SAC-01-015

กรอบการแก้ไขอัจฉริยะสำหรับการแปลงเอกสาร PDF เป็นข้อความภาษาไทย
Thai PDF-to-Text Intelligent Correction Framework

เปรมฤดี สาริยา[1], ศิรดา เมฆซ้อน[2], ภูริทรัพย์ เดชพิพัฒน์ประชา[3] และ อัครชัย อินทนิล[4]
Paemruedee Sariya[1], Sirada Mekson[2], Bhurisub Dejpipatpracha[3] and Akarachai Inthanil[4]
สาขาเทคโนโลยีดิจิทัล วิชาเอกวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏภูเก็ต[1, 2, 3]
และ ภาควิชาสถิติ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง[4]
Digital Technology Program, Faculty of Science and Technology, Phuket Rajabhat University[1,2,3]
and Department of Statistics, School of science, King Mongkut's Institute of Technology Ladkrabang[4]
e-mail: S6610886121@pkru.ac.th[1], S6610886129@pkru.ac.th[2], bhurisub@pkru.ac.th[3], akarachai.in@kmitl.ac.th[4]

บทคัดย่อ

งานวิจัยนี้เป็นการศึกษาเชิงทดลอง โดยมีวัตถุประสงค์เพื่อ (1) พัฒนากระบวนการแปลงเอกสาร PDF ภาษาไทย เป็นเอกสาร MS Word โดยคงความถูกต้องของข้อความและโครงสร้างของเอกสาร (2) ออกแบบกรอบการทำงานสำหรับแก้ไขข้อความภาษาไทยด้วยปัญญาประดิษฐ์ และ (3) ประเมินความถูกต้องของการแปลงเอกสารเทียบกับวิธีทั่วไป กลุ่มตัวอย่างคือเอกสาร PDF ภาษาไทย เครื่องมือที่ใช้ในการวิจัยคือกรอบการทำงาน Thai PDF-to-Text Intelligent Correction Framework (TPTICF) ซึ่งประกอบด้วย 4 ขั้นตอน ได้แก่ การแยกข้อมูลจากเอกสาร PDF การเตรียมข้อความ การแก้ไขข้อความด้วยปัญญาประดิษฐ์ และการจัดโครงสร้างเอกสารใหม่ โดยเปรียบเทียบแบบจำลองปัญญาประดิษฐ์ Gemini-2.5-Flash, Qwen3-14B และ Typhoon-v2.1-12B-Instruct การวิเคราะห์ข้อมูลใช้การเปรียบเทียบอัตราการแก้ไขตัวอักษร เวลาในการประมวลผล การใช้ทรัพยากรของระบบ และค่าความถูกต้องของข้อความจากการตรวจสอบโดยมนุษย์ ผลการวิจัยพบว่า (1) Gemini-2.5-Flash เป็นแบบจำลองที่เหมาะสมที่สุดสำหรับการประยุกต์ใช้ในกรอบการทำงาน TPTICF เนื่องจากสามารถรักษาสมดุลระหว่างการแก้ไขข้อความ ความถูกต้องของเอกสาร และเวลาในการประมวลผลได้ดีที่สุด (2) ผลการตรวจสอบโดยมนุษย์จากเอกสารที่จงใจสร้างข้อผิดพลาดจำนวน 10 ฉบับ พบว่าระบบมีค่าความถูกต้องเฉลี่ย 99.93% และมีเอกสารจำนวน 7 ฉบับที่มีค่าความถูกต้อง 100.00% และ (3) กรอบการทำงาน TPTICF สามารถเพิ่มความถูกต้องของการแปลงเอกสาร PDF เป็นเอกสาร MS Word จากค่าเฉลี่ย 95.08% ก่อนการแก้ไขด้วยปัญญาประดิษฐ์ เป็น 99.93% หลังการแก้ไข แสดงให้เห็นว่ากรอบการทำงานที่พัฒนาขึ้นช่วยเพิ่มความถูกต้องของการแปลงเอกสาร PDF ภาษาไทยเป็นเอกสาร MS Word

คำสำคัญ: การแปลงเอกสาร PDF ภาษาไทย การแก้ไขข้อความด้วยปัญญาประดิษฐ์ แบบจำลองภาษาขนาดใหญ่

Abstract

This research is an experimental study. The objectives were to (1) develop a process for converting Thai PDF documents into MS Word documents while preserving the accuracy of the text and document structure, (2) design a framework for correcting Thai text using artificial intelligence, and (3) evaluate the accuracy of the document conversion compared to conventional methods. The samples consisted of Thai PDF documents. The research instrument was the Thai PDF-to-Text Intelligent Correction Framework (TPTICF), which consisted of four steps: extracting data from PDF documents, preparing text, editing text using artificial intelligence, and restructuring the document. The artificial intelligence models Gemini-2.5-Flash, Qwen3-14B, and Typhoon-v2.1-12B-Instruct were compared. Data analysis involved comparing the character correction rate, processing time, system resource usage, and text accuracy based on human evaluation. The research results showed that (1) Gemini-2.5-Flash was the most suitable model for application in the TPTICF framework because it achieved the best balance between text correction, document accuracy, and processing time. (2) Human evaluation of 10 documents with intentionally introduced errors showed that the system had an average accuracy of 99.93% and 7 documents had an accuracy of 100.00% and (3) The TPTICF framework increased the accuracy of converting PDF documents to MS Word documents from an average of 95.08% before AI correction to 99.93% after correction, showing that the developed framework improves the accuracy of converting Thai-language documents.

Keywords: Thai PDF conversion, AI-based text correction, Large Language Models

บทนำ

การแปลงเอกสาร PDF เป็นเอกสาร MS Word สามารถดำเนินการโดยการใช้เทคนิค Optical Character Recognition (OCR) สำหรับเอกสารที่เป็นภาพ และการใช้ไลบรารีสำหรับดึงข้อมูลจากเอกสาร PDF โดยตรง เช่น pdf2docx, PyMuPDF และ pdfplumber ซึ่งสามารถดึงข้อความและจัดโครงสร้างเอกสารได้ดีในระดับหนึ่ง อย่างไรก็ตาม ความถูกต้องของผลลัพธ์ยังขึ้นอยู่กับลักษณะและความซับซ้อนของเอกสาร ทำให้ในบางกรณียังคงเกิดข้อผิดพลาดของข้อความหรือรูปแบบเอกสารที่ต้องได้รับการแก้ไขเพิ่มเติม (Donald et al., 2025) ทั้งนี้ ปัญหาการดึงข้อความจาก PDF ที่มีโครงสร้างซับซ้อนได้รับการศึกษามาอย่างต่อเนื่อง โดย Ramakrishnan และคณะ (2012) พบว่าเอกสารวิชาการที่มีหลายคอลัมน์ ตาราง และรูปภาพ มักทำให้ระบบดึงข้อความผิดลำดับหรือรวมข้อความจากส่วนต่าง ๆ เข้าด้วยกัน การระบุตำแหน่งและขอบเขตของแต่ละส่วนในเอกสารก่อนดึงข้อความจึงเป็นขั้นตอนสำคัญที่ช่วยเพิ่มความถูกต้องของผลลัพธ์ และ Adhikari และ Agarwal (2024) ได้เปรียบเทียบประสิทธิภาพของเครื่องมือแปลงและดึงข้อความจากเอกสาร PDF หลายตัวในเอกสารหลากประเภท พบว่าไม่มีเครื่องมือใดที่เหมาะสมกับทุกรูปแบบเอกสาร ในขณะที่ Lin (2024) เสนอแนวทางการวิเคราะห์โครงสร้าง PDF ที่ช่วยยกระดับคุณภาพการดึงข้อมูลสำหรับระบบที่ใช้ AI ค้นหาข้อมูลจากเอกสารก่อนสร้างคำตอบ Retrieval-Augmented Generation ซึ่งชี้ให้เห็นว่าการทำความเข้าใจโครงสร้างเอกสารเป็นพื้นฐานสำคัญของกระบวนการแปลงเอกสาร

