การสร้างแบบจำลองโดยใช้การเรียนรู้เชิงลึกเพื่อจำแนกข้อความการสนทนาจากแอปพลิเคชันไลน์

Creating a deep learning model for classifying conversation messages from a line application

ไพชยนต์ คงไชย[1*]
Phaichayon Kongchai[1*]

Received: 22 January 2023; **Revised**: 21 March 2023; **Accepted**: 18 April 2023

บทคัดย่อ

งานวิจัยนี้ได้นำเสนอวิธีการจำแนกข้อความจากกลุ่มแชทในแอปพลิเคชันไลน์ของคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี เพื่อแจ้งเตือนเฉพาะบางข้อความที่เหมาะสม ซึ่งจะช่วยลดจำนวนการแจ้งเตือนไปยังผู้เชี่ยวชาญ การทดลองการทำนายด้วยการเปรียบเทียบ 5 อัลกอริทึม ดังนี้ อัลกอริทึม Random Forest อัลกอริทึม Naïve Bayes อัลกอริทึม Logistic Regression อัลกอริทึม Support Vector Classification และเทคนิคการเรียนรู้เชิงลึก อัลกอริทึม Long Short-Term Memory จากผลการวิจัยพบว่า อัลกอริทึม Long Short-Term Memory มีค่าความถูกต้องในการจำแนกมากที่สุดเท่ากับร้อยละ 90.66 มีค่าความแม่นยำและค่าความถ่วงดุลมากที่สุด เมื่อจำแนกข้อความประเภทข้อความเฉพาะเจาะจงหรือคำถามที่ต้องการผู้เชี่ยวชาญ มีค่าความระลึกและค่าความถ่วงดุลมากที่สุดเมื่อจำแนกข้อความประเภทข้อความทั่วไป การวิจัยชี้ให้เห็นว่าวิธีนี้สามารถนำไปใช้กับการแชทกลุ่มอื่นที่คล้ายคลึงกัน เพื่อปรับปรุงประสิทธิภาพในการส่งการแจ้งเตือน

คำสำคัญ: การจำแนกประเภทข้อความ, ไลน์แอปพลิเคชัน, การเรียนรู้เชิงลึก

Abstract

This report presents a method for classifying text from the chat group within the Line application of the Faculty of Science, Ubon Ratchathani University, to notify only relevant messages and reduce the number of notifications to experts. This experiment compared five predictive algorithms: Random Forest, Naïve Bayes, Logistic Regression, Support Vector Classification and Deep Learning named Long Short-Term Memory. The results showed that the Long Short-Term Memory algorithm had the highest accuracy of 90.66%, with the highest precision and recall when classifying specific targeted messages or questions that require expert attention, and with the highest precision and F-measure when classifying general targeted messages. Additionally, the research demonstrated that this method could be applied to similar chat groups to improve the efficiency of notification.

Keywords: text classification, line application, deep learning

[1] อาจารย์, สาขาวิทยาการข้อมูลและนวัตกรรมซอฟต์แวร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี 34190
[1] Lecturer, Major of Data Science and Software Innovation, Faculty of Science, Ubon Ratchathani University 34190
* Corresponding Email: Phaichayon.k@ubu.ac.th

