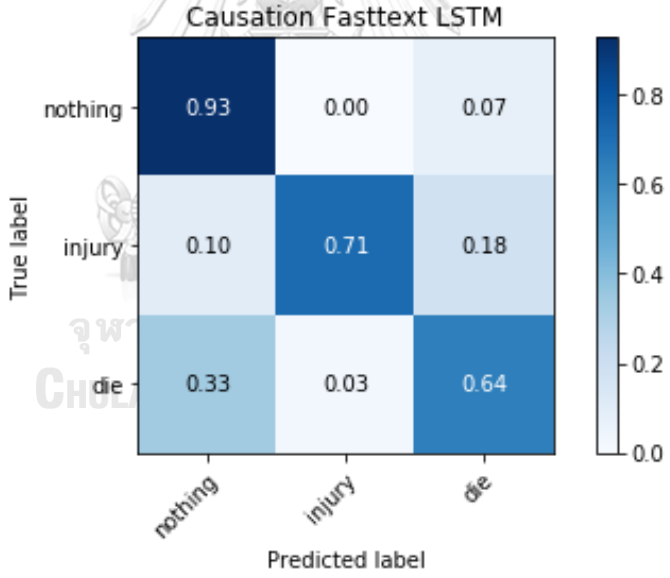
รูปที่ ก-3 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Thai2Vec LSTM



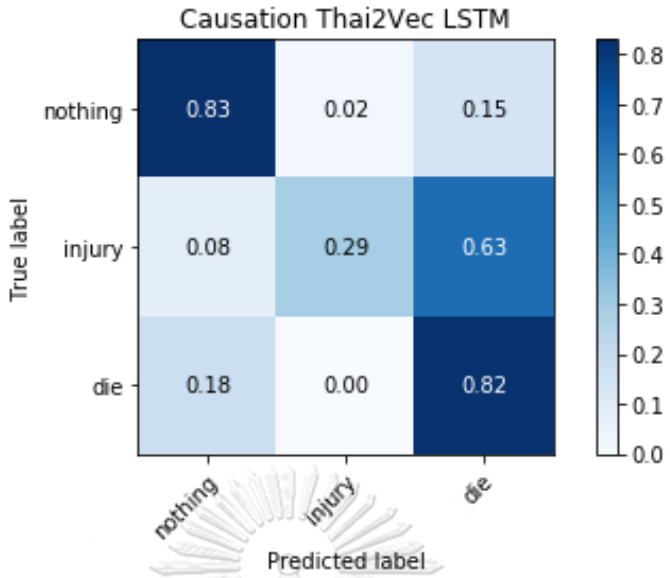
รูปที่ ก-4 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Crime2Vec LSTM



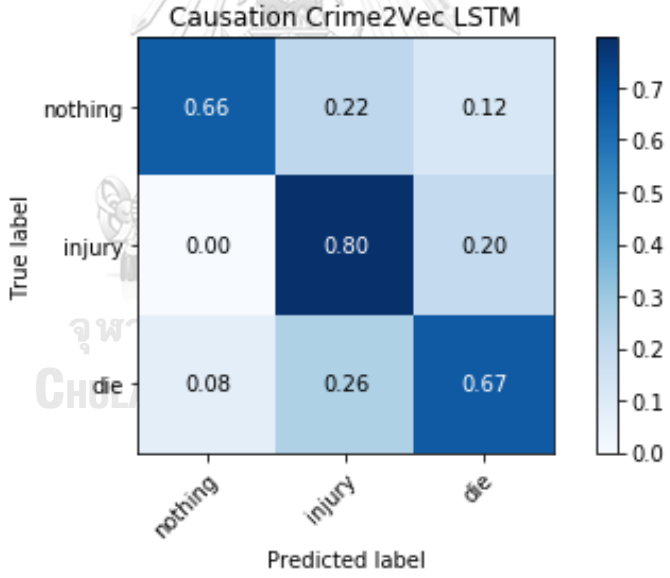
รูปที่ ก-5 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation LSTM



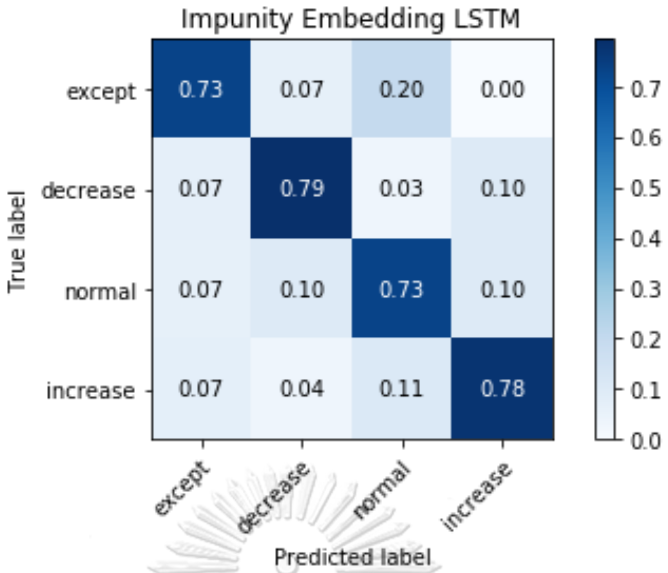
รูปที่ ก-6 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation FastText LSTM



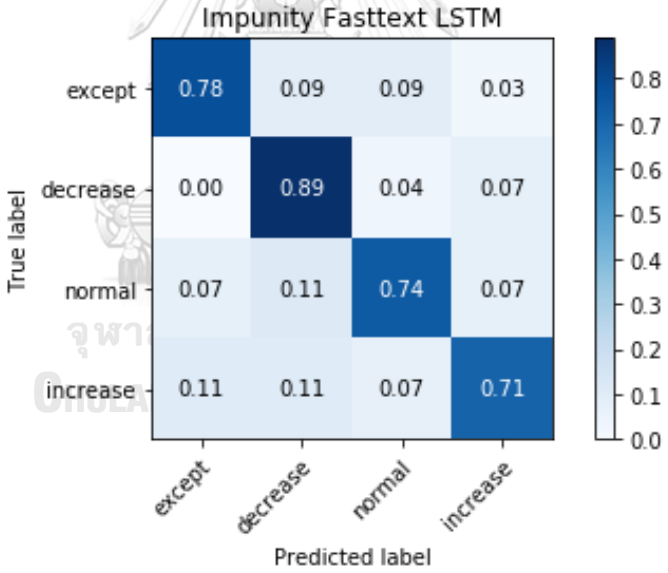
รูปที่ ก-7 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Thai2Vec LSTM