จากการทบทวนวรรณกรรมพบว่าปัญญาประดิษฐ์ถูกนำมาประยุกต์ใช้ในการแก้ไขข้อผิดพลาดจาก OCR โดย Rigaud และคณะ (2019) ได้จัดการแข่งขันระดับนานาชาติ ICDAR 2019 ที่ท้าทายให้นักวิจัยทั่วโลกพัฒนาระบบแก้ไขข้อผิดพลาดของข้อความที่ได้จาก OCR ซึ่งสะท้อนให้เห็นว่าปัญหาดังกล่าวได้รับความสนใจและถือเป็นโจทย์สำคัญในวงการวิจัยระดับสากล ต่อมา Rijhwani และคณะ (2021) เสนอแนวทาง semi-supervised learning สำหรับ OCR post-correction ที่ใช้ข้อมูลคำศัพท์ช่วยในการแก้ไข และ Guan และ Greene (2024) แสดงให้เห็นว่าการใช้ synthetic data ช่วยเพิ่มประสิทธิภาพของระบบ post-OCR correction ได้อย่างมีนัยสำคัญ สำหรับเอกสารที่มีเครื่องหมายกำกับเสียง Do และคณะ (2025) นำเสนอการประมวลผลหลัง OCR ด้วย LLM สำหรับข้อความที่มีเครื่องหมายกำกับเสียง เช่น สระและวรรณยุกต์ ในเอกสารประวัติศาสตร์ ซึ่งมีลักษณะใกล้เคียงกับภาษาไทย นอกจากนี้ Bourne (2025) พัฒนา CLOCR-C ซึ่งใช้ pre-trained language model สำหรับแก้ไขข้อผิดพลาดจาก OCR โดยอาศัยบริบทของข้อความโดยรอบ และการใช้แบบจำลอง Bi-LSTM with Attention สามารถลดค่า Word Error Rate จาก 11.99% เหลือ 4.03% (Mengkaw & Thiengburanathum, 2023) และการศึกษาของ Armaselu (2024) ประเมินการใช้ ChatGPT-4, Google Bard (Gemini) และ YouChat สำหรับการแก้ไขข้อความหลัง OCR ในเอกสารประวัติศาสตร์ พบว่าการออกแบบชุดคำสั่ง ที่เหมาะสมสามารถช่วยลดค่า Character Error Rate จาก 5.44% เหลือ 3.23%

งานวิจัยที่เกี่ยวข้องกับภาษาไทยยังมีจำนวนจำกัด แม้มีการพัฒนา Typhoon ซึ่งเป็น LLM สำหรับภาษาไทยโดยเฉพาะ (Pipatanakul et al., 2023) แต่ยังขาดงานวิจัยที่ครอบคลุมการแปลงเอกสาร PDF เป็น Word สำหรับภาษาไทยโดยเฉพาะ ซึ่งเป็นช่องว่างที่สำคัญ ดังนั้นงานวิจัยนี้จึงเสนอกรอบการทำงาน TPTICF ซึ่งผสานการใช้ไลบรารีร่วมกับปัญญาประดิษฐ์ใน 4 ขั้นตอน เพื่อยกระดับความถูกต้องของเอกสารภาษาไทยหลังการแปลงเอกสาร PDF โดยในการทบทวนวรรณกรรมนี้ได้สรุปงานวิจัยที่เกี่ยวข้องทั้งด้านการดึงข้อมูลโดยตรงและการใช้งานปัญญาประดิษฐ์ดังตารางที่ 1

ตารางที่ 1

การเปรียบเทียบวิธีการแปลงเอกสาร PDF เป็นเอกสาร MS Word

Reference	Library				OCR	AI			
	pymupdf	pdfplumber	pdf2docx	Other		GPT	Gemini	Deepseek	Other
Donald และคณะ (2025)	í	í	P	P	P	í	í	í	í
Lin (2024); Rijhwani และคณะ (2021)	í	í	í	í	P	í	í	í	P
Mei และคณะ (2018); Guan และ Greene (2024); Do และคณะ (2025)	í	í	í	í	P	í	í	í	í
Rigaud และคณะ (2019)	í	í	í	í	P	P	í	í	í
Armaselu (2024)	í	í	í	í	P	P	í	í	P
Adhikari และ Agarwal (2024)	P	P	í	P	P	í	í	í	P
Bourne (2025); Liu และคณะ (2022); Fang และคณะ (2023); Shen และคณะ (2021)	í	í	í	í	í	P	í	í	P
Maeng และคณะ (2023)	í	í	í	í	í	P	P	P	P
Pfitzmann และคณะ (2022); Li และคณะ (2025)	í	í	í	í	í	í	í	í	P
Pipatanakul และคณะ (2023)	í	í	í	P	P	í	í	í	P
Gemelli และคณะ (2024); OCR-D Consortium (n.d.); Ramakrishnan และคณะ (2012)	í	í	í	P	í	í	í	í	P

วัตถุประสงค์

1. เพื่อพัฒนากรอบการทำงาน TPTICF สำหรับแปลงเอกสาร PDF ภาษาไทยเป็นเอกสาร MS Word โดยผสานการใช้ไลบรารีดึงข้อมูลร่วมกับปัญญาประดิษฐ์ใน 4 ขั้นตอน

2. เพื่อเปรียบเทียบประสิทธิภาพของแบบจำลองปัญญาประดิษฐ์ Gemini-2.5-Flash, Qwen3-14B และ Typhoon-v2.1-12B-Instruct ในการแก้ไขข้อความภาษาไทยภายใต้กรอบการทำงาน TPTICF

3. เพื่อประเมินความถูกต้องของการแปลงเอกสาร PDF ภาษาไทยเป็นเอกสาร MS Word ด้วยกรอบการทำงาน TPTICF เทียบกับก่อนการแก้ไขด้วยปัญญาประดิษฐ์

