

การจำแนกระดับความซับซ้อนการพัฒนาซอฟต์แวร์จากข้อความเอกสารความต้องการโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติและปัญญาประดิษฐ์เชิงรู้สร้าง

นายวิทยเดช ใจกว้าง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาระบบสารสนเทศเพื่อการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

การจำแนกระดับความซับซ้อนการพัฒนาซอฟต์แวร์จากข้อความเอกสารความต้องการโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติและปัญญาประดิษฐ์เชิงรู้สร้าง

นายวิทยเดช ใจกว้าง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาระบบสารสนเทศเพื่อการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ



ใบรับรองการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การจำแนกระดับความซับซ้อนการพัฒนาซอฟต์แวร์จากข้อความเอกสารความต้องการโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติและปัญญาประดิษฐ์เชิงรู้สร้าง

โดย นายวิทยเดช ใจกว้าง

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ

.....
(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)
คณบดีคณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล

คณะกรรมการสอบการค้นคว้าอิสระ

.....
(ผู้ช่วยศาสตราจารย์ ดร.เดช ธรรมศิริ)
ประธาน

.....
(รองศาสตราจารย์ ดร.พยุง มีสัจ)
อาจารย์ที่ปรึกษา

.....
(ผู้ช่วยศาสตราจารย์ ดร.วัชรีวรรณ จิตต์สกุล)
กรรมการ

ชื่อ : นายวิทยเดช ใจกว้าง
ชื่อการค้นคว้าอิสระ : การจำแนกระดับความซับซ้อนการพัฒนาซอฟต์แวร์จากข้อความเอกสารความต้องการโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติและปัญญาประดิษฐ์เชิงรู้สร้าง
สาขาวิชา : ระบบสารสนเทศเพื่อการจัดการ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : : รองศาสตราจารย์ ดร.พยุง มีสัจ
ปีการศึกษา : 2568

บทคัดย่อ

การประเมินค่าความพยายามในการพัฒนาซอฟต์แวร์ถือเป็นหัวใจสำคัญของการวางแผนโครงการทั้งในด้านระยะเวลาและงบประมาณ โดยเฉพาะอย่างยิ่งในการจัดจ้างผู้ให้บริการภายนอกที่มักเสนอรายละเอียดในรูปแบบข้อความไร้โครงสร้าง ซึ่งทำให้กระบวนการตรวจสอบมีความซับซ้อน ต้องอาศัยประสบการณ์สูง และเสี่ยงต่อความล่าช้า งานวิจัยนี้จึงมีวัตถุประสงค์เพื่อพัฒนาแบบจำลองพยากรณ์ค่าความพยายามโดยประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (NLP) ร่วมกับโมเดลภาษาขนาดใหญ่ (LLMs) เริ่มต้นจากการขยายความบริบทของเนื้องานด้วยโมเดล Gemma และทำการสกัดฟีเจอร์เพื่อสร้าง Sentence Embeddings โดยในการศึกษานี้ได้ทำการเปรียบเทียบประสิทธิภาพระหว่างโมเดลภาษาไทย WangchanBERTa และโมเดลที่รองรับหลายภาษาอย่าง XLM-RoBERTa ร่วมกับการประยุกต์ใช้อัลกอริทึมการเรียนรู้ของเครื่อง (Machine Learning) พื้นฐาน จำนวน 5 รูปแบบ ได้แก่ HistGradientBoosting, RandomForestClassifier, Support Vector Classifier, KNeighborsClassifier และ ExtraTreesClassifier เพื่อจำแนกความเหมาะสมของค่า Man-Day ออกเป็นกลุ่มตามเกณฑ์ที่กำหนด ผลการทดสอบประสิทธิภาพของแบบจำลองด้วยค่า Accuracy, Precision, Recall และ F1-Score ซึ่งช่วยยกระดับความโปร่งใสและความเชื่อมั่นในกระบวนการตัดสินใจ

(มีจำนวนทั้งสิ้น 69 หน้า)

คำสำคัญ : การประมวลผลภาษาธรรมชาติ, โมเดลภาษาขนาดใหญ่, ปัญญาประดิษฐ์

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Wittayadet Jaikwang
Independent Study Title : Classification of Software Development Task Complexity Levels from Requirement Documents Using Natural Language Processing and Generative AI
Major Field : Management Information Systems
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : : Associate Professor Dr. Phayung Meesad
Academic Year : 2025

ABSTRACT

Software development effort estimation is a crucial component of project planning in terms of scheduling and budgeting. This is particularly evident in outsourcing, where external service providers frequently submit project details in unstructured text formats. Consequently, the verification process becomes complex, heavily reliant on expert experience, and susceptible to delays. Therefore, this research aims to develop an effort estimation model by integrating Natural Language Processing (NLP) techniques with Large Language Models (LLMs). The methodology begins with task context expansion using the Gemma model, followed by feature extraction to generate Sentence Embeddings. The study compares the performance of the Thai language model, WangchanBERTa, and the multilingual model, XLM-RoBERTa. These are utilized alongside Machine Learning algorithms namely HistGradientBoosting, RandomForestClassifier, Support Vector Classifier, KNeighborsClassifier, and ExtraTreesClassifier to classify the appropriateness of Man-Day values into groups based on predefined criteria. The model's performance is evaluated using Accuracy, Precision, Recall, and F1-Score, ultimately enhancing transparency and confidence in the decision-making process.

(Total 69 Pages)

Keywords: Natural Language Processing, Large Language Models, AI

_____ Advisor

กิตติกรรมประกาศ

การศึกษาค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาและความอนุเคราะห์อย่างดียิ่งจาก รองศาสตราจารย์ ดร.พยุง มีสัจ อาจารย์ที่ปรึกษาหลัก และผู้ช่วยศาสตราจารย์ ดร.วัชรีวรรณ จิตต์สกุล กรรมการสอบการศึกษาค้นคว้าอิสระ ตลอดจนผู้ช่วยศาสตราจารย์ ดร.เดช ธรรมศิริ ประธานสอบการศึกษาค้นคว้าอิสระ ที่ได้กรุณาสละเวลาให้คำปรึกษา ข้อเสนอแนะ และข้อคิดเห็นอันเป็นประโยชน์ รวมทั้งตรวจแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่ในทุกขั้นตอนของการดำเนินงานวิจัย คณะผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณคณาจารย์ทุกท่านในภาควิชาการจัดการเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ได้ถ่ายทอดองค์ความรู้ ประสบการณ์ และแนวคิดทางวิชาการอันทรงคุณค่า ซึ่งเป็นพื้นฐานสำคัญในการพัฒนางานวิจัยฉบับนี้ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบคุณเพื่อนร่วมรุ่น สาขาวิชาระบบสารสนเทศเพื่อการจัดการ ที่ได้ให้ความช่วยเหลือ คำแนะนำ แลกเปลี่ยนความคิดเห็น และเป็นกำลังใจที่ดีตลอดระยะเวลาการศึกษา

ขอกราบขอบพระคุณครอบครัวผู้เป็นกำลังใจสำคัญ ที่คอยสนับสนุน เข้าใจ และให้แรงใจอย่างสม่ำเสมอ รวมทั้งบุพการีและญาติพี่น้องทุกท่านที่ได้ให้ความรัก ความห่วงใย และการสนับสนุนในทุกด้าน ซึ่งเป็นพลังสำคัญที่ทำให้การศึกษาค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงด้วยดี

ท้ายที่สุดนี้ คุณค่าและประโยชน์อันพึงมีจากการศึกษาค้นคว้าอิสระฉบับนี้ คณะผู้วิจัยขอน้อมมอบเป็นกตัญญูกตเวทีแด่ผู้มีพระคุณทุกท่าน ตลอดจนบุพการีผู้ให้การอบรมสั่งสอน สนับสนุน และเป็นพลังสำคัญในการผลักดันให้การศึกษาในครั้งนี้สำเร็จลุล่วงด้วยดี

วิทยเดช ใจกว้าง

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูปภาพ ..... ซ
บทที่ 1 บทนำ 9
1.1 ความเป็นมาและความสำคัญของปัญหา 9
1.2 วัตถุประสงค์ของการวิจัย 10
1.3 ขอบเขตของการวิจัย 11
1.4 สมมติฐานการวิจัย 12
1.5 วิธีการวิจัย 12
1.6 ประโยชน์ที่คาดว่าจะได้รับ 13
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง 15
2.1 การประมาณการความพยายามในการพัฒนาซอฟต์แวร์ 15
2.2 การประมวลผลภาษาธรรมชาติและการแทนค่าข้อความ 16
2.3 สถาปัตยกรรมทรานส์ฟอร์เมอร์และแบบจำลองภาษาพหุรูปแบบ 17
2.4 การเรียนรู้ของเครื่องและการเรียนรู้แบบกลุ่ม 18
2.5 มาตรวัดประสิทธิภาพของแบบจำลอง 20
2.6 งานวิจัยที่เกี่ยวข้อง 21
2.7 การวิเคราะห์ช่องว่างทางการวิจัย 24
บทที่ 3 วิธีดำเนินการวิจัย 25
3.1 กรอบแนวคิดในการวิจัยและสถาปัตยกรรมระบบ 25

3.2 การรวบรวมและเตรียมข้อมูล 28
3.3 การสกัดคุณลักษณะด้วยโมเดลพหุรูปแบบ 29
3.4 การสร้างตัวแทนข้อมูลและวิศวกรรมคุณลักษณะ 31
3.5 การสร้างแบบจำลองและการหาค่าพารามิเตอร์ที่เหมาะสม 32
3.6 การประเมินประสิทธิภาพและสภาพแวดล้อมในการทดลอง 33
บทที่ 4 ผลการดำเนินการวิจัย 35
4.1 ภาพรวมการทดลองและการสำรวจข้อมูล 35
4.2 ผลการสกัดคุณลักษณะด้วยแบบจำลองภาษาขนาดใหญ่ 37
4.3 ผลการสร้างตัวแทนข้อความ 39
4.4 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองแปลงข้อความ 41
4.5 ผลการทดลองแบบจำลองการเรียนรู้ของเครื่องและการหาพารามิเตอร์ 42
4.6 การประเมินความสามารถในการแยกแยะด้วยกราฟเส้นโค้ง ROC 45
4.7 การวิเคราะห์เมทริกซ์ความสับสน 48
4.8 การวิเคราะห์กรณีศึกษาข้อผิดพลาด 50
4.9 การเชื่อมโยงผลการทดลองกับวัตถุประสงค์การวิจัย 52
บทที่ 5 สรุป อภิปรายผล และข้อเสนอแนะ 55
5.1 สรุปผลการวิจัย 55
5.2 การอภิปรายผล 56
5.3 ประโยชน์ของงานวิจัย 58
5.4 ข้อจำกัดของงานวิจัย 59
5.5 ข้อเสนอแนะสำหรับการทำวิจัยในอนาคต 59
บรรณานุกรม 61
ภาคผนวก ก 64
ประวัติผู้เขียน 67

สารบัญตาราง

หน้า

ตารางที่ 2-1 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้อง 23
ตารางที่ 3-1 ตัวอย่างการแปลงข้อมูลนำเข้าเป็นคำอธิบายขยายความ Gemma Text 30
ตารางที่ 4-1 แสดงตัวอย่างเปรียบเทียบระหว่างข้อความต้นฉบับและคุณลักษณะที่สกัดได้ 38
ตารางที่ 4-2 เปรียบเทียบประสิทธิภาพระหว่างแบบจำลอง WangchanBERTa และ XLM-RoBERTa . 41
ตารางที่ 4-3 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละอัลกอริทึม 42
ตารางที่ 4-4 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องกรณี WangchanBERTa 44
ตารางที่ 4-5 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องกรณี XLM-RoBERTa 44
ตารางที่ 4-6 สรุปการประเมินผลลัพธ์เทียบกับวัตถุประสงค์การวิจัย 53

พระจอมเกล้าพระนครเหนือ มหาวิทยาลัยเทคโนโลยี

สารบัญรูปภาพ

หน้า

ภาพที่ 2-1 Soft Voting diagram 20
ภาพที่ 3-1 กรอบแนวคิดการวิจัย 27
ภาพที่ 3-2 สถาปัตยกรรมการเรียนรู้ของเครื่องสำหรับการจำแนกข้อความ 27
ภาพที่ 4-1 สายงานการทดลอง 36
ภาพที่ 4-2 ภาพรวมของชุดข้อมูลหลังการทำความสะอาด 36
ภาพที่ 4-3 การกระจายตัวของความยาวข้อความ 36
ภาพที่ 4-4 สัดส่วนของกลุ่มข้อมูลความพยายามในการพัฒนา 37
ภาพที่ 4-5 การกระจายตัวของข้อมูลเวกเตอร์จาก WangchanBERTa ด้วยเทคนิค t-SNE แบบ 2 มิติ .. 40
ภาพที่ 4-6 การเปรียบเทียบประสิทธิภาพรวมระหว่างโมเดลภาษา 2 รูปแบบ 41
ภาพที่ 4-7 กราฟเส้นโค้ง ROC และค่า AUC สำหรับแบบจำลอง WangchanBERTa Ensemble..... 46
ภาพที่ 4-8 กราฟเส้นโค้ง ROC และค่า AUC สำหรับแบบจำลอง XLM-RoBERTa Ensemble 47
ภาพที่ 4-9 เมทริกซ์ความสับสนสำหรับแบบจำลอง WangchanBERTa 49
ภาพที่ 4-10 เมทริกซ์ความสับสนสำหรับแบบจำลอง XLM-RoBERTa 49

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการดำเนินโครงการพัฒนาระบบสารสนเทศหรือซอฟต์แวร์ในภาคธุรกิจและองค์กรภาครัฐ การประเมินความพยายามในการพัฒนา (Software Effort Estimation) ถือเป็นปัจจัยสำคัญที่ใช้กำหนดระยะเวลา งบประมาณ และทรัพยากรของโครงการ [5] โดยเฉพาะในกรณีที่องค์กรมีการจ้างผู้ให้บริการภายนอก (Vendor) มาดำเนินงานระบบการประเมิน Man-Day มักจะเริ่มจากการที่ Vendor ส่งเอกสารประเมินกลับมายังหน่วยงานเพื่อพิจารณาและอนุมัติ ซึ่งเอกสารเหล่านี้มักอยู่ในรูปของข้อความที่อธิบายรายละเอียดของความต้องการเชิงเทคนิคหรือ Business Requirement ควบคู่กับการประเมินจำนวนวันที่จำเป็น

อย่างไรก็ตาม ในทางปฏิบัติพบว่าข้อมูลจากเอกสารที่ส่งมาไม่ได้มีโครงสร้างที่แน่นอน และมักจะมีลักษณะเป็นภาษาธรรมชาติที่อาจตีความได้หลากหลาย ทำให้เจ้าหน้าที่ในทีมขององค์กรต้องใช้เวลานานในการอ่าน วิเคราะห์ และเปรียบเทียบกับงานในอดีต เพื่อพิจารณาว่าการประเมิน Man-Day ที่เสนอมานั้นมีความเหมาะสมหรือไม่ กระบวนการนี้ยังต้องอาศัยประสบการณ์ของผู้ประเมินเป็นหลัก และมีความเสี่ยงที่จะเกิดความคลาดเคลื่อนหรือความไม่สอดคล้องในการอนุมัติ [2, 6] นอกจากนี้ หากเกิดความไม่เห็นพ้องกันในเรื่องการประเมิน Man-Day อาจทำให้ต้องมีการหารือเพิ่มเติมหลายรอบ หรือแม้แต่การต่อรองใหม่ ซึ่งส่งผลต่อระยะเวลาของโครงการ และอาจกระทบต่อความสัมพันธ์ระหว่างผู้ว่าจ้างและผู้พัฒนา

ด้วยเหตุนี้ จึงมีความจำเป็นในการพัฒนาระบบที่สามารถช่วยวิเคราะห์ข้อความจากเอกสารประเมิน Man-Day โดยใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP) และการเรียนรู้ของเครื่อง (Machine Learning : ML) [9] มาประยุกต์ใช้โดยเปลี่ยนมุมมองจากการพยากรณ์ค่าตัวเลขที่มักมีความผิดพลาดสูง มาเป็นการ จำแนกระดับความพยายามแทน เช่น การแบ่งงานเป็นกลุ่มขนาดเล็ก (Small), กลาง (Medium), หรือใหญ่ (Large) ซึ่งงานวิจัยในอดีตระบุว่าเป็นแนวทางที่ให้ความแม่นยำสูงกว่าและเพียงพอสำหรับการตัดสินใจตรวจสอบความสมเหตุสมผลของราคาเบื้องต้น [7]

ดังนั้น การนำเทคโนโลยีปัญญาประดิษฐ์ โดยเฉพาะเทคนิคการประมวลผลภาษาธรรมชาติและการเรียนรู้แบบกลุ่ม (Ensemble Learning) [8] เพื่อวิเคราะห์ความซับซ้อนของงานจากเอกสารความต้องการทั้งในส่วนของข้อความและรูปภาพ ระบบที่พัฒนาขึ้นจะช่วยคัดกรองและจัดกลุ่มความยากง่ายของงานได้โดยอัตโนมัติ ช่วยลดภาระงานของเจ้าหน้าที่ ลดข้อโต้แย้งระหว่างคู่สัญญา และเพิ่มความโปร่งใสให้กับกระบวนการบริหารโครงการ

ที่ผ่านมางานวิจัยด้านการประเมินความพยายามในการพัฒนาซอฟต์แวร์มักพึ่งพาข้อมูลที่มีโครงสร้างชัดเจน (Structured Metrics) หรือวิจารณญาณของผู้เชี่ยวชาญ (Expert Judgment) [5], [6] แต่ยังขาดแคลนงานวิจัยที่ประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (NLP) ในการดึงข้อมูลจากเอกสารข้อกำหนดความต้องการ (Requirement Text) ที่เป็นข้อความอิสระโดยตรง ช่องว่างสำคัญที่งานวิจัยนี้ต้องการเติมเต็ม คือการใช้โมเดลภาษาขนาดใหญ่ (LLMs) เข้ามาสกัดคุณลักษณะ (Feature Extraction) [10], [12] และการเปรียบเทียบประสิทธิภาพของเวกเตอร์ข้อความ (Text Embedding) ระหว่างภาษาไทยและแบบพหุภาษา [14], [15] งานวิจัยนี้มีส่วนสนับสนุนและสร้างคุณูปการทางวิชาการ ดังนี้

1.1.1 การประยุกต์ใช้โมเดลภาษาขนาดใหญ่ (LLMs) อย่าง Gemma [10] ในการสกัดคุณลักษณะและวิเคราะห์ความซับซ้อนจากเอกสารความต้องการที่ไม่มีโครงสร้าง

1.1.2 การวิเคราะห์และเปรียบเทียบประสิทธิภาพการแปลงเวกเตอร์ข้อความ (Text Embedding) ระหว่างโมเดลภาษาไทย (WangchanBERTa) [14] และโมเดลพหุภาษา (XLM-RoBERTa) [15]

1.1.3 การพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) แบบ Classification ด้วยเทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) [8] เพื่อพยากรณ์และจัดกลุ่มระดับความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างอัตโนมัติ

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อประยุกต์ใช้ โมเดลภาษาขนาดใหญ่พหุรูปแบบ (Multimodal LLMs) ในการสกัดคุณลักษณะและอธิบายความซับซ้อน (Feature Extraction) จากเอกสารความต้องการซอฟต์แวร์ที่ไม่มีโครงสร้าง

1.2.2 เพื่อศึกษาและประยุกต์ใช้โมเดลภาษาขนาดใหญ่ (Large Language Model) คือ Gemma-3n [10] ในการทำหน้าที่เป็นนักวิเคราะห์ระบบอัตโนมัติ (AI System Analyst) เพื่อสกัดคุณลักษณะ (Feature Extraction) และอธิบายความซับซ้อนของเนื้องานจากเอกสารความต้องการโดยอาศัยความสามารถด้านพหุรูปแบบ (Multimodal) [11]

1.2.3 เพื่อพัฒนาแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) สำหรับจำแนกระดับความพยายามในการพัฒนาซอฟต์แวร์ (Effort Estimation Classification) โดยใช้เทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) [8]

1.2.4 เพื่อประเมินประสิทธิภาพ ของการแปลงข้อความเป็นเวกเตอร์ (Text Embedding) ระหว่างโมเดลที่ถูกฝึกสอนด้วยภาษาไทยเฉพาะทาง (WangchanBERTa) [14] และโมเดลพหุภาษา (XLM-RoBERTa) [15] เมื่อนำมาใช้ในบริบทของงานวิศวกรรมซอฟต์แวร์

1.2.5 เพื่อวิเคราะห์ อิทธิพลของคุณลักษณะทางภาษาศาสตร์และชนิดของเวกเตอร์ฝังตัวที่มีต่อความแม่นยำของแบบจำลอง ซึ่งจะนำไปสู่การสร้างองค์ความรู้ใหม่ด้านการประมวลผลภาษาธรรมชาติในงานวิศวกรรมซอฟต์แวร์ [9]

1.3 ขอบเขตของการวิจัย

1.3.1 ขอบเขตของข้อมูล

เอกสารข้อกำหนดความต้องการ (Requirement) ในรูปแบบไฟล์ PDF ซึ่งประกอบด้วยข้อมูลประเภทข้อความ (Text) และรูปภาพ (Image)
ข้อมูลจากเอกสารประเมินราคาของ Vendor ที่ประกอบด้วยรายละเอียดงาน (Task Description) และค่าประเมิน Man-Day

1.3.2 ขอบเขตของเครื่องมือและเทคนิค

พัฒนาด้วยภาษา Python โดยใช้ไลบรารี PyMuPDF [20] สำหรับจัดการเอกสาร PDF และ Transformers สำหรับจัดการโมเดลภาษา และไลบรารี PyThaiNLP [4] สำหรับการจัดการตัดคำภาษาไทย

การสกัดคุณลักษณะ (Feature Extraction) ใช้โมเดล Gemma-3n (Multimodal LLM) [10] วิเคราะห์ความสัมพันธ์ระหว่างข้อความและรูปภาพในเอกสาร เพื่อสร้างคำอธิบายบริบทของงาน (Contextual Analysis)

การแปลงข้อความเป็นเวกเตอร์ (Embedding) เปรียบเทียบระหว่าง WangchanBERTa (เน้นภาษาไทย) [14] และ XLM-RoBERTa (รองรับหลายภาษา) [15] โดยใช้เทคนิคการรวมข้อความ (Concatenation) ระหว่างรายละเอียดงานและบทวิเคราะห์จาก Gemma
แบบจำลองที่ใช้เป็นประเภท Classification Model โดยใช้เทคนิค Soft Voting Ensemble ที่ประกอบด้วยอัลกอริทึมหลากหลาย เช่น XGBoost (Extreme Gradient Boosting) [16], Random Forest [17], Support Vector Classifier (SVC), Extra Trees [21], และ K-Nearest Neighbors (KNN) ที่ผ่านการปรับจูนพารามิเตอร์ด้วย Optuna [18]

1.3.3 ขอบเขตของการวัดผล

วัดประสิทธิภาพการจำแนกประเภท (Classification Performance) ด้วยค่าทางสถิติ ได้แก่ ความแม่นยำ (Accuracy), ค่า Precision, Recall และ F1-Score

1.4 สมมติฐานการวิจัย

H_1 : ชนิดของโมเดลการแปลงข้อความเป็นเวกเตอร์ (Text Embedding Models) ที่แตกต่างกัน ส่งผลต่อประสิทธิภาพและความแม่นยำในการจำแนกระดับความพยายามที่แตกต่างกัน

H_2 : การสกัดคุณลักษณะและขยายบริบทความซับซ้อนด้วยโมเดลภาษาขนาดใหญ่ (LLMs Feature Extraction) มีส่วนช่วยเพิ่มความแม่นยำให้กับแบบจำลองการเรียนรู้ของเครื่อง

1.5 วิธีการวิจัย

1.5.1 การรวบรวมข้อมูล

รวบรวมไฟล์เอกสาร Requirement (PDF) และตารางรายการประเมิน Man-Day (CSV) จากโครงการพัฒนาระบบสารสนเทศในอดีต

1.5.2 การประมวลผลข้อมูลและวิเคราะห์ด้วย Multimodal LLM

ดึงข้อมูลดิบจากไฟล์ PDF โดยแยกส่วนที่เป็นข้อความและรูปภาพ นำรายละเอียดงานพร้อมด้วยบริบทจาก PDF (Text + Images) ส่งเข้าสู่โมเดล gemma-3n-E4B-it

กำหนดบทบาท (Prompting) ให้โมเดลทำหน้าที่เป็น System Analyst โดยใช้เทคนิค Chain-of-Thought [12] เพื่อวิเคราะห์และสร้างคำอธิบายขยายความ (Generated Description) ที่ระบุถึงความซับซ้อนทางเทคนิค และอุดช่องโหว่ความคลุมเครือ (Hidden Context) ของแต่ละงาน

ให้ gemma-3n-E4B-it ทำการสร้างคำอธิบายเพิ่มเติมหรือสรุปเนื้องาน (Generate Features) เพื่อขยายความเข้าใจในสิ่งที่ต้องพัฒนาให้ชัดเจนยิ่งขึ้นกว่าข้อความต้นฉบับเพียงอย่างเดียว

1.5.3 การเตรียมข้อมูลและวิศวกรรมคุณลักษณะ

Text Concatenation รวมข้อความต้นฉบับจาก Vendor และผลวิเคราะห์จาก Gemma เข้าด้วยกัน โดยใช้โทเคนพิเศษ [SEP] คั่นกลาง [13]

Label Encoding แปลงค่า Man-Day ที่เป็นตัวเลข ให้เป็นกลุ่มระดับความยาก (Class Labels) 3 ระดับ ได้แก่ ระดับง่าย (กลุ่ม 0), ปานกลาง (กลุ่ม 1), และยาก (กลุ่ม 2) พร้อมทั้งตรวจสอบและจัดการปัญหาความไม่สมดุลของข้อมูล (Class Imbalance) เพื่อให้แบบจำลองเรียนรู้ได้อย่างมีประสิทธิภาพ

1.5.4 การสร้างและฝึกสอนแบบจำลอง

แปลงข้อมูลข้อความที่เตรียมไว้เป็นเวกเตอร์ (Embedding) ด้วยโมเดล WangchanBERTa และ XLM-RoBERTa แยกเป็นการทดลอง 2 กรณี