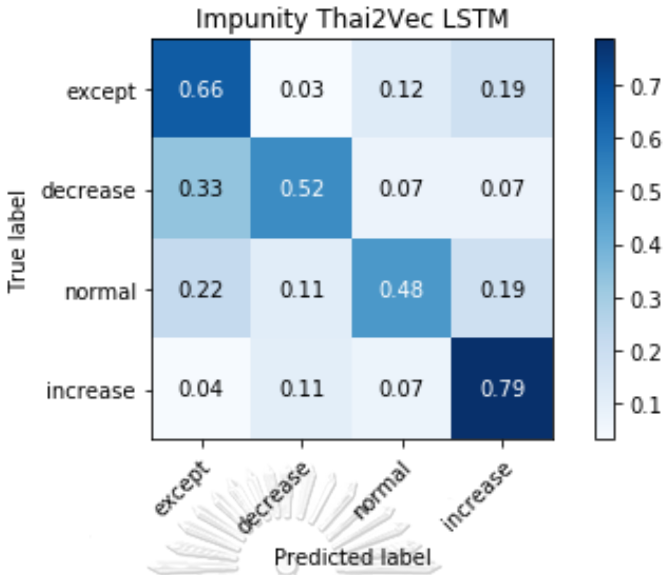
รูปที่ ก-8 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Crime2Vec LSTM



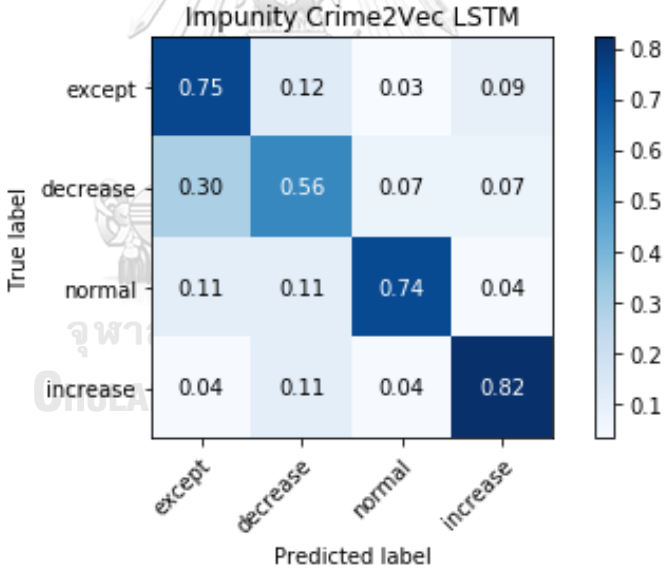
รูปที่ ก- 9 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity LSTM



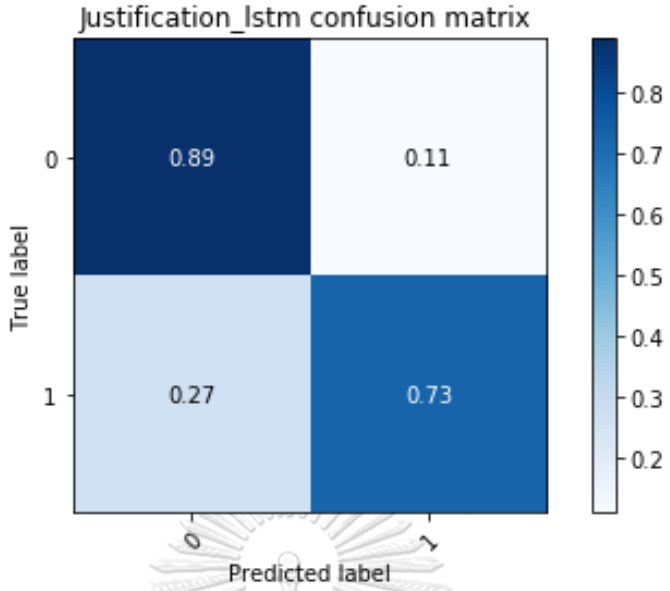
รูปที่ ก-10 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity FastText LSTM



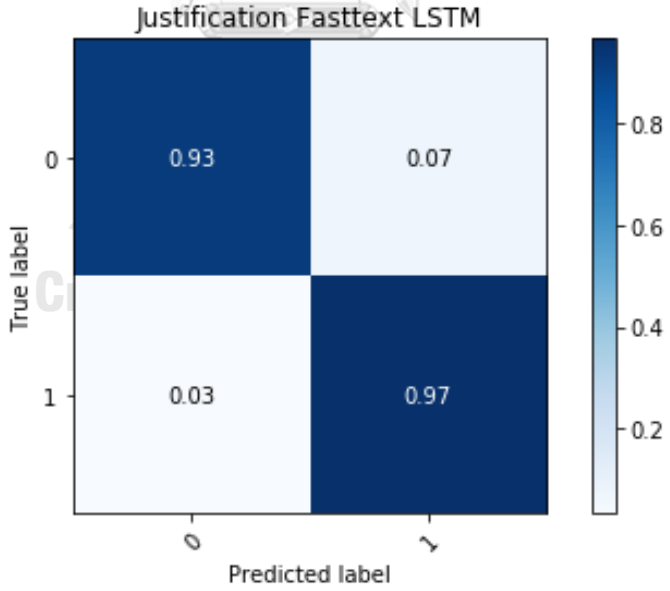
รูปที่ ก-11 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Thai2Vec LSTM