วิธีดำเนินการวิจัย

ชุดข้อมูลที่ใช้ในการประเมินระบบในงานวิจัยนี้ประกอบด้วยเอกสาร PDF ภาษาไทยจำนวน 40 ฉบับ ซึ่งรวบรวมให้ครอบคลุมลักษณะของเอกสารที่หลากหลาย ได้แก่ เอกสารข้อความทั่วไป เอกสารที่มีตาราง และเอกสารหลายคอลัมน์ เพื่อใช้เป็นชุดข้อมูลสำหรับการทดลอง ทั้งนี้ การวิเคราะห์ โครงสร้างของเอกสารถือเป็นองค์ประกอบสำคัญ ในกระบวนการแปลงเอกสาร ดังที่ Shen และคณะ (2021) นำเสนอ LayoutParser ซึ่งเป็น ชุดเครื่องมือสำหรับวิเคราะห์ภาพเอกสารด้วย deep learning และ Pfitzmann และคณะ (2022) พัฒนา DocLayNet ซึ่งเป็นชุดข้อมูลขนาดใหญ่สำหรับการวิเคราะห์ โครงสร้างของเอกสาร รวมถึง Gemelli และคณะ (2024) ที่นำเสนอ dataset และ annotation สำหรับการวิเคราะห์ layout ของบทความวิชาการโดยเฉพาะ ผู้วิจัยดำเนินการแปลงเอกสาร PDF เป็นเอกสาร MS Word โดยใช้ไลบรารี pdf2docx เนื่องจากสามารถแปลงเอกสารให้อยู่ในรูปแบบที่แก้ไขได้และสามารถรักษาโครงสร้างของเอกสารเดิมได้ อย่างไรก็ตาม เครื่องมือแปลงเอกสารยังมีข้อจำกัดในการประมวลผลเอกสารภาษาไทย ส่งผลให้เกิดข้อผิดพลาดของข้อความ เช่น สระหาย วรรณยุกต์ผิดตำแหน่ง อักขระเพี้ยน คำสะกดผิด และการจัดรูปแบบข้อความที่ไม่สมบูรณ์ เพื่อเพิ่มคุณภาพของข้อความก่อนการประมวลผลด้วยปัญญาประดิษฐ์ ผู้วิจัยจึงดำเนินการเตรียมข้อความ ซึ่งประกอบด้วยการปรับรูปแบบอักขระและข้อความภาษาไทย การแก้ปัญหาสระลอย การจัดรูปแบบช่องว่าง และการปรับข้อความให้อยู่ในรูปแบบมาตรฐาน เพื่อลดสัญญาณรบกวนของข้อมูลและเพิ่มความเหมาะสมของข้อความสำหรับการประมวลผล หลังจากนั้น ข้อความจะถูกส่งเข้าสู่กระบวนการแก้ไขด้วยปัญญาประดิษฐ์ งานวิจัยนี้เปรียบเทียบประสิทธิภาพของแบบจำลองภาษาขนาดใหญ่ Large Language Models: LLMs ได้แก่ Gemini-2.5-Flash, Qwen3-14B และ Typhoon-v2.1-12B-Instruct โดยการออกแบบชุดคำสั่ง สำหรับแต่ละแบบจำลองอ้างอิงแนวทางจาก Liu และคณะ (2022) ซึ่งได้สำรวจวิธีการออกแบบชุดคำสั่ง สำหรับการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) อย่างเป็นระบบ เพื่อประเมินความสามารถในการแก้ไขข้อความภาษาไทยที่เกิดข้อผิดพลาดจากการแปลงเอกสาร ทั้งการเติมสระหรือวรรณยุกต์ที่ขาดหาย การแก้ไขคำสะกดผิด และการปรับข้อความให้สอดคล้องกับบริบทของประโยค เมื่อการแก้ไขข้อความเสร็จสมบูรณ์ ระบบจะนำข้อความที่ได้รับการปรับแก้กลับไปจัดวางในโครงสร้างเอกสารเดิม เพื่อสร้างเอกสารผลลัพธ์ในรูปแบบเอกสาร MS Word โดยพยายามรักษารูปแบบของเอกสารต้นฉบับให้มากที่สุด ทั้งด้านข้อความ ตาราง และลำดับการจัดวางภายในเอกสาร

สำหรับการประเมินประสิทธิภาพของระบบ ผู้วิจัยได้จัดทำเอกสารอ้างอิงโดยเปรียบเทียบจากต้นฉบับ โดยกำหนดพารามิเตอร์ในการวัดผลจากอัตราส่วนของจำนวนตัวอักษรที่มีการแก้ไขต่อจำนวนตัวอักษรทั้งหมดในเอกสาร (Character Revision Rate: $\boldsymbol{CRR}$) อ้างอิงสามารถเขียนเป็นสูตรได้ดังนี้

$$\boldsymbol{CRR} = \frac{\boldsymbol{C_r}}{\boldsymbol{C_t}} \tag{1}$$

เมื่อ

CRR คืออัตราส่วนของจำนวนตัวอักษรที่มีการแก้ไขต่อจำนวนตัวอักษรทั้งหมด

C_r คือจำนวนตัวอักษรที่ถูกแก้ไข

C_t คือจำนวนตัวอักษรทั้งหมดในเอกสารอ้างอิง

ผู้วิจัยเลือกใช้ค่า CRR ในการเปรียบเทียบประสิทธิภาพของระบบ เนื่องจากเป็นตัวชี้วัดในระดับตัวอักษร ซึ่งเหมาะสมกับเอกสารภาษาไทยที่มีความซับซ้อนด้านตำแหน่งสระและวรรณยุกต์

สำหรับการประเมินผลด้านเวลานั้นผู้วิจัยได้ออกแบบวิธีวัดผลโดยกำหนดช่วงเวลาเริ่มต้นของการประมวลทั้งกระบวนการเริ่มตั้งแต่การแปลงเอกสาร PDF ด้วยไลบรารีจนถึงเวลาสุดท้าย ณ เวลาที่ได้เอกสาร MS Word ฉบับสมบูรณ์ โดยเขียนสมการได้เป็น