ทำการค้นหาค่าพารามิเตอร์ที่เหมาะสมที่สุด (Hyperparameter Tuning) ด้วย Optuna [18] สำหรับแต่ละอัลกอริทึม

สร้างแบบจำลองผสม (Ensemble Model) ด้วยวิธี Voting Classifier เพื่อรวมผลการทำนายจากโมเดลย่อยที่ดีที่สุดเพื่อลดค่าความแปรปรวน และเพิ่มเสถียรภาพในการทำนาย

1.5.5 การประเมินผลและการเปรียบเทียบ

ทดสอบโมเดลกับชุดข้อมูลทดสอบ (Test Set) เปรียบเทียบผลลัพธ์ระหว่างการใช้ WangchanBERTa และ XLM-RoBERTa ว่าแบบจำลองใดให้ค่าความแม่นยำ (Accuracy) และ F1-Score สูงกว่ากันในการจำแนกระดับความยากของงาน นอกจากนี้ยังมีการประเมินประสิทธิภาพเชิงลึกด้วยกราฟเส้นโค้ง ROC (ROC Curve) และเมทริกซ์ความสับสน (Confusion Matrix) เพื่อวิเคราะห์พฤติกรรมการจำแนกในแต่ละระดับความยาก

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ประโยชน์เชิงวิชาการ

1.6.1.1. ได้เพิ่มพูนองค์ความรู้ใหม่ด้านการประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (NLP) และการเรียนรู้ของเครื่อง (Machine Learning) ในการคัดกรองและจัดกลุ่มความซับซ้อนของงานพัฒนาซอฟต์แวร์ (Task Complexity Classification) แบบอัตโนมัติ

1.6.1.2. ได้ทราบถึงประสิทธิภาพของสถาปัตยกรรมโมเดลภาษาขนาดใหญ่พหุรูปแบบ (Multimodal LLMs) และการแปลงข้อความเป็นเวกเตอร์ (Text Embedding) ในการวิเคราะห์และสกัดคุณลักษณะจากเอกสารความต้องการซอฟต์แวร์ เทียบกับข้อมูลการประเมินจริง ตลอดจนพิสูจน์ศักยภาพในการขยายบริบทซ่อนเร้น (Hidden Context) ที่มนุษย์อาจละเลย

1.6.1.3. ได้ทราบถึงผลการเปรียบเทียบประสิทธิภาพของสถาปัตยกรรม Text Embedding ระหว่างโมเดลภาษาไทยและพหุภาษา เมื่อนำมาใช้กับชุดข้อมูลที่มีการสลับรหัสภาษา (Code-switching) ซึ่งเป็นลักษณะเฉพาะของเอกสารในอุตสาหกรรมซอฟต์แวร์ไทย

1.6.2 ประโยชน์เชิงปฏิบัติ

1.6.2.1. ได้เครื่องมือต้นแบบ (Prototype) ที่พัฒนาด้วยภาษา Python สำหรับช่วยผู้จัดการโครงการ (Project Manager) หรือผู้ตรวจรับงาน ในการตรวจสอบความสมเหตุสมผลของการประเมินราคาเบื้องต้น (Sanity Check) ว่าความยากของงานสอดคล้องกับค่า Man-Day ที่เสนอหรือไม่ รวมถึงใช้เป็นข้อมูลอ้างอิงเชิงประจักษ์ (Data-driven Evidence) เพื่อลดความขัดแย้งระหว่างองค์กรและผู้ให้บริการ (Vendor)

1.6.2.2. ช่วยลดภาระงาน ลดความผิดพลาด (Human Error) และความล่าช้าจากการที่บุคลากรต้องใช้วิจารณญาณอ่านและวิเคราะห์เอกสาร Requirement จำนวนมากด้วยตาเปล่า ทำให้การวางแผนโครงการมีประสิทธิภาพยิ่งขึ้น

1.6.2.3. เป็นการสกัดและบันทึกองค์ความรู้ฝังลึก (Tacit Knowledge) ของนักวิเคราะห์ระบบอาวุโสในการประเมินงาน ให้อยู่ในรูปแบบของอัลกอริทึมปัญญาประดิษฐ์ ซึ่งช่วยให้องค์กรสามารถรักษามาตรฐานการประเมินและถ่ายทอดองค์ความรู้ได้อย่างยั่งยืน

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ ผู้วิจัยนำเสนอทฤษฎี กรอบแนวคิด และองค์ความรู้พื้นฐานที่รองรับการพัฒนาแบบจำลองสำหรับการจำแนกระดับความซับซ้อนในการพัฒนาซอฟต์แวร์จากเอกสารความต้องการ โดยเนื้อหาเริ่มต้นจากการทบทวนหลักการประเมินความพยายามในการพัฒนาซอฟต์แวร์ (Software Effort Estimation) เพื่อให้เข้าใจถึงบริบท ปัญหา และข้อจำกัดของการประเมินแบบดั้งเดิมที่อาศัยเพียงวิจารณญาณของผู้เชี่ยวชาญ ลำดับถัดมาจะอธิบายถึงรากฐานทางเทคโนโลยีที่นำมาประยุกต์ใช้ ได้แก่ การประมวลผลภาษาธรรมชาติ (Natural Language Processing : NLP) ปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) สำหรับการสกัดและขยายบริบทความต้องการ ตลอดจนเทคนิคการแปลงข้อความเป็นเวกเตอร์ (Text Embedding) ทั้งในกลุ่มแบบจำลองเฉพาะทางภาษาไทยและแบบจำลองพหุภาษา นอกจากนี้ บทนี้ยังครอบคลุมถึงหลักการของอัลกอริทึมการเรียนรู้ของเครื่อง (Machine Learning) และเทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) ที่ใช้ในการสร้างแบบจำลองจำแนกประเภท รวมถึงทฤษฎีเกี่ยวกับมาตรวัดประสิทธิภาพ (Evaluation Metrics) ที่ใช้ประเมินผลการทดลอง ในส่วนท้ายของบทจะนำเสนองานวิจัยที่เกี่ยวข้องเพื่อวิเคราะห์ช่องว่างทางการวิจัย (Research Gap) ซึ่งจะเป็นรากฐานสำคัญในการออกแบบระเบียบวิธีวิจัยในบทถัดไปอย่างเป็นระบบ

2.1 การประมาณการความพยายามในการพัฒนาซอฟต์แวร์

การประมาณการความพยายาม (Effort Estimation) เป็นกระบวนการสำคัญในการบริหารจัดการโครงการซอฟต์แวร์ เพื่อคาดการณ์ทรัพยากรที่ต้องใช้ ซึ่งหน่วยวัดที่เป็นมาตรฐานสากลและนิยมใช้ในอุตสาหกรรมคือ Man-Day หรือ Man-Hour Boehm [5] ได้ระบุไว้ในตำรา Software Engineering Economics ว่าการประมาณการที่แม่นยำเป็นหัวใจสำคัญของความสำเร็จของโครงการ อย่างไรก็ตาม ในทางปฏิบัติมักพบปัญหาจากความไม่ชัดเจนของเอกสารความต้องการ (Requirement Ambiguity) และความแปรปรวนของทักษะผู้ประเมิน

2.1.1 ปัญหาจากเอกสารความต้องการ

วราพร จิระพันธุ์ทอง [2] ได้ศึกษาผลของการดำเนินโครงการซอฟต์แวร์และพบว่า ความล้มเหลวหรือความล่าช้าของโครงการส่วนใหญ่มีสาเหตุตั้งต้นมาจากการจัดการความต้องการ ที่ไม่มีประสิทธิภาพ เอกสารความต้องการที่คลุมเครือส่งผลให้การประเมินราคาผิดพลาด ซึ่งสอดคล้องกับงานวิจัยของ Moløkken และ Jørgensen [6] ที่พบว่าการประเมินโดยผู้เชี่ยวชาญมักมีความลำเอียง (Bias) และขาดความสม่ำเสมอ

2.1.2 ทฤษฎีการประเมินความพยายามด้วยการเรียนรู้ของเครื่อง

ในปัจจุบัน การประเมินความพยายามได้ก้าวข้ามจากการใช้ผู้เชี่ยวชาญคาดเดา มาสู่การประยุกต์ใช้อัลกอริทึมการเรียนรู้ของเครื่อง (Machine Learning) และการเรียนรู้เชิงลึก (Deep Learning) เพื่อเพิ่มความแม่นยำ โดยมีการใช้โมเดลหลากหลายรูปแบบ ได้แก่ โมเดลการถดถอย (Regression Models) เพื่อทำนายค่าความพยายามออกมาเป็นตัวเลขต่อเนื่อง (เช่น จำนวน Man-Day) รวมไปถึงการประยุกต์ใช้โมเดลต้นไม้ตัดสินใจอย่าง Random Forest และโครงข่ายประสาทเทียม (Neural Networks) ซึ่งมีความสามารถในการเรียนรู้ความสัมพันธ์ที่ซับซ้อนของข้อมูลโครงการในอดีตได้

2.2 การประมวลผลภาษาธรรมชาติและการแทนค่าข้อความ

2.2.1 วิวัฒนาการการแทนค่าข้อความ

การที่คอมพิวเตอร์จะเข้าใจข้อความในเอกสารความต้องการ (Requirement) ได้จำเป็นต้องมีการแปลงข้อความเป็นตัวเลขทางคณิตศาสตร์เสียก่อน ซึ่งวิวัฒนาการของการแทนค่าข้อความมีดังนี้

2.2.1.1. Bag of Words (BoW) และ TF-IDF เป็นเทคนิคยุคแรกที่ใช้วิธีนับความถี่ของคำ แต่มีข้อจำกัดคือไม่สามารถรับรู้ความหมายแฝงหรือลำดับคำในประโยคได้

2.2.1.2. Word Embedding เป็นการแปลงคำให้อยู่ในรูปเวกเตอร์ (เช่น Word2Vec) ซึ่งเริ่มจับความสัมพันธ์ของคำได้ แต่มีข้อจำกัดคือคำ ๆ เดียวกันจะมีเวกเตอร์ค่าเดียวเสมอแม้จะอยู่ต่างบริบทกัน

2.2.1.3. Contextual Embedding เป็นเทคโนโลยีขั้นสูงที่ใช้ในปัจจุบัน เช่น สถาปัตยกรรม BERT หรือ RoBERTa ที่สามารถสร้างเวกเตอร์ของคำโดยพิจารณาจาก “บริบทแวดล้อม” (Contextual Representation) ทำให้เข้าใจความหมายที่เปลี่ยนไปตามรูปประโยคได้อย่างแม่นยำ ในงานวิจัยนี้ มุ่งเน้นการใช้เทคนิค Sentence Embedding ซึ่งเป็นการนำข้อความ Requirement ทั้งประโยคหรือทั้งย่อหน้ามาแปลงเป็นเวกเตอร์ตัวเลขในปริภูมิเวกเตอร์ (Vector Space) แบบหลายมิติ โดยโมเดลจะพิจารณาความคล้ายคลึงทางความหมาย (Semantic Similarity) ทำให้ประโยคที่มีความหมายหรือบริบทความยากของงานใกล้เคียงกัน จะถูกจัดวางให้อยู่ในตำแหน่งที่ใกล้กันใน Vector Space ซึ่งเป็นหัวใจสำคัญก่อนส่งข้อมูลให้แบบจำลอง Machine Learning ทำการจัดกลุ่ม

2.2.2 ความท้าทายและการประมวลผลภาษาไทย

ภาษาไทยเป็นภาษาที่มีความซับซ้อนเนื่องจากไม่มีการเว้นวรรคระหว่างคำ และประโยคที่ชัดเจน นงเยาว์ สอนจะโปะ [1] และ วรพงศ์ บำรุงศรี [3] ได้ศึกษาการใช้กระบวนการประมวลผลภาษาธรรมชาติกับภาษาไทยและพบว่า การตัดคำ (Tokenization) ที่แม่นยำส่งผลโดยตรงต่อ

ประสิทธิภาพของโมเดล งานวิจัยนี้จึงอาศัยหลักการตัดคำภาษาไทยผ่านไลบรารี PyThaiNLP [4] ซึ่งเป็นเครื่องมือมาตรฐานสำหรับการจัดการภาษาไทย เพื่อเตรียมข้อมูลให้พร้อมก่อนนำเข้าสู่สถาปัตยกรรมระดับสูง

2.2.3 แบบจำลองภาษาที่ใช้ในการแปลงข้อความเป็นเวกเตอร์

การแปลงข้อความภาษาไทยและภาษาอังกฤษให้อยู่ในรูปแบบเวกเตอร์เป็นขั้นตอนที่สำคัญยิ่งในการวิจัยนี้ โดยผู้วิจัยได้เลือกใช้และเปรียบเทียบแบบจำลองดังต่อไปนี้

2.2.3.1. โมเดล WangchanBERTa เป็นโมเดลภาษาที่พัฒนาโดย Lowphansirikul et al. [14] บนสถาปัตยกรรม RoBERTa โดยได้รับการฝึกสอน (Pre-trained) ด้วยชุดข้อมูลภาษาไทยขนาดใหญ่ ทำให้มีความเข้าใจในบริบท ไวยากรณ์ และคำศัพท์เฉพาะของภาษาไทยได้ดีที่สุดในปัจจุบัน

2.2.3.2. โมเดล XLM-RoBERTa เนื่องจากเอกสารความต้องการด้านซอฟต์แวร์มักมีการใช้คำศัพท์เทคนิคภาษาอังกฤษปะปนกับภาษาไทย (Code-switching) งานวิจัยนี้จึงได้ศึกษาเปรียบเทียบกับ XLM-RoBERTa [15] ซึ่งเป็นโมเดลพหุภาษา (Multilingual) ที่ถูกฝึกสอนด้วยข้อมูลกว่า 100 ภาษา เพื่อทดสอบประสิทธิภาพในการจัดการกับเอกสารที่มีความหลากหลายทางภาษา

2.3 สถาปัตยกรรมทรานส์ฟอร์เมอร์และแบบจำลองภาษาพหุรูปแบบ

2.3.1 โครงสร้างสถาปัตยกรรมทรานส์ฟอร์เมอร์

แบบจำลองภาษาในปัจจุบันส่วนใหญ่ได้รับการพัฒนาต่อยอดจากสถาปัตยกรรมทรานส์ฟอร์เมอร์ (Transformer) ที่ถูกนำเสนอโดย Vaswani et al. [13] หัวใจสำคัญของสถาปัตยกรรมนี้คือ กลไกความสนใจ (Attention Mechanism หรือ Self-Attention) ที่ช่วยให้โมเดลสามารถให้น้ำหนักความสำคัญและเรียนรู้ความสัมพันธ์ของคำแต่ละคำในประโยคที่เกี่ยวข้องกันได้ ไม่ว่าจะอยู่ห่างกันแค่ไหน ส่งผลให้เกิดการสร้างตัวแทนบริบท (Contextual Representation) ที่ทรงพลัง ทำให้แบบจำลองเข้าใจบริบทได้ลึกซึ้งกว่าสถาปัตยกรรมแบบดั้งเดิม (เช่น RNN หรือ LSTM) อย่างมาก และได้กลายเป็นรากฐานสำคัญของโมเดลภาษาระดับโลกอย่าง GPT Architecture ที่โดดเด่นในปัจจุบัน

2.3.2 แบบจำลองภาษาพหุรูปแบบ Gemma-3n

งานวิจัยนี้เลือกใช้ Gemma-3n [10] ซึ่งเป็นแบบจำลองภาษาขนาดใหญ่แบบเปิด (Open Weights) ที่พัฒนาโดย Google DeepMind จุดเด่นสำคัญของ Gemma รุ่นใหม่คือความสามารถในการประมวลผลพหุรูปแบบ (Multimodal Processing) หมายความว่าโมเดลไม่เพียงแต่รับข้อมูลประเภทข้อความ (Text) เท่านั้น แต่สามารถบูรณาการข้อมูลภาพ (Image) เข้ามาประมวลผลร่วมกันได้ (Text + Image Integration) สอดคล้องกับการศึกษาของ Yin et al. [11] ที่ระบุว่าการผสานข้อมูลภาพเข้ากับข้อความช่วยให้แบบจำลองเข้าใจบริบทได้ลึกซึ้งขึ้น โดยเฉพาะในงานวิศวกรรม

ซอฟต์แวร์ที่เอกสารความต้องการมักประกอบด้วยแผนภาพ (Diagrams) ผังงาน (Flowcharts) หรือภาพหน้าจอ (UI Mockups) ซึ่งเป็นส่วนสำคัญที่โมเดลภาษาแบบข้อความเพียงอย่างเดียวไม่สามารถเข้าถึงได้ แม้ในขอบเขตการศึกษานี้จะเน้นที่การประมวลผลข้อความเป็นหลัก แต่การเลือกใช้สถาปัตยกรรมดังกล่าวได้ปูทางและรองรับการพัฒนาต่อยอดเพื่อให้อ่าน UI Mockup ควบคู่กับเอกสาร Requirement ในอนาคตได้อย่างไร้รอยต่อ

2.3.3 เทคนิควิศวกรรมคำสั่ง

เพื่อให้แบบจำลองสามารถวิเคราะห์ความซับซ้อนของงานได้เสมือนผู้เชี่ยวชาญ งานวิจัยนี้ได้ประยุกต์ใช้เทคนิควิศวกรรมคำสั่งแบบลูกโซ่ความคิด (Chain-of-Thought Prompting) ตามแนวทางของ Wei et al. [12] โดยมีการกำหนดบทบาทสมมติ (Role-playing) ให้แบบจำลองทำหน้าที่เป็นนักวิเคราะห์ระบบ (System Analyst) เพื่อบังคับให้โมเดลแสดงกระบวนการคิด การให้เหตุผล และการแยกแยะองค์ประกอบของงานทีละขั้นตอนก่อนที่จะสรุปผลลัพธ์ ซึ่งวิธีการนี้ช่วยเพิ่มประสิทธิภาพและความแม่นยำในการแก้ปัญหาที่มีความซับซ้อนสูงได้อย่างมีนัยสำคัญ

2.4 การเรียนรู้ของเครื่องและการเรียนรู้แบบกลุ่ม

งานวิจัยนี้ประยุกต์ใช้เทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) เพื่อเพิ่มความแม่นยำในการจำแนกประเภท โดยอ้างอิงแนวคิดจาก Pospieszny et al. [8] ที่พบว่าการรวมแบบจำลองหลายตัวเข้าด้วยกัน จะให้ผลลัพธ์และประสิทธิภาพที่ดีกว่าการใช้แบบจำลองเดี่ยวในการประมาณการโครงการซอฟต์แวร์

2.4.1 หลักการเรียนรู้แบบมีผู้สอนและการจำแนกประเภท

งานวิจัยนี้ออกแบบกระบวนการประเมินให้เป็นปัญหาการเรียนรู้แบบมีผู้สอน (Supervised Learning) ในรูปแบบการจำแนกประเภท (Classification Problem) โดยแบบจำลองจะทำการเรียนรู้จากข้อมูลเอกสารความต้องการ (Requirement) ในอดีตที่มี “เฉลย” หรือมีการระบุระดับความพยายามไว้แล้ว เพื่อสกัดเป็นกฎเกณฑ์ทางสถิติและนำไปใช้ทำนายหรือจัดกลุ่มระดับความยากของชุดข้อมูลใหม่

2.4.2 อัลกอริทึมพื้นฐานที่ใช้ในการสร้างแบบจำลอง

เพื่อครอบคลุมพฤติกรรมทางสถิติที่หลากหลายและเตรียมพร้อมสำหรับการทำแบบจำลองรวม ผู้วิจัยได้เลือกใช้อัลกอริทึมพื้นฐาน (Base Learners) จำนวน 5 อัลกอริทึม ผ่านการพัฒนาด้วยไลบรารี Scikit-learn [19] ดังนี้

2.4.2.1. XGBoost (Extreme Gradient Boosting) นำเสนอโดย Chen & Guestrin [16] เป็นอัลกอริทึมที่ใช้เทคนิค Boosting ในการสร้างต้นไม้ตัดสินใจ (Decision Trees) หลายต้นต่อกัน โดยแต่ละต้นจะเรียนรู้เพื่อแก้ไขข้อผิดพลาดของต้นก่อนหน้า มีจุดเด่นด้านความเร็วในการประมวลผลและความแม่นยำสูง

2.4.2.2. Random Forest ตามทฤษฎีของ Breiman [17] เป็นการใช้เทคนิค Bagging โดยสร้างต้นไม้ตัดสินใจหลายต้นอย่างอิสระจากการสุ่มชุดข้อมูลและคุณลักษณะ (Features) จากนั้นนำผลลัพธ์ของทุกต้นมาโหวตหาคำตอบสุดท้าย ซึ่งช่วยลดปัญหาการเรียนรู้เกิน (Overfitting) ได้อย่างมีประสิทธิภาพ

2.4.2.3. Extra Trees (Extremely Randomized Trees) นำเสนอโดย Geurts et al. [21] เป็นการต่อยอดจาก Random Forest โดยเพิ่มความสุ่มในการเลือกจุดตัด (Split) ของโหนดในต้นไม้แต่ละต้น ทำให้แบบจำลองสามารถลดความแปรปรวน (Variance) และหลีกเลี่ยงปัญหา Overfitting ได้ดียิ่งขึ้น โดยเฉพาะเมื่อจัดการกับข้อมูลที่มีสัญญาณรบกวน (Noise)

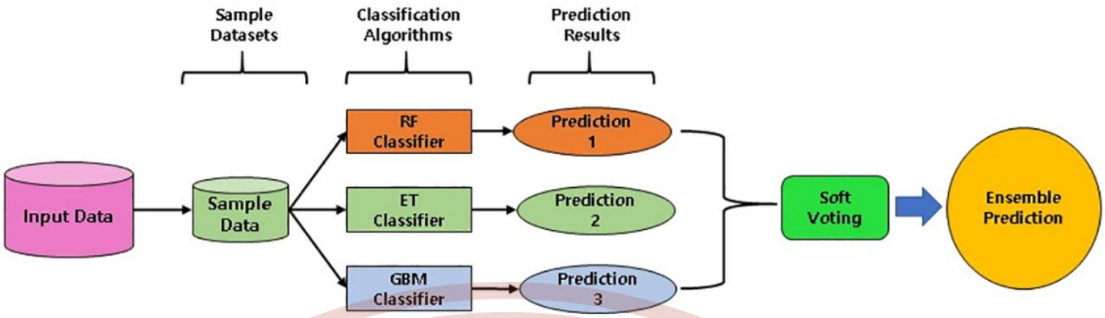
2.4.2.4. Support Vector Classifier (SVC) เป็นอัลกอริทึมที่ทำงานโดยการค้นหาระนาบเกิน (Hyperplane) ในพื้นที่หลายมิติ เพื่อแบ่งแยกกลุ่มข้อมูล (Class) ออกจากกันให้มีระยะห่าง (Margin) มากที่สุด เหมาะสำหรับการจัดการกับข้อมูลที่มีมิติสูง เช่น ข้อมูลเวกเตอร์ของข้อความ

2.4.2.5. K-Nearest Neighbors (KNN) เป็นอัลกอริทึมการเรียนรู้แบบอาศัยความจำ (Instance-based Learning) ที่ใช้หลักการคำนวณระยะห่าง (Distance Metric) เช่น Euclidean Distance หรือ Cosine Similarity ระหว่างข้อมูลที่ต้องการทำนายกับข้อมูลในชุดฝึกสอน โดยจะพิจารณาจากเพื่อนบ้านที่ใกล้ที่สุดจำนวน K ลำดับแรก และตัดสินใจเลือกคลาสเป้าหมายด้วยวิธีการโหวตเสียงข้างมาก (Majority Voting)

2.4.3 การเรียนรู้แบบกลุ่มด้วยเทคนิคการโหวต

การเรียนรู้แบบกลุ่ม (Ensemble Learning) คือเทคนิคการนำอัลกอริทึมพื้นฐานหลายตัวมารวมกันเพื่อแก้ปัญหาความเอนเอียง (Bias) และเพิ่มความแม่นยำ โดยอาศัยหลักการทำงานที่แตกต่างกันของอัลกอริทึม เช่น Bagging ของ Random Forest และ Boosting ของ XGBoost เข้ามาช่วยชดเชยจุดอ่อนซึ่งกันและกัน

ในงานวิจัยนี้ ผู้วิจัยได้ประยุกต์ใช้เทคนิค Soft Voting เพื่อรวมผลลัพธ์จากแบบจำลองย่อยทั้งหมด โดยจะพิจารณาคำนวณจากค่าความน่าจะเป็น (Probability) เฉลี่ยที่ได้จากแต่ละแบบจำลอง วิธีนี้ช่วยให้ระบบมีความทนทาน (Robustness) ต่อข้อมูลรบกวน และให้ผลการตัดสินใจที่ละเอียดอ่อนกว่าการโหวตแบบนับจำนวนเสียง (Hard Voting)



ภาพที่ 2-1 Soft Voting diagram

2.4.4 การปรับจูนพารามิเตอร์อัตโนมัติ

เพื่อให้แบบจำลองทุกตัวทำงานได้เต็มประสิทธิภาพสูงสุด งานวิจัยนี้จึงได้ใช้เฟรมเวิร์ก Optuna [18] ในการค้นหาค่าพารามิเตอร์ที่เหมาะสมที่สุด (Hyperparameter Tuning) ผ่านวิธีการหาค่าเหมาะสมที่สุดแบบเบย์ (Bayesian Optimization) ด้วยอัลกอริทึม Tree-structured Parzen Estimator (TPE) ซึ่งได้รับการพิสูจน์แล้วว่ามีกระบวนการค้นหาที่รวดเร็วและมีประสิทธิภาพสูงกว่าการค้นหาแบบครอบคลุม (Grid Search) แบบดั้งเดิม