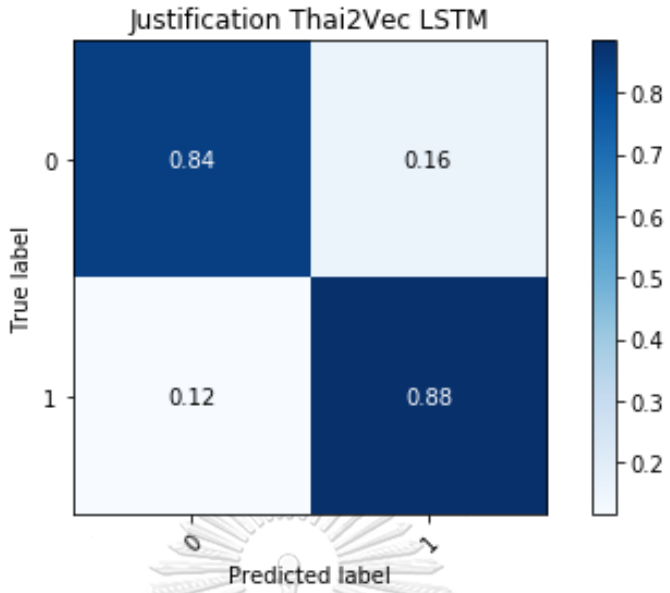
รูปที่ ก-12 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Crime2Vec LSTM



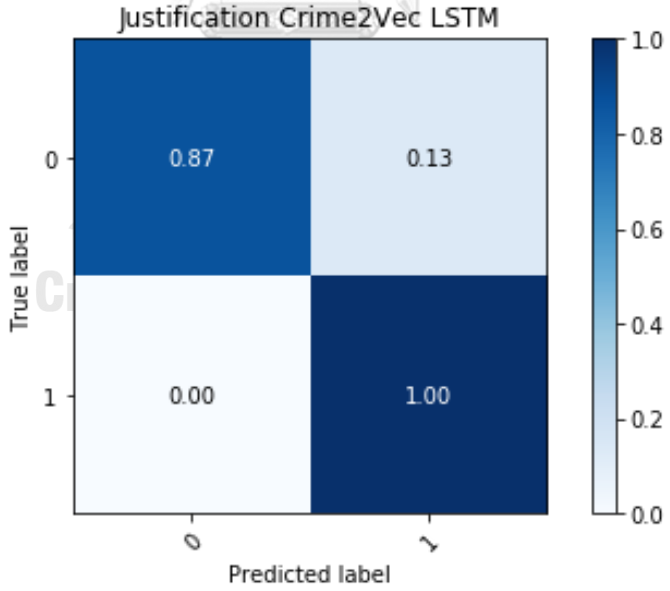
รูปที่ ก-13 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification LSTM



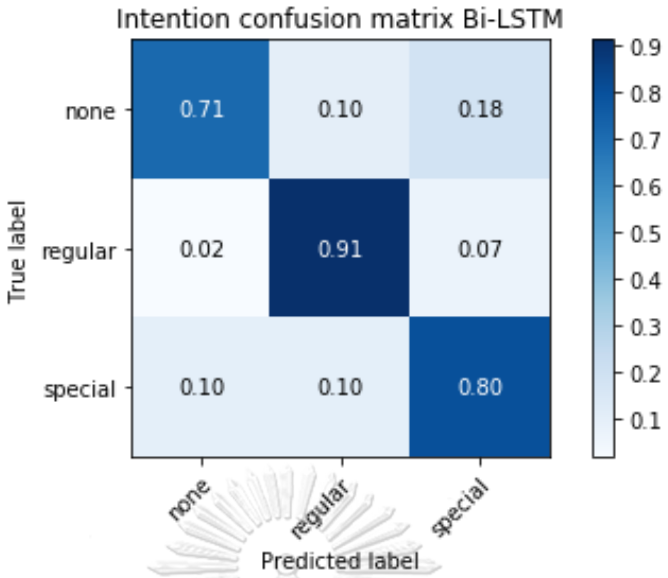
รูปที่ ก-14 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification FastText LSTM



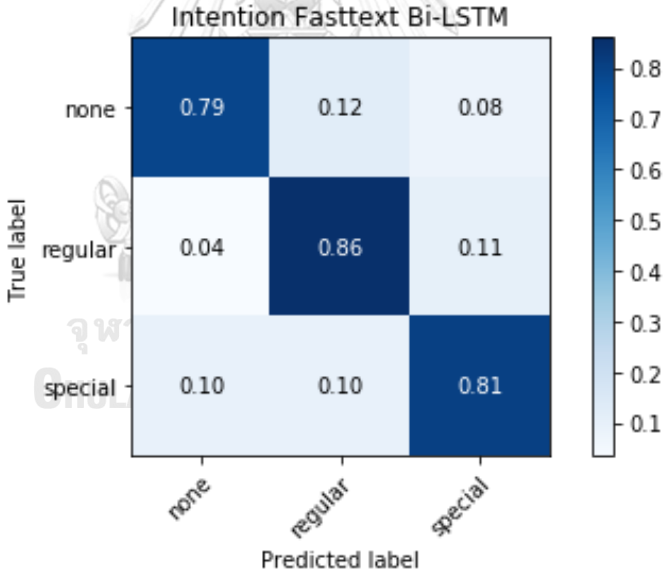
รูปที่ ก-15 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Thai2Vec LSTM



