การเพิ่มประสิทธิภาพและประเมินผล Retrieval-Augmented Generation (RAG) สำหรับเอกสารภาษาไทย

อรรถสิทธิ์ พิมพ์อินทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
บัณฑิตศึกษา สาขาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2568

OPTIMIZATION AND EVALUATION OF RETRIEVAL-AUGMENTED GENERATION (RAG) FOR THAI DOCUMENTS

Attasit Pimin

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate Studies
Thai-Nichi Institute of Technology
Academic Year 2025

หัวข้อวิทยานิพนธ์	การเพิ่มประสิทธิภาพและประเมินผล Retrieval-Augmented Generation (RAG) สำหรับเอกสารภาษาไทย
โดย	อรรถสิทธิ์ พิมพ์อินทร์
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ว่าที่ร้อยตรี ดร.พิชิตชัย คำอินทร์

บัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....รองอธิการบดีฝ่ายวิชาการ

(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล)

..... กรรมการ

(ดร.ประมุข บุญเสี่ยง)

..... กรรมการ

(ดร.ณปภัช วิชัยดิษฐ์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ว่าที่ร้อยตรี ดร. พิชิตชัย คำอินทร์)

อรรถสิทธิ์ พิมพ์อินทร์: การเพิ่มประสิทธิภาพและประเมินผล Retrieval-Augmented Generation (RAG) สำหรับเอกสารภาษาไทย. อาจารย์ที่ปรึกษา : ว่าที่ร้อยตรี ดร.พิชิตชัย คำอินทร์, 52 หน้า.

ปัจจุบันการจัดการองค์ความรู้ภายในองค์กรมีความสำคัญมาก โดยเฉพาะในสถาบันการศึกษาที่มีเอกสารระเบียบและประกาศจำนวนมากในรูปแบบ PDF ซึ่งการค้นหาข้อมูลยังคงใช้เวลานาน แม้โมเดลภาษาขนาดใหญ่ (LLMs) จะสามารถช่วยตอบคำถามได้ แต่ยังมีข้อจำกัด เช่น การให้ข้อมูลผิด (Hallucination) และไม่สามารถเข้าถึงข้อมูลภายในองค์กรได้โดยตรง เทคโนโลยี Retrieval-Augmented Generation (RAG) จึงถูกนำมาใช้เพื่อช่วยให้ระบบสามารถค้นหาและตอบคำถามจากข้อมูลจริงได้ อย่างไรก็ตาม ประสิทธิภาพของระบบไม่ได้ขึ้นอยู่กับโมเดลเพียงอย่างเดียว แต่ยังขึ้นอยู่กับคุณภาพของข้อมูลและวิธีเตรียมข้อมูลด้วย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาผลกระทบของคุณภาพข้อมูล 3 รูปแบบ ได้แก่ Text, PDF Text และ OCR รวมกับวิธีแบ่งข้อมูล (Chunking) ขนาด 250, 500 และ 1,000 โทเคน โดยใช้เอกสารจริงจำนวน 197 ฉบับเป็นกรณีศึกษา และใช้โมเดลช่วยจัดอันดับข้อมูล (Reranker) เพื่อเพิ่มความแม่นยำของการค้นหา การศึกษานี้มุ่งเน้นการประเมินผลแบบครบทั้งกระบวนการ เพื่อให้เข้าใจว่าปัจจัยใดมีผลต่อประสิทธิภาพของระบบมากที่สุด โดยเฉพาะเมื่อใช้งานกับเอกสารภาษาไทยที่มีลักษณะเฉพาะ

ผลการทดลองพบว่า คุณภาพของข้อมูลมีผลต่อความถูกต้องของคำตอบอย่างชัดเจน โดยข้อมูล Text ที่ผ่านการจัดเตรียมอย่างเหมาะสม และใช้การแบ่งข้อมูลขนาดใหญ่ (1,000 โทเคน) ให้ผลลัพธ์ดีที่สุด ในขณะที่ข้อมูลจาก OCR แม้จะช่วยให้ค้นหาข้อมูลได้มาก แต่มีสัญญาณรบกวนสูง ทำให้คำตอบคลาดเคลื่อน งานวิจัยนี้จึงสรุปว่า การเตรียมข้อมูลให้มีคุณภาพเป็นปัจจัยสำคัญที่สุดในการพัฒนาระบบ RAG และหากจำเป็นต้องใช้ข้อมูลจาก OCR ควรมีการคัดกรองข้อมูลก่อนนำไปใช้งาน เพื่อให้ระบบสามารถให้คำตอบได้อย่างถูกต้องและน่าเชื่อถือมากขึ้น

บัณฑิตศึกษา
สาขาวิชา เทคโนโลยีสารสนเทศ
ปีการศึกษา 2568

ลายมือชื่อนักศึกษา.....
ลายมือชื่ออาจารย์ที่ปรึกษา.....

ATTASIT PIMIN: OPTIMIZATION AND EVALUATION OF RETRIEVAL-AUGMENTED GENERATION (RAG) FOR THAI DOCUMENTS. ADVISOR: ACTING SUB LT. DR. PICHITCHAI KAMIN, 52 PP.

Knowledge management within organizations has become increasingly important, especially in educational institutions where a large number of regulations and announcements are stored in PDF format. Retrieving information from these documents is often time-consuming. Although Large Language Models (LLMs) can assist in answering questions, they still face limitations such as generating incorrect information (hallucination) and lacking access to internal organizational data. Retrieval-Augmented Generation (RAG) has been introduced to address these issues by enabling systems to retrieve and generate answers based on actual data. However, the effectiveness of RAG systems depends not only on the language models but also on the quality of the data and the data preparation process.

This study aims to investigate the impact of three types of data sources Text, PDF Text, and OCR combined with different chunk sizes (250, 500, and 1,000 tokens). A total of 197 real-world documents were used as the dataset, and a reranking model was applied to improve retrieval accuracy. The study evaluates the RAG system in an end-to-end manner to better understand which factors most significantly affect performance, particularly in the context of Thai-language documents, which have unique linguistic characteristics.

The results show that data quality has a significant impact on answer accuracy. Clean text data combined with a larger chunk size (1,000 tokens) produced the best performance. In contrast, OCR data, while improving retrieval coverage, introduced noise that reduced answer accuracy. This study concludes that high-quality data preparation is the most critical factor in developing effective RAG systems. When OCR data is unavoidable, additional data filtering steps should be applied to ensure more accurate and reliable responses.

Graduate Studies Student's Signature.....

Field of Study Information Technology Advisor's Signature.....