บทนำ

การคัดเลือกนักศึกษาใหม่เข้าเรียนในคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีวิธีการคัดเลือกนักศึกษาหลายรอบ เช่น รอบใช้แฟ้มสะสมผลงาน รอบโควตารอบแอดมิชชั่น และรอบรับตรงอิสระ โดยแต่ละรอบจะประกอบไปด้วยขั้นตอนการรับสมัคร สอบสัมภาษณ์ สอบคัดเลือก แจ้งความจำนงยืนยันสิทธิเพื่อเข้าศึกษา ซึ่งแต่ละขั้นตอน ผู้สมัครอาจจะมีข้อสงสัยหรือพบปัญหาที่ต้องการคำตอบอย่างเร่งด่วน เช่น ชำระเงินยืนยันสิทธิ์ไม่ได้ หมดเวลายืนยันสิทธิ์ตอนไหน ไม่ยืนยันสิทธิ์ได้ไหม สาขานี้ต้องสอบสัมภาษณ์ที่ไหน ซึ่งคำถามเหล่านี้อาจจะส่งผลให้พลาดโอกาสการเข้าศึกษาได้ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีทราบถึงปัญหาดังกล่าว จึงเปิดช่องทางให้นักศึกษาติดต่อหลายช่องทาง ไม่ว่าจะเป็นช่องทางเพจบนเฟซบุ๊ค ช่องทางโทรศัพท์ ช่องทางอีเมล์ และช่องทางไลน์แอปพลิเคชันที่มีผู้สมัครติดต่อมามากที่สุด โดยจะมีผู้เชี่ยวชาญคอยให้คำตอบ แต่การที่จะทำให้ผู้สมัครได้รับประสบการณ์ที่ดีและได้รับคำตอบรวดเร็วที่สุด อีกด้านของการตอบคำถามคือผู้เชี่ยวชาญจะต้องคอยตอบคำถามอยู่ตลอดเวลา ซึ่งอาจส่งผลให้ผู้เชี่ยวชาญมีปัญหาเกี่ยวกับสุขภาพจิตสุขภาพกาย หรือความสัมพันธ์กับครับครัวน้อยลง เพราะต้องคอยตอบคำถามจากนักเรียนผ่านกลุ่มแชทในแอปพลิเคชันไลน์ตลอดเวลา ผู้วิจัยจึงได้ทำการเก็บข้อมูลจากไลน์กลุ่มแชทงานรับเข้าคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ตั้งแต่วันที่ 18/08/2022 ถึงวันที่ 12/12/2022 (จำนวน 117 วัน) พบว่ามีจำนวนข้อความทั้งหมด 3179 ข้อความ (ไม่รวมข้อความที่แจ้งเตือนเมื่อมีสมาชิกใหม่เข้าร่วมกลุ่ม) โดยมีจำนวนข้อความแจ้งเตือนต่อวันแสดงดัง Figure 1 มีข้อความแจ้งเตือนตามช่วงเวลาราชการ 1767 ข้อความ และข้อความแจ้งเตือนนอกเวลาราชการ 1412 ข้อความ ดัง Table 1 ผู้วิจัยจึงได้ทำการจำแนกข้อความแจ้งเตือนตามช่วงเวลา ดัง Table 2 จากทั้ง 2 ตารางจะเห็นได้ว่ามีข้อความแจ้งเตือนเข้ามาตลอดทุกช่วงเวลา และมีข้อความแจ้งเตือนนอกเวลาราชการมากถึงร้อยละ 44.57 แต่มีข้อความที่ผู้เชี่ยวชาญต้องตอบทั้งหมดเพียง 569 ข้อความหรือประมาณร้อยละ 18 โดยวัดจากการตอบจริงในกลุ่มแชทการใช้ปัญญาประดิษฐ์เข้ามาช่วยจำแนกข้อความเพื่อแจ้งเตือนเฉพาะบางข้อความที่เหมาะสม จะช่วยลดจำนวนการอ่านข้อความของผู้เชี่ยวชาญลงได้ถึง 2610 ข้อความหรือประมาณร้อยละ 82

Table 1 The number of chats in Working Hours.

Parts of the Day	#Chats
Official Working Hours (08.00 - 16.59)	1767
Outside Official Working Hours (17.00 - 07.59)	1412
Total	**3179**

Table 2 The number of chats per time period.

Parts of the Day	#Chats
Early Morning (05.00 - 07.59)	316
Morning (08.00 - 11.59)	804
Noon (12.00 - 15.59)	772
Evening (16.00 - 19.59)	857
Night (20.00 - 23.59)	407
Late Night (00.00 - 04.59)	23
Total	**3179**

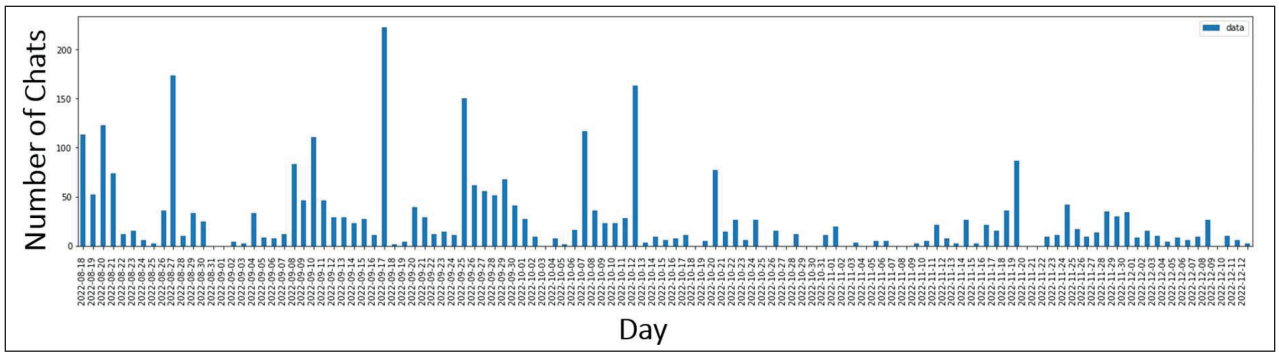

Figure 1 The number of chats per day (18/08/2022 to 12/12/2022).