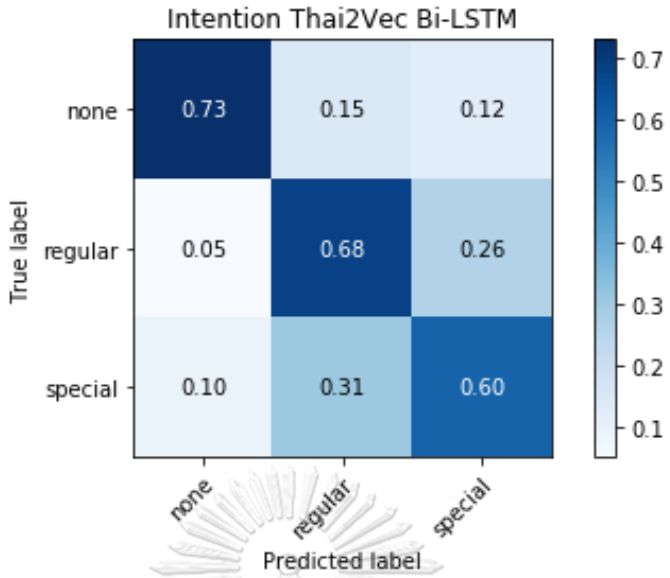
รูปที่ ก-16 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Crime2Vec LSTM



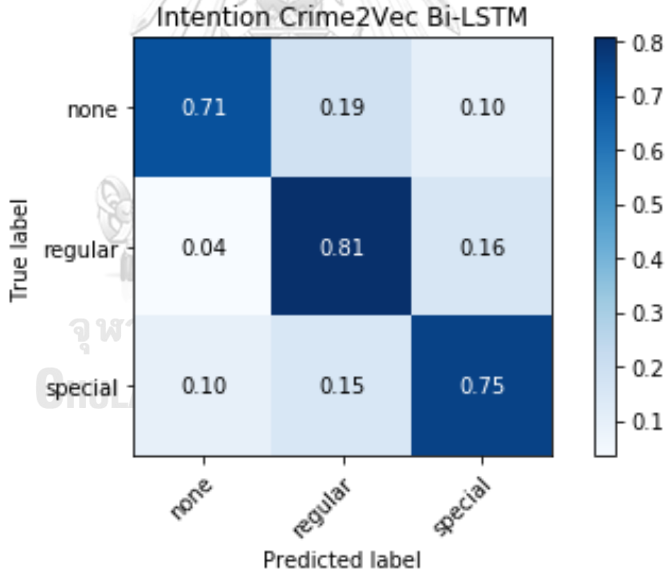
รูปที่ ก-17 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention BiLSTM



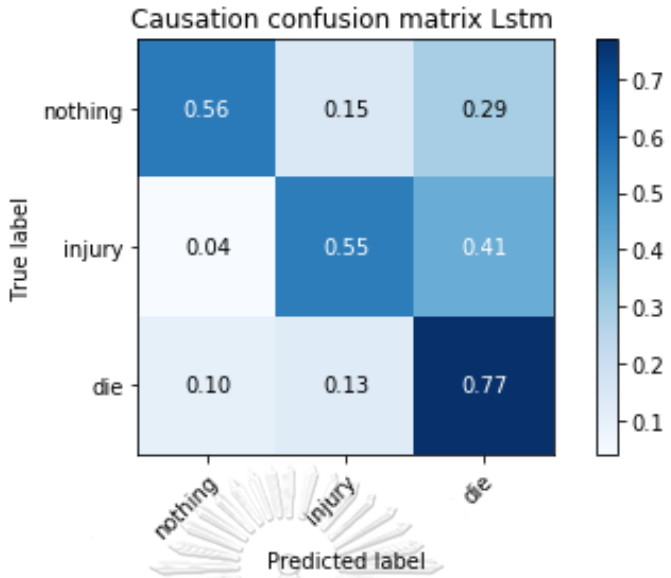
รูปที่ ก-18 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention FastText BiLSTM



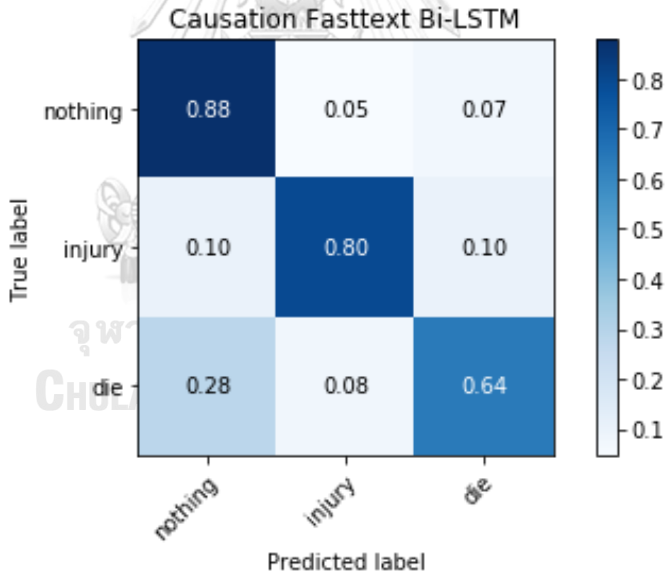
รูปที่ ก-19 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Thai2Vec BiLSTM



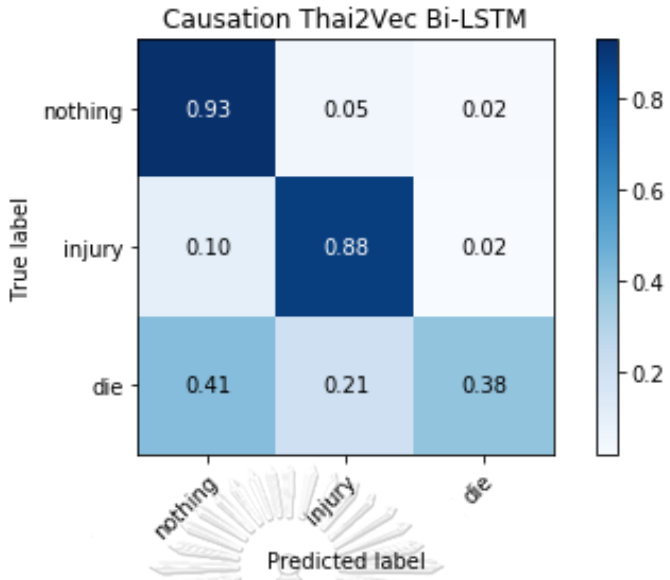
รูปที่ ก-20 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Intention Crime2Vec BiLSTM