Academic Year 2025

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณผู้มีพระคุณทุกท่านที่ได้ให้การสนับสนุนและช่วยเหลือเป็นอย่างดียิ่งตลอดระยะเวลาการดำเนินงาน โดยเฉพาะอย่างยิ่ง ว่าที่ร้อยตรี ดร.พิชิตชัย คำอินทร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้กรุณาสละเวลาอันมีค่าในการให้คำแนะนำ ตรวจสอบ แก้ไขข้อบกพร่อง และให้การสนับสนุนผู้วิจัยในทุกขั้นตอน จนวิทยานิพนธ์ฉบับนี้สำเร็จสมบูรณ์

ผู้วิจัยขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ได้แก่ รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล ดร.ประมุข บุญเสี่ยง และ ดร.ณปภัช วิชัยดิษฐ์ ที่กรุณาให้ข้อเสนอแนะอันมีคุณค่า และช่วยตรวจสอบแก้ไขข้อบกพร่องของวิทยานิพนธ์ฉบับนี้จนมีความสมบูรณ์ยิ่งขึ้น นอกจากนี้ขอขอบพระคุณคณาจารย์และเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศ และบัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่ให้ความอนุเคราะห์การอำนวยความสะดวก และให้คำปรึกษาอันเป็นประโยชน์ต่อการดำเนินงานวิจัยครั้งนี้

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณครอบครัว เพื่อนร่วมงาน และผู้เกี่ยวข้องทุกท่าน ที่ให้การสนับสนุนทั้งกำลังใจและความช่วยเหลือในทุกด้าน จนทำให้ผู้วิจัยสามารถดำเนินงานวิจัยฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

อรรถสิทธิ์ พิมพ์อินทร์

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันองค์กรต่าง ๆ ให้ความสำคัญของการจัดการความรู้ในองค์กร ในยุคที่ข้อมูลมีเพิ่มมากขึ้นเรื่อยๆ การบริหารจัดการองค์ความรู้ภายในองค์กร กลายเป็นปัจจัยสำคัญหลักในการสร้างความได้เปรียบทางการแข่งขันและการเพิ่มประสิทธิภาพในการดำเนินงานกับองค์กรทางการศึกษา รวมถึงสถาบันเทคโนโลยีไทย-ญี่ปุ่น ก็เป็นหนึ่งในหน่วยงานที่มีการผลิตและจัดเก็บข้อมูลจำนวนมากในแต่ละปี โดยเฉพาะข้อมูลเชิงโครงสร้างและระเบียบปฏิบัติ เช่น คำสั่งสถาบัน ประกาศ ข้อบังคับการเรียนการสอน และแนวปฏิบัติสำหรับบุคลากร ซึ่งส่วนใหญ่ถูกจัดเก็บและเผยแพร่ในรูปแบบเอกสารอิเล็กทรอนิกส์ PDF ผ่านระบบเครือข่ายภายในและเว็บไซต์ของสถาบันอย่างเป็นระบบ [1] เอกสารเหล่านี้เปรียบเสมือนคลังสมองขององค์กรที่บรรจุรายละเอียดสำคัญทางกฎหมายและระเบียบปฏิบัติที่บุคลากรและนักศึกษาจำเป็นต้องอ้างอิงอยู่เสมอ เมื่อปริมาณเอกสารสะสมเพิ่มมากขึ้นตามระยะเวลาที่ผ่านไป การสืบค้นและนำข้อมูลนั้นกลับมาใช้ใหม่จึงกลายเป็นความท้าทายที่ซับซ้อนและมีนัยสำคัญต่อความคล่องตัวของสถาบันฯ

การเข้าถึงข้อมูลแม้จะมีระบบการจัดเก็บที่เป็นระเบียบ แต่กระบวนการเข้าถึงข้อมูลเชิงลึกภายในเอกสารยังคงประสบปัญหาภาพรวมและการสรุปผลของเอกสารต่างๆ ของสถาบันฯ ที่ต้องอาศัยทักษะการค้นหาด้วยตนเองผ่านการค้นหาคำสำคัญแบบดั้งเดิม ผ่านการค้นหาเอกสารและสกัดข้อมูลที่ต้องการด้วยตนเอง บ่อยครั้งไม่สามารถเข้าถึงใจความสำคัญที่ซ่อนอยู่ในเอกสารที่มีความยาวหลายหน้าได้ ผู้ใช้ต้องใช้เวลาและทรัพยากรจำนวนมากในการอ่าน ทบทวน และสกัดข้อมูลเพื่อหาคำตอบในประเด็นที่เฉพาะเจาะจง ซึ่งกระบวนการนี้ไม่เพียงแต่ล่าช้า เป็นกระบวนการที่ใช้เวลานานและมีโอกาสเกิดความผิดพลาดสูง แม้ปัจจุบันจะมีการนำโมเดลภาษาขนาดใหญ่ เข้ามาช่วยในการประมวลผลภาษาธรรมชาติ โดยทั่วไปที่ได้รับการฝึกฝนจากข้อมูลสาธารณะยังคงมีข้อจำกัด 2 ประการ คือ การสร้างข้อมูลที่ผิดพลาด และการขาดความรู้ในบริบทเฉพาะ ของสถาบันฯ เนื่องจากโมเดลไม่สามารถเข้าถึงฐานข้อมูลเอกสารภายในที่มีการปรับปรุงใหม่ได้โดยตรง ส่งผลให้การนำมาใช้ในการตอบคำถามในบริบทองค์กรมีความขาดความน่าเชื่อถือของข้อมูลและความแม่นยำ

แนวทาง Retrieval Augmented Generation (RAG) [2] ได้รับความสนใจอย่างกว้างขวาง เนื่องจากสามารถผสานจุดแข็งของการสืบค้นและการสร้างภาษาเข้าด้วยกัน เพื่อแก้ปัญหาการสร้างข้อมูลเท็จ (Hallucination) และการที่ Large Language Models (LLMs) ไม่สามารถเข้าถึงข้อมูลภายในองค์กรได้โดยตรง การนำระบบ RAG ผสานเข้ากับจุดแข็งของการสืบค้น (Retrieval) และการ

สร้างภาษา (Generation) ให้ได้ประสิทธิภาพของระบบ RAG การจะสามารถนำมาใช้ในสถานการณ์จริง ไม่ได้ขึ้นอยู่กับคุณภาพของ LLMs เพียงอย่างเดียว แต่ยังขึ้นอยู่กับคุณภาพของบริบทข้อมูลที่ดึงมาสนับสนุนการตอบคำถามด้วย คุณภาพของบริบทข้อมูล โดยมีขั้นตอนการเตรียมข้อมูล 1. คุณภาพของแหล่งข้อมูล โดยเฉพาะอย่างยิ่งเมื่อเอกสารต้นทางเป็นไฟล์ PDF [3] ที่อาจเป็นได้ทั้งแบบข้อความและแบบภาพที่ต้องผ่านกระบวนการ OCR (Optical Character Recognition) [4] ซึ่งมักก่อให้เกิดสิ่งรบกวน (Noise) ตัวอักษรผิดเพี้ยน 2. การแบ่งส่วนข้อมูล (Chunking) เป็นการแบ่งเอกสารขนาดใหญ่ให้เป็นหน่วยย่อย (Chunks) ที่เหมาะสม ซึ่งขนาดของ Chunk เช่น 250, 500 และ 1,000 โทเคน (Tokens) มีผลโดยตรงต่อคุณลักษณะของบริบทข้อมูลที่ถูกดึงออกมา ร่วมกับการเลือกใช้โมเดลภาษาขนาดใหญ่ตระกูล Typhoon 2.x ซึ่งโดดเด่นในการประมวลผลภาษาไทย [5]