2.5 มาตรวัดประสิทธิภาพของแบบจำลอง

เพื่อประเมินประสิทธิภาพของแบบจำลองการจำแนกประเภท งานวิจัยนี้ได้เลือกใช้วิธีการวัดผลมาตรฐาน ได้แก่ เมทริกซ์ความสับสน (Confusion Matrix), ความแม่นยำรวม (Accuracy), ค่าความแม่นยำ (Precision), ค่าความจำ (Recall) และคะแนนเอฟวัน (F1-Score) โดย Sehra et al. [9] ได้สรุปแนวโน้มการวิจัยด้านนี้ไว้ว่า การพิจารณาค่า F1-Score ควบคู่ไปกับมาตรวัดอื่น มีความสำคัญอย่างยิ่งเมื่อวิเคราะห์ชุดข้อมูลที่มีความไม่สมดุล (Imbalanced Data) เพื่อให้มั่นใจว่าแบบจำลองไม่ได้ทำนายแม่นยำเฉพาะกับกลุ่มข้อมูลส่วนใหญ่เท่านั้น ซึ่งมาตรวัดแต่ละประเภทมีรายละเอียดดังนี้

2.5.1 เมทริกซ์ความสับสน

คือตารางเมทริกซ์ที่ใช้แสดงประสิทธิภาพของแบบจำลองการจำแนกประเภท โดยเปรียบเทียบผลการทำนาย (Predicted Values) กับค่าจริงเป้าหมาย (Actual Values) ทำให้เห็นรายละเอียดอย่างชัดเจนว่าแบบจำลองทำนายถูกต้องมากน้อยเพียงใด และเกิด “ความสับสน” หรือทายผิดข้ามคลาสไปที่กลุ่มใดบ้าง

2.5.2 ความแม่นยำรวม

คือสัดส่วนของข้อมูลที่แบบจำลองสามารถทำนายได้ถูกต้องทั้งหมด (ทั้งคลาสบวกและคลาสลบ) เมื่อเทียบกับจำนวนข้อมูลที่ใช้ทดสอบทั้งหมด ถือเป็นภาพรวมเบื้องต้นในการประเมินประสิทธิภาพของแบบจำลอง

2.5.3 ความแม่นยำ

คือสัดส่วนของข้อมูลที่แบบจำลองทำนายว่าอยู่ในคลาสเป้าหมาย (ทายว่าบวก) และถูกต้องจริงเมื่อเทียบกับค่าจริง (ทายว่าบวกแล้วเป็นบวกจริง) ค่านี้สะท้อนถึง “ความน่าเชื่อถือ” เมื่อแบบจำลองตัดสินใจทำนายคลาสนั้นๆ ออกมา

2.5.4 ความจำ

คือสัดส่วนของข้อมูลที่เป็นคลาสเป้าหมายจริงทั้งหมดในระบบ ที่แบบจำลองสามารถค้นหาและทายถูกได้ ค่านี้สะท้อนถึง “ความสามารถในการค้นหาความจริง” โดยไม่ปล่อยให้ข้อมูลที่ควรจะถูกจัดกลุ่มหลุดรอดไปได้

2.5.5 คะแนนเอฟวัน

คือค่าเฉลี่ยฮาร์โมนิก (Harmonic Mean) ระหว่างค่า Precision และ Recall ใช้สำหรับประเมินประสิทธิภาพโดยรวมของแบบจำลอง ค่านี้นำมาใช้เพื่ออุดช่องโหว่ของค่า Accuracy ในกรณีที่ชุดข้อมูลมีสัดส่วนของแต่ละคลาสไม่เท่ากัน หากค่า F1-Score สูง แสดงว่าแบบจำลองมีความสมดุลทั้งในแง่ของความน่าเชื่อถือและการค้นหาความจริง

2.6 งานวิจัยที่เกี่ยวข้อง

จากการสืบค้นงานวิจัยที่เกี่ยวข้อง สามารถสรุปประเด็นสำคัญได้ดังนี้

2.6.1 การใช้ Machine Learning ในการประมาณการ Effort

Pospieszny et al. (2018) [8] ได้เปรียบเทียบเทคนิค Ensemble Learning (Stacking, Bagging, Boosting) กับข้อมูลโครงการซอฟต์แวร์ พบว่าเทคนิค Ensemble ให้ค่าความแม่นยำสูงกว่าการใช้โมเดลเดี่ยวอย่างมีนัยสำคัญ และ Sehra et al. (2017) [9] ได้รวบรวมงานวิจัยในรอบทศวรรษและพบแนวโน้มที่ชัดเจนในการเปลี่ยนจากเทคนิคสถิติแบบเดิมมาใช้เทคนิค Soft Computing และ Machine Learning

2.6.2 การจำแนกช่วงความพยายาม (Effort Classification)

Bakir et al. (2016) [7] ได้เสนอแนวทางเปลี่ยนโจทย์จาก Regression เป็น Classification ผลการทดลองพบว่าโมเดลสามารถจำแนกโครงการตามขนาดความพยายามได้แม่นยำขึ้น และลดความเสี่ยงจากการระบุตัวเลขที่ผิดพลาด ซึ่งสอดคล้องกับแนวทางของงานวิจัยนี้

2.6.3 การประมวลผลภาษาธรรมชาติในงานวิศวกรรมซอฟต์แวร์

วรพงศ์ บำรุงศรี [3] และ นงเยาว์ สอนจะโปะ [1] ได้แสดงให้เห็นถึงความสำเร็จในการนำ NLP มาใช้จำแนกข้อความภาษาไทยในบริบทต่างๆ อย่างไรก็ตาม งานวิจัยส่วนใหญ่ในอดีตยังเน้นการวิเคราะห์เฉพาะ “ข้อความ” (Text-only)

2.6.4 การประเมินความพยายามในการพัฒนาซอฟต์แวร์ด้วยการเรียนรู้เชิงลึกและเทคนิคกลุ่ม

ในอดีต การประเมินความพยายามในการพัฒนาซอฟต์แวร์มักพึ่งพาโมเดลเชิงประจักษ์และการใช้ตัวแปรที่มีโครงสร้าง (Structured Metrics) เช่น โมเดล COCOMO [5] หรือการประเมินจากผู้เชี่ยวชาญ [6] ซึ่งมีข้อจำกัดเมื่อต้องเผชิญกับข้อมูลที่ไม่มีโครงสร้าง ต่อมา Bakir et al. (2016) [7] และ Pospieszny et al. (2018) [8] ได้นำเสนอการประยุกต์ใช้อัลกอริทึมการเรียนรู้ของเครื่อง (Machine Learning) และเทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) เพื่อช่วยลดความแปรปรวนในการพยากรณ์

นอกจากนี้ ในยุคปัจจุบัน งานวิจัยได้มุ่งเน้นไปที่การใช้ สถาปัตยกรรมการเรียนรู้เชิงลึกสำหรับการประเมินความพยายาม (Deep Learning for Effort Estimation) ซึ่งโครงข่ายประสาทเทียม (Neural Networks) มีความสามารถในการเรียนรู้ความสัมพันธ์ที่ซับซ้อน (Non-linear relationships) จากข้อมูลประวัติโครงการ อย่างไรก็ตาม งานวิจัยส่วนใหญ่ยังคงต้องการข้อมูลนำเข้าที่เป็นตัวเลขทางสถิติ และยังขาดการบูรณาการร่วมกับการประมวลผลข้อความจากเอกสารความต้องการโดยตรง [9]

2.6.5 การจำแนกข้อความด้วยสถาปัตยกรรมทรานส์ฟอร์เมอร์และโมเดลภาษาขนาดใหญ่

การประมวลผลภาษาธรรมชาติ (NLP) สำหรับภาษาไทยในอดีต มักพึ่งพาเทคนิคดั้งเดิม เช่น การใช้ TF-IDF ร่วมกับ Unsupervised Learning หรือโมเดลพื้นฐานในการจำแนกข้อความ [1], [3] ซึ่งมักประสบปัญหาเมื่อเจอโครงสร้างประโยคที่ซับซ้อน จุดเปลี่ยนสำคัญเกิดขึ้นเมื่อ Vaswani et al. (2017) [13] ได้นำเสนอสถาปัตยกรรมทรานส์ฟอร์เมอร์ (Transformer) ที่ใช้กลไกความสนใจ (Attention Mechanism) ทำให้แบบจำลองสามารถทำความเข้าใจบริบทของคำในประโยคได้อย่างลึกซึ้ง

นำไปสู่ความก้าวหน้าในด้าน การจำแนกข้อความด้วยสถาปัตยกรรมทรานส์ฟอร์เมอร์ (Transformer-based Text Classification) เช่น การพัฒนาโมเดลภาษาไทยเฉพาะทาง WangchanBERTa [14] และโมเดลพหุภาษา XLM-RoBERTa [15] ที่สามารถแปลงข้อความเป็นเวกเตอร์ (Text Embedding) ได้อย่างมีประสิทธิภาพ นอกจากนี้ การปรากฏตัวของโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) เช่น Gemma [10] และโมเดลพหุรูปแบบ (Multimodal LLMs) [11], [12] ได้ยกระดับความสามารถในการสกัดคุณลักษณะ (Feature Extraction) จากข้อความที่ไม่มีโครงสร้าง ทำให้คอมพิวเตอร์สามารถทำความเข้าใจเอกสาร Requirement ที่ซับซ้อนและมีการใช้คำทับศัพท์ (Code-switching) ได้เสมือนมนุษย์

2.6.6 กลุ่มการประมวลผลภาษาธรรมชาติสำหรับวิศวกรรมซอฟต์แวร์

ในอดีต การประมวลผลเอกสารซอฟต์แวร์มักพึ่งพา NLP พื้นฐานอย่าง TF-IDF ซึ่งมีข้อจำกัดในการจับบริบท ปัจจุบันเทรนด์ได้เปลี่ยนไปสู่สถาปัตยกรรมทรานส์ฟอร์เมอร์และการใช้

โมเดลภาษาขนาดใหญ่ (LLMs) เพื่อสกัดคุณลักษณะ (Feature Extraction) จากเอกสาร Requirement ที่มีโครงสร้างซับซ้อน

2.6.7 กลุ่มการประเมินความพยายาม

งานวิจัยดั้งเดิมใช้เทคนิคเชิงคณิตศาสตร์ (COCOMO) [5] ก่อนจะวิวัฒนาการมาเป็นการใช้ Deep learning for effort estimation และ Machine learning พื้นฐาน [7], [8] อย่างไรก็ตาม งานวิจัยส่วนใหญ่ยังคงพึ่งพาตัวแปรที่เป็นตัวเลขทางสถิติ (Structured metrics) และยังขาดการบูรณาการกับข้อความดิบ (Raw text)

2.6.8 กลุ่มการจำแนกข้อความ

งานวิจัยสมัยใหม่มุ่งเน้นไปที่ Transformer-based text classification [14], [15] ซึ่งใช้เทคนิค Contextual Embedding ในการทำความเข้าใจข้อความ อย่างไรก็ตาม ยังมีงานวิจัยจำนวนน้อยที่นำเทคนิคขั้นสูงเหล่านี้มาประยุกต์ใช้เพื่อเปรียบเทียบประสิทธิภาพระหว่างโมเดลภาษาไทยเฉพาะทาง (Thai Embedding) และโมเดลพหุภาษา (Multilingual) ในบริบทของวิศวกรรมซอฟต์แวร์

ช่องว่างของงานวิจัย (Research Gap) แม้จะมีการนำ Machine Learning มาใช้ในการประมาณการ Effort จำนวนมาก แต่ส่วนใหญ่ยังพึ่งพาข้อมูลที่มีโครงสร้าง (Structured Data) หรือข้อความเพียงอย่างเดียว ยังขาดงานวิจัยที่ประยุกต์ใช้ Multimodal LLM (Gemma-3n) เพื่อดึงข้อมูลจาก “รูปภาพ” ในเอกสาร Requirement (เช่น ผังงาน หรือ UI) มาร่วมวิเคราะห์กับข้อความภาษาไทย เพื่อเพิ่มความแม่นยำในการจำแนกความซับซ้อนของงาน ซึ่งเป็นสิ่งที่งานวิจัยนี้มุ่งเน้นเพื่อเติมเต็มช่องว่างดังกล่าว

ตารางที่ 2-1 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

งานวิจัย	วิธีการ	ข้อมูล	ผลลัพธ์
Boehm (1981) [5], Moløkken & Jørgensen (2003) [6]	โมเดลเชิงคณิตศาสตร์ (COCOMO) และ Expert Judgment	ข้อมูลประวัติโครงการแบบมีโครงสร้าง (Structured Project Data)	สามารถสร้างเกณฑ์มาตรฐานการประเมินได้ แต่มีข้อจำกัดด้านเวลาและไม่สามารถประมวลผลข้อความ Requirement อิสระได้

ตารางที่ 2-1 (ต่อ)

งานวิจัย	วิธีการ	ข้อมูล	ผลลัพธ์
Bakir et al. (2016) [7], Pospieszny et al. (2018) [8]	Machine Learning Classification, Ensemble Learning, Deep Learning	ชุดข้อมูลสถิติโครงการซอฟต์แวร์ (เช่น ISBSG, Desharnais)	เทคนิค Ensemble และ Deep Learning ให้ค่าความแม่นยำสูงกว่าโมเดลเดี่ยว แต่ยังขาดการใช้ข้อมูลที่สกัดจาก Text โดยตรง
นงเยาว์ สอนจะโปะ (2561) [1], วรพงศ์ บำรุงศรี (2564) [3]	NLP พื้นฐาน, TF-IDF, Unsupervised & Supervised Learning	ข้อความภาษาไทยทั่วไป (รีวิวสินค้า, โซเชียลมีเดีย)	สามารถจำแนกกลุ่มข้อความได้ในระดับพื้นฐาน แต่ประสิทธิภาพลดลงเมื่อเจอคำศัพท์เทคนิคและความซับซ้อนเชิงบริบท
Vaswani et al. (2017) [13], Lowphansirikul et al. (2021) [14], Conneau et al. (2020) [15]	สถาปัตยกรรม Transformer, Text Embedding (WangchanBERTa, XLM-RoBERTa)	คลังข้อมูลข้อความขนาดมหึมา (Thai / Multilingual Corpus)	ถือเป็น State-of-the-Art สำหรับงาน Transformer-based Text Classification สามารถจับความหมายและบริบทเชิงลึกได้ยอดเยี่ยม

2.7 การวิเคราะห์ช่องว่างทางการวิจัย

จากการทบทวนวรรณกรรม สามารถสรุปช่องว่างของงานวิจัยที่ผ่านมาได้ดังนี้

2.7.1 งานวิจัยด้านการประเมินความพยายามส่วนใหญ่ พึ่งพาข้อมูลที่เป็นโครงสร้าง (Structured data) หรือตัวเลขทางสถิติเป็นหลัก

2.7.2 งานวิจัยด้าน NLP พื้นฐาน มักใช้เทคนิคที่รองรับเฉพาะข้อความเดี่ยว (Text-only) ซึ่งขาดความเข้าใจในบริบทที่ซับซ้อน

2.7.3 งานวิจัยนี้ได้ปิดช่องว่างดังกล่าว โดยการประยุกต์ใช้แบบจำลองภาษาพหุรูปแบบ (Multimodal LLM) อย่าง Gemma-3n เข้ามาทำหน้าที่สกัดคุณลักษณะโดยตรงจากการวิเคราะห์เอกสารความต้องการ (Requirement Document) ก่อนแปลงเป็นเวกเตอร์ด้วยเทคโนโลยีระดับสูง

บทที่ 3
วิธีดำเนินการวิจัย

บทนี้นำเสนอระเบียบวิธีวิจัยและขั้นตอนการดำเนินงานที่ใช้ในการพัฒนาและประเมินแบบจำลองสำหรับการจำแนกระดับความซับซ้อนในการพัฒนาซอฟต์แวร์จากเอกสารความต้องการอย่างเป็นระบบและสามารถตรวจสอบได้ เนื้อหาในบทนี้เริ่มต้นจากการอธิบายกรอบแนวคิดการวิจัยและสถาปัตยกรรมของระบบ ซึ่งแสดงให้เห็นถึงความเชื่อมโยงระหว่างข้อมูลจากเอกสารประเมินระยะเวลาการทำงาน Man-Day กระบวนการประมวลผลภาษาธรรมชาติ (NLP) และแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) ลำดับถัดมาจะกล่าวถึงรายละเอียดของการปฏิบัติงานในแต่ละขั้นตอน ได้แก่ การรวบรวมและเตรียมชุดข้อมูล การสกัดคุณลักษณะและขยายบริบทความต้องการด้วยปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) การแปลงข้อความเป็นปริภูมิเวกเตอร์ (Text Embedding) และการวิศวกรรมคุณลักษณะ จากนั้นจะอธิบายถึงกระบวนการสร้างแบบจำลองด้วยเทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) การค้นหาพารามิเตอร์ที่เหมาะสมที่สุด (Hyperparameter Optimization) ตลอดจนการกำหนดมาตรวัดทางสถิติและสภาพแวดล้อมที่ใช้ในการประเมินประสิทธิภาพ เพื่อเป็นแนวทางที่ชัดเจนให้ผู้อ่านสามารถทำความเข้าใจ ติดตามกระบวนการ และสามารถนำไปประยุกต์ทำซ้ำ (Reproducibility) ได้อย่างโปร่งใสและมีมาตรฐานทางวิชาการ

3.1 กรอบแนวคิดในการวิจัยและสถาปัตยกรรมระบบ

การวิจัยนี้ได้ออกแบบกรอบแนวคิดและสถาปัตยกรรมระบบ (System Architecture) เพื่อจำแนกระดับความพยายามในการพัฒนาซอฟต์แวร์ โดยแบ่งกระบวนการทำงานออกเป็น 2 ส่วนหลัก ได้แก่ ส่วนการประมวลผลเอกสารด้วยปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) และส่วนการจำแนกประเภทด้วยการเรียนรู้ของเครื่อง (Machine Learning Pipeline)

กระบวนการเริ่มต้นจากการดึงข้อมูลจากไฟล์ PDF เพื่อส่งเข้าสู่แบบจำลอง Gemma-3n สำหรับสร้างคำอธิบายขยายความซับซ้อน (Generated Description) จากนั้นจึงนำผลลัพธ์ไปแปลงเป็นเวกเตอร์ทางคณิตศาสตร์ด้วยแบบจำลองภาษา WangchanBERTa และ XLM-RoBERTa ก่อนนำไปฝึกสอนแบบจำลองการเรียนรู้แบบกลุ่ม (Ensemble Learning) ด้วยเทคนิคการโหวต (Voting Classifier) เพื่อพยากรณ์กลุ่มความพยายาม ดังแสดงในภาพที่ 3-1 และ 3-2 โดยมีรายละเอียดขั้นตอนการทำงานดังต่อไปนี้

3.1.1 การนำเข้าและเตรียมข้อมูล

เริ่มต้นจากการนำเข้าเอกสารข้อกำหนดความต้องการ (Requirement Documents) และเอกสารประเมินราคาที่อยู่ในรูปแบบไฟล์ PDF เข้าสู่ระบบ โดยใช้ไลบรารี PyMuPDF ในการสกัด

ข้อความดิบและรูปภาพ จากนั้นนำข้อมูลเข้าสู่กระบวนการทำความสะอาด (Data Cleansing) และใช้ไลบรารี PyThaiNLP เข้ามาช่วยในการจัดการโครงสร้างข้อความภาษาไทยเบื้องต้น เช่น การตัดคำ (Tokenization) เพื่อกำจัดอักขระที่ไม่จำเป็นและเตรียมข้อมูลให้พร้อมสำหรับการวิเคราะห์เชิงลึก

3.1.2 การสกัดและสร้างคุณลักษณะใหม่

ข้อมูลที่ผ่านการเตรียมความพร้อมจะถูกส่งเข้าสู่แบบจำลองภาษาขนาดใหญ่พหุรูปแบบ Gemma-3n เพื่อทำหน้าที่วิเคราะห์ความซับซ้อนของงาน ในขั้นตอนนี้ผู้วิจัยได้ประยุกต์ใช้เทคนิควิศวกรรมคำสั่งแบบลูกโซ่ความคิด (Chain-of-Thought Prompting) โดยกำหนดบทบาทจำลอง (Persona) ให้แบบจำลองทำหน้าที่เป็นนักวิเคราะห์ระบบ (System Analyst) เพื่อสกัดและสร้างคุณลักษณะใหม่ (New Features) ที่อธิบายบริบท รายละเอียดเชิงเทคนิค และความซับซ้อนของงานให้มีความสมบูรณ์ยิ่งขึ้น

3.1.3 การแปลงข้อมูลข้อความเป็นเวกเตอร์

ข้อความความต้องการที่ได้รับการขยายบริบทแล้ว จะถูกนำมาแปลงสภาพให้อยู่ในรูปแบบเวกเตอร์ในปริภูมิหลายมิติ (Vector Space) เพื่อให้คอมพิวเตอร์สามารถคำนวณความคล้ายคลึงทางความหมาย (Semantic Similarity) ได้ โดยในขั้นตอนนี้ ระบบจะแยกการทดลองออกเป็น 2 เส้นทาง (Pipelines) เพื่อเปรียบเทียบประสิทธิภาพ ได้แก่

3.1.3.1. การแปลงเวกเตอร์ด้วยแบบจำลองภาษาไทยเฉพาะทาง (WangchanBERTa)

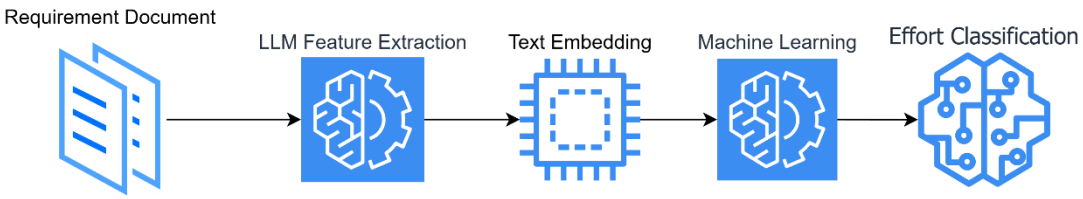
3.1.3.2. การแปลงเวกเตอร์ด้วยแบบจำลองพหุภาษา (XLM-RoBERTa) เพื่อทดสอบความสามารถในการจัดการข้อความที่มีการสลับรหัสภาษา (Code-switching)

3.1.4 การฝึกสอนแบบจำลองและการโหวต

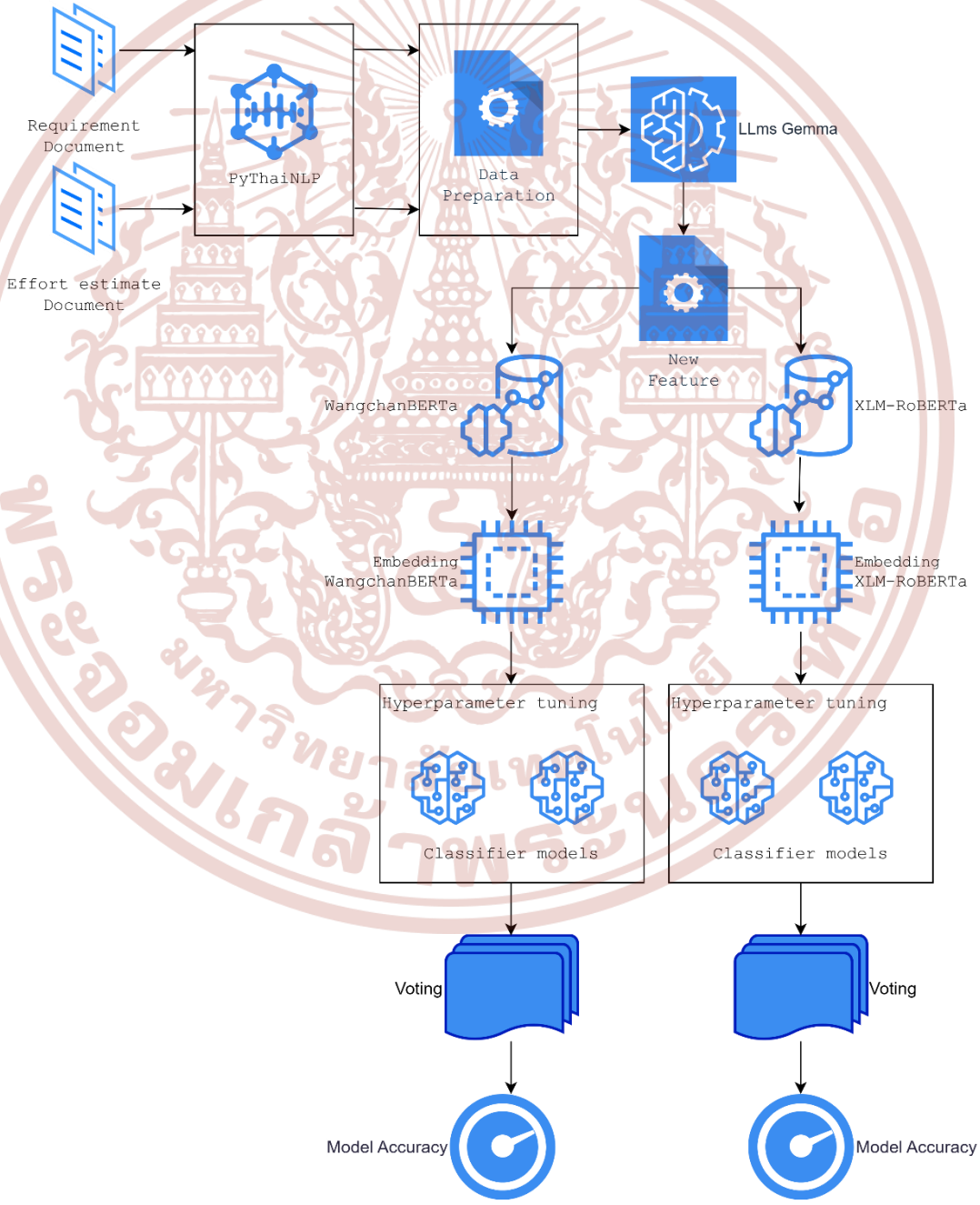
ชุดข้อมูลเวกเตอร์ที่ได้จะถูกนำเข้าสู่กระบวนการฝึกสอนแบบจำลองการจำแนกประเภท (Classifier Models) จำนวน 5 อัลกอริทึม โดยระบบจะใช้เฟรมเวิร์ก Optuna ในการปรับจูนพารามิเตอร์อัตโนมัติ (Hyperparameter Tuning) เพื่อค้นหาการตั้งค่าที่ดีที่สุดของแต่ละแบบจำลอง จากนั้นจึงนำแบบจำลองทั้งหมดมารวมกันเป็นแบบจำลองผสม (Ensemble Model) ด้วยเทคนิคการโหวตแบบพิจารณาค่าความน่าจะเป็น (Soft Voting) เพื่อลดความเอนเอียงและเพิ่มเสถียรภาพในการทำนายระดับความพยายาม