ดังนั้นผู้วิจัยจึงได้ทำการศึกษางานวิจัยที่ใกล้เคียง พบว่ามีหลายงานวิจัยที่มุ่งเน้นทำวิจัยเกี่ยวกับการประมวลผลภาษาไทย เช่น (อิทธิศักดิ์ ศรีดำ, 2565) ระบบการจัดหมวดหมู่ข้อคิดเห็นและข้อเสนอแนะของประชาชนที่มีต่อโครงการของรัฐโดยวิธีปัญญาประดิษฐ์ เสนอวิธีการใช้อัลกอริทึมต้นไม้ตัดสินใจในการจัดหมวดหมู่ข้อคิดเห็น (มุกตา หมานเหม และคณะ, 2563) โมเดลสำหรับจำแนกความรู้สึกของความคิดเห็นโดยใช้เทคนิคต้นไม้ตัดสินใจกรณีศึกษา เว็บไซต์จองโรงแรม โดยงานวิจัยนี้ได้นำเสนอการทดลองการใช้เครื่องมือในการตัดคำไทยที่มีหลากหลายรูปแบบ ซึ่งเครื่องมือที่มีประสิทธิภาพมากที่สุด คือ NewMM (วสวัตติ์ อินทร์แปลง และคณะ, 2563) การวิเคราะห์ความคิดเห็นต่อเกมมือถือผับจีด้วยเหมืองข้อความ ได้เก็บรวบรวมข้อมูลความคิดเห็นต่อเกมมือถือผับจีจำนวน 3,798 ข้อความ และใช้เทคนิคการปรับความสมดุลของข้อมูลด้วยวิธี SMOTE เพื่อทำให้ข้อมูลของคลาสสมดุลกัน จากนั้นนำไปสร้างแบบจำลองด้วย 5 อัลกอริทึม ดังนี้ อัลกอริทึม Random Forest อัลกอริทึม Naïve Bayes อัลกอริทึม C4.5 อัลกอริทึม Support Vector Machine และอัลกอริทึม K-Nearest Neighbor เพื่อหาอัลกอริทึมที่มีประสิทธิภาพสูงสุด จากการทดลองพบว่าอัลกอริทึม K-Nearest Neighbor มีประสิทธิภาพในการจำแนกข้อมูลมากที่สุด (ศรัญญา กาญจนวัฒนา และคณะ, 2565) การจำแนกอารมณ์ของมนุษย์จากการรู้จำเสียงพูดโดยใช้การเรียนรู้เชิงลึก เพื่อจำแนกข้อมูล 5 อารมณ์ประกอบด้วย โกรธ ปกติ ประหลาดใจ มีความสุข และเศร้า ผลการทดลองสรุปว่า Long Short-Term Memory (LSTM) เป็นอัลกอริทึมที่เหมาะสมกับการจำแนกอารมณ์จากเสียงพูดมากกว่า Convolution Neural Networks (CNN)

จากการศึกษางานวิจัย ผู้วิจัยจึงได้นำข้อดีของแต่ละงานมาประยุกต์ใช้ และได้นำเสนอการสร้างแบบจำลองโดยใช้เทคนิคการเรียนรู้เชิงลึก เพื่อจำแนกข้อความการสนทนาจากแอปพลิเคชันไลน์ โดยขั้นตอนการทดลองได้เปรียบเทียบ 5 อัลกอริทึม ดังนี้ อัลกอริทึม Random Forest อัลกอริทึม Naïve Bayes อัลกอริทึม Logistic Regression อัลกอริทึม Support Vector Classification และเทคนิคการเรียนรู้เชิงลึก อัลกอริทึม Long Short-Term Memory โดยมีรายละเอียดของขั้นตอนการทดลองในหัวข้อถัดไป

วัตถุประสงค์ของการวิจัย

เพื่อศึกษาการจำแนกข้อความจากกลุ่มแชทในแอปพลิเคชันไลน์

วิธีดำเนินการวิจัย

งานวิจัยนี้ประกอบด้วย 4 ขั้นตอน คือ การเก็บรวบรวมข้อมูล การเตรียมข้อมูล การจำแนกประเภทข้อมูล และการวัดผลและการประเมินผลลัพธ์ โดยมีรายละเอียดดังนี้

การเก็บรวบรวมข้อมูล

ผู้วิจัยได้ทำการเก็บข้อมูลจากไลน์กลุ่มแชท งานรับเข้าศึกษาคณะวิทย์ มหาวิทยาลัยอุบลราชธานี ตั้งแต่วันที่ 18/08/2022 ถึงวันที่ 12/12/2022 มีจำนวนข้อความทั้งหมด 3179 ข้อความ โดยมีตัวอย่างดัง Figure 2

	date	data
2	2022-08-18 13:08:00	Song\tขออนุญาตสอบถามค่าา มีใครไปจากศรีสะเกษไหม...
3	2022-08-18 13:08:00	K.💔\t
4	2022-08-18 13:10:00	Song\tเราขอไอดีได้ไหมคะ อยากคุยด้วย
5	2022-08-18 13:10:00	K.💔 unsent a message.
6	2022-08-18 13:12:00	พพ 🙅\tขออนุญาตนะคะ มีใครไปคนเดียวมั้ยคะ คือเรา...
...	...	...
3679	2022-12-23 09:30:00	Ig ;\tท่าไมถึงขึ้นเอกสารไม่ครบหรอคะ @Tutiyaporn
3680	2022-12-23 10:08:00	Tutiyaporn\t@Ig ; รอทางรับเข้าตรวจสอบเอกสารนะค...
3681	2022-12-23 10:09:00	Tutiyaporn\t@Chinnawat ตามประกาศถือว่าสละสิทธิ...
3682	2022-12-23 10:10:00	Chinnawat\tแล้วถือว่าสละสิทธิ์การเข้าเรียนไหมครับ
3683	2022-12-23 10:14:00	Tutiyaporn\t@Chinnawat นร.ที่มีรายชื่อเป็นผู้ผ...

3179 rows × 2 columns

Figure 2 Examples of data set.