งานวิจัยนี้มุ่งศึกษาผลกระทบเชิงปริมาณของ (1) คุณภาพแหล่งข้อมูลต้นทางที่ต่างรูปแบบกัน และ (2) กลยุทธ์การแบ่งส่วนข้อมูล (Chunking) ต่อประสิทธิภาพของระบบ Retrieval-Augmented Generation (RAG) สำหรับเอกสารภาษาไทย โดยประเมินทั้งการค้นคืนข้อมูล MRR, nDCG@K และคำตอบ ความถูกต้อง/ความสอดคล้องกับแหล่งอ้างอิง

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาผลกระทบของคุณภาพแหล่งข้อมูลเอกสารที่แตกต่างกัน ได้แก่ ข้อมูลจากไฟล์ PDF ข้อมูลที่ได้จากกระบวนการ OCR และข้อมูลในรูปแบบไฟล์ข้อความ (Text) ต่อประสิทธิภาพของระบบ Retrieval-Augmented Generation (RAG)

1.2.2 เพื่อเปรียบเทียบกลยุทธ์การแบ่งส่วนข้อมูล (Chunking Strategy) ที่มีขนาดแตกต่างกัน ว่าส่งผลต่อประสิทธิภาพของการค้นคืนข้อมูลและการจัดเก็บข้อมูลในฐานข้อมูลเวกเตอร์อย่างไร

1.2.3 เพื่อประเมินประสิทธิภาพและคุณภาพการจัดอันดับการค้นคืนข้อมูล (Retrieval Quality) ภายใต้ข้อจำกัดของข้อมูลที่มีสัญญาณรบกวน (Noise)

1.2.4 เพื่อประเมินคุณภาพความถูกต้องของคำตอบ ที่ได้โดยโมเดลภาษาขนาดเล็ก เมื่อได้รับข้อมูลบริบทจาก RAG

1.2.5 เพื่อวิเคราะห์ความสัมพันธ์ระหว่างคุณภาพของการค้นคืนข้อมูลและความถูกต้องของคำตอบ ในบริบทของเอกสารภาษาไทย

1.3 คำถามการวิจัย

1.3.1 คุณภาพของแหล่งข้อมูลเอกสารที่แตกต่างกัน (Text, PDF Text และ OCR) ส่งผลกระทบต่อประสิทธิภาพการทำงานของระบบ Retrieval-Augmented Generation อย่างไร

1.3.2 กลยุทธ์ขนาดของการแบ่งส่วนข้อมูล (Chunk Size) ที่แตกต่างกัน มีความเหมาะสมกับรูปแบบของแหล่งข้อมูลแต่ละประเภทอย่างไร เพื่อให้ได้คุณภาพการค้นคืนสูงสุด

1.3.3 ปริมาณและความหนาแน่นของข้อมูลที่ค้นคืนมาได้ สัมพันธ์กับความถูกต้องของคำตอบที่สร้างโดยโมเดลภาษา หรือไม่ เมื่อต้องมีข้อมูลสัญญาณรบกวน (Noise)

1.3.4 ระบบ Retrieval-Augmented Generation ที่ใช้โมเดลภาษา (Small Language Model: SLM) ร่วมกับการจัดอันดับใหม่ (Reranking) สามารถสร้างคำตอบที่มีความถูกต้องตามเกณฑ์ตัวชี้วัดที่กำหนดได้หรือไม่

1.4 ขอบเขตของการวิจัย

1.4.1 ชุดข้อมูลที่ใช้ในการวิจัยใช้เอกสารประกาศ คำสั่ง และระเบียบปฏิบัติภายในสถาบันเทคโนโลยีไทย-ญี่ปุ่น จำนวน 197 ฉบับ โดยเตรียมข้อมูลออกเป็น 3 รูปแบบจำลอง ได้แก่ ข้อมูลไฟล์ข้อความ (Text) ที่ผ่านการทำความสะอาด, ข้อมูลที่สกัดจากไฟล์ PDF โดยตรง (PDF Text) และข้อมูลที่ได้จากกระบวนการแปลงภาพเป็นข้อความ (OCR Text)

1.4.2 ชุดคำถามและเกณฑ์การประเมิน ทดลองโดยใช้ชุดคำถามอ้างอิงเชิงข้อเท็จจริง จำนวน 50 คำถาม

1.4.3 การแบ่งส่วนข้อมูล กำหนดขนาดการแบ่งชิ้นส่วนข้อมูลเป็น 3 ระดับ ได้แก่ 250, 500 และ 1,000 tokens โดยกำหนดค่า Overlap คงที่ที่ 20% ของขนาด Chunk

1.4.4 สถาปัตยกรรมโมเดล (Models)

1. การสร้างเวกเตอร์ข้อความ (Embedding) ใช้โมเดล BAAI/bge-m3
2. การปรับปรุงคุณภาพการจัดอันดับ (Reranking) ใช้โมเดล BAAI/bge-reranker-v2-m3
3. การสร้างคำตอบ (Generation) ใช้โมเดลภาษาตระกูล Typhoon 2.x (ได้แก่ Typhoon2.1-Gemma3-4B, Llama3.2-Typhoon2-3B และ Typhoon2.5-Qwen3-4B)

1.4.5 การทำงานของระบบ ระบบถูกออกแบบให้ทำงานแบบประมวลผลภายในเครื่อง (Local Environment) ผ่านเครื่องมือ Ollama เพื่อรักษาความปลอดภัยและความเป็นส่วนตัวของข้อมูลองค์กร

1.4.6 ตัวชี้วัดประสิทธิภาพ (Evaluation Metrics)

1. ด้านการค้นคืนข้อมูล (Retrieval) ประเมินด้วย Hit Rate, Mean Reciprocal Rank (MRR), และ Normalized Discounted Cumulative Gain (NDCG@K)

2. ด้านการสร้างคำตอบ (Generation) ประเมินความถูกต้องด้วย Cosine Similarity ภายใต้เกณฑ์ยืดหยุ่น (Lax) และเกณฑ์เข้มงวด (Strict)

1.4.7 การประยุกต์ใช้งาน เป็นการพัฒนาระบบต้นแบบ (Prototype) เพื่อรองรับการใช้งานของกลุ่มเป้าหมายหลัก ได้แก่ บุคลากรสายสนับสนุน คณาจารย์ ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้องค์ความรู้และข้อสรุปเชิงปริมาณที่ชัดเจนเกี่ยวกับผลกระทบของคุณภาพแหล่งข้อมูลและกลยุทธ์การแบ่งส่วนข้อมูล ที่มีความสำคัญต่อประสิทธิภาพของระบบ RAG สำหรับเอกสารภาษาไทย

1.5.2 ช่วยลดภาระงานของบุคลากรในองค์กร ในการตอบคำถามพื้นฐานหรือข้อซักถามเดิม ๆ ที่เกี่ยวข้องกับระเบียบปฏิบัติ ช่วยลดเวลาและภาระงานของเจ้าหน้าที่

1.5.3 เพิ่มประสิทธิภาพในการเข้าถึงข้อมูล ช่วยให้บุคลากรของสถาบันฯ สามารถสืบค้นและเข้าถึงข้อมูลระเบียบการ ประกาศ และข้อบังคับต่างๆ ได้อย่างรวดเร็ว และพร้อมใช้งานตลอดเวลา