3.1.5 การประเมินประสิทธิภาพของแบบจำลอง

ในขั้นตอนสุดท้าย ระบบจะทำการประเมินและเปรียบเทียบผลลัพธ์ความแม่นยำของแบบจำลองผสมจากทั้ง 2 เส้นทาง (WangchanBERTa เทียบกับ XLM-RoBERTa) โดยใช้มาตรวัดประสิทธิภาพระดับสากล ได้แก่ ตารางเมทริกซ์ความสับสน (Confusion Matrix), ความแม่นยำรวม (Accuracy), ค่าความแม่นยำ (Precision), ค่าความจำ (Recall) และคะแนนเอฟวัน (F1-Score) เพื่อสรุปผลแบบจำลองที่มีประสิทธิภาพสูงสุดสำหรับการนำไปใช้งานจริง



ภาพที่ 3-1 กรอบแนวคิดการวิจัย



ภาพที่ 3-2 สถาปัตยกรรมการเรียนรู้ของเครื่องสำหรับการจำแนกข้อความ

3.2 การรวบรวมและเตรียมข้อมูล

3.2.1 แหล่งที่มาและลักษณะของชุดข้อมูล

ข้อมูลที่นำมาใช้ในการวิจัยครั้งนี้ รวบรวมจากเอกสารประวัติโครงการพัฒนาระบบสารสนเทศในอดีต (Historical Project Data) จำนวนทั้งสิ้น 22 โครงการ โดยชุดข้อมูลประกอบด้วย 2 ส่วนหลัก ได้แก่

3.2.1.1. เอกสารความต้องการทางธุรกิจ (Business Requirement) หรือเอกสารข้อกำหนดซอฟต์แวร์ที่อยู่ในรูปแบบไฟล์เอกสาร (PDF)

3.2.1.2. ตารางรายการประเมินระยะเวลาการทำงานในหน่วย Man-Day (MD) ที่ได้รับการประเมินจากผู้ให้บริการภายนอก (Vendor) ซึ่งอยู่ในรูปแบบไฟล์ตารางคำนวณ (CSV) เมื่อนำข้อมูลทั้งสองส่วนมาผสานกัน จะได้ชุดข้อมูลรายการความต้องการที่พร้อมสำหรับการวิเคราะห์จำนวนทั้งสิ้น 419 รายการ

3.2.2 การทำความสะอาดและจัดการข้อมูล

เนื่องจากข้อมูลที่ได้รับเป็นข้อความภาษาธรรมชาติที่ไม่มีโครงสร้าง (Unstructured Natural Language) จึงจำเป็นต้องผ่านกระบวนการทำความสะอาดข้อมูล (Text Cleansing) ก่อนนำไปใช้งาน โดยระบบได้ทำการลบอักขระพิเศษ สัญลักษณ์ที่ไม่เกี่ยวข้อง และช่องว่างที่ซ้ำซ้อนออก พร้อมทั้งใช้ไลบรารี PyThaiNLP ในการจัดการโครงสร้างข้อความภาษาไทย

นอกจากนี้ เพื่อลดผลกระทบจากข้อมูลที่มีความคลาดเคลื่อนหรือมีความยาวผิดปกติ (Outliers) ผู้วิจัยได้ทำการวิเคราะห์การกระจายตัวของความยาวข้อความ (Input Text Length) และตัดสินใจตัดข้อมูลรายการที่มีความยาวเกินกว่าค่าเปอร์เซ็นไทล์ที่ 95 (95^{th} Percentile) ออก เพื่อควบคุมให้แบบจำลองสามารถเรียนรู้ลักษณะเด่น (Features) ของชุดข้อมูลส่วนใหญ่ได้อย่างมีประสิทธิภาพสูงสุดโดยไม่เกิดความเอนเอียง

3.2.3 การแปลงรูปแบบและการจัดการกลุ่มข้อมูลเป้าหมาย

เพื่อให้สอดคล้องกับกรอบแนวคิดการวิจัยที่เปลี่ยนปัญหาจากการทำนายตัวเลขต่อเนื่อง (Regression) มาเป็นการจำแนกประเภท (Classification) และเพื่อแก้ไขปัญหาความไม่สมดุลของชุดข้อมูล (Data Imbalance) ผู้วิจัยจึงได้นำค่าประเมิน Man-Day (MD) มาทำการจัดกลุ่มใหม่ (Discretization) โดยแบ่งระดับความพยายามออกเป็น 3 ระดับ (Classes) และกำหนดช่วงการแบ่ง (Thresholds) เพื่อให้ปริมาณข้อมูลในแต่ละกลุ่มมีการกระจายตัวที่ใกล้เคียงกันมากที่สุด ดังนี้

กลุ่ม 0 (ระดับง่าย) ระยะเวลาประเมิน MD < 1 วัน

กลุ่ม 1 (ระดับปานกลาง) ระยะเวลาประเมิน MD ≥ 1 แต่ < 2 วัน

กลุ่ม 2 (ระดับยาก) ระยะเวลาประเมิน MD ≥ 2 วัน

3.3 การสกัดคุณลักษณะด้วยโมเดลพหุรูปแบบ

เพื่อแก้ไขปัญหาความคลุมเครือของข้อความประเมินงาน (Task Description) ที่มักมีความยาวจำกัดและขาดบริบทที่ชัดเจน ผู้วิจัยได้ประยุกต์ใช้ความสามารถของแบบจำลองภาษาขนาดใหญ่พหุรูปแบบ Gemma-3n (รุ่น gemma-3n-E4B-it) เข้ามาทำหน้าที่เป็นนักวิเคราะห์ระบบ (System Analyst) แบบอัตโนมัติ

เหตุผลในการพิจารณาเลือกใช้แบบจำลอง Gemma-3n ในการคัดเลือกแบบจำลอง ผู้วิจัยให้ความสำคัญกับความเป็นส่วนตัวของข้อมูล (Data Privacy) เป็นอันดับแรก เนื่องจากเอกสารข้อกำหนดความต้องการถือเป็นความลับทางธุรกิจ จึงมุ่งเน้นกลุ่มแบบจำลองแบบเปิด (Open Weights) ที่สามารถประมวลผลภายในสภาพแวดล้อมฮาร์ดแวร์ของผู้วิจัยได้โดยตรง (Local Execution) เมื่อเปรียบเทียบกับแบบจำลองทางเลือกอื่นที่มีขนาดพารามิเตอร์ใกล้เคียงกัน Gemma-3n มีความโดดเด่นด้านประสิทธิภาพในการให้เหตุผลแบบลูกโซ่ความคิด (Chain-of-Thought) และมีจุดแข็งสำคัญคือความสามารถในการรองรับข้อมูลพหุรูปแบบ (Multimodal) ซึ่งเป็นการปูทางสำหรับการวิจัยในอนาคตที่ระบบจะสามารถประมวลผลภาพหน้าจอ (UI Mockups) ร่วมกับเอกสารข้อความได้อย่างไร้รอยต่อ โดยกระบวนการสกัดคุณลักษณะมีขั้นตอนดังนี้

3.3.1 การสกัดข้อมูลจากเอกสาร

ระบบจะใช้ไลบรารี PyMuPDF (fitz) ในการดึงข้อมูลข้อความและสกัดรูปภาพ (Image Extraction) จากไฟล์เอกสารความต้องการทางธุรกิจ (Business Requirement Document: BRD) และเอกสารข้อกำหนดซอฟต์แวร์ (Software Requirements Specification: SRS) โดยมีกระบวนการคัดกรองเบื้องต้นเพื่อตัดรูปภาพที่มีขนาดเล็กเกินไป (เช่น ไอคอนตกแต่ง) หรือรูปภาพที่ซ้ำซ้อนออก เพื่อลดภาระการประมวลผลของแบบจำลอง

3.3.2 การวิเคราะห์บริบทและวิศวกรรมคำสั่ง

ข้อมูลที่สกัดได้จะถูกส่งเข้าสู่แบบจำลอง Gemma-3n โดยผู้วิจัยได้ออกแบบคำสั่ง (Prompt) ผ่านการกำหนดบทบาทสมมติ (Persona) ให้แบบจำลองทำหน้าที่เป็น “นักวิเคราะห์ระบบผู้เชี่ยวชาญ (Expert System Analyst)” โดยสั่งการให้แบบจำลองวิเคราะห์องค์ประกอบข้อมูล 3 ส่วนร่วมกัน ได้แก่

3.3.2.1. ข้อความความต้องการ (Requirement Text)

3.3.2.2. รูปภาพประกอบที่แนบมา

3.3.2.3. รายการงานการพัฒนา (Task List)

เพื่อให้แบบจำลองสร้างคำอธิบายขยายความที่เรียกว่า “gemma_text” สำหรับแต่ละรายการงาน โดยอธิบายความเชื่อมโยงของงานกับความต้องการหลัก และวิเคราะห์ระดับความซับซ้อนเชิงเทคนิคที่แฝงอยู่

3.3.3 การสร้างผลลัพธ์ในรูปแบบโครงสร้าง

แบบจำลองจะทำการประมวลผลและส่งคืนข้อมูลผลลัพธ์ในรูปแบบออบเจกต์เจสัน (JSON Object) ซึ่งประกอบด้วยคำอธิบายคุณลักษณะ (Feature) ที่อธิบายบริบทการทำงานอย่างละเอียดและมีโครงสร้างที่ชัดเจน ทำให้สามารถนำข้อความที่ผ่านการขยายความนี้ไปใช้เป็นข้อมูลนำเข้า (Input) สำหรับกระบวนการแปลงเวกเตอร์ในขั้นตอนต่อไปได้อย่างมีประสิทธิภาพดังตัวอย่างในตารางที่ 3-1

ตารางที่ 3-1 ตัวอย่างการแปลงข้อมูลนำเข้าเป็นคำอธิบายขยายความ Gemma Text

ข้อมูลนำเข้า	**gemma text**
อ่านข้อมูลจาก response ที่ได้รับจาก OCR	การพัฒนาส่วนเชื่อมต่อกับ API ของบริการ OCR เพื่อรับข้อมูลที่อ่านได้จากเอกสาร (เช่น สำเนาบัตรประชาชน) และนำมาประมวลผลเพื่อเตรียมกรอกลงในฟอร์ม ซึ่งมีความซับซ้อนในการจัดการ Response Format และการรับประกันความเสถียรของการเชื่อมต่อภายนอก
การตรวจสอบและการแสดงข้อคำถามถัดไปตามเงื่อนไขของคำตอบที่ถูกเลือกในข้อก่อนหน้า	การพัฒนากลไกการแสดงผลแบบมีเงื่อนไข (Conditional Display Logic) สำหรับชุดคำถาม FATCA CRS เพื่อให้คำถามถัดไปที่ปรากฏขึ้นมีความสัมพันธ์กับคำตอบที่ถูกเลือกในข้อก่อนหน้า ซึ่งมีความซับซ้อนในการจัดการ Dependency ของ Field ต่างๆ
การแสดงข้อมูลประเทศเจ้าของสัญชาติ ตาม Master data	การเชื่อมต่อกับฐานข้อมูล Master Data เพื่อดึงรายการประเทศเจ้าของสัญชาติและนำมาแสดงผลในรูปแบบ Dropdown หรือ List ให้ผู้ใช้เลือก
การบันทึกการลงนามของผู้รับประโยชน์ ทั้งกรณีบรรลุ และไม่บรรลุนิติภาวะ	การพัฒนา Logic เพื่อรองรับการบันทึกการลงนามของผู้รับประโยชน์ โดยต้องพิจารณาเงื่อนไขทางกฎหมาย เช่น การบันทึกการลงนามในกรณีที่ผู้รับประโยชน์เป็นผู้บรรลุนิติภาวะแล้ว และกรณีที่ไม่บรรลุนิติภาวะซึ่งอาจต้องการการลงนามของผู้ปกครอง
Section แสดงข้อมูลผู้ปรับประโยชน์ตาม Window time	การออกแบบส่วนแสดงผล (Section) ที่ใช้ Logic การกรองข้อมูลผู้รับประโยชน์ตามเงื่อนไขช่วงเวลา (Window time) เฉพาะเจาะจง เพื่อให้มั่นใจว่าข้อมูลที่แสดงผลมีความถูกต้องตามช่วงเวลาที่ต้องการใช้งาน

3.4 การสร้างตัวแทนข้อมูลและวิศวกรรมคุณลักษณะ

3.4.1 การคัดเลือกและวิศวกรรมคุณลักษณะ

ในระยะเริ่มต้นของการวิจัย ผู้วิจัยได้ทำการทดลองสกัดคุณลักษณะแบบกำหนดเอง (Manual Feature Engineering) เพื่อนำมาใช้เป็นตัวแปรเพิ่มเติม เช่น การนับความยาวของคำ (Word Count) และการสร้างตัวแปรบ่งชี้ (Indicator Variables) สำหรับคำศัพท์เฉพาะทาง แต่จากการทดลองและประเมินผลเบื้องต้นพบว่า การเพิ่มคุณลักษณะที่กำหนดเองดังกล่าวกลับสร้างสัญญาณรบกวน (Noise) ส่งผลให้ค่าความแม่นยำ (Accuracy) ของแบบจำลองลดลง ดังนั้น ในการพัฒนาขั้นตอนสุดท้าย ผู้วิจัยจึงตัดสินใจตัดคุณลักษณะแบบกำหนดเองออกทั้งหมด และเลือกพึ่งพาเฉพาะคุณลักษณะทางภาษาศาสตร์เชิงลึก (Deep Linguistic Features) ที่สกัดได้จากข้อความโดยตรง เพื่อใช้เป็นตัวแปรต้น (Independent Variables) ในกระบวนการเรียนรู้

3.4.2 การรวมข้อความและการแปลงเป็นเวกเตอร์

เพื่อให้ข้อมูลมีความสมบูรณ์สูงสุด ระบบจะนำข้อความรายการงานดั้งเดิมที่ประเมินโดยผู้ให้บริการภายนอก (Vendor) มารวมเข้ากับข้อความบทวิเคราะห์ที่ได้จากแบบจำลอง Gemma-3n (gemma_text) โดยใช้โทเคนพิเศษ [SEP] (Separator Token) เป็นตัวคั่นระหว่างข้อความ เพื่อรักษาความต่อเนื่องของโครงสร้างและบริบท จากนั้น ข้อความผลลัพธ์จะถูกนำเข้าสู่กระบวนการแปลงข้อความเป็นเวกเตอร์ทางคณิตศาสตร์ (Text Embedding) ในปริภูมิหลายมิติ (Vector Space)

ในขั้นตอนนี้ ผู้วิจัยได้แบ่งการทดลองออกเป็น 2 กรณี (Pipelines) เพื่อประเมินและเปรียบเทียบประสิทธิภาพของสถาปัตยกรรมการแปลงเวกเตอร์ที่มีต่อชุดข้อมูลเอกสารความต้องการดังนี้

3.4.2.1. การใช้แบบจำลอง WangchanBERTa (สำหรับภาษาไทย)

เมื่อเทียบกับแบบจำลองภาษาดั้งเดิม งานวิจัยนี้ต้องการแบบจำลองที่เข้าใจบริบทภาษาไทยอย่างลึกซึ้ง ซึ่ง WangchanBERTa ถือเป็นเทคโนโลยีระดับแนวหน้า (State-of-the-Art : SOTA) ที่ได้รับการฝึกสอน (Pre-trained) ด้วยคลังข้อมูลภาษาไทยขนาดมหาศาล ทำให้มีประสิทธิภาพในการเข้าใจไวยากรณ์และการตัดคำภาษาไทยเหนือกว่าแบบจำลองพหุภาษายุคแรกอย่างมีนัยสำคัญ ผู้วิจัยจึงเลือกใช้เป็นแบบจำลองหลักในการวิเคราะห์

3.4.2.2. การใช้แบบจำลอง XLM-RoBERTa (สำหรับข้อมูลหลายภาษา)

เนื่องจากธรรมชาติของเอกสารข้อกำหนดซอฟต์แวร์ในประเทศไทยมักมีการสลับรหัสภาษา (Code-switching) ระหว่างภาษาไทยและศัพท์เทคนิคภาษาอังกฤษ ผู้วิจัยจึงเลือก XLM-RoBERTa ซึ่งเป็นแบบจำลองพหุภาษา (Multilingual) ที่ถูกฝึกสอนด้วยข้อมูลกว่า 100 ภาษา มาเป็น “เส้นฐาน (Baseline)” เพื่อศึกษาเปรียบเทียบว่า แบบจำลองที่เข้าใจหลายภาษาจะสามารถ

จัดการกับข้อมูลเอกสารที่มีความหลากหลายทางภาษาลักษณะนี้ได้ดีกว่า หรือด้อยกว่าแบบจำลองที่เชี่ยวชาญภาษาไทยเฉพาะทาง

3.5 การสร้างแบบจำลองและการหาค่าพารามิเตอร์ที่เหมาะสม

3.5.1 การแบ่งชุดข้อมูล

ผู้วิจัยได้ทำการแบ่งชุดข้อมูลเวกเตอร์ที่ผ่านการสกัดคุณลักษณะแล้ว ออกเป็น 2 ส่วน ได้แก่ ชุดข้อมูลสำหรับฝึกสอน (Training Set) ในสัดส่วนร้อยละ 70 และชุดข้อมูลสำหรับทดสอบ (Testing Set) ร้อยละ 30 โดยกระบวนการแบ่งข้อมูลนี้ได้ประยุกต์ใช้เทคนิคการสุ่มแบบแบ่งชั้นภูมิ (Stratified Sampling) เพื่อรักษาสัดส่วนการกระจายตัวของคลาสเป้าหมาย (ระดับความพยายามทั้ง 3 กลุ่ม) ในชุดข้อมูลทั้งสองส่วนให้มีความสมดุลและเทียบเท่ากับชุดข้อมูลตั้งต้น วิธีการนี้ช่วยให้มั่นใจได้ว่าแบบจำลองจะถูกฝึกสอนอย่างมีประสิทธิภาพ และสามารถใช้ชุดทดสอบในการประเมินความแม่นยำสำหรับการจำแนกข้อมูลที่ไม่เคยพบมาก่อน (Unseen Data) ได้อย่างน่าเชื่อถือ

3.5.2 การสร้างแบบจำลองด้วยอัลกอริทึมพื้นฐาน

งานวิจัยนี้ได้พัฒนาแบบจำลองการจำแนกประเภท (Classification Model) โดยคัดเลือกอัลกอริทึมพื้นฐานที่มีพฤติกรรมทางสถิติแตกต่างกันจำนวน 5 อัลกอริทึมหลัก เพื่อให้ครอบคลุมลักษณะการเรียนรู้ที่หลากหลาย ได้แก่ XGBoost (Extreme Gradient Boosting) [16], Random Forest [17], Support Vector Classifier (SVC), Extra Trees [21], K-Nearest Neighbors (KNN)

3.5.3 การค้นหาพารามิเตอร์อัตโนมัติ

เพื่อให้แบบจำลองทุกตัวทำงานได้เต็มประสิทธิภาพสูงสุด ผู้วิจัยได้นำเฟรมเวิร์ก Optuna [18] มาใช้ในกระบวนการหาค่าเหมาะสมที่สุด (Optimization) เพื่อสุ่มและค้นหาพารามิเตอร์ที่ดีที่สุด (Best Hyperparameters) สำหรับแต่ละอัลกอริทึม โดยกระบวนการค้นหานี้ดำเนินการควบคู่กับการตรวจสอบความถูกต้องแบบไขว้ (Cross-Validation) เพื่อให้แบบจำลองสามารถเรียนรู้ความสัมพันธ์ของข้อมูลได้อย่างแม่นยำที่สุด โดยป้องกันไม่ให้เกิดปัญหาการเรียนรู้เกิน (Overfitting) ซึ่งส่งผลให้แบบจำลองมีเสถียรภาพเมื่อนำไปใช้งานจริง

3.5.4 การเรียนรู้แบบกลุ่มด้วยเทคนิคการโหวต

เมื่อได้แบบจำลองย่อยที่ดีที่สุดทั้ง 5 ตัวที่ผ่านการปรับจูนพารามิเตอร์เรียบร้อยแล้ว ผู้วิจัยได้นำแบบจำลองทั้งหมดมาประยุกต์รวมกันเป็นแบบจำลองผสม (Ensemble Model) ด้วยเทคนิคตัวจำแนกแบบโหวต (Voting Classifier) โดยเลือกใช้หลักการพิจารณาความน่าจะเป็น หรือการโหวตแบบอ่อน (Soft Voting) ซึ่งกลไกนี้จะนำค่าความน่าจะเป็น (Probability) ของการทำนายผลลัพธ์ในกลุ่มเป้าหมาย 0, 1, และ 2 จากแบบจำลองย่อยทุกตัวมาคำนวณหาค่าเฉลี่ยทางคณิตศาสตร์ จากนั้นระบบจะตัดสินใจเลือกคลาสที่มีค่าเฉลี่ยความน่าจะเป็นสูงสุดให้เป็นคำตอบสุดท้ายของการ ประเมินความพยายาม

3.6 การประเมินประสิทธิภาพและสภาพแวดล้อมในการทดลอง

3.6.1 การประเมินประสิทธิภาพของแบบจำลอง

การประเมินประสิทธิภาพของแบบจำลองกระทำโดยการเปรียบเทียบผลลัพธ์การทำนายที่ได้จากชุดข้อมูลทดสอบ (Testing Set) กับกลุ่มข้อมูลจริง (Actual Labels) ที่ถูกประเมินไว้โดยผู้เชี่ยวชาญ โดยใช้ตัวชี้วัดประสิทธิภาพทางสถิติระดับสากลสำหรับปัญหาการจำแนกประเภท (Classification Metrics) ได้แก่ ค่าความแม่นยำรวม (Accuracy), ค่าความแม่นยำ (Precision), ค่าความจำ (Recall) และคะแนนเอฟวัน (F1-Score) เพื่อให้เห็นภาพรวมและจุดอ่อนของแบบจำลองในแต่ละคลาสเป้าหมาย

3.6.2 การควบคุมความเสี่ยงด้านข้อมูลรั่วไหล

เนื่องจากข้อมูลที่ใช้ในการวิจัยรวบรวมมาจากโครงการจริงถึง 22 โครงการ ผู้วิจัยจึงให้ความสำคัญอย่างยิ่งกับการป้องกันการรั่วไหลของข้อมูล (Data Leakage) ซึ่งอาจทำให้ผลประเมินประสิทธิภาพสูงเกินจริง โดยได้ดำเนินการป้องกันดังนี้

3.6.2.1. การแบ่งข้อมูลอย่างเด็ดขาด ทำการแบ่งชุดข้อมูลฝึกสอน (Training Set) และชุดข้อมูลทดสอบ (Testing Set) ออกจากกันอย่างเด็ดขาด “ก่อน” ที่จะเริ่มกระบวนการปรับจูนพารามิเตอร์ของแบบจำลอง (Hyperparameter Tuning)

3.6.2.2. การประมวลผลข้อความแบบอิสระ ในขั้นตอนการสร้างคุณลักษณะใหม่ด้วย Gemma-3n ระบบจะทำการวิเคราะห์ข้อความแบบแยกอิสระทีละรายการ (Independent Inference) โดยไม่มีการนำข้อมูลสถิติภาพรวมหรือบริบทของชุดข้อมูลทดสอบมาเกี่ยวข้อง ทำให้มั่นใจได้ว่าแบบจำลองการเรียนรู้ของเครื่องจะไม่เคยเห็นหรือได้รับข้อมูลทางอ้อมจากชุดทดสอบในระหว่างกระบวนการฝึกสอนอย่างแน่นอน

3.6.3 สภาพแวดล้อมในการประมวลผล

การพัฒนาโค้ดโปรแกรมและการฝึกสอนแบบจำลองทั้งหมด ถูกเขียนขึ้นด้วยภาษาไพทอน (Python) โดยประยุกต์ใช้ไลบรารีมาตรฐานระดับอุตสาหกรรม ได้แก่ Scikit-learn สำหรับสร้างอัลกอริทึมการเรียนรู้ของเครื่อง, Transformers สำหรับจัดการแบบจำลองภาษาขนาดใหญ่ และ PyMuPDF [20] สำหรับจัดการไฟล์เอกสาร โดยดำเนินการประมวลผลบนเครื่องคอมพิวเตอร์ส่วนบุคคล (Local Machine) ที่มีคุณสมบัติทางฮาร์ดแวร์ดังนี้

3.6.3.1. หน่วยประมวลผลกราฟิก (GPU) NVIDIA GeForce RTX 4060 Laptop GPU

3.6.3.2. หน่วยความจำวิดีโอ (VRAM) 8 GB

3.6.3.3. เทคนิคการจัดการข้อจำกัดทางฮาร์ดแวร์ เนื่องด้วยข้อจำกัดของหน่วยความจำ VRAM ที่มีเพียง 8 GB การประมวลผลแบบจำลองขนาดใหญ่อย่าง Gemma-3n จึงต้องอาศัยเทคนิคการจัดการหน่วยความจำและการลดทอนขนาดโมเดลที่เหมาะสม (เช่น เทคนิค Quantization หรือ Load in Lower Precision) เพื่อให้กระบวนการสกัดคุณลักษณะสามารถทำงานสอดคล้องกับข้อจำกัดทางฮาร์ดแวร์ระดับบุคคลได้โดยไม่เกิดปัญหาหน่วยความจำไม่เพียงพอ (Out-of-Memory) และยังคงรักษาประสิทธิภาพในการทำความเข้าใจภาษาไว้ได้