$$\boldsymbol{T_i} = \boldsymbol{T_s} - \boldsymbol{T_e} \tag{2}$$

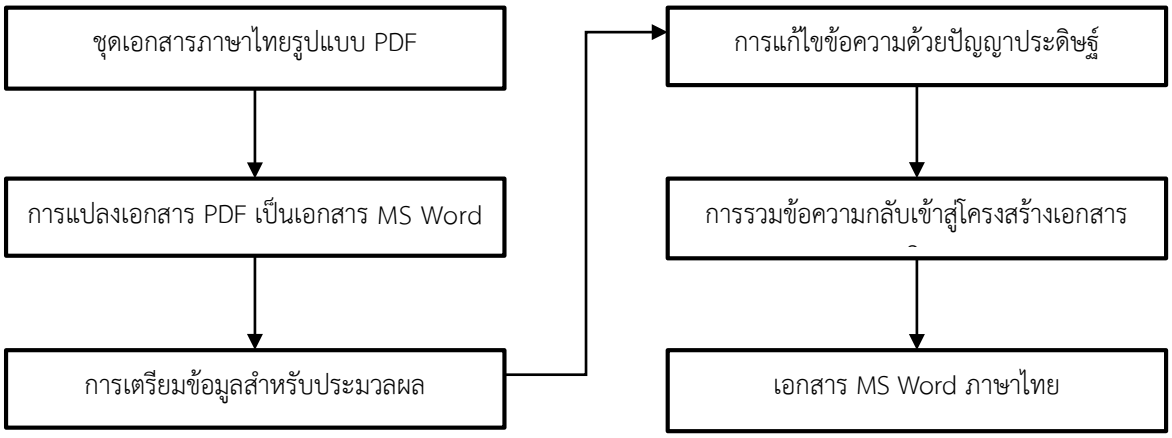
เมื่อ

T_i คือระยะเวลาในการประมวลผล

T_s คือเวลาเริ่มต้นในการใช้งานไลบรารี

T_e คือเวลาสุดท้ายที่ได้เอกสาร MS Word ฉบับสมบูรณ์

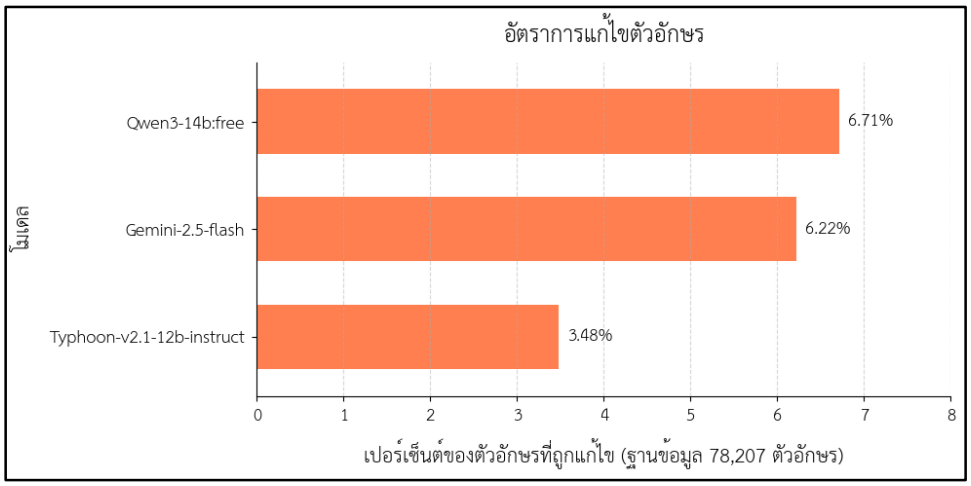
สำหรับการประเมินประสิทธิภาพของระบบ งานวิจัยนี้พิจารณาจากหลายตัวชี้วัด ได้แก่ ระยะเวลาในการประมวลผล การใช้ทรัพยากรของระบบ ความถูกต้องของการแปลงเอกสาร และความถูกต้องของการแก้ไขข้อความ การประเมินด้านเวลาในการประมวลผลคำนวณจากช่วงเวลาตั้งแต่เริ่มกระบวนการแปลงเอกสารจนได้เอกสาร MS Word ที่สมบูรณ์ ดังสมการที่ (2) ขณะที่การใช้ทรัพยากรของระบบประเมินจากค่าเฉลี่ยการใช้หน่วยความจำ และหน่วยประมวลผลกลาง ตลอดช่วงเวลาการประมวลผลของเอกสารแต่ละฉบับ เพื่อใช้เป็นตัวชี้วัดประสิทธิภาพของระบบ นอกจากนี้ ยังมีการประเมินความถูกต้องของการแปลงเอกสาร PDF เป็นเอกสาร MS Word โดยเปรียบเทียบผลลัพธ์ของ TPTICF กับเครื่องมือแปลงเอกสารอื่นจากชุดข้อมูลจำนวน 40 เอกสาร และการประเมินความถูกต้องของการแก้ไขข้อความใช้วิธีการตรวจสอบโดยมนุษย์ โดยกำหนดข้อความต้นฉบับที่ถูกต้องเป็นข้อมูลอ้างอิง จากนั้นจงใจสร้างข้อผิดพลาดที่มักเกิดจากการแปลงเอกสาร PDF เป็นเอกสาร MS Word เช่น การสะกดคำผิด การขาดหรือเกินของสระและวรรณยุกต์ การเว้นวรรคผิดตำแหน่ง และอักขระที่ผิดเพี้ยน เพื่อทดสอบความสามารถของระบบปัญญาประดิษฐ์ในการตรวจจับและแก้ไขข้อผิดพลาดดังกล่าว ภายหลังการประมวลผล ผู้วิจัยตรวจสอบผลลัพธ์ด้วยการอ่านและเปรียบเทียบข้อความที่ระบบแก้ไขกับข้อความต้นฉบับทีละคำและทีละประโยค เพื่อตรวจสอบว่าระบบสามารถแก้ไขข้อผิดพลาดที่จงใจสร้างขึ้นให้ถูกต้องได้ครบถ้วน รวมทั้งตรวจสอบว่าระบบไม่มีการแก้ไขข้อความที่ถูกต้องอยู่แล้วให้เกิดความผิดพลาดใหม่ โดยพิจารณาทั้งความถูกต้องของคำ ความต่อเนื่องของประโยค และความสอดคล้องของความหมายกับข้อความต้นฉบับ ก่อนนำผลการตรวจสอบมาคำนวณเป็นค่าความถูกต้อง ของการแก้ไขข้อความในแต่ละเอกสาร เพื่อใช้ประเมินประสิทธิภาพของกรอบการทำงาน TPTICสามารถสรุปได้ดังภาพที่ 1



ภาพที่ 1 กระบวนการทำงานของ TPTICF

ผลการวิจัย

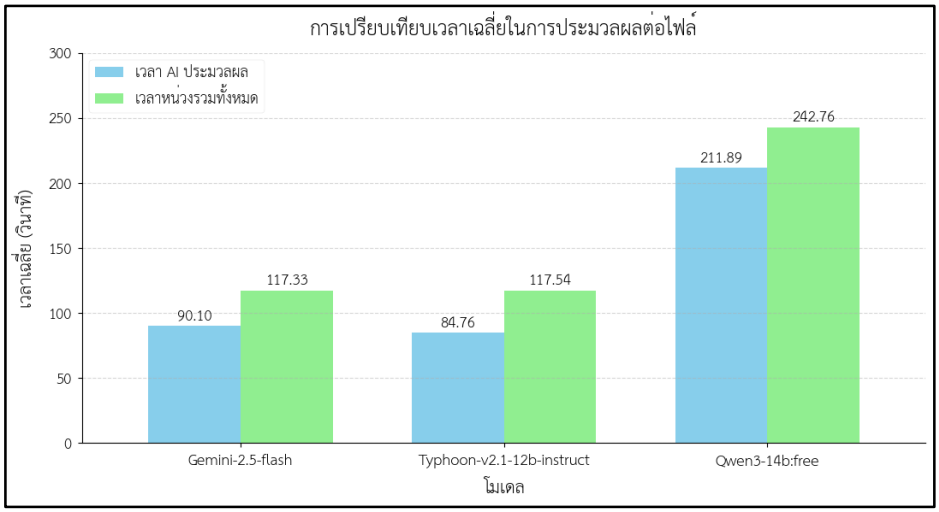
ผลการวิจัยนำเสนอตามลำดับการประเมิน ได้แก่ อัตราการแก้ไขตัวอักษร เวลาในการประมวลผล การใช้ทรัพยากรระบบ ความถูกต้องของการแยกข้อความ และความถูกต้องหลังการแก้ไขด้วยปัญญาประดิษฐ์ ดังภาพที่ 2–7