1.6 ข้อจำกัดของการวิจัย

1.6.1 ข้อจำกัดด้านรูปแบบข้อมูล เอกสารที่มีตารางซับซ้อน เช่น ตารางงบประมาณ เมื่อผ่านกระบวนการสกัดข้อความ อาจสูญเสียโครงสร้างเดิม ส่งผลให้ความแม่นยำในการดึงข้อมูลลดลง ซึ่งส่งผลโดยตรงต่อประสิทธิภาพของขั้นตอน Embedding และ Retrieval

1.6.2 สัญญาณรบกวนจากกระบวนการ OCR เช่น การอ่านอักขระผิด หรือวรรณยุกต์ผิดตำแหน่ง อาจทำให้ความหมายของข้อความคลาดเคลื่อน ส่งผลให้เวกเตอร์ที่สร้างขึ้นไม่สอดคล้องกับข้อมูลจริง

1.6.3 ข้อจำกัดของโมเดลภาษา (LLM) โมเดลภาษาอาจสร้างคำตอบที่ไม่ตรงกับข้อมูลจริง (Hallucination) หากข้อมูลที่ค้นคืนมาไม่ครอบคลุม

1.7 นิยามศัพท์เฉพาะ

1.7.1 Retrieval-Augmented Generation (RAG) หมายถึง แนวทางการทำงานของระบบปัญญาประดิษฐ์ที่ผสมผสานกระบวนการค้นคืนข้อมูล (Retrieval) เข้ากับการสร้างข้อความของโมเดลภาษา (Generation) โดยโมเดลภาษาจะใช้ข้อมูลที่ค้นคืนมาเป็นบริบทในการสร้างคำตอบ

1.7.2 Chunking หมายถึง กระบวนการแบ่งข้อความขนาดใหญ่ให้เป็นส่วนย่อย เพื่อให้สามารถนำไปสร้างเวกเตอร์และจัดเก็บในฐานข้อมูลเวกเตอร์ได้อย่างมีประสิทธิภาพ

1.7.3 Embedding หมายถึง กระบวนการแปลงข้อความให้เป็นเวกเตอร์เชิงตัวเลข ซึ่งสามารถใช้ในการวัดความคล้ายคลึงของข้อความ

1.7.4 Vector Database หมายถึง ฐานข้อมูลที่ใช้จัดเก็บเวกเตอร์ของข้อความและสามารถค้นหาเวกเตอร์ที่มีความใกล้เคียงกันได้อย่างรวดเร็ว

1.7.5 Reranking หมายถึง กระบวนการจัดอันดับเอกสารใหม่หลังจากขั้นตอน Retrieval เพื่อเพิ่มความแม่นยำของเอกสารที่เกี่ยวข้อง

1.7.6 Large Language Model (LLM) หมายถึง โมเดลภาษาขนาดใหญ่ที่ถูกฝึกด้วยข้อมูลจำนวนมากและสามารถสร้างข้อความหรือคำตอบจากข้อมูลบริบทได้

1.7.7 Token หมายถึง หน่วยย่อยของข้อความที่โมเดลภาษาใช้ในการประมวลผล ซึ่งอาจเป็นคำ ตัวอักษร หรือส่วนของคำ ขึ้นอยู่กับวิธีการตัดคำ (Tokenizer) ของโมเดล

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการศึกษาบนพื้นฐานของแนวคิดและทฤษฎีที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางในการทำวิจัย ดังนี้

2.1 Artificial Intelligence

ปัญญาประดิษฐ์ เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อให้ระบบคอมพิวเตอร์และเครื่องจักรมีความสามารถในการเลียนแบบสติปัญญาและพฤติกรรมของมนุษย์ โดยครอบคลุมกระบวนการที่สำคัญ ได้แก่ การเรียนรู้ (Learning) จากข้อมูล การคิดวิเคราะห์เพื่อหาเหตุผล การตัดสินใจ (Decision-making) และการแก้ปัญหา (Problem-solving) เทคโนโลยีปัญญาประดิษฐ์ในปัจจุบันถูกนำมาประยุกต์ใช้อย่างแพร่หลายและเป็นรากฐานสำคัญของการพัฒนานวัตกรรมที่ช่วยลดภาระงานของมนุษย์และเพิ่มประสิทธิภาพในการจัดการข้อมูลขนาดใหญ่

2.2 Machine Learning

การเรียนรู้ของเครื่อง เป็นแขนงหนึ่งของเทคโนโลยีปัญญาประดิษฐ์ (AI) ที่มุ่งเน้นการสร้างอัลกอริทึมให้คอมพิวเตอร์สามารถเรียนรู้และปรับปรุงการทำงานได้จากข้อมูลโดยไม่ต้องพึ่งพาการเขียนโปรแกรมหรือกฎคำสั่งทุกขั้นตอน การเรียนรู้ของเครื่องแบ่งออกเป็น 3 ประเภท ดังนี้

2.2.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นการเรียนรู้จากชุดข้อมูลที่มีการระบุป้ายกำกับ (Labeled data) เพื่อให้โมเดลสามารถทำนายหรือจำแนกข้อมูลใหม่ได้ เช่น รูปภาพ ข้อความ และเสียง

2.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เป็นการเรียนรู้จากชุดข้อมูลที่ไม่มีป้ายกำกับ (Unlabeled data) โดยโมเดลจะทำหน้าที่ค้นหาโครงสร้าง รูปแบบ หรือการจัดกลุ่มที่ซ่อนอยู่ในข้อมูลนั้น

2.2.3 การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) เป็นการเรียนรู้ผ่านการทดลองผิดทดลองถูก โดยระบบจะปรับเปลี่ยนพฤติกรรมตามผลลัพธ์ที่เป็นรางวัล หรือการลงโทษ

2.3 Artificial Neural Networks

โครงข่ายประสาทเทียม เป็นโมเดลทางคณิตศาสตร์ที่จำลองแบบมาจากโครงสร้างการทำงานของระบบประสาทและสมองของมนุษย์ โดยอาศัยหน่วยประมวลผลย่อยที่เรียกว่า โหนด (Neuron) ซึ่งเชื่อมต่อกันเป็นเครือข่าย โครงสร้างพื้นฐานของ ANN ประกอบด้วย 3 ชั้นหลัก ได้แก่ ชั้นข้อมูลขาเข้า (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นข้อมูลขาออก (Output Layer) การทำงานที่ประสานกันของโหนดเหล่านี้ทำให้โมเดลสามารถเรียนรู้รูปแบบที่ซับซ้อนได้ และ ANN ถือเป็นรากฐานที่สำคัญอย่างยิ่งของการพัฒนาเทคนิคการเรียนรู้เชิงลึก (Deep Learning) ในเวลาต่อมา

2.4 Deep Learning

การเรียนรู้เชิงลึก เป็นเทคนิคขั้นสูงของ Machine Learning ที่อาศัยโครงข่ายประสาทเทียมที่มีความลึกหรือมีชั้นซ่อน (Hidden Layers) จำนวนหลายชั้น (Deep Neural Networks) ในการเรียนรู้และสกัดจุดเด่นของข้อมูลที่มีความซับซ้อนสูงและมีปริมาณมหาศาลได้อย่างอัตโนมัติ เทคโนโลยีนี้มีบทบาทสำคัญในการผลักดันงานด้านปัญญาประดิษฐ์ในปัจจุบัน เช่น การรู้จำภาพ (Image Recognition) การรู้จำเสียง (Speech Recognition) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing)