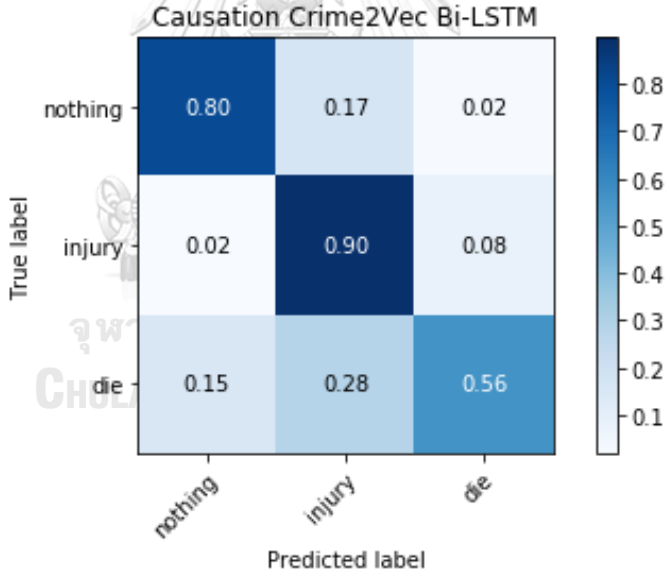
รูปที่ ก-21 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation BiLSTM



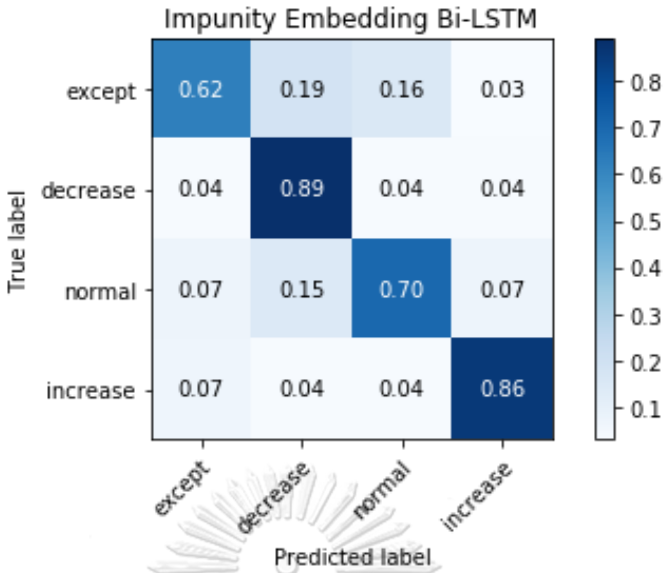
รูปที่ ก-22 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation FastText BiLSTM



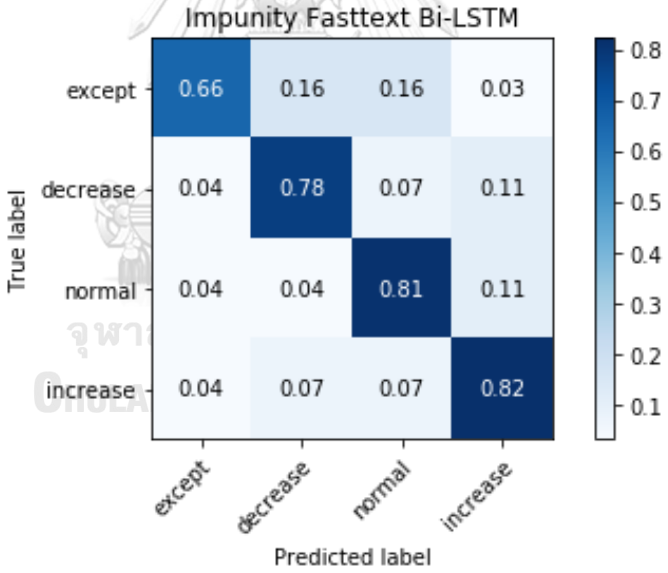
รูปที่ ก-23 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Thai2Vec BiLSTM



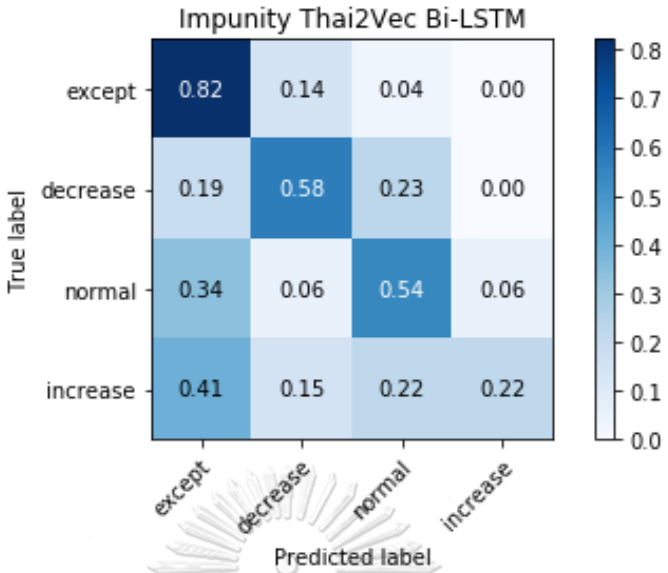
รูปที่ ก-24 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Causation Crime2Vec BiLSTM



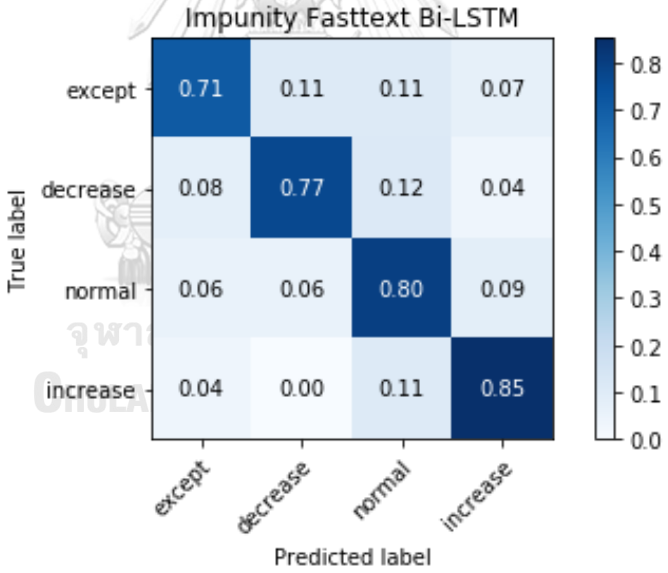
รูปที่ ก-25 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity BiLSTM