การเตรียมข้อมูล

1) ตรวจสอบความถูกต้องและติดฉลากข้อมูล นำข้อมูลจาก Figure 2 มาทำการตัดข้อความ “Unsent a message” (ข้อความที่มีการลบออก) และข้อความที่มีเฉพาะรูปภาพออก (เนื่องจากงานวิจัยนี้ยังไม่สามารถประมวลผลข้อมูลรูปภาพได้) ทำให้เหลือข้อความทั้งหมด 2784 ข้อความ จากนั้นทำความสะอาดข้อมูลด้วยการกำจัด ชื่อ สัญลักษณ์ที่ปรากฏในข้อความ แล้วทำการติดฉลากให้กับข้อความ (data labelling) ดัง Table 3 ในคอลัมน์ Class โดยหมายเลข 0 แทนข้อความทั่วไป (general chats) มีจำนวน 2215 ข้อความ และหมายเลข 1 แทนข้อความที่เป็นข้อคำถามที่ผู้เชี่ยวชาญต้องตอบ (specific chats) มีจำนวน 569 ข้อความ

2) การตัดคำ กำจัดคำหยุดและแก้ไขคำที่เรียงผิด งานวิจัยนี้ได้เลือกไลบรารี PyThaiNLP ในด้านการประมวลผลข้อมูลภาษาไทย โดยใช้เทคนิคการตัดคำด้วยอัลกอริทึม maximum matching (NewMM) กำจัดคำหยุดด้วย thai_stopwords ในคลังข้อมูล PyThaiNLP และแก้ไขคำที่เรียงผิดด้วยฟังก์ชัน Normalize (Phatthiyaphaibun *et al*, 2023)

Table 3 Examples of data labeling.

Chats	Class
มีใครยุมุกดาหารไหมงับ	0
เราอยู่อุบลค่า	0
ขอสอบถามหน่อยค่ะ เกรดสะสมเอาแค่ 4 เทอมหรอ	1
หนูสามารถโอนชำระได้ไหมคะ	1
สาขาไหนเธอ	0

Table 4 Examples of bag of words.

Rows	เกรด (1)	ชำระ (2)	อุบล (3)	สอบ (4)
1	0	0	0	0
2	0	0	1	0
3	1	0	0	0
4	0	1	0	0
5	0	0	0	0

3) การแปลงคุณลักษณะของข้อมูล ขั้นตอนนี้เป็นการแปลงคุณลักษณะของข้อมูล ให้อยู่ในรูปแบบที่สามารถนำไปประมวลผลเพื่อสร้างแบบจำลอง ซึ่งงานวิจัยนี้ได้เลือกการสร้างคลังคำศัพท์ (bag of word) ซึ่งเป็นการแปลงคำที่ไม่ซ้ำกันให้เป็นตัวบ่งชี้ของคำ โดยที่คำศัพท์ใดไม่ปรากฏในประโยค ตัวบ่งชี้ในประโยคนั้นจะมีค่าเป็น 0 แต่ถ้าคำศัพท์ใดปรากฏในประโยค ตัวบ่งชี้ในประโยคนั้นจะมีค่าเป็น 1 แสดงตัวอย่างดัง Table 4

การจำแนกประเภทข้อมูล

งานวิจัยนี้ได้เลือก 5 อัลกอริทึม โดยมี 4 อัลกอริทึมเป็นการเรียนรู้ของเครื่องแบบดั้งเดิม (traditional machine learning) ได้แก่ อัลกอริทึม Random Forest (RF) อัลกอริทึม Naïve Bayes (NB) อัลกอริทึม Logistic Regression (LR) และอัลกอริทึม Support Vector Classification (SVC) ด้วยค่าพารามิเตอร์ที่กำหนดไว้แล้ว (default parameter) และอีกหนึ่งอัลกอริทึมเป็นการเรียนรู้เชิงลึก (deep learning) คือ อัลกอริทึม Long Short-Term Memory (LSTM) โดยการทำงานของแต่ละอัลกอริทึมมีรายละเอียดดังนี้

อัลกอริทึม RF ใช้แนวคิดของการสร้างต้นไม้ตัดสินใจ (decision tree) ในการสร้างตัวแบบ โดยจะสร้างต้นไม้ตัดสินใจจำนวน N ต้น ตามผู้ใช้กำหนด งานวิจัยนี้ได้ใช้มาตรวัด Gini Index ดังสมการที่ (1) จากนั้นนำผลลัพธ์จากการทำนายของแต่ละต้นมาคิดเป็นคำตอบของตัวแบบ โดยเลือกคำตอบที่ซ้ำกันมากที่สุด (Pal, 2005)

$$\text{Gini}(t) = 1 - \sum_{j=1}^{n} p_j^{\,2} \qquad (1)$$

โดยที่

t แทนข้อมูลสำหรับฝึกและ p_j คือ สัดส่วนของจำนวนข้อมูลที่อยู่ในแต่ละกลุ่ม โดย j เป็นจำนวนกลุ่ม นอกจากนี้งานวิจัยนี้มีการมีใช้เทคนิค Grid Search เพื่อหาค่าพารามิเตอร์ที่เหมาะสม โดยมีค่า n_estimators เริ่มต้น 10 ถึง 100 (ปรับค่าเพิ่มทีละ 10) ค่า max_depth เริ่มต้น 5 ถึง 20 (ปรับค่าเพิ่มทีละ 5) และไม่จำกัดความลึก