2.5 Natural Language Processing

การประมวลผลภาษาธรรมชาติ เป็นเทคโนโลยีปัญญาประดิษฐ์แขนงหนึ่งที่ทำหน้าที่เป็นสะพานเชื่อมระหว่างคอมพิวเตอร์และภาษามนุษย์ ทำให้คอมพิวเตอร์สามารถทำความเข้าใจ ตีความ วิเคราะห์ และประมวลผลภาษาธรรมชาติที่มนุษย์ใช้สื่อสารได้อย่างมีประสิทธิภาพ ตัวอย่างของการประยุกต์ใช้ NLP ที่สำคัญคือ เทคโนโลยีการแปลภาษาด้วยเครื่อง (Machine Translation) ซึ่งมีวิวัฒนาการตั้งแต่การแปลโดยอาศัยกฎทางภาษาศาสตร์ (Rule-Based Machine Translation: RBMT) ไปจนถึงการใช้สถิติจากข้อมูลจำนวนมาก (Statistical Machine Translation: SMT) ปัจจุบัน NLP เป็นเทคโนโลยีเบื้องหลังที่สำคัญของระบบค้นหาข้อมูลอัตโนมัติและโมเดลภาษาขนาดใหญ่

2.6 Transformer Architecture

สถาปัตยกรรม Transformer เป็นรูปแบบของเครือข่ายประสาทเทียม ที่ถูกออกแบบมาเพื่อเพิ่มประสิทธิภาพในการจัดการกับข้อมูลประเภทลำดับ เช่น ข้อความ โดยมีจุดเด่นสำคัญคือความสามารถในการประมวลผลข้อมูลได้พร้อมกันทั้งหมดในรูปแบบขนาน ซึ่งช่วยให้โมเดลสามารถเรียนรู้และจับความสัมพันธ์ระหว่างคำที่อยู่ห่างกันในประโยค ได้อย่างมีประสิทธิภาพกว่าสถาปัตยกรรมแบบดั้งเดิม

โครงสร้างของ Transformer [6] ประกอบด้วยสองส่วนหลัก ได้แก่ ส่วนเข้ารหัส (Encoder) และส่วนถอดรหัส (Decoder) โดยแต่ละส่วนจะประกอบด้วยเลเยอร์ย่อย หลายชั้นเรียงซ้อนกัน การทำงานในสถาปัตยกรรมนี้คือกลไก Self-Attention ซึ่งเป็นกระบวนการคำนวณค่าน้ำหนักความสำคัญ ของคำแต่ละคำในประโยค เพื่อพิจารณาว่าคำใดมีความเกี่ยวข้องหรือส่งผลต่อกันมากที่สุด กลไกนี้ช่วยให้โมเดลสามารถทำความเข้าใจบริบทและความหมายโดยรวมของประโยคได้โดยไม่จำเป็นต้องให้ความสำคัญกับทุกคำอย่างเท่าเทียมกัน แต่จะเน้นไปที่คำที่มีความสัมพันธ์กันทางความหมายในบริบทนั้น ๆ ในปัจจุบัน สถาปัตยกรรม Transformer ได้กลายเป็นรากฐานสำคัญในการพัฒนาโมเดลภาษาขนาดใหญ่ นี้ เช่น GPT [7] , Qwen [8], Gemini [9] และ Llama [10]

2.7 Large Language Model (LLM)

โมเดลภาษาที่ถูกฝึกฝนด้วยชุดข้อมูลขนาดใหญ่ และมีโครงสร้างเครือข่ายประสาทเทียมที่ซับซ้อน โดยทั่วไปมีจำนวนพารามิเตอร์หลายพันล้านถึงหลายแสนล้าน ขึ้นอยู่กับสถาปัตยกรรมและการใช้งาน ทำให้โมเดลมีความสามารถในการ "เรียนรู้ในบริบท" และมีความสามารถเข้าใจและตอบคำถามที่ซับซ้อน งานเขียนเชิงสร้างสรรค์ และการแก้ปัญหาทางตรรกะได้ดี โมเดลที่นิยมใช้งานกันในปัจจุบัน เช่น GPT (OpenAI), Gemini (Google), Claude Sonnet (Anthropic) เป็นต้น การใช้งานในการประมวลผลโมเดลภาษาขนาดใหญ่จำเป็นต้องใช้ทรัพยากรการประมวลผลจำนวนมาก เช่น เซิร์ฟเวอร์ GPU ขนาดใหญ่ ซึ่งส่งผลให้มีต้นทุนในการดำเนินงานที่สูง

2.8 Small Language Model (SLM)

โมเดลภาษาที่ได้รับการปรับปรุงประสิทธิภาพให้มีขนาดเล็ก โดยทั่วไปจะมีจำนวนพารามิเตอร์ ขนาดเล็กมากไม่ถึงพันล้านพารามิเตอร์ ทำให้สามารถทำงานได้บนอุปกรณ์ที่มีทรัพยากรจำกัด เช่น คอมพิวเตอร์ส่วนบุคคล (PC), แล็ปท็อป หรือ Edge Device โดยยังคงรักษาประสิทธิภาพทางภาษาในระดับที่น่าพอใจ เหมาะสำหรับการใช้งานภายในองค์กรที่ต้องการรักษาความเป็นส่วนตัวของข้อมูล และต้องการความรวดเร็วในการประมวลผล เช่น Meta-Llama-3-8B, Mistral-7B, รวมไปถึงโมเดลภาษาไทยตระกูล Typhoon 2.x (เช่น Typhoon2-3B, Typhoon2.5-Qwen-4B) แม้จะ

ประหยัดทรัพยากร แต่ SLM มักมีคลังความรู้รอบตัวน้อยกว่า LLM เนื่องจากถูกฝึกด้วยข้อมูลจำนวนน้อยกว่า แต่สามารถแก้ไขได้ด้วยการนำมาประยุกต์ใช้ร่วมกับเทคนิค Retrieval-Augmented Generation (RAG) ซึ่งเป็นการเพิ่มความรู้เฉพาะทางจากภายนอกเข้าไปในกระบวนการตอบคำถาม ทำให้ SLM ทำงานเฉพาะทางได้ดีเพิ่มขึ้น

2.9 Thai LLM

แม้ว่าโมเดลภาษาขนาดใหญ่ (Large Language Models) จะมีความก้าวหน้าอย่างรวดเร็วและแสดงศักยภาพสูงในการประมวลผลภาษาธรรมชาติ แต่โมเดลส่วนใหญ่ยังคงได้รับการฝึกฝนจากชุดข้อมูลภาษาอังกฤษเป็นหลัก ส่งผลให้ประสิทธิภาพในการประมวลผลภาษาไทย (Thai Natural Language Processing) ยังมีข้อจำกัดในหลายมิติ เช่น การตัดคำ ความเข้าใจโครงสร้างไวยากรณ์ที่ซับซ้อน และการตีความบริบททางวัฒนธรรมที่เฉพาะเจาะจงของภาษาไทย โดยเฉพาะในเอกสารทางการและเอกสารองค์กร