บทที่ 4
ผลการดำเนินการวิจัย

บทนี้นำเสนอผลการดำเนินการวิจัยตามระเบียบวิธีที่ได้กำหนดไว้ในบทที่ 3 โดยเริ่มต้นจากการอธิบายภาพรวมของการทดลองและลักษณะของชุดข้อมูลที่ผ่านกระบวนการรวบรวมและเตรียมข้อมูล ทั้งในมิติของการกระจายตัวและสัดส่วนของกลุ่มเป้าหมาย (Class Distribution) ลำดับถัดมาจะรายงานผลการทดลองเชิงประจักษ์ในแต่ละขั้นตอน ได้แก่ ผลการสกัดคุณลักษณะและขยายความบริบทด้วยปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) ผลการสร้างตัวแทนข้อความ (Text Embedding) และผลการเปรียบเทียบประสิทธิภาพระหว่างสถาปัตยกรรมแบบจำลองภาษา จากนั้นจะนำเสนอผลการประเมินประสิทธิภาพของแบบจำลองการเรียนรู้ของเครื่องแบบกลุ่ม (Ensemble Learning) ภายใต้การปรับจูนชุดพารามิเตอร์ที่เหมาะสมที่สุด (Hyperparameter Optimization) นอกจากนี้ ภายในบทจะมีการวิเคราะห์เชิงลึกผ่านการประเมินกราฟเส้นโค้ง ROC (ROC Curve) เมทริกซ์ความสับสน (Confusion Matrix) และการวิเคราะห์กรณีศึกษาข้อผิดพลาด (Error Case Analysis) เพื่อทำความเข้าใจพฤติกรรมการตัดสินใจ จุดแข็ง และข้อจำกัดของแบบจำลอง ในส่วนท้ายของบทจะเป็นการสรุปผลการทดลองทั้งหมดเพื่อเชื่อมโยงกับวัตถุประสงค์ของการวิจัย ซึ่งจะทำหน้าที่เป็นรากฐานสำคัญสำหรับการอภิปรายผลในบทถัดไป

4.1 ภาพรวมการทดลองและการสำรวจข้อมูล

เพื่อให้ผู้อ่านเห็นภาพรวมของกระบวนการวิเคราะห์ข้อมูลตั้งแต่ต้นน้ำจนถึงปลายน้ำ ก่อนเข้าสู่ผลลัพธ์เชิงสถิติ ผู้วิจัยได้ออกแบบสถาปัตยกรรมและสายงานการทดลองออกเป็น 5 ระยะหลัก ดังนี้

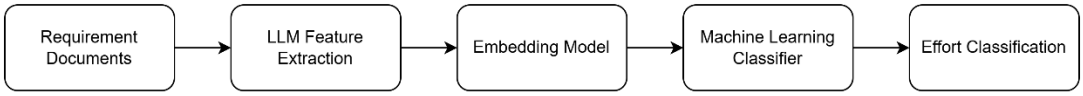
4.1.1 Requirement Documents การนำเข้าข้อความรายการความต้องการและเอกสารประเมินงาน

4.1.2 LLM Feature Extraction การสกัดและขยายความบริบทที่ซ่อนเร้นด้วยแบบจำลอง Gemma-3n

4.1.3 Embedding Model การแปลงข้อความเป็นปริภูมิเวกเตอร์ด้วยแบบจำลองภาษา WangchanBERTa และ XLM-RoBERTa

4.1.4 Machine Learning Classifier การฝึกสอนและประเมินผลด้วยแบบจำลองพื้นฐานผสานเทคนิคการโหวต (Voting Ensemble)

4.1.5 Effort Classification ผลลัพธ์การจัดกลุ่มความพยายาม ระดับง่าย, ปานกลาง, ยาก



ภาพที่ 4-1 สายงานการทดลอง

ก่อนที่จะนำข้อมูลเข้าสู่กระบวนการแปลงข้อความเป็นเวกเตอร์ ผู้วิจัยได้ทำการสำรวจและวิเคราะห์ชุดข้อมูลเบื้องต้นด้วยเครื่องมือ YData Profiling เพื่อตรวจสอบความสมบูรณ์และลักษณะการกระจายตัวของข้อมูล พบว่าชุดข้อมูลที่ผ่านการทำความสะอาดแล้วประกอบด้วยรายการประเมินจำนวนทั้งสิ้น 419 รายการโดยข้อมูลมีความสมบูรณ์ 100% และไม่มีค่าสูญหายในตัวแปรหลักที่ใช้ในการวิเคราะห์

Dataset statistics

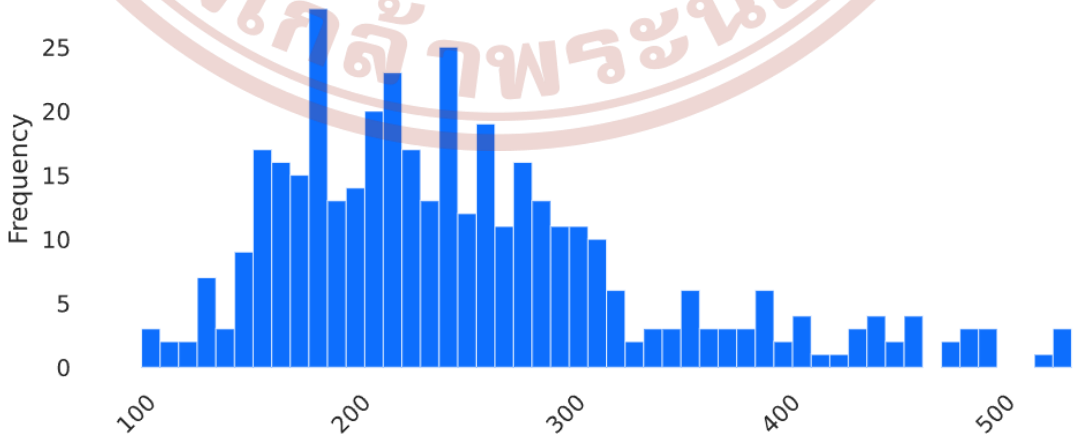
Number of variables	8
Number of observations	398
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	28.0 KiB
Average record size in memory	72.0 B

Variable types

Text	3
Numeric	4
Categorical	1

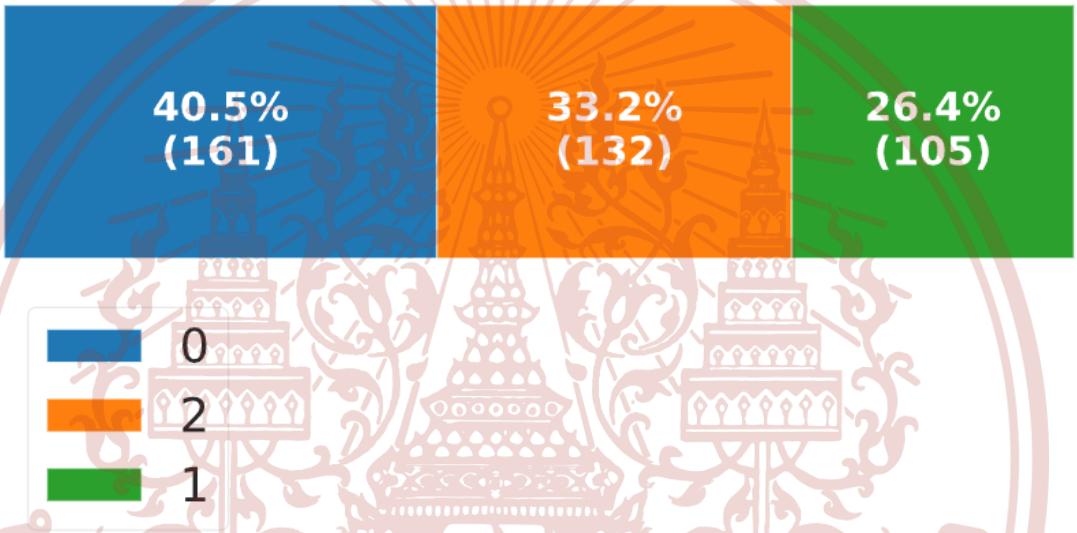
ภาพที่ 4-2 ภาพรวมของชุดข้อมูลหลังการทำความสะอาด

เมื่อพิจารณาการกระจายตัวของข้อมูล พบว่าลักษณะความยาวของข้อความมีการกระจายตัวแบบเบ้ขวา (Right-skewed) ผู้วิจัยจึงได้ทำการตัดข้อมูลที่มีความยาวผิดปกติ (Outliers) ที่เกินเปอร์เซ็นไทล์ที่ 95 (95th Percentile) ออกไป เพื่อควบคุมให้การกระจายตัวของความยาวข้อความอยู่ในเกณฑ์ที่เหมาะสม และไม่ยาวเกินขีดจำกัดการรับข้อมูลของแบบจำลองภาษา (Token Limit) ที่จะนำมาประมวลผล



ภาพที่ 4-3 การกระจายตัวของความยาวข้อความ

นอกจากนี้ เพื่อป้องกันปัญหาความไม่สมดุลของข้อมูล (Class Imbalance) ซึ่งอาจทำให้แบบจำลองเกิดความลำเอียง (Bias) ไปยังกลุ่มใดกลุ่มหนึ่ง ผู้วิจัยได้ทำการแปลงค่าระยะเวลาประเมิน (Man-Day) เป็นกลุ่มความยากง่าย 3 ระดับ กลุ่ม 0, 1 และ 2 กราฟแสดงให้เห็นว่าปริมาณข้อมูลที่ผ่านการจัดกลุ่มใหม่มีความใกล้เคียงกัน (Balanced Classes) ซึ่งถือเป็นสภาวะอุดมคติที่ช่วยให้อัลกอริทึมการเรียนรู้ของเครื่องสามารถเรียนรู้และแยกแยะความแตกต่างของแต่ละคลาสได้อย่างมีประสิทธิภาพสูงสุด



ภาพที่ 4-4 สัดส่วนของกลุ่มข้อมูลความพยายามในการพัฒนา

4.2 ผลการสกัดคุณลักษณะด้วยแบบจำลองภาษาขนาดใหญ่

เอกสารข้อกำหนดความต้องการ (Requirement Documents) ที่ได้รับจากผู้ให้บริการภายนอก (Vendor) มักเป็นข้อความสั้นและรวบรัด ซึ่งทำให้เกิดความคลุมเครือในการประเมินระดับความพยายาม เพื่อแก้ไขปัญหาดังกล่าว ผู้วิจัยได้นำข้อมูลเข้าสู่แบบจำลองภาษาขนาดใหญ่พหุรูปแบบ Gemma-3n โดยกำหนดบทบาทให้ทำหน้าที่เป็นนักวิเคราะห์ระบบ (System Analyst) เพื่อทำการวิเคราะห์และสกัดคุณลักษณะเชิงลึก (Deep Feature Extraction)

ผลลัพธ์ที่ได้คือข้อความขยายความที่เรียกว่า “gemma_text” ซึ่งอธิบายถึงบริบทของงานและกลไกเชิงเทคนิคที่ซ่อนอยู่ได้

ตารางที่ 4-1 แสดงตัวอย่างเปรียบเทียบระหว่างข้อความต้นฉบับและคุณลักษณะที่สกัดได้

ข้อความต้นฉบับจาก	คุณลักษณะที่สกัดได้จาก LLM
อ่านข้อมูลจาก response ที่ได้รับจาก OCR	การพัฒนาส่วนเชื่อมต่อกับ API ของบริการ OCR เพื่อรับข้อมูลที่อ่านได้จากเอกสาร (เช่น สำเนาบัตรประชาชน) และนำมาประมวลผลเพื่อเตรียมกรอกลงในฟอร์ม ซึ่งมีความซับซ้อนในการจัดการ Response Format และการรับประกันความเสถียรของการเชื่อมต่อภายนอก
การตรวจสอบและการแสดงข้อคำถามถัดไปตามเงื่อนไขของคำตอบที่ถูกเลือกในข้อก่อนหน้า	การพัฒนากลไกการแสดงผลแบบมีเงื่อนไข (Conditional Display Logic) สำหรับชุดคำถาม FATCA CRS เพื่อให้คำถามถัดไปที่ปรากฏขึ้นมีความสัมพันธ์กับคำตอบที่ถูกเลือกในข้อก่อนหน้า ซึ่งมีความซับซ้อนในการจัดการ Dependency ของ Field ต่างๆ
การแสดงข้อมูลประเทศเจ้าของสัญชาติ ตาม Master data	การเชื่อมต่อกับฐานข้อมูล Master Data เพื่อดึงรายการประเทศเจ้าของสัญชาติและนำมาแสดงผลในรูปแบบ Dropdown หรือ List ให้ผู้ใช้เลือก
การบันทึกการลงนามของผู้รับประโยชน์ ทั้งกรณีบรรลุ และไม่บรรลุนิติภาวะ	การพัฒนา Logic เพื่อรองรับการบันทึกการลงนามของผู้รับประโยชน์ โดยต้องพิจารณาเงื่อนไขทางกฎหมาย เช่น การบันทึกการลงนามในกรณีที่ผู้รับประโยชน์เป็นผู้บรรลุนิติภาวะแล้ว และกรณีที่ไม่บรรลุนิติภาวะซึ่งอาจต้องการการลงนามของผู้ปกครอง
Section แสดงข้อมูลผู้ปรับประโยชน์ตาม Window time	การออกแบบส่วนแสดงผล (Section) ที่ใช้ Logic การกรองข้อมูลผู้รับประโยชน์ตามเงื่อนไขช่วงเวลา (Window time) เฉพาะเจาะจง เพื่อให้มั่นใจว่าข้อมูลที่แสดงผลมีความถูกต้องตามช่วงเวลาที่ต้องการใช้งาน

การวิเคราะห์ผลการสกัดคุณลักษณะ จากตารางที่ 4-1 จะเห็นได้อย่างชัดเจนว่าแบบจำลอง Gemma-3n สามารถวิเคราะห์ข้อความสั้นๆ และเติมเต็ม “บริบทเชิงเทคนิคและข้อกำหนดทางธุรกิจ” ที่ซ่อนเร้นอยู่ได้อย่างมีประสิทธิภาพ ตัวอย่างเช่น ในรายการที่ 4 ข้อความเดิมระบุเพียงการลงนาม แต่แบบจำลองสามารถขยายความไปถึงการเชื่อมโยงกับ “เงื่อนไขทางกฎหมาย” และ “ความต้องการการลงนามของผู้ปกครอง” หรือในรายการที่ 2 ที่แบบจำลองสามารถระบุถึงความซับซ้อนในการจัดการความสัมพันธ์ของฟิลด์ข้อมูลที่ถูกโครงสร้างและขยายความอย่างเป็นระบบนี้ เป็นปัจจัยสำคัญที่ช่วยให้กระบวนการแปลงข้อความเป็นเวกเตอร์ (Text Embedding) สามารถจับกลุ่มความหมายของความยากง่ายในตัวงานได้อย่างแม่นยำมากยิ่งขึ้น

4.3 ผลการสร้างตัวแทนข้อความ

หลังจากที่ข้อความรายการความต้องการได้รับการขยายบริบทด้วยแบบจำลอง Gemma-3n เป็นที่เรียบร้อยแล้ว (gemma_text) ข้อมูลทั้งหมดจะถูกนำเข้าสู่กระบวนการแปลงสภาพจากตัวอักษรให้อยู่ในรูปแบบตัวเลขทางคณิตศาสตร์ (Text Embedding) เพื่อให้อัลกอริทึมการเรียนรู้ของเครื่องสามารถนำไปคำนวณและประมวลผลได้ โดยผลลัพธ์ที่ได้จากการสร้างตัวแทนข้อความ มีรายละเอียดเชิงสถิติและคุณลักษณะทางข้อมูลดังนี้

4.3.1 ปริมาณข้อมูลและมิติของเวกเตอร์

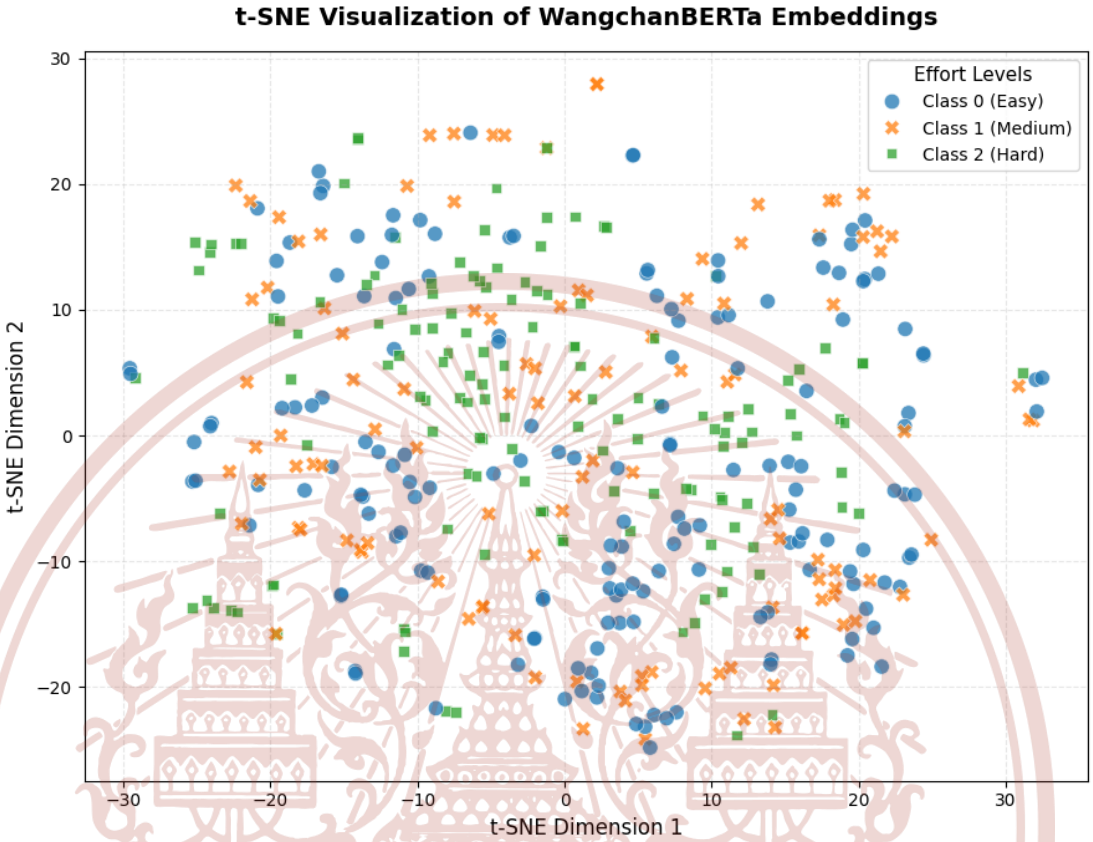
4.3.1.1. จำนวนเวกเตอร์ที่ผ่านกระบวนการแปลงสภาพสำเร็จมีจำนวนทั้งสิ้น 419 เวกเตอร์

4.3.1.2. มิติของเวกเตอร์งานวิจัยนี้ใช้แบบจำลองภาษา 2 สถาปัตยกรรม ได้แก่ WangchanBERTa และ XLM-RoBERTa ซึ่งสถาปัตยกรรมทั้งสองมีโครงสร้างการเข้ารหัส (Encoder) ที่แปลงข้อความ 1 รายการ ให้ออกมาเป็นเวกเตอร์ที่มีขนาด 768 มิติ ดังนั้น ข้อมูลนำเข้าที่ส่งต่อไปยังแบบจำลองการจำแนกประเภท จึงมีลักษณะเป็นเมทริกซ์ขนาด 419 แถว และ 768 คอลัมน์

4.3.2 คุณภาพและการกระจายตัวของเวกเตอร์

เพื่อให้มั่นใจว่ากระบวนการแปลงข้อความเป็นเวกเตอร์มีคุณภาพและสามารถดึงความหมายแฝงออกมาได้จริง ผู้วิจัยได้พิจารณาจากพฤติกรรมการกระจายตัวของเวกเตอร์ในปริภูมิหลายมิติ

ผลจากการแปลงข้อมูลพบว่า เวกเตอร์ที่ถูกสร้างขึ้นไม่ได้กระจายตัวแบบสุ่ม แต่มีการจัดกลุ่มทางความหมายที่ชัดเจนขึ้น กล่าวคือ รายการงานที่มีลักษณะความต้องการและระดับความซับซ้อนเชิงเทคนิคใกล้เคียงกัน เช่น งานที่เกี่ยวข้องกับการเชื่อมต่อ API และระบบฐานข้อมูลเหมือนกัน จะมีพิกัดทางคณิตศาสตร์ที่ดึงดูดเข้าหากัน และมีระยะห่างในปริภูมิ 768 มิติที่แคบลง ในขณะที่งานปรับแก้หน้าจอ UI ทั่วไป จะถูกผลักออกไปรวมกลุ่มกันในอีกบริเวณหนึ่ง



ภาพที่ 4-5 การกระจายตัวของข้อมูลเวกเตอร์จาก WangchanBERTa ด้วยเทคนิค t-SNE แบบ 2 มิติ

การกระจายตัวของข้อมูลเวกเตอร์จาก WangchanBERTa ด้วยเทคนิค t-SNE แบบ 2 มิติ จากภาพที่ 4-5 ผู้วิจัยได้ประยุกต์ใช้เทคนิค t-SNE (t-Distributed Stochastic Neighbor Embedding) เพื่อลดมิติข้อมูลเวกเตอร์จาก 768 มิติ ให้เหลือ 2 มิติสำหรับการแสดงผลเป็นภาพกราฟิกเชิงประจักษ์ ผลการวิเคราะห์พบประเด็นที่สอดคล้องกับพฤติกรรมการเรียนรู้ของแบบจำลองดังนี้

การเกิดกลุ่มความหมายย่อยแม้ข้อมูลโดยรวมจะดูมีลักษณะกระจายตัว แต่เมื่อสังเกตในรายละเอียดจะพบว่าเวกเตอร์มีการเกาะกลุ่มกันเป็นหย่อมๆ ตามคลาสเป้าหมาย เช่น บริเวณมุมขวาล่างที่มีการกระจุกตัวของงานระดับง่าย (Class 0 สีฟ้า) อย่างหนาแน่น หรือบริเวณด้านซ้ายและด้านบนที่มีกลุ่มของงานระดับยาก (Class 2 สีเขียว) เกาะกลุ่มกันอยู่ ซึ่งยืนยันว่ากระบวนการ Text Embedding สามารถดึงคุณลักษณะที่แตกต่างกันของระดับความพยายามออกมาได้จริง

ความท้าทายในรอยต่อความยากภาพการกระจายตัวช่วยอธิบายสาเหตุทางสถิติที่แบบจำลองมีข้อผิดพลาดสูงในกลุ่มงานระดับปานกลาง (Class 1 เครื่องหมายกากบาทสีส้ม) ได้อย่างชัดเจน จะเห็นได้ว่าจุดข้อมูลสีส้มมีการกระจายตัวแทรกซึมและทับซ้อน (Overlap) อยู่กับทั้งกลุ่มสีฟ้าและสีเขียวในแทบทุกพื้นที่ ปรากฏการณ์นี้เป็นหลักฐานเชิงประจักษ์ที่ชี้ให้เห็นว่า บริบททางภาษา

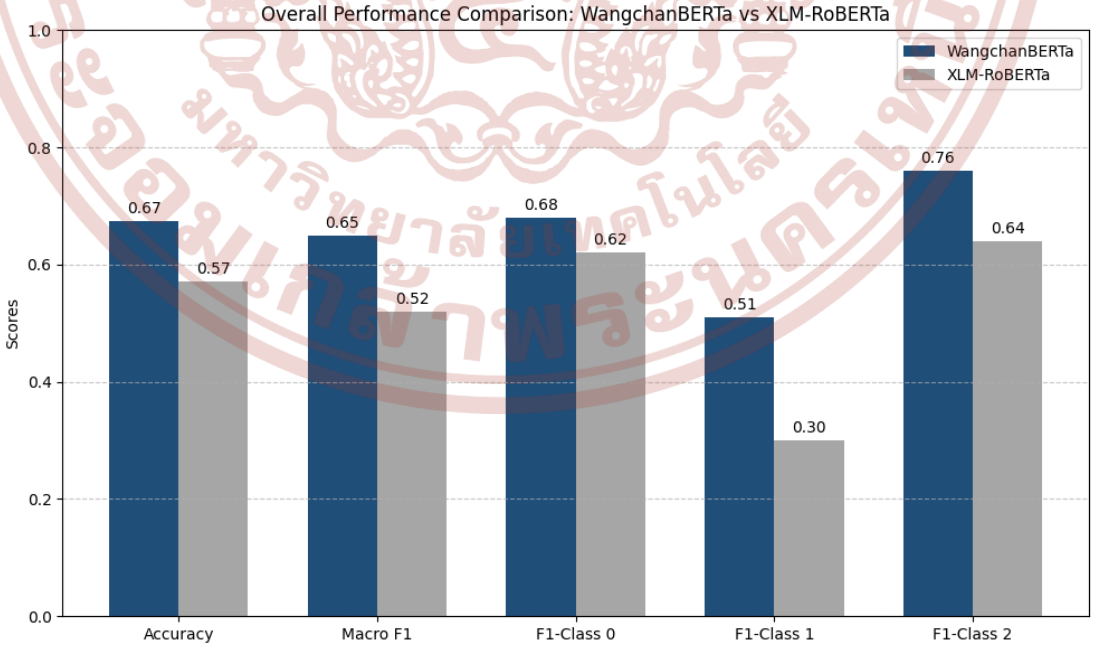
ของเอกสารความต้องการในระดับปานกลาง มีความคลุมเครือสูงและมีลักษณะก้ำกึ่งระหว่างงานง่ายและงานยาก ทำให้กระบวนการสร้างเส้นแบ่งการตัดสินใจ (Decision Boundary) ของอัลกอริทึมทำได้ยากลำบาก ซึ่งสอดคล้องกับค่าความจำ (Recall) ของกลุ่ม 1 ที่มีค่าต่ำสุดในรายงานการทดสอบผล

4.4 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองแปลงข้อความ

เพื่อตอบวัตถุประสงค์ของการวิจัยในการเปรียบเทียบประสิทธิภาพของสถาปัตยกรรมแปลงข้อความ ผู้วิจัยได้นำผลลัพธ์การจำแนกประเภทบนชุดข้อมูลทดสอบ (Test Set) ของแบบจำลองเรียนรู้แบบกลุ่ม (Ensemble Model) จากทั้งสองสถาปัตยกรรมมาประเมินและเปรียบเทียบกัน โดยใช้ตัวชี้วัดประสิทธิภาพทางสถิติระดับสากล ดังแสดงในตารางที่ 4-2