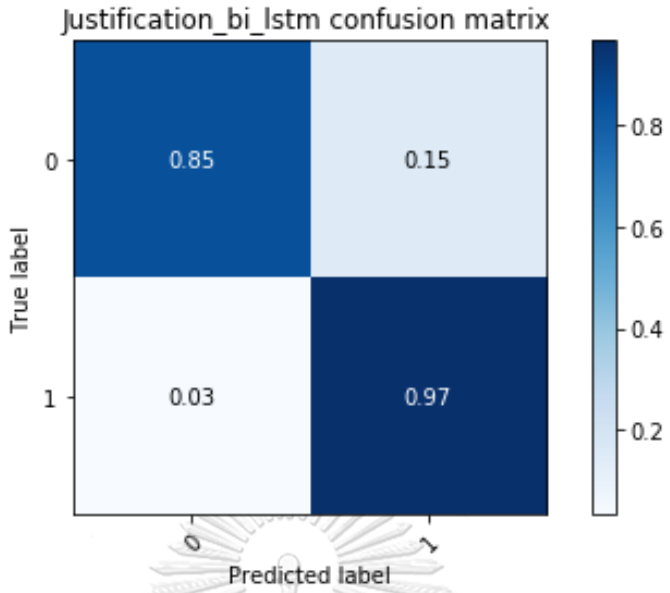
รูปที่ ก-26 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity FastText BiLSTM



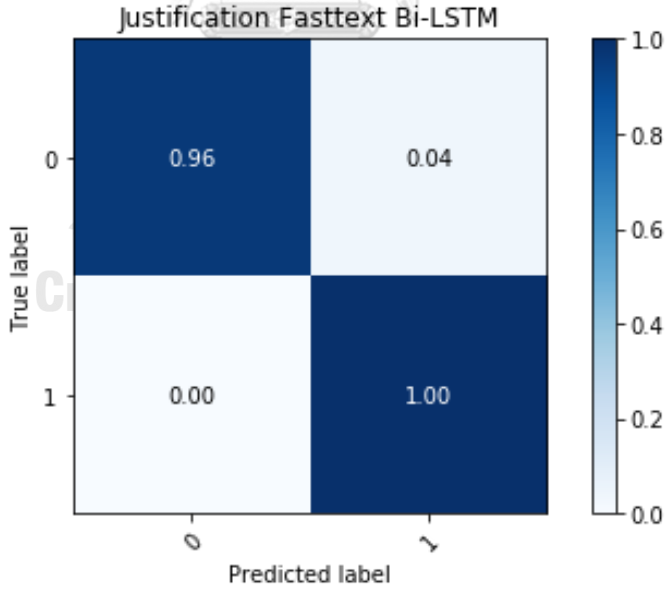
รูปที่ ก-27 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Thai2Vec BiLSTM



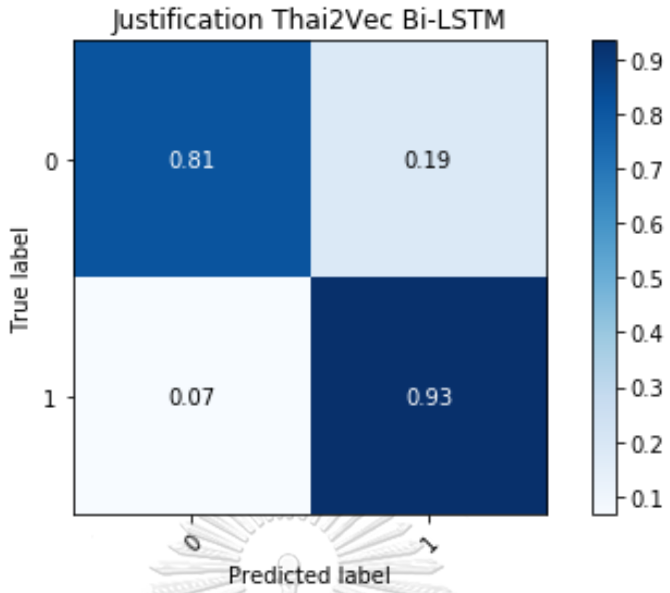
รูปที่ ก-28 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Impunity Crime2Vec BiLSTM



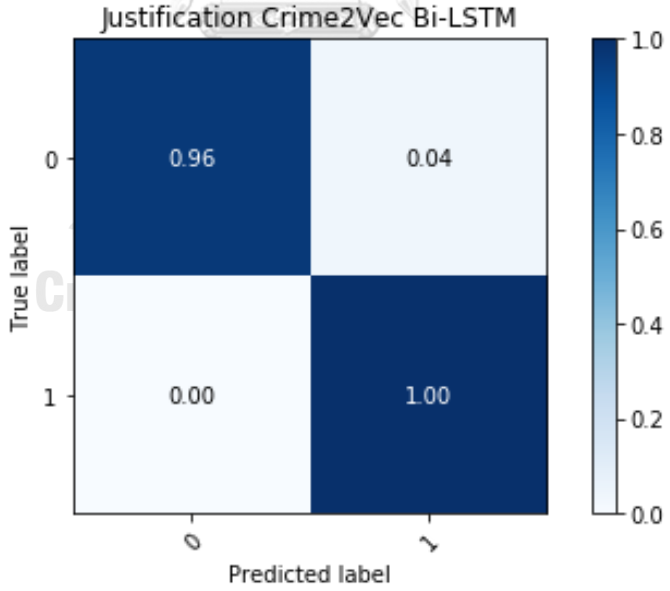
รูปที่ ก-29 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification BiLSTM