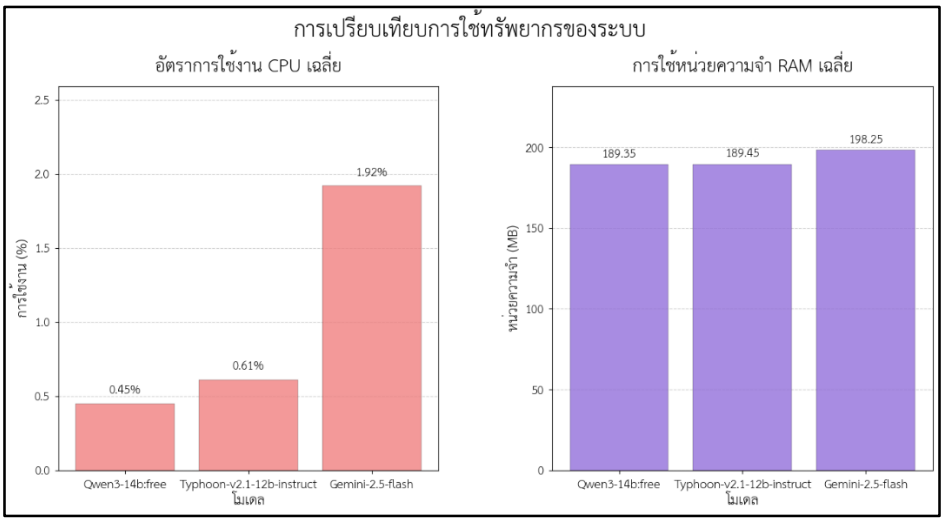
ภาพที่ 2 อัตราการแก้ไขตัวอักษรของแบบจำลองปัญญาประดิษฐ์ต่าง ๆ

จากภาพที่ 2 แสดงค่า CRR ของแบบจำลองปัญญาประดิษฐ์แต่ละตัว พบว่าแบบจำลอง Qwen3-14B:Free และ Gemini-2.5-Flash มีค่า CRR เท่ากับ 6.71% และ 6.22% ตามลำดับ อย่างไรก็ตาม Qwen3-14B:Free มีค่า CRR ที่สูงแต่ ไม่ได้หมายความว่ามีความถูกต้องสูงที่สุด จึงต้องพิจารณาร่วมกับความถูกต้องของข้อความหลังการแก้ไข เวลาในการประมวลผล การใช้ทรัพยากรของระบบ และผลการประเมินความถูกต้องของเอกสารโดยรวม เพื่อให้สามารถสะท้อนประสิทธิภาพของแบบจำลองได้อย่างครบถ้วน ส่วน Typhoon-v2.1-12B-Instruct มีค่า CRR ต่ำที่สุดในการทดลองครั้งนี้ คือ 3.48% โดยการประเมินดังกล่าวคำนวณจากเอกสารทดลองจำนวน 40 เอกสาร ซึ่งพิจารณาจากจำนวนอักขระที่มีการแก้ไขในเอกสาร MS Word หลังผ่านกระบวนการประมวลผล อย่างไรก็ตาม ค่า CRR เป็นตัวชี้วัดเชิงปริมาณที่แสดงอัตราส่วนของจำนวนอักขระที่มีการแก้ไขต่อจำนวนอักขระทั้งหมดเท่านั้น จึงไม่สามารถใช้สรุปคุณภาพหรือความถูกต้องของการแก้ไขข้อความได้โดยตรง จำเป็นต้องพิจารณาร่วมกับค่าความถูกต้องของข้อความและผลการประเมินด้านอื่น ๆ เพื่อให้สามารถวิเคราะห์ประสิทธิภาพของแบบจำลองได้อย่างครบถ้วน

สำหรับปัจจัยด้านเวลาในการประมวลผลถือเป็นตัวชี้วัดสำคัญสำหรับการนำระบบไปใช้งานจริง โดยผลการทดลองในภาพที่ 3 พบว่า Typhoon-v2.1-12B-Instruct ใช้เวลาเฉลี่ยน้อยที่สุดที่ 84.76 วินาทีต่อไฟล์ รองลงมาคือ Gemini-2.5-Flash ที่ 90.10 วินาทีต่อไฟล์ และ Qwen3-14B:Free ใช้เวลามากที่สุดที่ 211.89 วินาทีต่อไฟล์ แม้ว่า Typhoon-v2.1-12B-Instruct จะมีความโดดเด่นด้านความเร็ว แต่ Gemini-2.5-Flash สามารถรักษาสมดุลระหว่างความเร็วในการประมวลผลและประสิทธิภาพในการแก้ไขข้อความได้ดีกว่า ดังนั้น การเลือกแบบจำลองที่เหมาะสมควรพิจารณาทั้งด้านความเร็วและความถูกต้องของการแก้ไขข้อความร่วมกัน