ด้วยข้อจำกัดดังกล่าว จึงเกิดความพยายามในการพัฒนาโมเดลภาษาที่มุ่งเน้นภาษาไทยโดยเฉพาะ หนึ่งในแนวทางที่สำคัญคือโครงการ OpenThaiGPT [11] ซึ่งเป็นโครงการโอเพนซอร์สที่มีเป้าหมายในการพัฒนาโมเดลภาษาไทยโดยใช้ชุดข้อมูลภาษาไทยเป็นหลัก เพื่อยกระดับขีดความสามารถของโมเดลให้สามารถรองรับงานด้านภาษาไทยได้อย่างมีประสิทธิภาพและทัดเทียมกับโมเดลระดับสากล

นอกจากนี้ ตระกูลโมเดล Typhoon [12] ซึ่งพัฒนาโดย SCB 10X เป็นอีกหนึ่งตัวอย่างของการพัฒนา Thai LLM ที่สำคัญ โดยอาศัยสถาปัตยกรรมพื้นฐานจากโมเดลขนาดใหญ่ระดับสากล เช่น Llama, Gemini และ Qwen และนำมาฝึกฝนต่อด้วยชุดข้อมูลภาษาไทยขนาดใหญ่ พร้อมทั้งปรับแต่งด้วยเทคนิค Instruction Tuning เพื่อเพิ่มความสามารถในการทำความเข้าใจคำสั่งและบริบทการใช้งานจริง ผลลัพธ์ที่ได้คือโมเดลตระกูล Typhoon 2.x ที่มีความสามารถในการเข้าใจโครงสร้างภาษาไทย บริบทเชิงวัฒนธรรม และรูปแบบภาษาทางการได้ดีกว่าโมเดลทั่วไป

การเลือกใช้โมเดลภาษาไทยที่ได้รับการออกแบบและปรับแต่งเฉพาะด้าน จึงเป็นปัจจัยสำคัญต่อการพัฒนาระบบถาม–ตอบจากเอกสารภาษาไทย โดยเฉพาะในระบบ Retrieval-Augmented Generation (RAG) ซึ่งคุณภาพของการสร้างคำตอบในขั้นตอน Generation มีความอ่อนไหวต่อความถูกต้องของภาษาและบริบทข้อมูล การใช้โมเดลตระกูล Typhoon 2.x เป็นโมเดลหลักในการสร้างคำตอบ เพื่อให้สอดคล้องกับลักษณะของเอกสารองค์กรภาษาไทยและเพิ่มความน่าเชื่อถือของผลการทดลอง

2.10 Thai Tokenization

โทเค็นภาษาไทยมีลักษณะเฉพาะทางภาษาศาสตร์ที่แตกต่างจากภาษาอังกฤษ โดยมีความท้าทายหลักคือ การไม่มีการเว้นวรรคระหว่างคำ ข้อความภาษาไทยมักถูกเขียนเป็นคำอักขระต่อเนื่อง ทำให้ไม่สามารถใช้ช่องว่างเป็นตัวแบ่งคำได้โดยตรง ส่งผลให้การทำ Tokenization [13] [14] ระดับคำ แบบดั้งเดิมไม่สามารถนำมาใช้งานได้โดยตรง ซึ่งทำให้คำมีความคลุมเครือ เช่น “การศึกษาความเป็นไปได้อย่างมากที่จะเรียนรู้ไม่สิ้นสุด” สามารถถูกตีความและแบ่งคำได้หลายรูปแบบ หากการแบ่งคำผิดพลาด จะส่งผลต่อความหมายโดยตรง และส่งผลต่อประสิทธิภาพของโมเดลในขั้นตอนถัดไป

2.10.1 ข้อจำกัดของ Word Tokenization กับภาษาไทย การใช้ Word Tokenization กับภาษาไทยมักต้องพึ่งพาพจนานุกรม หรือกฎเชิงไวยากรณ์ ซึ่งมีข้อจำกัดสำคัญ ดังนี้

1. ปัญหาคำศัพท์นอกคลัง (Out-of-Vocabulary - OOV) ภาษาไทยมีการสร้างคำใหม่และคำเฉพาะทางอยู่ตลอดเวลา เช่น คำทางกฎหมาย คำทางราชการ หรือคำทับศัพท์ หากคำเหล่านี้ไม่อยู่ในพจนานุกรม ระบบจะไม่สามารถแบ่งคำได้อย่างถูกต้อง

2. ความคลุมเครือของการแบ่งคำ (Segmentation Ambiguity) ประโยคเดียวกันอาจแบ่งคำได้หลายแบบ และการเลือกแบบใดแบบหนึ่งอาจไม่สอดคล้องกับความหมายที่ต้องการ มักมีโครงสร้างประโยคซับซ้อน ใช้ศัพท์เฉพาะ และมีรูปแบบการเขียนที่ไม่สอดคล้องกับภาษาพูดทั่วไป

2.10.2 Character Tokenization กับภาษาไทย เป็นแนวทางที่สามารถหลีกเลี่ยงปัญหาการแบ่งคำของภาษาไทยได้โดยตรง เนื่องจากแบ่งข้อความออกเป็นตัวอักษรแต่ละตัว ภาษาไทยมีระบบอักขระที่ซับซ้อน ประกอบด้วยพยัญชนะ สระ วรรณยุกต์ และเครื่องหมายกำกับเสียง ซึ่งการแบ่งในระดับตัวอักษรเพียงอย่างเดียวไม่สามารถสะท้อนหน่วยความหมายที่ของภาษาได้ โมเดลต้องเรียนรู้ความสัมพันธ์ของลำดับอักขระที่ยาวขึ้นอย่างมาก ส่งผลให้ภาระการคำนวณเพิ่มสูง และยากต่อการจับความหมายเชิงบริบท เช่น เอกสารประกาศ ระเบียบ หรือข้อบังคับขององค์กร

2.10.3 Sub-word Tokenization สามารถจัดการกับสมดุลระหว่างความละเอียดและความหมายของภาษา การเรียนรู้หน่วยย่อยของคำจากข้อมูลจริง แทนการพึ่งพาการแบ่งคำตามพจนานุกรม เช่น คำว่า “การดำเนินงาน” อาจถูกแบ่งออกเป็นโทเค็นย่อย เช่น “การ”, “ดำเนิน”, “งาน” หรือในกรณีของคำเฉพาะทาง ระบบสามารถแบ่งเป็นหน่วยย่อยที่พบได้บ่อยในชุดข้อมูลฝึก ทำให้โมเดลสามารถเรียนรู้ความหมายได้แม้ไม่เคยพบคำทั้งคำมาก่อน ช่วยลดปัญหา OOV คำใหม่หรือคำเฉพาะทางสามารถถูกแยกเป็นหน่วยย่อยที่โมเดลได้ รักษาความหมายเชิงบริบทได้ดีกว่า Character Tokenization หน่วยย่อยมีขนาดใหญ่พอที่จะสะท้อนความหมายบางส่วนของคำ Sub-word Tokenization ช่วยให้การฝังข้อมูล (Embedding) สามารถจับความหมายของข้อความภาษาไทยได้ดีขึ้น ส่งผลโดยตรงต่อคุณภาพการค้นคืน (Retrieval) และการสร้างคำตอบ (Generation) ดีขึ้น