รูปที่ ก-30 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification FastText BiLSTM



รูปที่ ก-31 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Thai2Vec BiLSTM



รูปที่ ก-32 ตารางคอนฟิวชันก์เมทริกต์ของแบบจำลอง Justification Crime2Vec BiLSTM

ภาคผนวก ข

รายละเอียดการทดสอบข่าวอาชญากรรม

วิทยานิพนธ์นี้ใช้ข้อมูลข่าวอาชญากรรมออนไลน์ [6] จำนวน 100 ข่าว รายละเอียดตามตารางที่ ข-1 โดยในส่วนเว็บไซต์ข่าว เครื่องหมาย “/” หมายถึงที่มาของเนื้อหาข่าวอาชญากรรมจากเว็บไซต์ https://dailynews.co.th/crime และตามด้วยที่อยู่ของเนื้อหาข่าว เช่น ในข่าวที่ 1 ระบุว่า “/666630” หมายถึง ข่าวที่นำมาวิเคราะห์อยู่ที่ URL : https://dailynews.co.th/crime/666630 เป็นต้น ส่วนตัวอักษรย่อในตารางมีความหมายดังนี้

1) แบบจำลองเจตนา (Intention) กำหนดให้ N: ไม่มีเจตนา (None), R: เจตนาปกติ (Regular) และ S: เจตนาพิเศษ (Special)

2) แบบจำลองผลการกระทำ (Causation) กำหนดให้ N: ไม่เป็นอะไร (Nothing), I: บาดเจ็บ (Injury) และ D: ตาย (Death)

3) แบบจำลองพิจารณาโทษ (Impunity) กำหนดให้ E: ยกเว้นโทษ (Except), D: ลดโทษ (Decrease), N: ปกติ (Normal) และ I: เพิ่มโทษ (Increase)

4) แบบจำลองการพิจารณความผิด (Justification) กำหนดให้ N: ไม่มีความผิด (Notwrong) และ W: มีความผิด (Wrong)

5) กลุ่มความผิด มีรายละเอียด ดังนี้

A ไม่มีความผิดและไม่ต้องรับโทษ

B พยายามฆ่าผู้อื่นเจตนามีเหตุลดโทษ

C พยายามฆ่าผู้อื่น

D ฆ่าผู้อื่นโดยเจตนามีเหตุลดโทษ

E ฆ่าผู้อื่นโดยเจตนา

F พยายามฆ่าผู้อื่นโดยไตร่ตรอง

G ฆ่าผู้อื่นโดยไตร่ตรอง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข- 1 การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย

ลำดับ	เว็บไซต์ข่าว	ความเห็นนักกฎหมาย					การทำนายของแบบจำลอง					
		Intention	Causation	Impunity	Justification	กลุ่มความผิด	Intention	Causation	Impunity	Justification	กลุ่มความผิด	ความแม่นยำ
1	/666630	R	I	N	W	C	R	I	N	W	C	1
2	/702524	R	I	N	W	C	R	I	N	W	C	1
3	/697206	R	I	N	W	E	R	D	N	W	E	1
4	/744950	S	D	I	W	G	R	N	I	W	C	0
5	/697109	S	I	I	W	F	R	I	I	W	C	0
6	/744746	R	D	N	W	E	R	D	N	W	E	1
7	/696720	S	D	I	W	G	S	D	I	W	G	1
8	/696644	R	D	N	W	E	R	D	N	W	E	1
9	/744843	S	D	N	W	G	S	D	N	W	G	1
10	/697444	R	D	D	W	D	S	D	N	W	G	0
11	/697393	R	D	N	w	E	R	D	N	W	E	1
12	/695859	S	D	I	W	G	S	D	I	W	G	1
13	/695796	R	D	D	W	E	R	D	N	W	E	1
14	/695674	S	D	I	W	G	S	D	I	W	G	1
15	/725184	R	D	I	W	E	R	D	N	W	E	1
16	/695286	R	D	N	W	E	S	D	N	W	G	0
17	/744828	S	D	N	W	E	R	I	N	W	C	0
18	/695152	R	D	N	W	E	R	D	N	W	E	1
19	/695101	S	D	N	W	G	R	D	N	W	E	0
20	/744641	R	D	I	W	E	R	D	I	W	E	1
21	/694626	R	D	N	W	E	R	D	I	W	E	1
22	/742914	N	D	E	N	A	R	D	I	W	E	0
23	/755219	S	D	I	W	G	R	D	I	W	E	0
24	/753970	R	D	N	W	E	R	D	N	W	E	1

ตารางที่ ข-1 การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย (ต่อ)

ลำดับ	เว็บไซต์ข่าว	ความเห็นนักกฎหมาย					การทำนายของแบบจำลอง					
		Intention	Causation	Impunity	Justification	กลุ่มความผิด	Intention	Causation	Impunity	Justification	กลุ่มความผิด	ความแม่นยำ
25	/742749	R	D	N	W	E	R	I	N	W	A	0
26	/756231	R	D	N	W	E	R	D	N	W	E	1
27	/742607	S	D	N	W	G	S	I	N	W	F	0
28	/744348	R	D	N	W	E	R	N	E	W	B	0
29	/742971	R	D	I	W	E	S	D	I	W	G	0
30	/756498	S	D	I	W	G	R	I	I	W	C	0
31	/744195	S	D	N	W	G	S	D	N	W	G	1
32	/755891	S	D	N	W	G	S	D	N	W	G	1
33	/744598	R	D	N	W	E	R	D	N	W	E	1
34	/755449	R	D	I	W	G	R	D	N	W	E	0
35	/755411	S	D	N	W	G	S	D	N	W	G	1
36	/734348	N	D	E	N	A	R	D	I	W	E	0
37	/753499	S	D	I	W	G	R	N	N	W	C	0
38	/753487	S	D	N	W	G	S	D	N	W	G	1
39	/753477	R	D	E	W	D	R	D	N	W	E	0
40	/752743	R	D	N	W	E	R	D	N	W	E	1
41	/744658	R	D	I	W	G	R	I	I	W	C	0
42	/744185	R	I	N	W	C	R	I	N	W	C	1
43	/744183	R	D	N	W	E	R	D	N	W	E	1
44	/744132	S	I	N	W	F	R	N	N	W	C	0
45	/752090	R	D	I	W	C	R	D	N	W	E	0
46	/744074	R	D	I	W	G	R	I	I	W	C	0
47	/743784	R	D	D	W	D	R	I	N	W	C	0
48	/743797	R	D	N	W	E	R	D	N	W	E	1