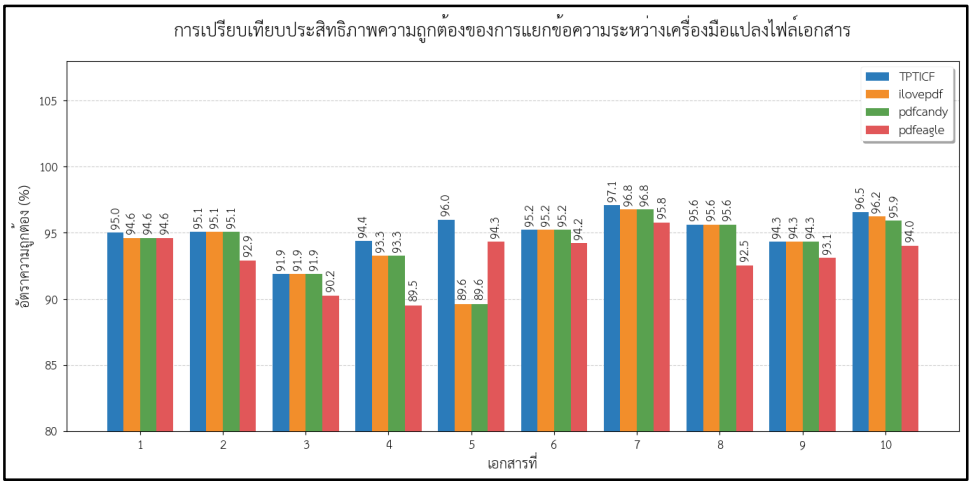


ภาพที่ 3 เวลาเฉลี่ยในการประมวลผลต่อไฟล์ของระบบสำหรับเอกสารจำนวน 40 เอกสาร

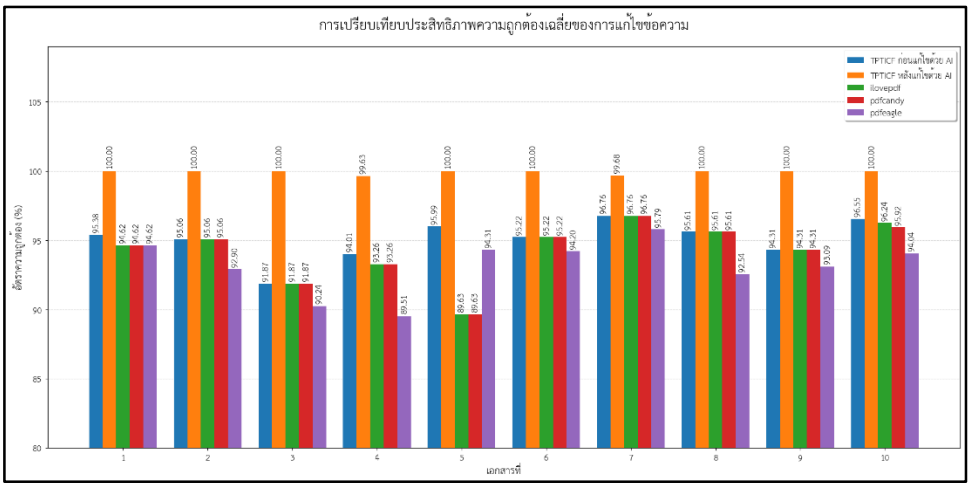


ภาพที่ 4 การใช้ทรัพยากรของระบบโดยเฉลี่ย

จากภาพที่ 4 ผลการวิเคราะห์พบว่าการใช้หน่วยความจำของแบบจำลองปัญญาประดิษฐ์ทั้งสามตัวมีค่าใกล้เคียงกัน โดยมีค่าเฉลี่ยอยู่ระหว่าง 189–198 MB แสดงให้เห็นว่าปริมาณหน่วยความจำที่ใช้ไม่ได้เป็นปัจจัยหลักที่ทำให้ประสิทธิภาพของแต่ละแบบจำลอง อย่างไรก็ตาม เมื่อพิจารณาการใช้งานหน่วยประมวลผลกลาง พบว่ามีความแตกต่างระหว่างแบบจำลองโดย Gemini-2.5-Flash มีการใช้ CPU สูงที่สุดที่ประมาณ 1.92% ขณะที่ Qwen3-14B:Free ซึ่งมีความเร็วในการประมวลผลช้าที่สุด กลับใช้ CPU ต่ำที่สุดที่ประมาณ 0.45% ผลที่เกิดขึ้นอาจสะท้อนถึงความแตกต่างของสถาปัตยกรรมภายในและวิธีการประมวลผลของแต่ละแบบจำลอง ซึ่งส่งผลต่อรูปแบบการใช้ทรัพยากรของระบบในระหว่างการทำงาน

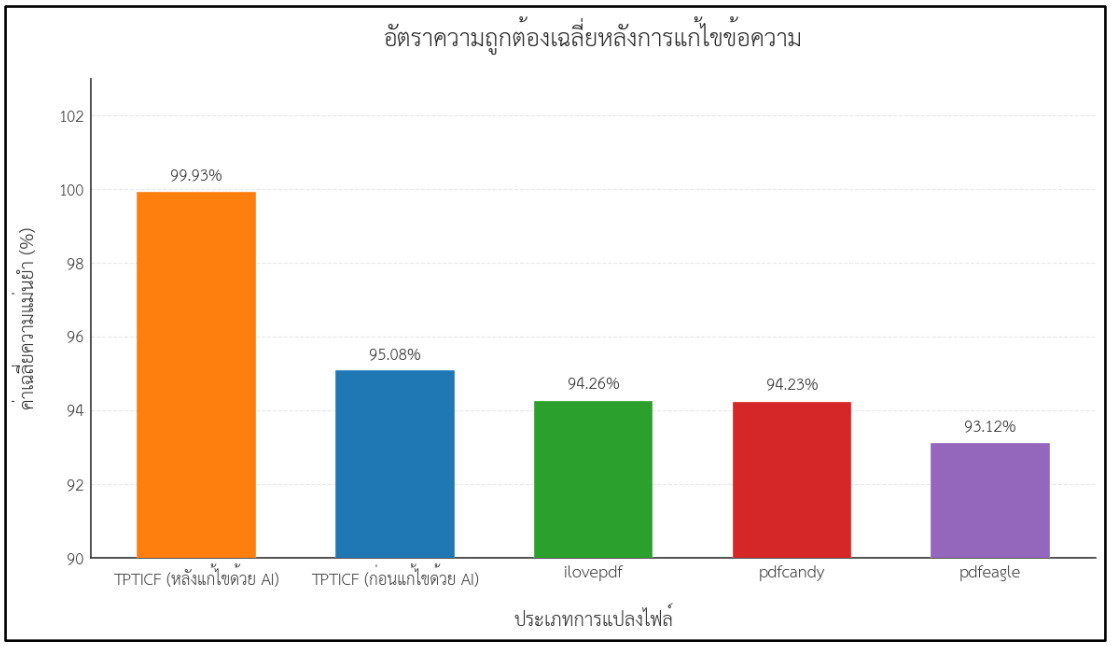


ภาพที่ 5 ความถูกต้องของการแยกข้อความระหว่างเครื่องมือแปลงเอกสารPDF เป็นเอกสาร MS Word



ภาพที่ 6 ค่าความถูกต้องของการแก้ไขข้อความจากการตรวจสอบโดยมนุษย์

ภาพที่ 5 และภาพที่ 6 แสดงการเปรียบเทียบความถูกต้องของข้อความในแต่ละขั้นตอนของกรอบการทำงาน TPTICF โดยภาพที่ 5 แสดงผลการดึงข้อความจากเอกสาร PDF ก่อนเข้าสู่ขั้นตอนการแก้ไขด้วยปัญญาประดิษฐ์ ซึ่งใช้เป็นค่าพื้นฐาน สำหรับเปรียบเทียบกับเครื่องมือแปลงเอกสารออนไลน์ ได้แก่ iLovePDF, PDFCandy และ PDF Eagle ผลการทดลองพบว่า TPTICF มีค่าความถูกต้องของการดึงข้อความมากกว่าร้อยละ 94 ในเอกสารทุกฉบับ และสามารถรักษาความถูกต้องของข้อความได้อย่างสม่ำเสมอในเอกสารหลายรูปแบบ ขณะที่เครื่องมือออนไลน์บางตัวมีค่าความถูกต้องลดลงในเอกสารที่มีรูปแบบซับซ้อน เมื่อข้อความผ่านขั้นตอนการแก้ไขด้วยปัญญาประดิษฐ์ ผลการทดลองในภาพที่ 6 แสดงให้เห็นว่าความถูกต้องของข้อความเพิ่มขึ้นอย่างชัดเจน โดยกระบวนการ TPTICF ที่ใช้ Gemini-2.5-Flash มีค่าความถูกต้องเฉลี่ย 99.93% และมีเอกสารจำนวน 7 ฉบับจากทั้งหมด 10 ฉบับที่มีค่าความถูกต้อง 100.00% แสดงให้เห็นว่าการประยุกต์ใช้ปัญญาประดิษฐ์ร่วมกับกระบวนการเตรียมข้อความสามารถลดข้อผิดพลาดที่เกิดจากการแปลงเอกสาร และเพิ่มความถูกต้องของเอกสาร MS Word เมื่อเทียบกับการใช้เครื่องมือแปลงเอกสารเพียงอย่างเดียว