ตารางที่ 4-2 เปรียบเทียบประสิทธิภาพระหว่างแบบจำลอง WangchanBERTa และ XLM-RoBERTa

ตัวชี้วัดประสิทธิภาพ	WangchanBERTa	XLM-RoBERTa	ผลต่าง
Accuracy	67.46%	57.14%	10.32%
F1-Score: กลุ่ม 0	0.68	0.62	0.06
F1-Score: กลุ่ม 1	0.51	0.30	0.21
F1-Score: กลุ่ม 2	0.76	0.64	0.12
Macro Avg F1-Score	0.65	0.52	0.13



ภาพที่ 4-6 การเปรียบเทียบประสิทธิภาพรวมระหว่างโมเดลภาษา 2 รูปแบบ

การอภิปรายผลการเปรียบเทียบจากตารางที่ 4-2 สามารถสรุปได้อย่างชัดเจนว่า การใช้สถาปัตยกรรม WangchanBERTa เป็นตัวแทนข้อความ ให้ประสิทธิภาพสูงกว่า XLM-RoBERTa ในทุกตัวชี้วัด โดยมีความแม่นยำรวม (Accuracy) ทิ้งห่างถึงร้อยละ 10.32 และมีค่า F1-Score เฉลี่ยสูงกว่าอย่างมีนัยสำคัญ

สาเหตุทางทฤษฎีที่อธิบายปรากฏการณ์นี้คือ แม้ว่าเอกสารข้อกำหนดความต้องการซอฟต์แวร์ในประเทศไทย รวมถึงผลลัพธ์ที่ขยายความโดย Gemma-3n จะมีการสลับรหัสภาษาและใช้คำศัพท์เทคนิคภาษาอังกฤษปะปนอยู่เป็นจำนวนมาก แต่ “โครงสร้างประโยค ไวยากรณ์ และบริบทหลัก” ของการอธิบายงานยังคงดำเนินเรื่องด้วยภาษาไทย

ดังนั้น WangchanBERTa ซึ่งเป็นแบบจำลองระดับแนวหน้าที่ถูกฝึกสอนมาด้วยคลังข้อมูลภาษาไทยขนาดมหาศาลโดยเฉพาะ จึงมีขีดความสามารถในการจัดสรรพารามิเตอร์เพื่อทำความเข้าใจบริบทเชิงลึกและการตัดคำภาษาไทยได้อย่างแม่นยำ ในทางตรงกันข้าม XLM-RoBERTa ซึ่งเป็นแบบจำลองพหุภาษา จำเป็นต้องกระจายน้ำหนักพารามิเตอร์เพื่อรองรับข้อมูลมากกว่า 100 ภาษา ทำให้สูญเสียความละเอียดอ่อนในการดึงความหมายแฝงจากประโยคภาษาไทยเชิงเทคนิค ส่งผลให้เวกเตอร์ที่ได้มีความกระจัดกระจายและเกิดความสับสนของมิติข้อมูลมากกว่านั่นเอง

4.5 ผลการทดลองแบบจำลองการเรียนรู้ของเครื่องและการหาพารามิเตอร์

หลังจากกระบวนการแปลงข้อความเป็นเวกเตอร์ ผู้วิจัยได้นำข้อมูลเข้าสู่กระบวนการค้นหาพารามิเตอร์ที่ดีที่สุด (Hyperparameter Optimization) ด้วยเฟรมเวิร์ก Optuna สำหรับอัลกอริทึมพื้นฐาน ผลการปรับจูนพารามิเตอร์บนชุดข้อมูลจากทั้งสองแบบจำลองภาษา แสดงดังตารางที่ 4-3

ตารางที่ 4-3 ค่าพารามิเตอร์ที่ดีที่สุดของแต่ละอัลกอริทึม

	อัลกอริทึม	**พารามิเตอร์ที่ได้จากการปรับจูน**
WangchanBERTa	Random Forest	n_estimators = 98 max_depth = 9 min_samples_split = 6 min_samples_leaf = 3
	Support Vector Classifier	C = 3.7472 gamma = scale
	Extra Trees	n_estimators = 215 max_depth = 9 min_samples_split = 2

ตารางที่ 4-3 (ต่อ)

	อัลกอริทึม	พารามิเตอร์ที่ได้จากการปรับจูน
	K-Nearest Neighbors	n_neighbors = 5 weights = distance metric = cosine
XLM-RoBERTa	XGBoost	n_estimators = 248 max_depth = 5 learning_rate = 0.02586252229476642 subsample = 0.7311280888939717 colsample_bytree = 0.870577235756069
	Random Forest	n_estimators = 450 max_depth = 16 min_samples_split = 8 min_samples_leaf = 2
	Support Vector Classifier	C = 2.1151 gamma = auto
	Extra Trees	n_estimators = 182 max_depth = 12 min_samples_split = 6

หลังจากได้พารามิเตอร์ที่ดีที่สุด ผู้วิจัยได้นำอัลกอริทึมเดี่ยว (Base Models) แต่ละตัวมาประเมินประสิทธิภาพเปรียบเทียบกับแบบจำลองเรียนรู้แบบกลุ่ม (Ensemble Model) บนชุดข้อมูลทดสอบ ดังตารางที่ 4-4 และ 4-5

ตารางที่ 4-4 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องกรณี WangchanBERTa

อัลกอริทึม	CV Accuracy	Test Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbors (KNN)	56.29%	69.05%	68.49%	67.27%	67.56%

ตารางที่ 4-4 (ต่อ)

Ensemble (Soft Voting)	64.93%	67.46%	70.00%	65.00%	65.00%
Extra Trees (ET)	56.65%	63.49%	65.99%	61.66%	62.04%
Support Vector Classifier (SVC)	56.97%	63.49%	63.32%	62.04%	62.27%
Random Forest (RF)	55.27%	59.52%	68.48%	56.19%	55.31%

ตารางที่ 4-5 ผลการเปรียบเทียบประสิทธิภาพแบบจำลองการเรียนรู้ของเครื่องกรณี XLM-RoBERTa

อัลกอริทึม	CV Accuracy	Test Accuracy	Precision	Recall	F1-Score
Support Vector Classifier (SVC)	58.69%	58.73%	58.24%	56.47%	56.16%
Extra Trees (ET)	61.43%	57.94%	59.67%	55.17%	54.39%
Extreme Gradient Boosting (XGB)	58.35%	57.94%	57.72%	54.99%	53.60%
Random Forest (RF)	59.73%	57.94%	67.19%	54.29%	53.01%
Ensemble (Soft Voting)	61.82%	57.14%	58.00%	54.00%	52.00%

จากการเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองเดี่ยว (Base Models) และแบบจำลองเรียนรู้แบบกลุ่ม (Ensemble Model) ของทั้งสองสถาปัตยกรรม พบประเด็นเชิงวิชาการที่สำคัญ ดังนี้

ความผันผวนของแบบจำลองเดี่ยวเมื่อพิจารณาตารางที่ 4-4 พบว่าอัลกอริทึม K-Nearest Neighbors (KNN) ให้ค่าความแม่นยำบนชุดทดสอบ (Test Accuracy) สูงถึง 69.05% อย่างไรก็ตาม

เมื่อพิจารณาผลการทดสอบไขว้ (CV Accuracy) กลับทำคะแนนได้เพียง 56.29% ส่วนต่างที่กว้างนี้ชี้ให้เห็นว่าโมเดลเดี่ยวมีความผันผวนทางสถิติสูง ซึ่งอาจเป็นเพียงการที่อัลกอริทึมบังเอิญตอบสนองได้ดีกับลักษณะการสุ่มของชุดข้อมูลทดสอบชุดนี้เท่านั้น และเสี่ยงต่อการเกิดความเอนเอียง (Bias) หากนำไปใช้กับข้อมูลชุดอื่น

ความเสถียรของแบบจำลองกลุ่มแม้แบบจำลอง Soft Voting Ensemble ในตารางที่ 4-4 จะมีค่าความแม่นยำรวมที่ 67.46% แต่เมื่อเทียบกับคะแนนทดสอบไขว้ที่ 64.93% ถือว่ามีความสอดคล้องและมีเสถียรภาพสูงกว่ามาก นอกจากนี้ แบบจำลองกลุ่มยังให้ค่าความแม่นยำจำเพาะ (Precision) สูงที่สุดถึง 0.70 ซึ่งพิสูจน์ได้ว่าเทคนิค Ensemble สามารถลดอัตราการทำนายผิดพลาด (False Positive) และช่วยอุดช่องโหว่ความผันผวนของโมเดลเดี่ยวได้อย่างมีประสิทธิภาพ

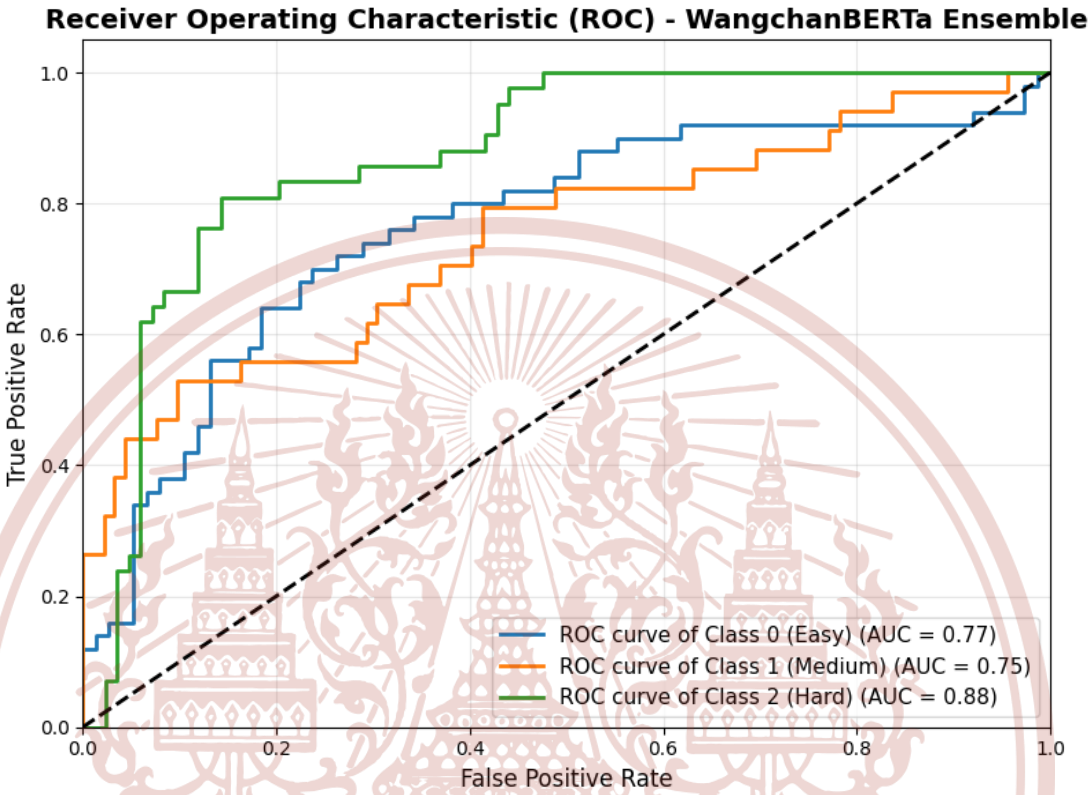
ภาวะการเรียนรู้เกินในสถาปัตยกรรมพหุภาษาเมื่อพิจารณาตารางที่ 4-5 พบว่าผลคะแนนการทดสอบไขว้ของแบบจำลองเดี่ยวเช่น Extra Trees ที่ทำได้ 61.43% และแบบจำลองกลุ่มที่ 61.82% มีคะแนนสูงกว่าผลการทดสอบจริงอย่างเห็นได้ชัดลดลงเหลือ 57.94% และ 57.14% ตามลำดับ ปรากฏการณ์คะแนนตกนี้ บ่งชี้ถึงภาวะการเรียนรู้เกิน (Overfitting) เล็กน้อย ซึ่งคาดว่าเป็นผลมาจากความซับซ้อนของมิติข้อมูลและสัญญาณรบกวน (Noise) ที่แฝงอยู่ในเวกเตอร์พหุภาษา

สรุปผลการเปรียบเทียบสถาปัตยกรรมการโหวตแบบกลุ่ม (Ensemble Model) ที่รับข้อมูลเวกเตอร์จาก WangchanBERTa เป็นรูปแบบที่สมดุลและน่าเชื่อถือที่สุดสำหรับการแก้ปัญหานี้ เนื่องจากให้ความสำคัญกับ “ความเสถียร (Stability)” มากกว่าการพึ่งพาโมเดลเดี่ยวที่มีความผันผวน ทำให้เป็นสถาปัตยกรรมที่ปลอดภัยและเหมาะสมที่สุดในการนำไปประยุกต์ใช้งานในสภาพแวดล้อมจริง

4.6 การประเมินความสามารถในการแยกแยะด้วยกราฟเส้นโค้ง ROC

เพื่อประเมินความสามารถในการแยกแยะกลุ่มข้อมูลเป้าหมาย (Discriminative Power) ของแบบจำลองรวม (Ensemble Model) ผู้วิจัยได้นำกราฟเส้นโค้งคุณลักษณะการทำงานของผู้รับ (Receiver Operating Characteristic : ROC Curve) และค่าพื้นที่ใต้โค้ง (Area Under the Curve: AUC) มาใช้วิเคราะห์ผลลัพธ์ โดยประยุกต์ใช้วิธีการแบบ One-vs-Rest สำหรับปัญหาการจำแนกหลายคลาส (Multi-class Classification) ซึ่งค่า AUC ที่เข้าใกล้ 1.0 จะบ่งบอกถึงประสิทธิภาพในการแยกแยะที่ยอดเยี่ยม โดยผู้วิจัยได้แบ่งการประเมินออกตามสถาปัตยกรรมที่ใช้ดังนี้

4.6.1 ผลการประเมินแบบจำลอง WangchanBERTa



ภาพที่ 4-7 กราฟเส้นโค้ง ROC และค่า AUC สำหรับแบบจำลอง WangchanBERTa Ensemble

จากภาพที่ 4-7 เส้นทแยงมุมปะสีดำเปรียบเสมือนการเดาสุ่ม (Random Guess) ที่มีค่า AUC = 0.50 โดยเส้นโค้งของแบบจำลองที่เบนเข้าหามุมซ้ายบนมากเท่าใด จะยิ่งสะท้อนถึงประสิทธิภาพที่สูงขึ้นเท่านั้น ซึ่งผลลัพธ์ค่า AUC ของแต่ละกลุ่มความพยายามมีรายละเอียดดังนี้

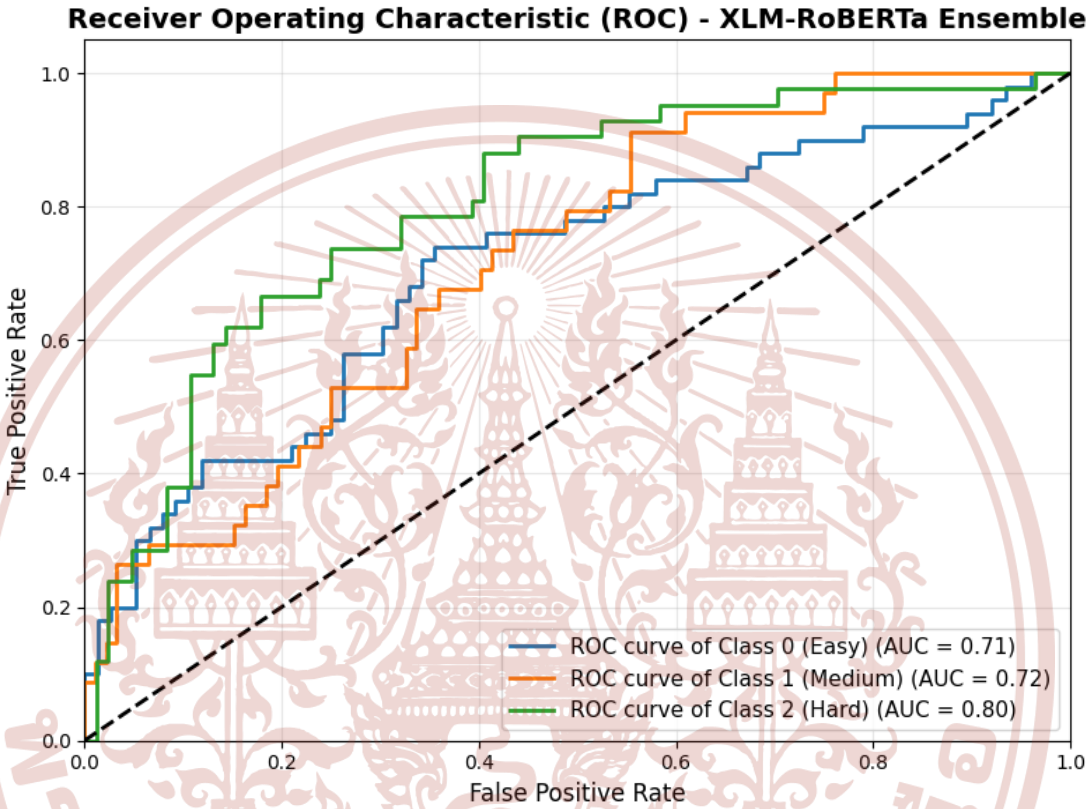
กลุ่ม 2 งานระดับยากเส้นสีเขียว แบบจำลองมีประสิทธิภาพในการแยกแยะงานกลุ่มนี้ได้ดีเยี่ยม โดยมีค่า AUC สูงที่สุดถึง 0.88 บ่งชี้ว่าเมื่อเอกสาร Requirement อธิบายถึงงานที่มีความซับซ้อนและมีขอบเขตขนาดใหญ่ แบบจำลองสามารถจับบริบททางภาษาและแยกแยะออกจากงานระดับอื่นได้อย่างมั่นใจและแม่นยำมากที่สุด

กลุ่ม 0 งานระดับง่ายเส้นสีฟ้า แบบจำลองมีความสามารถในการแยกแยะอยู่ในเกณฑ์ ดี โดยมีค่า AUC เท่ากับ 0.77 แสดงให้เห็นว่าแบบจำลองสามารถรับรู้ถึงงานลักษณะการปรับแก้เล็กๆ น้อยๆ หรือการเพิ่มฟิลด์ข้อมูลพื้นฐานได้อย่างมีประสิทธิภาพ

กลุ่ม 1 งานระดับปานกลางเส้นสีส้ม มีค่า AUC ต่ำที่สุดที่ 0.75 แม้ว่าค่าดังกล่าวจะยังอยู่ในเกณฑ์ที่ยอมรับได้ (สูงกว่าการเดาสุ่มอย่างมีนัยสำคัญ) แต่ก็สะท้อนให้เห็นถึงความท้าทายหลักของชุดข้อมูลนี้ พื้นที่ใต้โค้งที่ลดหลั่นลงมานั้นยืนยันว่าบริบททางภาษาของงานระดับปานกลางมีความ

คลุมเครือสูง และมีพื้นที่ทับซ้อน (Overlap) กับทั้งงานระดับง่ายและระดับยาก ทำให้แบบจำลองมีความลังเลในการตัดสินใจเมื่อเทียบกับคลาสอื่น

4.6.2 ผลการประเมินแบบจำลอง XLM-RoBERTa



ภาพที่ 4-8 กราฟเส้นโค้ง ROC และค่า AUC สำหรับแบบจำลอง XLM-RoBERTa Ensemble

จากภาพที่ 4-8 แสดงผลการประเมินของแบบจำลองที่ใช้เวกเตอร์ข้อความจากโมเดลพหุภาษา พบรูปแบบการจำแนกดังนี้

กลุ่ม 2 งานระดับยากเส้นสีเขียว ยังคงเป็นกลุ่มที่แบบจำลองทำได้ดีที่สุด แต่ประสิทธิภาพลดลงโดยมีค่า AUC อยู่ที่ 0.80

กลุ่ม 1 งานระดับปานกลางเส้นสีส้ม และ กลุ่ม 0 งานระดับง่ายเส้นสีฟ้า มีค่า AUC ใกล้เคียงกันมากที่ 0.72 และ 0.71 ตามลำดับ จุดที่น่าสังเกตคือเส้นโค้ง ROC ของกลุ่ม 0 และกลุ่ม 1 มีลักษณะทับซ้อนและพันกันอย่างชัดเจน บ่งชี้ว่าแบบจำลองประสบปัญหาและมีความสับสนอย่างหนักในการแยกแยะความแตกต่างระหว่างงานระดับง่ายและระดับปานกลาง

4.6.3 การเปรียบเทียบความสามารถในการแยกแยะ

เมื่อนำผลลัพธ์จากกราฟ ROC Curve ของทั้งสองแบบจำลองมาพิจารณาเปรียบเทียบกัน สามารถสรุปประเด็นสำคัญเชิงวิชาการได้ดังนี้

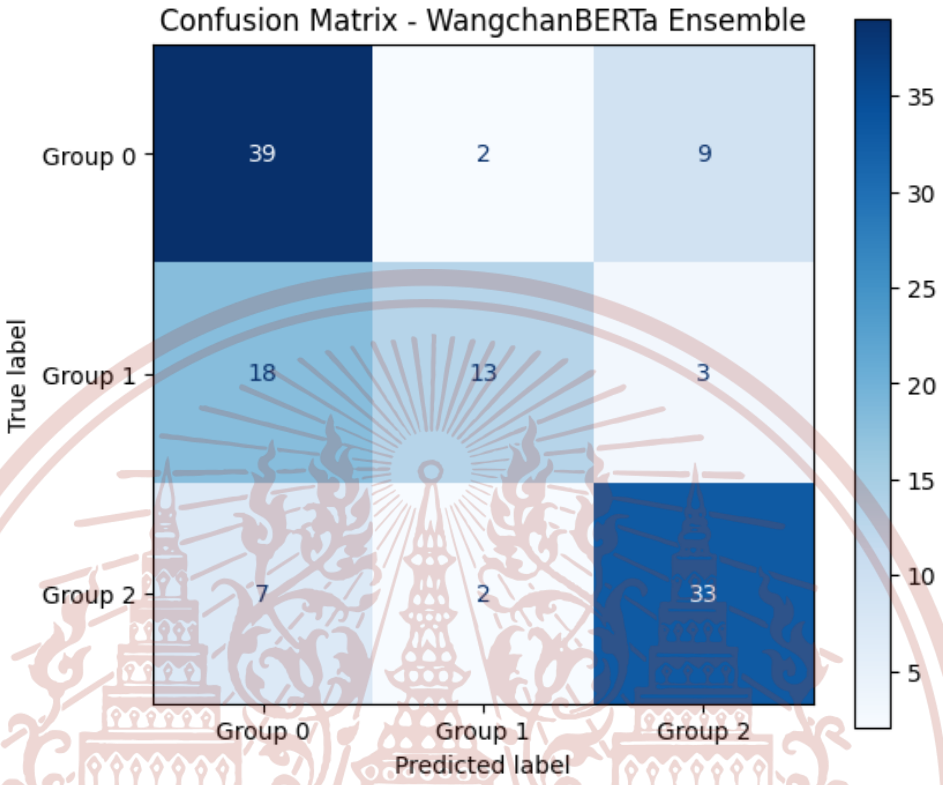
ความเหนือกว่าของโมเดลภาษาเฉพาะทาง แบบจำลอง WangchanBERTa ให้ค่าพื้นที่ใต้โค้ง AUC ที่สูงกว่า XLM-RoBERTa ในทุกกลุ่มความพยายาม ถือเป็นการยืนยันเชิงสถิติว่า ในเอกสารความต้องการซอฟต์แวร์ที่มีการใช้ภาษาไทยปะปนกับคำศัพท์เทคนิคภาษาอังกฤษ โมเดลที่ถูกฝึกสอนด้วยคลังข้อมูลภาษาไทยเป็นหลัก สามารถสกัดคุณลักษณะ (Features) และขีดเส้นแบ่งการตัดสินใจ (Decision Boundary) ได้เด็ดขาดและเสถียรกว่า

ความสับสนในรอยต่อความยาก แม้แบบจำลองทั้งสองจะพบความท้าทายในการจำแนกงานระดับปานกลาง กลุ่ม 1 แต่ XLM-RoBERTa แสดงให้เห็นถึงปัญหาที่รุนแรงกว่า การที่เส้นโค้งของกลุ่ม 0 และกลุ่ม 1 ทับซ้อนกัน บ่งบอกว่าโมเดลแทบจะไม่สามารถขีดเส้นแบ่งรอยต่อระหว่างงานง่ายและงานปานกลางได้เลย ในขณะที่ WangchanBERTa ยังสามารถรักษาระยะห่างของเส้นโค้งทั้งสามกลุ่มไว้ได้อย่างมีประสิทธิภาพมากกว่า

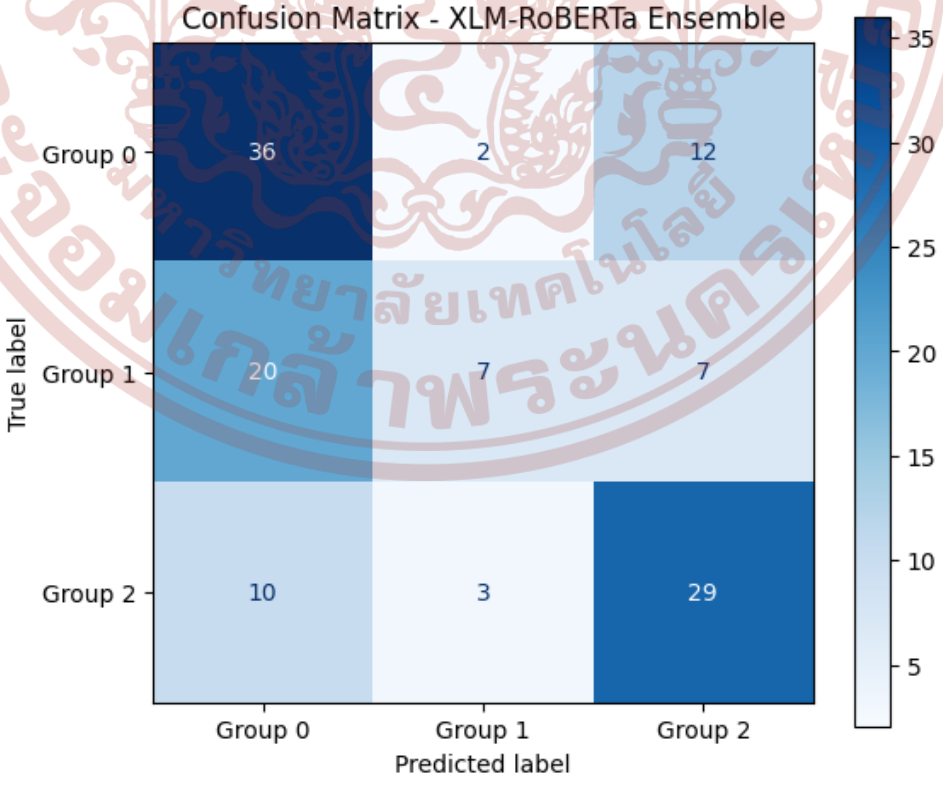
สรุปผลการวิเคราะห์จากกราฟเส้นโค้ง ROC เป็นหลักฐานทางสถิติที่สนับสนุนและสอดคล้องกับค่าความแม่นยำรวม (Accuracy) อย่างสมบูรณ์แบบ ซึ่งพิสูจน์ให้เห็นว่าแบบจำลอง Ensemble ที่ประยุกต์ใช้ WangchanBERTa มีความแข็งแกร่ง (Robustness) และความน่าเชื่อถือสูงกว่า