อัลกอริทึม NB ใช้แนวคิดความน่าจะเป็นด้วยทฤษฎีของเบย์ โดยการหาความสัมพันธ์ของระหว่างตัวแปร เพื่อใช้ในการสร้างตัวแบบ (Rish, 2001) ดังสมการที่ (2)

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \quad (2)$$

โดยที่

$P(C_i \mid X)$ คือ ค่าความน่าจะเป็นที่เกิดแอททริบิวต์ X ก่อนความน่าจะเป็น C_i

$P(C_i)$ คือ ค่าความน่าจะเป็นในการเกิดคลาส C_i

P(X) คือ ค่าความน่าจะเป็นในการเกิดแอททริบิวต์ X

งานวิจัยนี้ได้สร้างตัวแบบจาก Multinomial Naive Bayes และมีการเทคนิคใช้ Grid Search เพื่อหาค่าพารามิเตอร์ที่เหมาะสม โดยมีค่า Alpha เริ่มต้น 0.01 ถึง 10.00 (ปรับค่าเพิ่มทีละ 10 เท่า)

อัลกอริทึม LR ใช้แนวคิดทางสถิติที่วิเคราะห์สมการแบบถดถอย เพื่อทำนายโอกาสที่จะเกิดเหตุการณ์ที่สนใจ แล้วใช้ฟังก์ชันทดสอบสมมติฐานเพื่อจำแนกข้อมูล (Antipov & Pokryshevskaya, 2010)

$$\log\left(\frac{P(y)}{1-P(y)}\right) = b_0 + b_1x_1 + \ldots b_nx_n \quad (3)$$

โดยที่

P(y) คือ ความน่าจะเป็นในการเกิดเหตุการณ์ที่สนใจ

Q(y) คือ ความน่าจะเป็นเหตุการณ์ที่สนใจไม่เกิด

b คือ ค่าสัมประสิทธิ์การถดถอย

x คือ แอททริบิวต์

ซึ่งการประมาณค่าสัมประสิทธิ์การถดถอยจะใช้ความน่าจะเป็นสูงสุด แล้วเทียบกับผลการทำนาย เพื่อหาค่าทำนายของตัวแปรให้ใกล้เคียงกับข้อมูลจริงมากที่สุด นอกจากนี้งานวิจัยนี้มีการมีใช้เทคนิค Grid Search เพื่อหาค่าพารามิเตอร์ที่เหมาะสม โดยมีค่า Penalty เป็น L1 และ L2 ค่า C เริ่มต้น 0.01 ถึง 100 (ปรับค่าเพิ่มทีละ 10 เท่า)

อัลกอริทึม SVC (Cortes *et al*, 1995) ใช้แนวคิดการหาสัมประสิทธิ์ของสมการ เพื่อสร้างเส้นแบ่งแยกข้อมูล (hyperplane) โดยมีเส้นขอบ (margin) ของเส้นตรงที่เป็นเส้นแบ่ง เส้นขอบที่แบ่งกลุ่มกว้างมากที่สุดของทั้งสองกลุ่ม แต่ถ้าข้อมูลไม่เป็นเชิงเส้นจะใช้เคอร์แนว (Kernel) ฟังก์ชันเพื่อให้สามารถจำแนกข้อมูลบนระนาบได้หลายมิติ และเวกเตอร์ที่อยู่ข้างระนาบจะเรียกว่าเวกเตอร์สนับสนุน (support vectors) นอกจากนี้งานวิจัยนี้มีการมีใช้เทคนิค Grid Search (Bui *et al*, 2020). เพื่อหาค่าพารามิเตอร์ที่เหมาะสม โดยมีค่า C อยู่ระหว่าง 0.01 ถึง 100 (ปรับค่าเพิ่มทีละ 10 เท่า) ค่า gamma อยู่ระหว่าง 0.01 ถึง 100 (ปรับค่าเพิ่มทีละ 10 เท่า) และ Kernel มีค่าเป็น linear, rbf และ poly

อัลกอริทึม LSTM (Schmidhuber *et al*, 1997)ใช้แนวคิดของโครงข่ายประสาทเทียม (artificial neural network) โดยเลียนแบบการทำงานคล้ายกับสมองมนุษย์ มีพื้นฐานมาจากอัลกอริทึม Recurrent Neural Network (RNN) ซึ่งเป็นตัวแบบประมวลผลข้อมูลที่มีการเชื่อมต่อกันอยู่ระหว่างชั้นของข้อมูล ซึ่งจะช่วยให้ตัวแบบสามารถรับรู้ความสัมพันธ์ของข้อมูลได้อย่างมีประสิทธิภาพ อัลกอริทึม LSTM ถูกพัฒนาให้มีความสามารถในการจดจำข้อมูลในระยะเวลายาว ๆ โดยมีการเพิ่มชั้นของข้อมูล ซึ่งจะช่วยให้แบบจำลองสามารถจดจำข้อมูลนานขึ้นได้ ปกติจะใช้ในงานประมวลผลข้อมูลภาษามนุษย์ที่มีความซับซ้อนสูง โดยผู้วิจัยได้ออกแบบปรับแต่งขั้นตอนการทำงานของอัลกอริทึม LSTM ดัง Figure 3 มีการปรับอัตราการเรียนรู้ที่ 0.0001 ขนาดของ batch เป็น 32 และประมวลผล 100 อีพ็อค ด้วย Adam Optimizer

```python
def simple_LSTM(MaxLength):
  inputlayer = Input((MaxLength,))
  x = Embedding(NumWords+2, 400)(inputlayer)
  x = Conv1D(32,3,padding='same',activation='relu')(x)
  x = MaxPool1D()(x)

  x = LSTM(32)(x)
  x = Dropout(0.65)(x)
  out = Dense(2, activation="softmax")(x)

  return out,inputlayer
```