ภาพที่7 อัตราความถูกต้องหลังการแก้ไขข้อความ

ภาพที่ 7 แสดงผลสรุปของกรอบการทำงาน TPTICF หลังเลือกใช้ Gemini-2.5-Flash เป็นแบบจำลองหลักในการแก้ไขข้อความ โดยพบว่าความถูกต้องเฉลี่ยของข้อความเพิ่มขึ้นจาก 95.08% ในขั้นตอนก่อนการแก้ไขด้วยปัญญาประดิษฐ์ เป็น 99.93% หลัง

ผ่านกระบวนการแก้ไขข้อความ คิดเป็นการเพิ่มความถูกต้องประมาณ 4.85 เปอร์เซ็นต์ ผลการทดลองสะท้อนให้เห็นว่า Gemini-2.5-Flash สามารถแก้ไขข้อผิดพลาดที่เกิดจากการแปลงเอกสารภาษาไทยได้อย่างมีประสิทธิภาพ พร้อมทั้งรักษาความถูกต้องของข้อความเดิมไว้ได้เป็นอย่างดี ผลการทดลองดังกล่าวแสดงให้เห็นว่า Gemini-2.5-Flash เป็นแบบจำลองที่มีความเหมาะสมที่สุดสำหรับกรอบการทำงาน TPTICF เนื่องจากสามารถแก้ไขข้อผิดพลาดของข้อความภาษาไทยที่เกิดจากการแปลงเอกสารได้อย่างมีประสิทธิภาพ พร้อมทั้งรักษาความถูกต้องของข้อความและโครงสร้างเอกสารได้อย่างสม่ำเสมอ ส่งผลให้กรอบการทำงานที่พัฒนาขึ้นมีความเหมาะสมต่อการประยุกต์ใช้ในการแปลงเอกสาร PDF ภาษาไทยเป็นเอกสาร MS Word ในการใช้งานจริง

สรุปผลการวิจัย

งานวิจัยนี้ได้พัฒนากระบวนการแปลงเอกสารภาษาไทยจากเอกสาร PDF ให้อยู่ในรูปแบบเอกสาร MS Word โดยยังคงความถูกต้องของข้อความและรูปแบบเอกสารเดิมให้ได้มาก ซึ่งประกอบด้วย การแยกข้อมูลจากเอกสาร PDF การเตรียมข้อความ การแก้ไขข้อความด้วยปัญญาประดิษฐ์ และการนำข้อความกลับเข้าสู่เอกสารเดิม จากการเปรียบเทียบแบบจำลองปัญญาประดิษฐ์ที่นำมาใช้ในกรอบการทำงาน พบว่า Qwen3-14B:Free มีอัตราการแก้ไขตัวอักษรสูงที่สุดที่ 6.71% ขณะที่ Typhoon-v2.1-12B-Instruct ใช้เวลาประมวลผลน้อยที่สุดที่ 84.76 วินาทีต่อไฟล์ และ Gemini-2.5-Flash สามารถรักษาสมดุลระหว่างความเร็วและประสิทธิภาพได้ดีที่สุด ผลการประเมินพบว่า TPTICF สามารถเพิ่มความถูกต้องของข้อความได้สูงถึง 99.93% มีเอกสารจำนวน 7 ฉบับจากทั้งหมด 10 ฉบับที่มีความถูกต้อง 100% และทุกเอกสารมีความถูกต้องไม่ต่ำกว่า 99.5% แสดงให้เห็นว่ากรอบการทำงานที่พัฒนาขึ้นสามารถช่วยเพิ่มประสิทธิภาพและความถูกต้องของการแปลงเอกสาร PDF ภาษาไทยได้มากขึ้น

การอภิปรายผล

ผลการวิจัยแสดงให้เห็นว่าการประยุกต์ใช้ปัญญาประดิษฐ์ร่วมกับกระบวนการแปลงเอกสารสามารถช่วยเพิ่มความถูกต้องของข้อความภาษาไทยหลังการแปลงเอกสาร PDF ซึ่งสอดคล้องกับงานวิจัยที่ประยุกต์ใช้ปัญญาประดิษฐ์ในการตรวจจับและแก้ไขข้อผิดพลาดของข้อความจาก OCR โดยอาศัยกระบวนการเตรียมและปรับปรุงข้อความก่อนเข้าสู่ขั้นตอนการแก้ไข เพื่อเพิ่มประสิทธิภาพของแบบจำลองในการแก้ไขคำผิด (Mengkaw & Thiengburanathum, 2023) อีกทั้งยังสอดคล้องกับผลการศึกษาที่แสดงให้เห็นว่าปัญญาประดิษฐ์สามารถช่วยลดข้อผิดพลาดของข้อความที่เกิดจาก OCR ได้ (Armaselu, 2024) งานวิจัยนี้ได้ต่อยอดแนวคิดดังกล่าวโดยพัฒนากรอบการทำงาน TPTICF สำหรับเอกสารภาษาไทย ซึ่งเพิ่มขั้นตอนการเตรียมข้อความก่อนการประมวลผลด้วยปัญญาประดิษฐ์ ได้แก่ การปรับรูปแบบอักขระและข้อความภาษาไทย การแก้ปัญหาสระลอย และการจัดรูปแบบช่องว่าง เพื่อลดความผิดปกติของข้อความที่เกิดจากการแปลงเอกสารก่อนส่งเข้าสู่แบบจำลองปัญญาประดิษฐ์ ส่งผลให้แบบจำลองสามารถแก้ไขข้อผิดพลาดที่มักเกิดจากการแปลงเอกสารภาษาไทย เช่น สระหาย วรรณยุกต์ผิดตำแหน่ง และอักขระเพี้ยน อย่างไรก็ตาม งานวิจัยนี้มีความแตกต่างจาก Armaselu (2024) ซึ่งมุ่งศึกษาศักยภาพของ ChatGPT-4, Google Bard (Gemini) และ YouChat ในการแก้ไขข้อผิดพลาดของข้อความจาก OCR สำหรับเอกสารประวัติศาสตร์ภาษาฝรั่งเศส โดยประเมินผลผ่านตัวชี้วัด CER และ WER ขณะที่งานวิจัยนี้มุ่งพัฒนากรอบการทำงานสำหรับเอกสารภาษาไทยที่มีการเตรียมข้อความก่อนการประมวลผลด้วยปัญญาประดิษฐ์ และประเมินผลด้วยการตรวจสอบโดยมนุษย์ ซึ่งพบว่าระบบมีค่าความถูกต้องเฉลี่ยของข้อความหลังการแก้ไขสูงถึง 99.93% ความแตกต่างของผลการทดลองจึงอาจเกิดจากลักษณะของชุดข้อมูล วัตถุประสงค์ของการศึกษา วิธีการประเมินผล และแนวทางการประมวลผลที่แตกต่างกันของทั้งสองงานวิจัย นอกจากนี้ เมื่อเปรียบเทียบกับงานวิจัยที่ใช้ LLM สำหรับการแก้ไขข้อผิดพลาดทางไวยากรณ์ พบว่า Fang และคณะ (2023) แสดงให้เห็นว่า ChatGPT มีความคล่องแคล่วในการแก้ไขข้อผิดพลาดทางไวยากรณ์ภาษาอังกฤษ และ Maeng และคณะ (2023) พบว่า ChatGPT มีประสิทธิภาพในการแก้ไขข้อผิดพลาดทางไวยากรณ์ภาษาเกาหลีซึ่งเป็นภาษาที่มีระบบการเขียนที่ซับซ้อนใกล้เคียงกับภาษาไทย ผลเหล่านี้สนับสนุนว่า LLM มีศักยภาพในการแก้ไขข้อผิดพลาดของภาษาที่ไม่ใช่ Latin script ได้อย่างมีประสิทธิภาพ ซึ่งสอดคล้องกับผลที่ได้จากงานวิจัยนี้ ในด้านแนวทางการพัฒนาระบบในอนาคต Li และคณะ (2025) นำเสนอ DREAM ซึ่งเป็นแบบจำลอง end-to-end autoregressive สำหรับการสร้างเอกสารใหม่โดยตรง แนวทางดังกล่าวอาจเป็นทิศทางที่น่าสนใจสำหรับการพัฒนากรอบการทำงานในอนาคต เพื่อลดขั้นตอนการประมวลผลและเพิ่มความแม่นยำของการแปลงเอกสารภาษาไทยให้มากยิ่งขึ้น