2.11 Byte-Pair Encoding (BPE)

Byte-Pair Encoding (BPE) [15] เป็นอัลกอริทึมสำหรับการทำ Tokenization แบบซับเวิร์ด (Sub-word Tokenization) ที่ถูกใช้อย่างแพร่หลายในงานด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) โดยเฉพาะในโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) แนวคิดของ BPE มีพื้นฐานมาจากเทคนิคการบีบอัดข้อมูล ซึ่งมุ่งรวมหน่วยย่อยที่ปรากฏบ่อยเข้าด้วยกัน เพื่อสร้างคลังคำศัพท์ที่มีขนาดเหมาะสมและยืดหยุ่น ความสามารถในการจัดการกับคำศัพท์ที่ไม่เคยพบมาก่อน (Out-of-Vocabulary: OOV) โดยการแยกคำออกเป็นหน่วยย่อยที่โมเดลคุ้นเคย ทำให้ไม่จำเป็นต้องเพิ่มคำใหม่เข้าไปในคลังคำศัพท์ทั้งหมด เนื่องจากไม่ต้องพึ่งพาพจนานุกรม แต่เรียนรู้หน่วยย่อยจากข้อมูลจริง เช่น “การศึกษา การศึกษา การเรียนรู้ การเรียนรู้เชิงลึก”

ขั้นที่ 1 แยกข้อความออกเป็นอักขระเดี่ยว ในระยะแรก BPE จะมองคำทุกคำเป็นลำดับของตัวอักษร พร้อมเพิ่มสัญลักษณ์สิ้นสุดคำ </w>

การศึกษา เป็น ก า ร ศ ึ ก ษ า </w>

การเรียนรู้ เป็น ก า ร เ ร ี ย น ร ู ้ </w>

คลังคำศัพท์เริ่มต้นจะประกอบด้วยอักขระที่ไม่ซ้ำกัน เช่น ก, า, ร, ศ, ึ, ก, ษ, เ, ร, ี, ย, น, ู, ้, </w>

ขั้นที่ 2 รวมคู่ตัวอักษรที่พบบ่อยที่สุด จากชุดข้อมูลตัวอย่าง คู่ตัวอักษร “ก า ร” จะปรากฏซ้ำหลายครั้งในคำว่า “การศึกษา” และ “การเรียนรู้” BPE จะเริ่มรวมคู่ที่พบบ่อยทีละขั้น เช่น

ก + า เป็น กา

กา + ร เป็น การ

หลังจากรวมแล้ว คำจะถูกแทนด้วยโทเค็นย่อย เช่น การศึกษา เป็น การ ศ ึ ก ษ า </w> การเรียนรู้ เป็น การ เ ร ี ย น ร ู ้ </w>

ขั้นที่ 3 รวมหน่วยย่อยที่มีความถี่สูงต่อเนื่อง เมื่อทำซ้ำหลายรอบ BPE จะเรียนรู้หน่วยย่อยที่มีความหมายมากขึ้น เช่น

การ + ศึกษา เป็น การศึกษา

เรียน + รู้ เป็น เรียนรู้

คำศัพท์สุดท้ายอาจประกอบด้วย

ตัวอักษรเดี่ยว: ก, า, ร, ศ, ึ, ก, ษ, เ, ร, ี, ย, น, ู, ้

โทเค็นซับเวิร์ด: การ, ศึกษา, เรียน, รู้, การศึกษา, เรียนรู้

สัญลักษณ์พิเศษ: </w>

2.12 Text Chunking

2.12.1 การแบ่งข้อความออกเป็นส่วนย่อย หรือ Text Chunking คือกระบวนการแบ่งเอกสารขนาดใหญ่ออกเป็นหน่วยย่อยที่มีขนาดเล็กลง เพื่อให้สามารถจัดการและประมวลผลร่วมกับโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) ได้อย่างมีประสิทธิภาพ เนื่องจาก LLMs มีข้อจำกัดด้านขนาดบริบท (Context Window) ซึ่งไม่สามารถรับข้อความยาวทั้งหมดได้ในครั้งเดียว การแบ่งข้อความออกเป็น Chunks จึงเป็นแนวทางสำคัญในการเตรียมข้อมูลก่อนนำเข้าสู่โมเดล

2.12.2 การทำ Text Chunking เป็นการลดขนาดข้อมูลให้อยู่ในขอบเขตที่เหมาะสมสำหรับการประมวลผลของโมเดลภาษา ควบคู่กับการเพิ่มประสิทธิภาพในการค้นคืนข้อมูล โดยช่วยให้ระบบสามารถดึงเฉพาะส่วนของเอกสารที่มีความเกี่ยวข้องกับคำถามได้อย่างแม่นยำ การแบ่งข้อความที่เหมาะสมยังมีบทบาทสำคัญในการรักษาคุณภาพของบริบท (Context Quality) ซึ่งเป็นปัจจัยหลักที่สนับสนุนการสร้างคำตอบที่ถูกต้องและสอดคล้องกับแหล่งข้อมูลจริง ด้วยเหตุนี้ Text Chunking จึงถือเป็นขั้นตอนสำคัญในสถาปัตยกรรม Retrieval-Augmented Generation (RAG) ซึ่งทำหน้าที่เชื่อมโยงระหว่างเอกสารต้นทาง ระบบค้นคืน และกระบวนการสร้างคำตอบของ LLMs

2.12.3 แนวทางการแบ่งข้อความที่ใช้กันทั่วไปคือการกำหนดขนาดของ Chunk แบบคงที่ตามจำนวนอักขระหรือจำนวนโทเค็น เช่น การแบ่งทุก ๆ 500 หรือ 1,000 ตัวอักษร (Fixed-size Chunking) วิธีการนี้มีความง่ายในการนำไปใช้งานและสามารถควบคุมขนาดข้อมูลได้อย่างชัดเจน อย่างไรก็ตาม ข้อจำกัดสำคัญของการแบ่งแบบขนาดคงที่คือไม่คำนึงถึงโครงสร้างและความหมายของเนื้อหา

ในบริบทของเอกสารที่ต้องการความต่อเนื่องของเหตุผลและโครงสร้างเชิงตรรกะ เช่น เอกสารทางกฎหมาย ระเบียบ ข้อบังคับ หรือเอกสารเชิงองค์กร การแบ่งข้อความแบบขนาดคงที่อาจทำให้บริบทของเนื้อหาถูกตัดขาดกลางประโยคหรือกลางย่อหน้า ส่งผลให้เหตุผลทางกฎหมายหรือเงื่อนไขสำคัญสูญเสียความเชื่อมโยง และชิ้นส่วนข้อมูลที่ได้ไม่สอดคล้องกับโครงสร้างของเอกสารโดยรวม ผลที่ตามมาคือโมเดลภาษาขนาดใหญ่อาจตีความบริบทผิดพลาด และนำไปสู่การสร้างคำตอบที่คลาดเคลื่อนจากข้อเท็จจริงได้

การแบ่งข้อความแบบลำดับชั้น (Hierarchical/Recursive Splitting) เป็นแนวทางที่คำนึงถึงโครงสร้างและความหมายของเนื้อหา โดยระบบจะพยายามแบ่งข้อความตามหน่วยที่มีความหมายก่อน เช่น ย่อหน้า หรือประโยค หากยังไม่สามารถควบคุมขนาดของข้อความให้อยู่ภายในขอบเขตที่กำหนดได้ จึงค่อยแบ่งย่อยลงตามหน่วยที่เล็กลง วิธีการนี้ช่วยลดปัญหาการตัดข้อความกลางประโยคหรือกลางย่อหน้า ทำให้บริบทในแต่ละ Chunk มีความต่อเนื่องและสอดคล้องกับโครงสร้างของเอกสารมากกว่าการแบ่งแบบขนาดคงที่ อย่างไรก็ตาม ในงานวิจัยนี้กำหนดขนาดของ