Figure 3 LSTM python code.

Table 5 LSTM Description.

Line	Describe
3	ทำการแปลงคำให้เป็นเวกเตอร์ด้วย Word Embedding โดยมีข้อมูลอินพุตเท่ากับจำนวนคำทั้งหมดของประโยคมากที่สุดและข้อมูลเอาต์พุตเท่ากับ 400
4	ชั้นคอลโวลูชันแบบ 1 มิติ ประกอบด้วย 32 filter ความกว้างของ kernel เป็น 3 และกำหนด padding เป็น same ที่ใช้ในการเพิ่มขนาดของ input โดยไม่เปลี่ยนขนาดของ output และฟังก์ชันกระตุ้นเป็น relu
5	ทำการเปลี่ยนขนาดของข้อมูลให้เล็กลงและเอาเฉพาะค่าที่สำคัญมาเก็บไว้ ด้วย MaxPool ขนาด 1 มิติ
7	ชั้น LSTM ที่มีขนาด hidden state เท่ากับ 32
8	ทำการ Dropout ด้วยอัตราส่วน 0.65 (ตัวเลขนี้ได้จากการทดลองใน Table 6) เพื่อป้องกันแบบจำลองเกิดการ Overfitting
9	สุดท้าย คือ ชั้นที่เชื่อมโยงกับข้อมูลชุด x ก่อนหน้า ซึ่งจะมีจำนวนโหนด 2 โหนด และฟังก์ชันกระตุ้นเป็น softmax เป็นฟังก์ชันที่คำนวณความน่าจะเป็นของแต่ละคลาส และค่าที่คำนวณจะอยู่ในช่วง [0,1]

จาก Figure 3 ผู้วิจัยได้กำหนดตัวแปรเริ่มต้นของ MaxLength มีค่าเท่ากับ 30 (ความยาวของประโยคมากที่สุด) และ NumWords มีค่าเท่ากับ 20000 (จำนวนคำทั้งหมด) โดยคำสั่งที่สำคัญอธิบายดัง Table 5 และสรุปจำนวนพารามิเตอร์ทั้งหมดที่ใช้ ดัง Figure 4

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 30)]              0

embedding (Embedding)        (None, 30, 400)           8000800

conv1d (Conv1D)              (None, 30, 32)            38432

max_pooling1d (MaxPooling1D  (None, 15, 32)            0
)