ตารางที่ ข-1 การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย (ต่อ)

ลำดับ	เว็บไซต์ข่าว	ความเห็นนักกฎหมาย					การทำนายของแบบจำลอง					
		Intention	Causation	Impunity	Justification	กลุ่มความผิด	Intention	Causation	Impunity	Justification	กลุ่มความผิด	ความแม่นยำ
49	/751532	R	D	N	W	E	R	D	I	W	E	1
50	/751515	S	D	N	W	G	R	D	N	W	E	0
51	/751405	R	D	N	W	E	R	I	N	W	C	0
52	/751222	R	D	N	W	E	R	D	I	W	E	1
53	/751151	S	D	I	W	G	R	D	I	W	E	0
54	/751106	R	D	N	W	E	R	D	N	W	E	1
55	/744387	S	D	I	W	G	R	I	N	W	C	0
56	/750568	S	N	N	W	F	S	I	N	W	F	1
57	/750503	S	I	I	W	F	S	N	I	W	F	1
58	/750410	S	D	N	W	G	S	D	N	W	G	1
59	/744563	S	D	N	W	G	R	D	N	W	G	1
60	/750357	R	D	N	W	E	R	D	N	W	E	1
61	/750237	S	D	N	W	G	R	D	N	W	E	0
62	/744447	R	D	I	W	E	R	D	I	W	E	1
63	/744327	R	D	N	W	E	R	D	N	W	E	1
64	/750097	R	D	N	W	E	R	D	N	W	E	1
65	/743741	S	D	N	W	G	S	D	N	W	G	1
66	/743691	S	I	I	W	F	S	I	N	W	F	1
67	/743681	R	D	N	W	E	R	D	N	W	E	1
68	/749871	S	D	I	W	G	R	D	I	W	E	0
69	/743693	S	D	I	W	G	R	I	I	W	C	0
70	/744373	R	D	N	W	E	R	I	N	W	C	0
71	/749540	S	D	I	W	G	S	D	I	W	G	1
72	/748958	S	D	N	W	G	S	D	N	W	G	1

ตารางที่ ข-1 การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย (ต่อ)

ลำดับ	เว็บไซต์ข่าว	ความเห็นนักกฎหมาย					การทำนายของแบบจำลอง					
		Intention	Causation	Impunity	Justification	กลุ่มความผิด	Intention	Causation	Impunity	Justification	กลุ่มความผิด	ความแม่นยำ
73	/748819	R	I	N	W	C	S	N	N	W	F	0
74	/743576	S	D	I	W	G	R	I	N	W	C	0
75	/748720	R	D	N	W	E	R	D	N	W	E	1
76	/744764	S	D	N	W	G	R	I	N	W	C	0
77	/748395	S	D	I	W	G	R	D	I	W	E	0
78	/743507	S	D	I	W	G	R	I	N	W	C	0
79	/748172	R	D	N	W	E	R	D	N	W	G	0
80	/748125	S	D	I	W	G	S	D	I	N	G	1
81	/747870	R	D	N	W	E	R	D	N	W	E	1
82	/743394	R	I	D	W	B	R	N	N	W	C	0
83	/740779	S	D	I	W	G	R	D	I	W	E	0
84	/747335	S	D	N	W	G	R	D	N	W	E	0
85	/747248	S	D	N	W	G	S	D	N	W	G	1
86	/743261	R	D	N	W	E	R	D	I	W	E	1
87	/740605	R	D	D	W	D	R	I	N	W	E	0
88	/742423	R	D	I	W	E	R	D	I	W	E	1
89	/742151	R	D	N	W	E	R	D	N	W	E	1
90	/736436	S	D	N	W	G	S	D	N	W	G	1
91	/736277	R	D	N	W	E	R	D	N	W	E	1
92	/739944	R	I	N	W	C	R	I	N	W	C	1
93	/734766	R	D	N	W	E	R	D	N	W	E	1
94	/746016	R	D	N	W	E	R	D	N	W	E	1
95	/745813	S	D	N	W	G	S	D	N	W	G	1
96	/745805	R	D	N	W	E	R	D	I	W	E	1

ตารางที่ ข-1 การทดสอบประสิทธิภาพของแบบจำลองเทียบกับความเห็นของนักกฎหมาย (ต่อ)

ลำดับ	เว็บไซต์ข่าว	ความเห็นนักกฎหมาย					การทำนายของแบบจำลอง					
		Intention	Causation	Impunity	Justification	กลุ่มความผิด	Intention	Causation	Impunity	Justification	กลุ่มความผิด	ความแม่นยำ
97	/745398	R	D	N	W	E	R	D	N	W	E	1
98	/738863	S	D	I	W	G	S	I	N	W	F	0
99	/738869	R	D	I	W	E	R	D	I	W	E	1
100	/744964	R	D	N	W	E	R	D	N	W	E	1
ร้อยละความแม่นยำ												**59**

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	
วัน เดือน ปี เกิด	15 มกราคม 2527
สถานที่เกิด	บึงกาฬ
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
ที่อยู่ปัจจุบัน	183/19318 ถ.นาวงประชาพัฒนา แขวงสีกัน เขตดอนเมือง กรุงเทพฯ 10210
ผลงานตีพิมพ์	W. Krungklang and S. Sinthupinyo, "An Analysis of Natural Language Text Relating to Thai Criminal Law," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 2020, pp. 1-6.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University