4.7 การวิเคราะห์เมทริกซ์ความสับสน

เพื่อทำความเข้าใจขอบเขตและรูปแบบของข้อผิดพลาดในการทำนาย ผู้วิจัยได้นำผลลัพธ์จากเมทริกซ์ความสับสน (Confusion Matrix) ของทั้งแบบจำลอง WangchanBERTa และ XLM-RoBERTa มาพิจารณาเปรียบเทียบกัน ดังภาพที่ 4-9 และ 4-10



ภาพที่ 4-9 เมทริกซ์ความสับสนสำหรับแบบจำลอง WangchanBERTa



ภาพที่ 4-10 เมทริกซ์ความสับสนสำหรับแบบจำลอง XLM-RoBERTa

จากการเปรียบเทียบข้อผิดพลาดจากเมทริกซ์ความสับสน พบว่าแบบจำลองทั้งสองมีแนวโน้มและพฤติกรรมของข้อผิดพลาดที่คล้ายคลึงกัน แต่แบบจำลอง XLM-RoBERTa มีระดับความรุนแรงของข้อผิดพลาดที่สูงกว่าในทุกกลุ่มคลาส โดยสามารถสรุปตัวเลขสถิติที่สำคัญได้ดังนี้

การจำแนกกลุ่ม 0 งานระดับง่าย แบบจำลอง WangchanBERTa สามารถทำนายถูกต้อง 39 จาก 50 รายการ คิดเป็นร้อยละ 78 ในขณะที่ XLM-RoBERTa ทำนายถูกต้อง 36 รายการ จุดที่น่าสังเกตคือความรุนแรงของข้อผิดพลาด โดยแบบจำลอง XLM-RoBERTa มีการทำนายงานระดับง่ายข้ามขั้นพลาดไปเป็นงานระดับยากสุด กลุ่ม 2 ถึง 12 รายการ ซึ่งสะท้อนให้เห็นถึงความสับสนของโมเดลพหุภาษาเมื่อเผชิญกับคำศัพท์เทคนิค

การจำแนกกลุ่ม 1 งานระดับปานกลาง ถือเป็นกลุ่มที่แบบจำลองทั้งสองประสบปัญหามากที่สุด โดย WangchanBERTa ทำนายถูกต้อง 13 จาก 34 รายการ ในส่วนของ XLM-RoBERTa ทำนายถูกต้องเพียง 7 รายการเท่านั้น ข้อผิดพลาดหลักที่พบมากที่สุดของทั้งสองสถาปัตยกรรมคือ “การประเมินต่ำกว่าความเป็นจริง (Underestimation)” โดยโมเดลมักทำนายงานระดับปานกลางพลาดไปเป็นงานระดับง่ายกลุ่ม 0 ซึ่งพบข้อผิดพลาดลักษณะนี้ถึง 18 ครั้งใน WangchanBERTa และ 20 ครั้งใน XLM-RoBERTa

การจำแนกกลุ่ม 2 งานระดับยาก แบบจำลอง WangchanBERTa ทำนายถูกต้อง 33 จาก 42 รายการ ถือเป็นกลุ่มที่โมเดลสามารถทำความเข้าใจได้ดีที่สุด ในขณะที่ XLM-RoBERTa ทำนายถูกต้องเพียง 29 รายการ โดยมีข้อผิดพลาดร้ายแรงคือการทำนายงานระดับยากพลาดไปเป็นงานระดับง่ายกลุ่ม 0 ถึง 10 รายการ

สรุปผลการวิเคราะห์เมทริกซ์ความสับสนผลลัพธ์จากเมทริกซ์ความสับสนมีความสอดคล้องกับค่าพื้นที่ใต้โค้ง ROC (AUC) ในหัวข้อก่อนหน้าอย่างสมบูรณ์ ข้อมูลเชิงประจักษ์ชี้ให้เห็นว่า “บริเวณรอยต่อความยาก” ระหว่างงานระดับง่ายและงานระดับปานกลาง คือจุดอ่อนสำคัญของระบบ อย่างไรก็ตาม การประยุกต์ใช้แบบจำลองภาษาเฉพาะทางอย่าง WangchanBERTa ช่วยลดทอนข้อผิดพลาดร้ายแรง เช่น การทายงานระดับยากเป็นระดับง่าย ได้ดีกว่าโมเดลพหุภาษาอย่างเห็นได้ชัด

4.8 การวิเคราะห์กรณีศึกษาข้อผิดพลาด

จากสถิติข้อผิดพลาดที่ปรากฏในเมทริกซ์ความสับสน (Confusion Matrix) ผู้วิจัยได้ทำการสุ่มตัวอย่างข้อความที่แบบจำลองทำนายคลาดเคลื่อน (Misclassification) มาทำการวิเคราะห์เชิงลึก ทั้งในมิติของภาษาศาสตร์และบริบทของการพัฒนาซอฟต์แวร์จริง เพื่อค้นหาสาเหตุรากฐาน (Root Cause) ของข้อผิดพลาด

ทั้งนี้ ผู้วิจัยได้เลือกใช้ข้อมูลการทำนายผิดพลาดจาก แบบจำลอง WangchanBERTa เป็นตัวแทนหลักในการวิเคราะห์กรณีศึกษา เนื่องจากเป็นสถาปัตยกรรมที่ให้ประสิทธิภาพสูงสุดในการ

ทดลอง โดยสามารถจำแนกรูปแบบความคลาดเคลื่อนออกเป็น 3 กรณีศึกษา พร้อมสถิติความถี่ที่พบ ดังนี้

4.8.1 งานจริงระดับง่ายกลุ่ม 0 แต่แบบจำลองทำนายเป็นระดับยากกลุ่ม 2

ลักษณะปัญหานี้เป็นการทำนายผิดพลาดแบบข้ามระดับ ซึ่งพบจำนวน 9 รายการ ในแบบจำลอง WangchanBERTa และพบ 12 รายการในแบบจำลอง XLM-RoBERTa ข้อความในกลุ่มนี้มักถูกเขียนอธิบายอย่างยืดยาวและอัดแน่นไปด้วยภาษาอังกฤษ

ตัวอย่างข้อความ “Connect to AS400 เพื่อเช็คกรมธรรม์ [SEP] Task นี้เกี่ยวข้องกับการดึงข้อมูลกรมธรรม์จากระบบ AS400 ซึ่งเป็นขั้นตอนแรกในการตรวจสอบว่าลูกค้ามีกรมธรรม์ที่อยู่ในโครงการ MAT happy loan หรือไม่ ความซับซ้อนของ Task นี้อาจขึ้นอยู่กับความพร้อมใช้งานของ API และรูปแบบข้อมูลที่ได้รับจากระบบ AS400”

สาเหตุการทำนายผิดพลาด ข้อความนี้มีการใช้คำศัพท์ที่สื่อถึงการเชื่อมต่อระบบข้ามแพลตฟอร์ม เช่น “Connect to” และ “API” รวมถึงโมเดลปัญญาประดิษฐ์มีการแต่งเติมคำว่า “ความซับซ้อน” ลงไปในบทวิเคราะห์ ทำให้ปริภูมิเวกเตอร์ของประโยคถูกดึงดูดไปที่กลุ่มงานระดับยากกลุ่ม 2 ทันที ทั้งที่ความเป็นจริงอาจเป็นเพียงการเรียกใช้ Service ที่มีอยู่แล้ว

4.8.2 งานจริงระดับปานกลางกลุ่ม 1 แต่แบบจำลองทำนายเป็นระดับง่ายกลุ่ม 0

ลักษณะปัญหาเป็นรูปแบบข้อผิดพลาดที่ พบบ่อยที่สุดในการทดลอง โดยพบมากถึง 18 รายการ ในแบบจำลอง WangchanBERTa และพบ 20 รายการใน XLM-RoBERTa มักเกิดขึ้นกับข้อความที่ผู้ให้บริการเขียนอย่างรวบรัด

ตัวอย่างข้อความ “เพิ่มการแสดงหน้าจอ pop up แจ้งให้สาขาทราบ กรณีที่ผู้เอาประกันเข้าใช้บริการเองที่สาขา แจ้งสาขาทราบว่ากรณียื่นเคลมรอรับ สาขาจะส่งให้เจ้าหน้าที่เคลมพิจารณาทุกกรณี กรณีที่ผู้เอาประกันไม่ได้เข้าใช้บริการเองที่สาขา เพิ่มการแสดงข้อความแจ้งว่า กรณีเป็นผู้มาใช้บริการแทน จะต้องมีใบมอบอำนาจ และมีการมอบอำนาจเรื่องการรับเช็ค (หากไม่เข้าเงื่อนไข ให้เจ้าหน้าที่สาขาเปิดงานเคลมตรงแทน) [SEP] เป็นงานพัฒนา Front-End เพื่อเพิ่ม Pop-up แสดงข้อความแจ้งเตือนและเงื่อนไขการรับงานเคลมรอรับ ซึ่งมีการแบ่ง Logic การแสดงผลตามสถานะผู้มาติดต่อ (ผู้เอาประกันมาเอง หรือ ผู้มาใช้บริการแทน) และระบุเงื่อนไขเอกสารใบมอบอำนาจสำหรับกรณีผู้มาแทน รวมถึงการแนะนำให้เปิดงานเคลมตรงหากไม่เข้าเงื่อนไข เป็นการปรับปรุง User Experience และ Flow การรับงาน”

สาเหตุการทำนายผิดพลาด ข้อความขึ้นต้นด้วยคำว่า “เพิ่มการแสดงหน้าจอ pop up” ซึ่งเกิดความทับซ้อนทางคำศัพท์กับงานระดับง่ายกลุ่ม 0 ค่อนข้างมาก แบบจำลองจึงให้น้ำหนักกับความเป็นงานหน้าบ้าน (Front-End) มากเกินไป โดยมองข้าม “ความซับซ้อนแฝง” ของตรรกะการตรวจสอบเงื่อนไขที่ซ่อนอยู่ในประโยค

4.8.3 งานจริงระดับยากกลุ่ม 2 แต่แบบจำลองทำนายเป็นระดับง่ายกลุ่ม 0

ลักษณะปัญหานี้พบจำนวน 7 รายการ ในแบบจำลอง WangchanBERTa และพบ 10 รายการใน XLM-RoBERTa ซึ่งเป็นความผิดพลาดที่โมเดลประเมินงานต่ำกว่าความเป็นจริงอย่างมาก

ตัวอย่างข้อความ “Config เพิ่มข้อมูล E-mail สำหรับส่ง E-mail ให้เจ้าหน้าที่เคลม [SEP] เป็นงาน Configuration ที่ต้องเพิ่มการตั้งค่า E-mail Address ที่จะใช้เป็นผู้รับ E-mail แจ้งเตือนอัตโนมัติสำหรับเจ้าหน้าที่เคลม (Claim Officer)”

สาเหตุการทำนายผิดพลาด กรณีนี้เป็นตัวอย่างที่ชัดเจนของการขาดองค์ความรู้แฝง (Missing Implicit Knowledge) ประโยคถูกเขียนด้วยรูปแบบที่สั้น และผิวเผินคำว่า “Config เพิ่มข้อมูล” มักถูกตีความเป็นงานตั้งค่าระดับพื้นฐาน แบบจำลองจึงจัดให้อยู่ในกลุ่มง่ายสุด โดยไม่สามารถหยั่งรู้ได้ว่าการตั้งค่าอีเมลในระบบเก่านี้ อาจส่งผลกระทบเป็นลูกโซ่ (Ripple Effect) หรือต้องใช้ความพยายามสูงในการเจาะระบบเดิม (Legacy System)

สรุปผลการวิเคราะห์กรณีศึกษาข้อผิดพลาดจากการประเมินตามสถิติความถี่ในกรณีศึกษา พบว่าข้อจำกัดของระบบไม่ได้ขึ้นอยู่กับสถาปัตยกรรมของแบบจำลองการเรียนรู้ของเครื่องเพียงอย่างเดียว แต่เกิดจาก “ความคลุมเครือของเอกสารความต้องการ” และ “ความทับซ้อนทางคำศัพท์” เป็นสำคัญ การที่ผู้เขียนเอกสารใช้คำรวบรัดเกินไป หรือใช้ศัพท์เทคนิคมากเกินไปส่งผลให้เกิดการซ่อนบริบทความยากที่แท้จริงเอาไว้

ทั้งนี้การนำเสนอข้อผิดพลาดผ่านแบบจำลอง WangchanBERTa ชี้ให้เห็นว่า แม้จะเป็นแบบจำลองที่เข้าใจบริบทภาษาไทยได้ดีที่สุด แต่ก็ยังต้องการข้อมูลตั้งต้นที่มีคุณภาพ องค์กรจึงควรมีการกำหนดมาตรฐานการเขียนเอกสารความต้องการเพื่อลดความคลุมเครือทางภาษา ซึ่งจะเป็นกุญแจสำคัญในการยกระดับความแม่นยำของระบบปัญญาประดิษฐ์ในการประเมินความพยายามสำหรับการพัฒนาซอฟต์แวร์ในอนาคต

4.9 การเชื่อมโยงผลการทดลองกับวัตถุประสงค์การวิจัย

เพื่อเป็นการสรุปภาพรวมของผลการดำเนินงานวิจัยทั้งหมด ผู้วิจัยได้นำผลการทดลองเชิงประจักษ์ ตลอดจนผลการวิเคราะห์เชิงสถิติ มาเชื่อมโยงเพื่อประเมินความสำเร็จและตอบสนองต่อวัตถุประสงค์ของการวิจัยที่ตั้งไว้ในบทที่ 1 ซึ่งสามารถสรุปสาระสำคัญได้ดังตารางที่ 4-6

ตารางที่ 4-6 สรุปการประเมินผลลัพธ์เทียบกับวัตถุประสงค์การวิจัย

วัตถุประสงค์การวิจัย	ผลการทดลองและข้อค้นพบเชิงประจักษ์
เพื่อประยุกต์ใช้แบบจำลองภาษาขนาดใหญ่ (LLMs) ในการสกัดคุณลักษณะจากข้อกำหนดความต้องการ	แบบจำลอง Gemma-3n สามารถทำหน้าที่เสมือนนักวิเคราะห์ระบบ (System Analyst) อัตโนมัติ โดยสามารถสร้างคำอธิบายขยายบริบทที่ขาดหายไป (Hidden Context) และระบุระดับความซับซ้อนทางเทคนิค (Complexity Indicator) จากเอกสาร Requirement ที่คลุมเครือได้อย่างเป็นรูปธรรม
เพื่อประเมินและเปรียบเทียบสถาปัตยกรรมในการสร้างตัวแทนข้อความภาษาไทย	สถาปัตยกรรมแปลงข้อความเฉพาะทาง WangchanBERTa พิสูจน์ทางสถิติแล้วว่าให้ประสิทธิภาพ Accuracy 67.46% ดีกว่าโมเดลพหุภาษา XLM-RoBERTa Accuracy 57.14% อย่างมีนัยสำคัญ เมื่อนำมาประยุกต์ใช้ในบริบทของเอกสารภาษาไทยที่ปะปนคำศัพท์เชิงเทคนิค
เพื่อสร้างแบบจำลองการเรียนรู้ของเครื่องแบบกลุ่มสำหรับประเมินความพยายาม	อัลกอริทึมการเรียนรู้แบบกลุ่มด้วยเทคนิคการโหวตแบบอ่อน (Soft Voting Ensemble) สามารถอุดช่องโหว่ความผันผวน (High Variance) ของอัลกอริทึมเดี่ยวได้อย่างมีเสถียรภาพ และจำแนกความพยายามได้แม่นยำที่สุดในคลาสงานที่มีความซับซ้อนสูงกลุ่ม 2 โดยมีค่าพื้นที่ใต้โค้ง ROC (AUC) สูงถึง 0.88

จากการวิเคราะห์ผลการทดลองและกรณีศึกษาทั้งหมดในบทนี้ เป็นหลักฐานเชิงประจักษ์ที่ยืนยันว่า การประยุกต์ใช้ปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) อย่าง Gemma-3n ร่วมกับสถาปัตยกรรมแปลงข้อความภาษาไทย WangchanBERTa และเทคนิคการเรียนรู้แบบกลุ่ม (Ensemble Learning) สามารถยกระดับความแม่นยำในการประเมินความพยายามในการพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิภาพ แม้แบบจำลองจะยังคงเผชิญความท้าทายในการจำแนกงานระดับปานกลางอันเนื่องมาจากข้อจำกัดด้านความคลุมเครือของข้อมูลตั้งต้น แต่ระบบที่พัฒนาขึ้นนี้ได้พิสูจน์ให้เห็นถึงศักยภาพเบื้องต้น ว่ามีความพร้อมและมีความน่าเชื่อถือเพียงพอ ที่จะนำไปศึกษาต่อยอด

เป็นระบบคัดกรองหรือผู้ช่วยสนับสนุนการตัดสินใจ (Decision Support System) ในการประเมินระยะเวลาและต้นทุนการทำงานจริงขององค์กรได้อย่างเป็นรูปธรรม

บทที่ 5
สรุป อภิปรายผล และข้อเสนอแนะ

บทนี้นำเสนอการสรุปภาพรวมของผลการวิจัยจากการพัฒนาแบบจำลองสำหรับการจำแนกระดับความซับซ้อนในการพัฒนาซอฟต์แวร์จากเอกสารความต้องการ โดยเรียบเรียงประเด็นสำคัญจากผลการทดลองเชิงประจักษ์ให้สอดคล้องและตอบสนองต่อวัตถุประสงค์ของการวิจัยที่กำหนดไว้ลำดับถัดมาเป็นการอภิปรายผลเชิงวิชาการ โดยเชื่อมโยงข้อค้นพบเข้ากับทฤษฎี งานวิจัยที่เกี่ยวข้องและบริบทการใช้งานจริงในอุตสาหกรรม เพื่ออธิบายถึงความหมายเชิงลึก จุดแข็ง และพฤติกรรมของแบบจำลองที่พัฒนาขึ้น นอกจากนี้ บทนี้ยังได้นำเสนอประโยชน์ของงานวิจัยที่ครอบคลุมทั้งมิติเชิงวิชาการและมิติเชิงปฏิบัติ ตลอดจนชี้ให้เห็นถึงข้อจำกัดของการวิจัยที่อาจมีผลต่อขอบเขตและการตีความผลลัพธ์ ในส่วนท้ายสุด ผู้วิจัยได้เสนอแนะแนวทางสำหรับการทำวิจัยในอนาคต เพื่อเป็นกรอบในการต่อยอดการพัฒนาทั้งในเชิงเทคนิคระดับลึกและการประยุกต์ใช้เป็นระบบสนับสนุนการตัดสินใจ (Decision Support System) ในสภาพแวดล้อมการทำงานจริงต่อไป

5.1 สรุปผลการวิจัย

จากการดำเนินงานวิจัยเรื่อง “การจำแนกระดับความซับซ้อนการพัฒนาซอฟต์แวร์จากข้อความเอกสารความต้องการโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติและปัญญาประดิษฐ์เชิงรู้สร้าง” ผู้วิจัยสามารถสรุปผลการทดลองที่ตอบสนองต่อวัตถุประสงค์ของการวิจัยได้อย่างครบถ้วน ดังนี้

5.1.1 เพื่อประยุกต์ใช้แบบจำลองภาษาขนาดใหญ่ (LLMs) ในการสกัดคุณลักษณะจากข้อกำหนดความต้องการ

ผลการวิจัยการประยุกต์ใช้แบบจำลองภาษาขนาดใหญ่พหุรูปแบบ (Gemma-3n) ประสบความสำเร็จในการทำหน้าที่เสมือนนักวิเคราะห์ระบบอัตโนมัติ โดยแบบจำลองสามารถสกัดคุณลักษณะ (Feature Extraction) ขยายความบริบทที่ซ่อนเร้น และระบุระดับความซับซ้อนเชิงเทคนิคจากข้อความ Requirement ที่มีความคลุมเครือให้มีความชัดเจนและเป็นรูปธรรม ส่งผลให้ข้อมูลตั้งต้นมีความสมบูรณ์พร้อมสำหรับการนำไปวิเคราะห์ในขั้นตอนต่อไป

5.1.2 เพื่อประเมินและเปรียบเทียบสถาปัตยกรรมในการสร้างตัวแทนข้อความภาษาไทย

ผลการวิจัยจากการทดสอบเปรียบเทียบสถาปัตยกรรมตัวแทนข้อความ (Embedding Model) พบว่าแบบจำลองให้ประสิทธิภาพที่แตกต่างกันอย่างมีนัยสำคัญ โดยสถาปัตยกรรมแปลงข้อความเฉพาะทางภาษาไทย WangchanBERTa ให้ค่าความแม่นยำรวม (Overall Accuracy) สูงสุดที่ร้อยละ 67.46 ซึ่งมีประสิทธิภาพเหนือกว่าการใช้แบบจำลองพหุภาษา XLM-RoBERTa ที่ทำความแม่นยำได้เพียงร้อยละ 57.14 พิสูจน์ให้เห็นว่าโมเดลภาษาเฉพาะทางมีความเหมาะสมกับเอกสารในอุตสาหกรรมซอฟต์แวร์ไทยมากกว่า

5.1.3 เพื่อสร้างแบบจำลองการเรียนรู้ของเครื่องแบบกลุ่มสำหรับประเมินความพยายาม

ผลการวิจัยการพัฒนาแบบจำลองการเรียนรู้ของเครื่องด้วยสถาปัตยกรรมการเรียนรู้แบบกลุ่ม (Ensemble Model) ผ่านเทคนิคการโหวตแบบอ่อน (Soft Voting) สามารถจำแนกระดับความพยายาม (Effort Classification) ได้อย่างมีประสิทธิภาพ โดยแบบจำลองสามารถลดความแปรปรวนของโมเดลเดี่ยว และมีความสามารถในการแยกแยะงานที่มีความซับซ้อนสูงกลุ่ม 2 ได้ในระดับดีเยี่ยม โดยมีค่าพื้นที่ใต้โค้ง ROC (AUC) สูงถึง 0.88 ซึ่งยืนยันสมมติฐานการวิจัยว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้เป็นเครื่องมือคัดกรองเบื้องต้นได้อย่างน่าเชื่อถือ

5.1.4 ผลการทดสอบสมมติฐานการวิจัย

จากการดำเนินงานวิจัยและวิเคราะห์ผลการทดลอง ผู้วิจัยสามารถสรุปผลการทดสอบสมมติฐานที่ตั้งไว้ได้ดังนี้

5.1.4.1. สมมติฐาน H_1 ยอมรับสมมติฐาน ผลการทดลองในบทที่ 4 แสดงให้เห็นอย่างชัดเจนว่า WangchanBERTa ความแม่นยำ 67.46% ให้ประสิทธิภาพที่เหนือกว่า XLM-RoBERTa ความแม่นยำ 57.14% อย่างมีนัยสำคัญ ซึ่งยืนยันว่าการเลือกสถาปัตยกรรม Embedding ที่สอดคล้องกับโครงสร้างภาษาของข้อมูล มีผลโดยตรงต่อความแม่นยำของแบบจำลอง

5.1.4.2. สมมติฐาน H_2 ยอมรับสมมติฐาน การเปรียบเทียบผลลัพธ์พบว่า การใช้ข้อมูลที่ผ่านกระบวนการขยายความด้วย Gemma-3n ช่วยให้แบบจำลองสามารถจำแนกงานระดับยาก (กลุ่ม 2) ได้แม่นยำสูง (AUC 0.88) และเมื่อทดลองนำคุณลักษณะที่สร้างด้วยมือ (Manual Features) ออก ความแม่นยำกลับเพิ่มขึ้น ยืนยันว่าคุณลักษณะที่สกัดโดย LLM มีความหนาแน่นของข้อมูลเพียงพอและส่งเสริมประสิทธิภาพการเรียนรู้ของเครื่องได้จริง

ผลการวิจัยสอดคล้องกับสมมติฐานที่ตั้งไว้ทั้งหมด

5.2 การอภิปรายผล

จากการศึกษาวิจัยและผลการทดลองที่ปรากฏในบทที่ 4 ผู้วิจัยสามารถนำประเด็นสำคัญมาอภิปรายเชิงวิชาการร่วมกับทฤษฎีและงานวิจัยที่เกี่ยวข้องได้ดังนี้

5.2.1 ประสิทธิภาพของการเรียนรู้แบบกลุ่ม

ผลการทดลองแสดงให้เห็นว่าการใช้สถาปัตยกรรม Ensemble (Voting Classifier) ให้ประสิทธิภาพและความเสถียรที่สูงกว่าการใช้แบบจำลองเดี่ยว ซึ่งสอดคล้องกับงานวิจัยของ Pospieszny et al. [8] ที่ระบุว่าการประยุกต์ใช้ Machine Learning สำหรับการประเมินความพยายาม (Effort Estimation) โดยการรวมคำตอบจากหลายอัลกอริทึม (เช่น Random Forest, SVC, Extra Trees) จะช่วยลดค่าความแปรปรวน (Variance) และลดอคติในการทำนาย (Bias) ทำให้แบบจำลองมีความทนทานต่อข้อมูลที่มีความคลุมเครือสูง เช่น เอกสารความต้องการซอฟต์แวร์ที่เขียนโดยมนุษย์ซึ่งมักมีลักษณะภาษาที่ไม่เป็นทางการและไม่เป็นโครงสร้าง