lstm (LSTM)                  (None, 32)                8320

dropout (Dropout)            (None, 32)                0

dense (Dense)                (None, 2)                 66

=================================================================
Total params: 8,047,618
Trainable params: 8,047,618
Non-trainable params: 0
```

Figure 4 All of the parameters.

Table 6 Dropout testing with LSTM algorithm (ACC: accuracy).

No.	Dropout	Train/ACC	Test/ACC
1	0.00	100.00	88.27
2	0.10	100.00	**90.66***
3	0.20	100.00	90.19
4	0.30	100.00	90.31
5	0.40	100.00	89.92
6	0.50	100.00	90.19
7	0.60	100.00	89.59
8	0.70	100.00	89.83
9	0.80	99.96	89.83
10	0.90	98.80	89.95

การวัดผลและการประเมินผลลัพธ์

งานวิจัยนี้ได้ทำการเปรียบเทียบประสิทธิภาพของตัวแบบที่สร้างขึ้นด้วยการใช้เกณฑ์ค่าความถูกต้อง (accuracy) ค่าความแม่นยำ (precision) ค่าความระลึก (recall) และค่าความถ่วงดุล (F-measure) ดังสมการที่ 4, 5, 6 และ 7 ตามลำดับ ข้อมูลสำหรับการทดสอบเป็นข้อมูลสำหรับฝึกปริมาณ 2 ใน 3 ของข้อมูลทั้งหมด และข้อมูลสำหรับทดสอบปริมาณ 1 ใน 3 ของข้อมูลทั้งหมด

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

โดยที่

TP คือ จำนวนที่ทำนายถูกคลาส Positive

TN คือ จำนวนที่ทำนายถูกคลาส Negative

FP คือ จำนวนที่ทำนายผิดคลาส Positive

FN คือ จำนวนที่ทำนายผิดคลาส Negative

ผลการทดลอง

งานวิจัยนี้ได้ทำการจำแนกข้อความงานรับเข้า เพื่อแจ้งเตือนเฉพาะบางข้อความที่ต้องให้ผู้เชี่ยวชาญตอบ โดยใช้ 5 อัลกอริทึม ดังนี้ อัลกอริทึม RF อัลกอริทึม NB อัลกอริทึม LR อัลกอริทึม SVC และอัลกอริทึม LSTM การทดลองโดยเขียนคำสั่งด้วยภาษา Python ไลบรารี่ Scikit-Learn และ Keras ข้อมูลที่ใช้ในงานวิจัยนี้เก็บข้อมูลจากกลุ่มแชทในแอปพลิเคชันไลน์ งานรับเข้าคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ตั้งแต่วันที่ 18/08/2022 ถึงวันที่ 12/12/2022 มีจำนวนข้อความทั้งหมด 3179 ผ่านการเตรียมข้อมูลแล้วเหลือ 2784 ข้อความ แบ่งเป็นข้อมูลสำหรับฝึกด้วยวิธีการสุ่มได้ 1948 ข้อความ และเป็นข้อมูลสำหรับทดสอบสุ่มได้ 836 ข้อความ แต่ด้วยชุดข้อมูลสำหรับฝึกมีคลาสที่ไม่สมดุลกัน โดยคลาส 0 มีจำนวนข้อมูล 1550 ข้อความ ส่วนคลาส 1 มีจำนวนข้อมูล 398 ข้อความ ผู้วิจัยได้ใช้เทคนิคการปรับข้อมูลให้สมดุล SMOTE (Chawla *et al*, 2002) โดยการสุ่มเพิ่มข้อมูลจากคลาสจำนวนน้อยให้มีค่าเท่ากันกับคลาสจำนวนมาก (over sampling) และวัดประสิทธิภาพของแบบจำลองด้วยค่าความถูกต้อง (ค่าความแม่นยำ และค่าความระลึก แสดงใน Table 7 - 9

Table 7 The testing results of accuracy.

Algorithms	Test/ACC
RF	84.80
NB	78.82
LR	85.52
SVC	83.97
LSTM	90.66*

จาก Table 7 แสดงให้เห็นว่าอัลกอริทึม LSTM มีค่าความถูกต้องมากที่สุด คือ ร้อยละ 90.66 และอัลกอริทึม LR อัลกอริทึม RF อัลกอริทึม SVC และอัลกอริทึม NB มีค่าความถูกต้องรองลงมาตามลำดับ

Table 8 The testing results of general chats (Class = 0).

Algorithms	Precision	Recall	F-measure
RF	**97.00***	84.00	90.00
NB	96.00	76.00	85.00
LR	95.00	87.00	90.00
SVC	94.00	85.00	89.00
LSTM	93.00	94.00*	94.00*

จาก Table 8 อัลกอริทึม RF มีค่าความแม่นยำในการทำนายคลาสข้อความทั่วไปมากที่สุด คือ ร้อยละ 97 และอัลกอริทึม LSTM มีค่าความระลึกและค่าความถ่วงดุลที่สูงสุดระดับที่เท่ากัน คือ ร้อยละ 94

Table 9 The testing results of specific chats (Class = 1).

Algorithms	Precision	Recall	F-measure
RF	58.00	**89.00***	71.00
NB	49.00	89.00	63.00
LR	61.00	81.00	70.00
SVC	58.00	78.00	67.00
LSTM	**75.00***	74.00	**75.00***

จาก Table 9 อัลกอริทึม RF มีค่าความความระลึกในการทำนายคลาสข้อความเฉพาะเจาะจงมากที่สุด คือ ร้อยละ 89 และอัลกอริทึม LSTM มีค่าความแม่นยำและความถ่วงดุลสูงที่สุดระดับที่เท่ากัน คือ ร้อยละ 75

สรุปผลและอภิปรายผล