ข้อเสนอแนะ

งานวิจัยนี้ยังมีข้อจำกัดด้านประเภทของเอกสารที่ใช้ในการทดลอง ซึ่งส่วนใหญ่เป็นเอกสาร PDF ภาษาไทยประเภทข้อความทั่วไป ตาราง และหลายคอลัมน์เท่านั้น ยังไม่ครอบคลุมเอกสารประเภทอื่น เช่น เอกสารสแกน หรือเอกสารที่มีรูปภาพประกอบจำนวนมาก ดังนั้นในการศึกษาครั้งต่อไปควรขยายชุดข้อมูลให้ครอบคลุมเอกสารที่หลากหลายประเภทมากขึ้น นอกจากนี้ การพึ่งพาการทำงานผ่าน API ภายนอกทำให้มีข้อจำกัดด้านระยะเวลาในการประมวลผลและจำนวนครั้งในการเรียกใช้งาน การศึกษาในอนาคตอาจพิจารณา

การนำแบบจำลองมาติดตั้งและประมวลผลภายในระบบ (On-premise) เพื่อลดการพึ่งพา API ภายนอกและเพิ่มความยืดหยุ่นในการใช้งานจริง

เอกสารอ้างอิง

Adhikari, N. S., & Agarwal, S. (2024). *A comparative study of PDF parsing tools across diverse document categories*. arXiv preprint. https://arxiv.org/pdf/2410.09871

Armaselu, F. (2024). Small-scale testing on generative AI and post-OCR correction in historical datasets. In *Digital Humanities Benelux Conference 2024*. https://orbilu.uni.lu/bitstream/10993/61414/1/DHB2024_GenAI_Post-OCR_Abstract.pdf

Bourne, J. (2025). *CLOCR-C: Context leveraging OCR correction with pre-trained language models*. arXiv preprint. https://arxiv.org/pdf/2408.17428

Do, T., Tran, D. P., Vo, A., & Kim, D. (2025). Reference-based post-OCR processing with LLM for precise diacritic text in historical document recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v39i27.35012

Donald, A., Panghantiwar, S., Gande, N., Dadgelwar, M., Teware, S., & Gargelwar, L. (2025). Rapid document conversion. *International Journal of Advanced Research in Science, Communication and Technology*, *5*(5). https://www.ijarsct.co.in/Paper27749.pdf

Fang, T., Yang, S., Lan, K., Wong, D. F., Hu, J., Chao, L. S., & Zhang, Y. (2023). *Is ChatGPT a highly fluent grammatical error correction system? A comprehensive evaluation*. arXiv preprint. https://arxiv.org/pdf/2304.01746

Gemelli, A., Marinai, S., Pisaneschi, L., & Santoni, F. (2024). Datasets and annotations for layout analysis of scientific articles. *International Journal on Document Analysis and Recognition*, *27*, 683-705. https://doi.org/10.1007/s10032-024-00461-2

Guan, S., & Greene, D. (2024). *Advancing post-OCR correction: A comparative study of synthetic data*. arXiv preprint. https://arxiv.org/pdf/2408.02253v1

Li, X., Gong, M., Wu, Y., Dai, J., Guo, A., Jiang, X., Cao, H., Liu, Y., Jiang, D., & Sun, X. (2025). *DREAM: Document reconstruction via end-to-end autoregressive model*. arXiv preprint. https://arxiv.org/abs/2507.05805

Lin, D. (2024). *Revolutionizing retrieval-augmented generation with enhanced PDF structure recognition*. arXiv preprint. https://arxiv.org/pdf/2401.12599

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2022). *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*. ACM Computing Surveys. https://doi.org/10.1145/3560815

Mei, J., Islam, A., Moh'd, A., Wu, Y., & Milios, E. (2018). MiBio: A dataset for OCR post-processing evaluation. *Data in Brief*, *21*, 251–255. https://doi.org/10.1016/j.dib.2018.09.051

Mengkaw, J., & Thiengburanathum, P. (2023). Thai spelling correction investigation framework based on OCR word extraction. *Data Science and Engineering Record*, *5*(1). https://datascience.cmu.ac.th/storage/articles/53.pdf

OCR-D Consortium. (n.d.). OCR-D evaluation specification. https://ocr-d.de/en/spec/ocrd_eval.html

Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A. S., & Staar, P. (2022). DocLayNet: A large human-annotated dataset for document-layout analysis. In *Proceedings of ACM*. https://doi.org/10.1145/3534678.3539043

Pipatanakul, K., Jirabovonvisut, P., Manakul, P., Sripaisarnmongkol, S., Patomwong, R., Chokchainant, P., & Tharnpipitchai, K. (2023). *Typhoon: Thai large language models*. arXiv preprint. https://arxiv.org/pdf/2312.13951

Ramakrishnan, C., Patnia, A., Hovy, E., & Burns, G. A. P. (2012). Layout-aware text extraction from full-text PDF of scientific articles. *Journal of Biomedical Informatics*, *45*(6), 1159–1170. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3441580/

Rigaud, C., Doucet, A., Coustaty, M., & Moreux, J.-P. (2019). ICDAR 2019 competition on post-OCR text correction. In *Proceedings of the 15th International Conference on Document Analysis and Recognition* (pp. 1588–1593). https://hal.science/hal-02304334

Rijhwani, S., Rosenblum, D., Anastasopoulos, A., & Neubig, G. (2021). *Lexically-aware semi-supervised learning for OCR post-correction*. arXiv preprint. https://arxiv.org/pdf/2111.02622

Shen, Z., Zhang, R., Dell, M., Lee, B. C. G., Carlson, J., & Li, W. (2021). *LayoutParser: A unified toolkit for deep learning based document image analysis*. arXiv preprint. https://arxiv.org/pdf/2103.15348