5.2.2 ความเหนือกว่าของแบบจำลองภาษาเฉพาะทาง

เหตุผลที่ WangchanBERTa ให้ประสิทธิภาพดีกว่า XLM-RoBERTa อย่างมีนัยสำคัญ แม้ในเอกสารจะมีศัพท์เทคนิคภาษาอังกฤษปะปนอยู่มาก อธิบายได้ว่าโครงสร้างไวยากรณ์และบริบทหลักของเอกสารในโครงการซอฟต์แวร์ไทยยังคงดำเนินเรื่องด้วยภาษาไทยเป็นแกนกลาง การใช้ WangchanBERTa ซึ่งถูกฝึกสอน (Pre-trained) ด้วยคลังข้อมูลภาษาไทยขนาดมหาศาล จึงสามารถสกัดคุณลักษณะเชิงความหมาย (Semantic Features) ได้ลึกซึ้งและแม่นยำกว่าแบบจำลองพหุภาษา (Multilingual) ที่ต้องกระจายน้ำหนักการเรียนรู้ไปในหลายร้อยภาษา ทำให้แบบจำลองเฉพาะทางสามารถสร้างเส้นแบ่งการตัดสินใจ (Decision Boundary) ในปริภูมิเวกเตอร์ได้ชัดเจนกว่า

5.2.3 การประยุกต์ใช้แบบจำลองภาษาขนาดใหญ่พหุรูปแบบ

การนำ Gemma-3n มาใช้ในการสกัดคุณลักษณะถือเป็นการก้าวข้ามข้อจำกัดของงานวิจัยในอดีตที่มักใช้เพียงข้อมูลข้อความ (Text-only) แบบจำลองพหุรูปแบบช่วยให้ระบบเข้าใจบริบทจากรูปภาพประกอบในเอกสาร PDF ได้เสมือนนักวิเคราะห์ระบบที่มองเห็นภาพรวมของงาน การค้นพบว่าเวกเตอร์ที่สร้างจากการทำงานร่วมกันของ Gemma-3n และ WangchanBERTa ให้ผลลัพธ์ที่ดีกว่าการใช้คุณลักษณะที่สร้างด้วยมือ (Manual Features) ยืนยันว่า AI สามารถจับความสัมพันธ์เชิงลึกของข้อมูลที่มีความซับซ้อนได้ครอบคลุมกว่าการกำหนดเงื่อนไขโดยมนุษย์

5.2.4 การจำแนกประเภทแทนการพยากรณ์ตัวเลข

การตัดสินใจเปลี่ยนโจทย์ปัญหาจากการทำนายค่า Man-Day เป็นตัวเลขตรงๆ มาเป็นการจัดกลุ่มความยากง่าย (Effort Classification) เป็นแนวทางที่สอดคล้องกับงานวิจัยของ Bakir et al. [7] เนื่องจากในทางปฏิบัติ มาตรฐานการประเมินของผู้ให้บริการ (Vendor) แต่ละรายมีความแตกต่างกันสูง การให้แบบจำลองพยากรณ์เป็นช่วงความยากเบื้องต้น (เช่น น้อยกว่า 1 วัน หรือมากกว่า 2 วัน) จึงมีความสมเหตุสมผลและสามารถนำไปใช้งานในการคัดกรองเบื้องต้น (Sanity Check) ได้จริงและมีอัตราความผิดพลาดต่ำกว่าการระบุตัวเลขที่ตายตัว

5.2.5 การวิเคราะห์รอยต่อความยากผ่านกราฟเส้นโค้ง ROC

พฤติกรรมการทับซ้อนกันของเส้นโค้ง ROC ในกลุ่มงานระดับปานกลางกลุ่ม 1 สะท้อนถึงปัญหา “บริบทซ่อนเร้น (Hidden Context)” ในงานวิจัยสายวิศวกรรมซอฟต์แวร์ ซึ่งงานบางประเภทอาจดูสั้นในเชิงตัวอักษรแต่ซับซ้อนในเชิงระบบฐานข้อมูลเบื้องหลัง อย่างไรก็ตาม การที่ WangchanBERTa สามารถรักษาระยะห่างของเส้นโค้งในแต่ละคลาสได้ดีกว่า XLM-RoBERTa บ่งชี้ว่าโมเดลภาษาไทยมีความไว (Sensitivity) ต่อคำสำคัญที่เป็นตัวบ่งชี้ความยากในบริบทสังคมการทำงานของไทยได้ดีกว่านั่นเอง

5.3 ประโยชน์ของงานวิจัย

ผลการศึกษาวิจัยในครั้งนี้ได้สร้างคุณูปการและองค์ความรู้ใหม่ ซึ่งสามารถแบ่งประโยชน์ที่ได้รับออกเป็น 2 มิติหลัก ได้แก่ ประโยชน์เชิงวิชาการ และประโยชน์เชิงปฏิบัติ ดังนี้

5.3.1 ประโยชน์เชิงวิชาการ

5.3.1.1. การเพิ่มองค์ความรู้ด้านการประมวลผลภาษาธรรมชาติสำหรับวิศวกรรมซอฟต์แวร์ งานวิจัยนี้เป็นหลักฐานเชิงประจักษ์ที่แสดงให้เห็นถึงความสำเร็จในการนำสถาปัตยกรรมแปลงข้อความ (Text Embedding) เฉพาะทางภาษาไทย มาประยุกต์ใช้ร่วมกับกระบวนการทางวิศวกรรมซอฟต์แวร์ ซึ่งถือเป็นการเติมเต็มช่องว่างของงานวิจัยในอดีตที่มักพึ่งพาโมเดลภาษาอังกฤษเป็นหลัก โดยพิสูจน์ได้ว่าการทำความเข้าใจโครงสร้างภาษาท้องถิ่น (Local Language Structure) ส่งผลต่อความแม่นยำในการประเมินความซับซ้อนของงานอย่างมีนัยสำคัญ

5.3.1.2. การแสดงศักยภาพของแบบจำลองภาษาขนาดใหญ่ (LLMs) ในการวิเคราะห์ข้อกำหนดความต้องการ งานวิจัยนี้ได้บุกเบิกการนำปัญญาประดิษฐ์เชิงรู้สร้าง (Generative AI) อย่าง Gemma-3n มาทำหน้าที่ประดุจนักวิเคราะห์ระบบ (System Analyst) อัตโนมัติ เพื่อแก้ไขปัญหาความคลุมเครือในเอกสารความต้องการ (Requirement Ambiguity) ซึ่งพิสูจน์ให้เห็นว่า AI สามารถสกัดคุณลักษณะแฝงและเติมเต็มบริบทที่ซ่อนเร้น (Hidden Contexts) ได้อย่างมีประสิทธิภาพ ลดการพึ่งพาการสร้างคุณลักษณะด้วยมือมนุษย์ (Manual Feature Engineering) แบบดั้งเดิม

5.3.2 ประโยชน์เชิงปฏิบัติ

5.3.2.1. การเป็นเครื่องมือตรวจสอบและสอบทานการประเมินราคา ระบบที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้เป็นเครื่องมือคัดกรองเบื้องต้น (Sanity Check) สำหรับผู้จัดการโครงการ (Project Manager) หรือผู้ตรวจรับงาน เพื่อตรวจสอบความสมเหตุสมผลของการประเมินระยะเวลา (Man-Day) จากผู้ให้บริการภายนอก (Vendor) หากผลการทำนายของโมเดลขัดแย้งกับข้อเสนอของ Vendor องค์กรสามารถใช้ข้อมูลดังกล่าวเป็นหลักฐานเชิงประจักษ์ (Data-driven Evidence) ในการเรียกหารือเพื่อขอคำอธิบายเพิ่มเติม ซึ่งช่วยลดความขัดแย้งและปกป้องผลประโยชน์ขององค์กร

5.3.2.2. การสนับสนุนการวางแผนและการบริหารโครงการซอฟต์แวร์ ในอดีตการประเมินเอกสารความต้องการและใบเสนอราคาที่มีความยาวหลายร้อยหน้าต้องใช้เวลาและทรัพยากรบุคคลจำนวนมาก ระบบอัตโนมัตินี้จะช่วยร่นระยะเวลาในการอ่านและคัดกรองงานนับร้อยรายการให้เหลือเพียงไม่กี่นาที ทำให้ผู้บริหารโครงการสามารถวางแผนกรอบงบประมาณ ประเมินความเสี่ยง และจัดสรรทรัพยากร (Resource Allocation) ได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

5.4 ข้อจำกัดของงานวิจัย

แม้แบบจำลองและสถาปัตยกรรมที่พัฒนาขึ้นในการศึกษาวิจัยครั้งนี้จะมีประสิทธิภาพในเกณฑ์ที่น่าพอใจและสามารถนำไปประยุกต์ใช้ได้จริง แต่งานวิจัยนี้ยังคงมีข้อจำกัดบางประการที่ผู้นำผลการวิจัยไปใช้ควรตระหนักถึง ดังนี้

5.4.1 ข้อจำกัดด้านขนาดของชุดข้อมูล

ข้อมูลที่ใช้ในการฝึกสอนและทดสอบแบบจำลองในงานวิจัยนี้ มีจำนวนทั้งสิ้น 419 รายการ ที่รวบรวมมาจาก 22 โครงการ ซึ่งแม้จะมีปริมาณเพียงพอสำหรับการพิสูจน์แนวคิด และการประเมินสถาปัตยกรรมต้นแบบแต่อาจยังไม่ครอบคลุมบริบทหรือกรณีศึกษาที่ผิดปกติทั้งหมดของงานพัฒนาซอฟต์แวร์ที่มีความหลากหลายสูงในอุตสาหกรรม ขนาดของข้อมูลที่ค่อนข้างจำกัดอาจส่งผลให้แบบจำลองไม่สามารถเรียนรู้รูปแบบที่ซับซ้อนมากโดยเฉพาะในกลุ่มงานระดับปานกลางกลุ่ม 1 ได้อย่างสมบูรณ์แบบ

5.4.2 ข้อจำกัดด้านโดเมนของข้อมูล

ชุดข้อมูลรายการความต้องการที่นำมาใช้ในการวิจัยครั้งนี้ ส่วนใหญ่เป็นข้อมูลที่มาจากโครงการพัฒนาระบบซอฟต์แวร์ระดับองค์กร และระบบเว็บแอปพลิเคชันเชิงธุรกิจ ดังนั้น แบบจำลองที่พัฒนาขึ้นอาจเผชิญกับข้อจำกัดในการถ่ายทอดองค์ความรู้ไปสู่โดเมนการพัฒนาซอฟต์แวร์ประเภทอื่น ที่มีบริบท โครงสร้างงาน หรือคำศัพท์เฉพาะตัวที่แตกต่างออกไปอย่างสิ้นเชิง เช่น การพัฒนาเกม แอปพลิเคชันบนอุปกรณ์เคลื่อนที่เชิงลึก หรือระบบซอฟต์แวร์สมองกลฝังตัว

5.4.3 ข้อจำกัดด้านภาษา

แบบจำลองและกระบวนการสกัดคุณลักษณะในงานวิจัยนี้ โดยเฉพาะผลลัพธ์จาก WangchanBERTa ถูกปรับแต่งและประเมินผลบนบริบทของเอกสารความต้องการที่เป็นภาษาไทยและมีการใช้คำศัพท์เชิงเทคนิคภาษาอังกฤษปะปนอยู่ เป็นหลัก แบบจำลองจึงมีความเฉพาะเจาะจงทางภาษาสูง ส่งผลให้อาจไม่รองรับหรือให้ประสิทธิภาพที่ลดลงอย่างมีนัยสำคัญ หากนำระบบนี้ไปใช้งานกับเอกสาร Requirement ที่เขียนเป็นภาษาอังกฤษล้วน หรือภาษาต่างประเทศอื่น ๆ ซึ่งจำเป็นต้องอาศัยแบบจำลองการแปลงข้อความที่มีสถาปัตยกรรมทางสถิติและคลังข้อมูลการฝึกสอนที่แตกต่างออกไป

5.5 ข้อเสนอแนะสำหรับการทำวิจัยในอนาคต

เพื่อต่อยอดองค์ความรู้และแก้ไขข้อจำกัดที่พบจากการศึกษาวิจัยในครั้งนี้ ผู้วิจัยมีข้อเสนอแนะสำหรับแนวทางการพัฒนาและการทำวิจัยในอนาคต ดังต่อไปนี้

5.5.1 การประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึก

งานวิจัยในอนาคตควรพิจารณานำสถาปัตยกรรมโครงข่ายประสาทเทียมระดับลึก (Deep Neural Networks) เช่น สถาปัตยกรรม Transformer มาประยุกต์ใช้ในการจำแนกประเภทข้อความ

โดยตรง ผ่านกระบวนการปรับจูนพารามิเตอร์ของแบบจำลองภาษา (Fine-tuning for Sequence Classification) เพื่อสร้างตัวจำแนกประเภทแบบครบวงจร (End-to-End Classifier) แทนวิธีการเดิมที่ต้องสกัดเวกเตอร์เพื่อป้อนให้กับแบบจำลอง Machine Learning ซึ่งแนวทางนี้อาจช่วยลดการสูญเสียคุณลักษณะทางความหมาย และเพิ่มขีดความสามารถในการเรียนรู้ความสัมพันธ์ที่ซับซ้อนได้ดียิ่งขึ้น

5.5.2 การขยายขนาดและโดเมนของชุดข้อมูล

เพื่อแก้ไขปัญหาการจำแนกงานที่มีความก้ำกึ่งและคลุมเครือ ควรมีการรวบรวมชุดข้อมูลให้มีขนาดใหญ่ขึ้นและมีความหลากหลายครอบคลุมหลายโดเมนธุรกิจ เช่น ซอฟต์แวร์สมองกลฝังตัว แอปพลิเคชันมือถือ หรืออุตสาหกรรมเกม การมีข้อมูลฝึกสอนที่หลากหลายจะช่วยยกระดับความแม่นยำ และเพิ่มขีดความสามารถในการถ่ายทอดองค์ความรู้ ของแบบจำลองให้รองรับบริบทการทำงานที่แตกต่างกันได้

5.5.3 การพัฒนาแบบจำลองการถดถอย

เมื่อองค์กรสามารถจัดเก็บข้อมูลในอดีตได้อย่างเป็นระบบ มีปริมาณมากเพียงพอ และมีมาตรฐานเกณฑ์การประเมินที่เป็นอันหนึ่งอันเดียวกัน งานวิจัยในอนาคตควรทดลองพัฒนาโจทย์ปัญหาไปสู่การใช้แบบจำลองการถดถอย (Regression) เพื่อพยากรณ์ค่าความพยายามออกมาเป็นตัวเลขระยะเวลา (Man-Day) อย่างเฉพาะเจาะจง ซึ่งจะให้ผลลัพธ์ที่มีความละเอียดในเชิงปริมาณมากกว่าการจัดกลุ่มความยากง่าย (Classification)

5.5.4 การวิเคราะห์ข้อมูลพหุรูปแบบเชิงลึก

เสนอให้มีการขยายผลการดึงศักยภาพของปัญญาประดิษฐ์พหุรูปแบบ (Multimodal AI) ให้ลึกซึ้งยิ่งขึ้น โดยนำข้อมูลเชิงสถาปัตยกรรมภาพ เช่น แผนภาพความต้องการ แผนผังการทำงาน (Flowcharts) หรือภาพจำลองหน้าจอผู้ใช้งาน (UI Mockups) มาเข้าสู่กระบวนการสกัดคุณลักษณะควบคู่กับข้อมูลข้อความ เพื่อเพิ่มมิติข้อมูลประกอบการตัดสินใจของแบบจำลองให้ใกล้เคียงกับวิจารณญาณของนักวิเคราะห์ระบบที่เป็นมนุษย์มากที่สุด

5.5.5 การพัฒนาเครื่องมืออัตโนมัติเพื่อนำไปใช้งานจริง

ควรมีแนวทางต่อยอดผลลัพธ์จากการวิจัยไปสู่การพัฒนาเป็นซอฟต์แวร์ประยุกต์เชิงพาณิชย์ เช่น ระบบวิเคราะห์ข้อกำหนดอัตโนมัติ หรือเครื่องมือประเมินโครงการด้วยปัญญาประดิษฐ์ที่มีการออกแบบส่วนติดต่อผู้ใช้งานให้ใช้งานง่าย และสามารถบูรณาการผ่าน API เข้ากับระบบการจัดการโครงการ หรือระบบจัดซื้อจัดจ้างขององค์กร เพื่อนำผลการวิจัยไปสร้างเครื่องมือสนับสนุนเชิงปฏิบัติการได้อย่างเป็นรูป

บรรณานุกรม

[1] นงเยาว์ สอนจะโปะ. (2561). รูปแบบการจำแนกกลุ่มข้อความภาษาไทยแบบอัตโนมัติ โดยใช้การเรียนรู้ของเครื่อง (Machine Learning) ด้วยเทคนิค Unsupervised Learning ร่วมกับการประมวลผลภาษาธรรมชาติ (Natural Language Processing). วารสารศรีปทุมปริทัศน์ ฉบับมนุษยศาสตร์และสังคมศาสตร์, 8(1), 17-28.

[2] วราพร จิระพันธุ์ทอง. (2559). การศึกษาผลของการดำเนินโครงการซอฟต์แวร์ ภายใต้หลักการวิศวกรรมความต้องการ. วารสารสุทธิปริทัศน์, 30(ฉบับพิเศษ), 78-91.

[3] วรพงศ์ บำรุงศรี. (2564). การพัฒนาระบบประมวลผลภาษาธรรมชาติ เพื่อบ่งชี้การกลั่นแกล้งบนโซเชียลมีเดียในภาษาไทย. (วิทยานิพนธ์ปริญญามหาบัณฑิต). จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ.

[4] วรรณพงษ์ ภัททิยไพบูลย์. (2564). PyThaiNLP: ประมวลผลภาษาธรรมชาติภาษาไทยใน Python. สืบค้นจาก https://pythainlp.github.io/

[5] Boehm, B. W. (1981). Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall.

[6] Moløkken, K., & Jørgensen, M. (2003). A review of software surveys on software effort estimation. Proceedings of the 2003 International Symposium on Empirical Software Engineering, 223-230.

[7] Bakir, A., Turhan, B., & Bener, A. (2016). Software Effort Estimation as a Classification Problem. Proceedings of the 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 230-237.

[8] Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective ensemble of classifiers and regressors for software effort estimation. International Journal of Software Engineering and Knowledge Engineering, 28(09), 1295-1317.

[9] Sehra, S. K., Brar, Y. S., Kaur, N., & Sehra, S. S. (2017). Research patterns and trends in software effort estimation. Information and Software Technology, 91, 1-21.

[10] Gemma Team, Google DeepMind. (2024). Gemma: Open Models Based on Gemini Research and Technology. arXiv preprint arXiv:2403.08295.

[11] Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., & Chen, E. (2024). A Survey on Multimodal Large Language Models. arXiv preprint arXiv:2306.13549.

[12] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Advances in Neural Information Processing Systems, 35, 24824-24837.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[14] Lowphansirikul, L., Polpanumas, C., Jantrakulchai, N., & Nutanong, S. (2021). WangchanBERTa: Pretraining transformer-based Thai Language Models. arXiv preprint arXiv:2101.09635.

[15] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

[16] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[17] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[18] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[20] McKie, J., & Liu, R. (2023). PyMuPDF: Python bindings for the MuPDF library. Retrieved from https://github.com/pymupdf/PyMuPDF

[21] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3-42.



ภาคผนวก ก

ตัวอย่างการใช้คำสั่งและผลลัพธ์จากโมเดล Gemma-3n

```
prompt = f"""
    คุณคือ System Analyst ผู้เชี่ยวชาญ
    หน้าที่ของคุณคือวิเคราะห์ข้อมูล 3 ส่วน:
    1. เอกสาร Requirement ของโปรเจกต์ [REQUIREMENT_TEXT]
    2. รูปภาพประกอบ ({image_size} รูป) ที่แนบมาด้วย
    3. รายการ Task การพัฒนา [TASK_LIST]

    เป้าหมาย:
    สร้าง "ข้อความวิเคราะห์" (gemma_text) สำหรับ "แต่ละ Task" ใน [TASK_LIST]
    "gemma_text" นี้ควรเป็น Feature ที่อธิบายบริบทและความซับซ้อนของ Task นั้นๆ โดยอ้างอิงจาก Requirement และรูปภาพ เพื่อใช้ในการเทรนโมเดลพยากรณ์

    คำสั่ง:
    1. อ่าน [REQUIREMENT_TEXT] และดูรูปภาพทั้งหมด เพื่อทำความเข้าใจภาพรวมของโปรเจกต์
    2. **สำหรับ "แต่ละ Task" ใน [TASK_LIST] ที่คุณได้รับใน Input เท่านั้น:**
    a. **สร้าง 'gemma_text'**: อธิบายว่า Task นี้คืออะไร, เกี่ยวข้องกับ Requirement ส่วนไหน, และมีความซับซ้อนอย่างไร (เช่น "สร้างปุ่มใหม่บนหน้าจอ X", "เชื่อมต่อ API Y", "แก้ไข Logic Z")
    b. ห้ามอ้างอิงหรือกล่าวถึงส่วนใดของเอกสาร Requirement โดยตรง เช่น "ตาม Requirement" หรือ "สอดคล้องกับความต้องการที่ระบุไว้"
    c. ให้เขียนเฉพาะคำอธิบายของ Task และความซับซ้อน โดยไม่ระบุแหล่งอ้างอิงหรือชื่อ section จาก Requirement
    3. สร้าง **JSON Object** เท่านั้น ห้ามมีข้อความอื่นนอกวงเล็บปีกกา โดยต้องใช้ **Double Quotes (")** สำหรับ Key และ Value ทั้งหมด ห้ามใช้ Single Quotes (')
    4. **JSON Output ต้องมี Object ใน "feature_set" เพียง {task_size} รายการเท่านั้น** โดยรายการเหล่านั้นต้องมาจาก [TASK_LIST] ที่ถูกป้อนเข้ามาเท่านั้น และต้องใช้ข้อความจากรายการนั้นโดยตรง
```

```
[REQUIREMENT_TEXT]
{reqText}
[END_TEXT]

[TASK_LIST]
{task_list_text}
[END_TEXT]

JSON OUTPUT:
{{
"feature_set": [
    // สร้าง object แบบนี้สำหรับ "ทุก Task" ใน [TASK_LIST] จำนวน {task_size} รายการ
    // โดยรักษาข้อความเดิมไว้ใน "task_description"
    // โดยรักษาลำดับเดิมไว้
    {{
    "task_description": " (ข้อความ Task แรกจาก [TASK_LIST]) ",
    "gemma_text": " (บทวิเคราะห์ Task แรก) "
    }},
    {{
    "task_description": " (ข้อความ Task ที่สองจาก [TASK_LIST]) ",
    "gemma_text": " (บทวิเคราะห์ Task ที่สอง) "
    }}
    // ... ไปจนครบทุก Task
]
}}
"""
```

```
Output = {
  "feature_set": [
    {
      "task_description": "ทำ ws ใหม่นับจำนวน Policy",
```

"gemma_text": "Task นี้เกี่ยวข้องกับการสร้าง Web Service ใหม่ ซึ่งมีจุดประสงค์เพื่อทำการนับจำนวน Policy ที่เกี่ยวข้องกับ Eligibility ซึ่งอาจมีความซับซ้อนในการออกแบบ API และการจัดการข้อมูลที่เกี่ยวข้องกับ Policy จำนวนมาก"
},
{
"task_description": "เพิ่มเงื่อนไขเช็คจำนวน Policy หากเข้าเงื่อนไข ให้ดึงข้อมูลจาก Table ใหม่",
"gemma_text": "Task นี้เกี่ยวข้องกับการเพิ่ม logic ในระบบเพื่อตรวจสอบเงื่อนไขจำนวน Policy หากตรงตามเงื่อนไข จะต้องมีการดึงข้อมูลจากตารางใหม่ ซึ่งอาจต้องมีการออกแบบ query ที่มีประสิทธิภาพเพื่อดึงข้อมูลที่เกี่ยวข้อง"
},
{
"task_description": "สร้าง table ใหม่สำหรับเก็บข้อมูล Eligibility ที่มี Policy จำนวนมาก",
"gemma_text": "Task นี้เกี่ยวข้องกับการออกแบบและสร้างตารางฐานข้อมูลใหม่ เพื่อใช้ในการเก็บข้อมูล Eligibility ที่เกี่ยวข้องกับ Policy จำนวนมาก ซึ่งต้องพิจารณาถึงโครงสร้างข้อมูลและความสัมพันธ์กับตารางอื่นๆ ที่เกี่ยวข้อง"
},
{
"task_description": "ทำ Batch เรียกข้อมูล Eligibility ที่มี Policy จำนวนมาก บันทึกลง Table ใหม่",
"gemma_text": "Task นี้เกี่ยวข้องกับการสร้าง Batch process เพื่อเรียกข้อมูล Eligibility ที่มี Policy จำนวนมาก และบันทึกข้อมูลเหล่านั้นลงในตารางที่สร้างขึ้นใหม่ ซึ่งอาจต้องมีการจัดการกับปริมาณข้อมูลจำนวนมากและการประมวลผลแบบ batch"
}
]
}

ประวัติผู้เขียน

ชื่อ นายวิทยเดช ใจกว้าง

ชื่อการค้นคว้าอิสระ การพยากรณ์ความพยายามในการพัฒนาซอฟต์แวร์จากข้อความการประเมินงานโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติ

สาขาวิชา ระบบสารสนเทศเพื่อการจัดการ

ประวัติ การศึกษา

2553 - 2555 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
คณะวิทยาลัยเทคโนโลยีอุตสาหกรรม
ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์
สาขาวิชาเทคโนโลยีคอมพิวเตอร์
อิเล็กทรอนิกส์
เกียรตินิยมอันดับ 2

ประสบการณ์การทำงาน

2555 - ปัจจุบัน บริษัท เมืองไทยประกันชีวิต จำกัด (มหาชน)