งานวิจัยนี้ได้นำเสนอวิธีการจำแนกข้อความจากกลุ่มแชทในแอปพลิเคชันไลน์ งานรับเข้าคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี เพื่อแจ้งเตือนเฉพาะบางข้อความที่เหมาะสม ช่วยลดจำนวนการอ่านข้อความของผู้เชี่ยวชาญลง โดยการทดลองผู้วิจัยได้ทำการทดสอบด้วยวิธีการเปรียบเทียบมาตรวัด ค่าความถูกต้อง ค่าความแม่นยำ ค่าความระลึก และค่าความถ่วงดุล กับ 5 อัลกอริทึม ดังนี้ อัลกอริทึม RF อัลกอริทึม NB อัลกอริทึม LR อัลกอริทึม SVC และอัลกอริทึม LSTM จากผลการวิจัยพบว่าอัลกอริทึม LSTM มีประสิทธิภาพมากที่สุดในการจำแนกข้อความ เพราะมีค่าความถูกต้องมากที่สุด ค่าความแม่นยำมากที่สุด เมื่อคลาสเป็น 1 ค่าความระลึกมากที่สุด เมื่อคลาสเป็น 0 และค่าความถ่วงดุลมากที่สุด เมื่อคลาสเป็น 0 และ 1 จะเห็นได้ชัดว่าอัลกอริทึม LSTM ไม่ได้มีค่าความแม่นยำสูงที่สุดในคลาสเป็น 0 แต่ในคลาสเป็น 1 อัลกอริทึม LSTM มีค่าความแม่นยำสูงที่สุด (คลาสเป็น 1 มีอีกหนึ่งความหมายคือข้อความที่ผู้เชี่ยวชาญต้องตอบ) อัลกอริทึม LSTM สามารถจำแนกข้อมูลได้ดีในทั้งสองคลาสมากกว่าอัลกอริทึมอื่นที่สามารถจำแนกข้อมูลได้ดีในคลาสเดียว เพราะมีจำนวนพารามิเตอร์มากถึงแปดล้านตัว ดังนั้นอัลกอริทึม LSTM จึงเหมาะสมนำไปสร้างแบบจำลอง เพื่อใช้งานในการจำแนกข้อความกลุ่มแชทในแอปพลิเคชันไลน์ งานรับเข้าคณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ซึ่งสอดคล้องกับงานวิจัย (ศรัญญา กาญจนวัฒนา และคณะ, 2565) ที่มีค่าความถูกต้องมากที่สุดในการจำแนกข้อมูลที่เป็นข้อความซับซ้อน แต่อัลกอริทึม LSTM ไม่สามารถจำแนกข้อมูลได้ถูกต้องทั้งหมด มีบางข้อความที่ทำนายผิดพลาด คือ 1) ข้อความที่ต้องการผู้เชี่ยวชาญแต่แบบจำลองทำนายเป็นข้อมูลทั่วไป เช่น “สาขาวิทยาการข้อมูล ต้องยื่นพอร์ตทุกคนใช่มั้ยคะรวมถึงเด็กซิ่ว” 2) ข้อความทั่วไปที่ผู้ใช้ถามเพื่อนในกลุ่ม แต่แบบจำลองทำนายเป็นข้อความที่ต้องการผู้เชี่ยวชาญ เช่น “มีกลุ่มไลน์วิทชีวะไหมคะ” การทำวิจัยต่อไปควรนำข้อความคำถามที่ได้จากการทำนายของแบบจำลอง ไปแยกประเภทคำถามย่อยเฉพาะด้าน เพื่อให้โปรแกรมสามารถตอบคำถามได้อย่างอัตโนมัติแทนผู้เชี่ยวชาญ จะช่วยลดงานของผู้เชี่ยวชาญ และส่งผลทางอ้อมจะช่วยลดความเครียดจากการทำงานของผู้เชี่ยวชาญ

เอกสารอ้างอิง

มุกตา หมานเหม, สิทธิพงศ์ ด่ล๊ะ, สารภี จุลแก้ว, และสุภาวดี มากอ้น. (2563). โมเดลสำหรับจำแนกความรู้สึกของความคิดเห็นโดยใช้เทคนิคต้นไม้ตัดสินใจกรณีศึกษาเว็บไซต์จองโรงแรม. *วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา, 2*(1), 69-79.

วสวัตติ์ อินทร์แปลง, จารี ทองคำ. (2563). การวิเคราะห์ความคิดเห็นต่อเกมมือถือพับจีด้วยเหมืองข้อความ. *วารสารวิทยาศาสตร์และเทคโนโลยีมหาวิทยาลัยมหาสารคาม, 39*(5), 523-531.

ศรัญญา กาญจนวัฒนา, อัษฎายุธ จารัตน์ และปัญญ์ ชลีปราณีตพลกรัง. (2565). การจำแนกอารมณ์ของมนุษย์จากการรู้จำเสียงพูดโดยใช้การเรียนรู้เชิงลึก. *วารสาร วิทยาศาสตร์และเทคโนโลยีมหาวิทยาลัยราชภัฏศรีสะเกษ, 2*(2), 1-11.

อิทธิศักดิ์ ศรีดำ. (2565). ระบบการจัดหมวดหมู่ข้อคิดเห็นและข้อเสนอแนะของประชาชนที่มีต่อโครงการของรัฐโดยวิธีปัญญาประดิษฐ์. *วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยมหาสารคาม, 41(3)*, 134-141.

Antipov, E. & Pokryshevskaya, E. (2010). Applying CHAID for logistic regression diagnostics and classification accuracy improvement. *Journal of Targeting, Measurement and Analysis for Marketing, 18*(2), 109-117.

Bui, D. T. & Tsangaratos, P. & Nguyen, V. T. & Van Liem, N., & Trinh, P. T. (2020). Comparing the prediction performance of a Deep Learning Neural Network model with conventional machine learning models in landslide susceptibility assessment. *Catena, 188*, 104426.

Chawla, N. V. & Bowyer, K. W. & Hall, L. O. & Kegelmeyer, P. W. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321-357.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273-297.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing, 26*(1), pp. 217-222.

Phatthiyaphaibun, W., Chaovavanich, K., Polpanumas, C., Suriyawongkul, A., Lowphansirikul, L., & Chormai, P. (22 january 2023). *PyThaiNLP: Thai natural language processing in Python*. https://pythainlp.github.io/docs/3.1/

Rish, I. (2001). An empirical study of the naive Bayes classifier. *Proceedings of the International Joint Conference on Artificial Intelligence, 3*(22), 41-46.

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