Chunk เป็นหน่วยโทเค็น (tokens) เพื่อให้สอดคล้องกับการนับความยาวด้วย tokenizer ของโมเดลฝังความหมาย และเพื่อให้การทดลองสามารถควบคุมตัวแปรได้อย่างเป็นระบบ

นอกจากนี้ การกำหนด ขนาดของ Chunk (Chunk Size) และ ช่วงซ้อนทับของข้อความ (Overlap) ถือเป็นปัจจัยสำคัญที่ส่งผลโดยตรงต่อคุณภาพของบริบทที่ถูกส่งเข้าสู่ระบบ RAG โดย Chunk ที่มีขนาดเล็กเกินไปอาจขาดข้อมูลเชิงบริบทที่จำเป็นสำหรับการสังเคราะห์คำตอบ ขณะที่ Chunk ที่มีขนาดใหญ่เกินไปอาจนำข้อมูลที่ไม่เกี่ยวข้องหรือสัญญาณรบกวน (Noise) เข้ามาปะปนและเพิ่มภาระการประมวลผลของโมเดล การเลือกใช้ขนาด Chunk ที่เหมาะสมร่วมกับการกำหนด Overlap ระหว่าง Chunks เพื่อรักษาความต่อเนื่องของเนื้อหา โดยเฉพาะในกรณีที่แนวคิดหรือเหตุผลสำคัญถูกแบ่งอยู่ระหว่างขอบเขตของ Chunk

การใช้ Overlap ช่วยให้ข้อมูลสำคัญที่ปรากฏบริเวณรอยต่อของ Chunks ไม่สูญหาย และช่วยเพิ่มโอกาสที่ระบบค้นคืนจะดึงบริบทที่ครบถ้วนและเชื่อมโยงกันได้ ส่งผลให้ขั้นตอนการสืบค้น มีความแม่นยำมากขึ้น และสนับสนุนการสร้างคำตอบในขั้นตอน Generation ให้สอดคล้องกับเนื้อหาต้นทาง โดยเฉพาะในเอกสารที่มีโครงสร้างซับซ้อนและต้องการความต่อเนื่องของเหตุผล เช่น เอกสารทางกฎหมาย ระเบียบ และเอกสารเชิงองค์กร ซึ่งความคลาดเคลื่อนของบริบทเพียงเล็กน้อยอาจนำไปสู่การตีความที่ผิดพลาดได้

ดังนั้น งานวิจัยจึงนำขนาดของ Chunk และค่า overlap มาเป็นพารามิเตอร์ในการออกแบบชุดการทดลอง เพื่อศึกษาผลต่อคุณภาพการค้นคืนและคุณภาพการสร้างคำตอบในระบบ RAG

2.13 Embedding model

การแปลงข้อความให้อยู่ในรูปของเวกเตอร์เชิงความหมาย (Dense Vector Representation) เพื่อให้คอมพิวเตอร์สามารถเข้าใจความหมายเชิงลึกของภาษามนุษย์ได้ ในโมเดล Embedding ได้แก่ BGE-M3 [16] ซึ่งทำหน้าที่แปลงทั้งคำถามและเอกสารให้อยู่ในพื้นที่เวกเตอร์หลายมิติ (High-dimensional Vector Space) โดยข้อความที่มีความหมายหรือบริบทใกล้เคียงกันจะถูกแทนด้วยเวกเตอร์ที่อยู่ใกล้กัน กลไกดังกล่าวช่วยให้ระบบสามารถค้นคืนข้อมูลตามความหมาย (Semantic Search) ได้อย่างมีประสิทธิภาพ แม้ว่าคำถามและเอกสารจะไม่ใช้คำศัพท์เดียวกัน ซึ่งแตกต่างจากการค้นหาแบบคีย์เวิร์ดดั้งเดิม และเป็นขั้นตอนพื้นฐานที่ส่งผลโดยตรงต่อความแม่นยำของการค้นคืนและการสร้างคำตอบในระบบ RAG

ตัวอย่างประกอบแนวคิด Embedding และ Semantic Search พิจารณาคำถามตัวอย่าง

“นักศึกษาสามารถลาพักการเรียนได้กี่ภาคการศึกษา”

แม้เอกสารต้นทางจะใช้ถ้อยคำว่า

“นักศึกษามีสิทธิ์ขอพักการศึกษาได้ไม่เกินสองภาคการศึกษาติดต่อกัน”

ซึ่งไม่ได้มีการใช้คำศัพท์เดียวกันทั้งหมด เช่น “ลาพักการเรียน” และ “พักการศึกษา” แต่เมื่อข้อความทั้งสองถูกแปลงเป็นเวกเตอร์ด้วยโมเดล BGE-M3 เวกเตอร์ของคำถามและเอกสารจะอยู่ใกล้กันในพื้นที่เวกเตอร์เชิงความหมาย ทำให้ระบบสามารถค้นคืนเอกสารที่เกี่ยวข้องได้อย่างถูกต้อง กลไกนี้แสดงให้เห็นถึงความสามารถของ Embedding Model ในการจับคู่ความหมายเชิงบริบทมากกว่าการจับคู่คำแบบตรงตัว ซึ่งมีความสำคัญอย่างยิ่งสำหรับเอกสารเชิงองค์กรและเอกสารทางกฎหมายที่มักใช้ถ้อยคำหลากหลายแต่มีความหมายสอดคล้องกัน

2.14 Retrieval

การค้นคืนข้อมูล (Retrieval) เป็นกระบวนการและกลไกสำคัญในสถาปัตยกรรม Retrieval-Augmented Generation (RAG) ที่ทำหน้าที่ในการดึงสารสนเทศจากแหล่งความรู้ภายนอกมาประกอบเป็นบริบท (Context) กระบวนการนี้ช่วยแก้ไขข้อจำกัดของโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) ในเรื่องการหลอนข้อมูล (Hallucination) หรือการสร้างคำตอบที่ไม่สามารถตรวจสอบแหล่งที่มาได้ โดยระบบจะบังคับให้โมเดลภาษาอ้างอิงข้อมูลจากการค้นคืนเท่านั้น ซึ่งกระบวนการค้นคืนข้อมูลในระบบ RAG สมัยใหม่ ประกอบด้วยกลไกและแนวคิดที่สำคัญ ดังต่อไปนี้

2.14.1 การค้นคืนข้อมูลด้วยเวกเตอร์ (Vector Search และ Bi-Encoder)

ในขั้นตอนแรกของการค้นคืน ระบบจะทำการแปลงเอกสารความรู้และคำถามของผู้ใช้ (User Query) ให้อยู่ในรูปแบบของเวกเตอร์เชิงความหมาย (Vector Embeddings) จากนั้นระบบจะใช้หลักการวัดความคล้ายคลึงเชิงความหมาย เช่น วิธี Cosine Similarity ซึ่งเป็นการคำนวณค่ามุมระหว่างเวกเตอร์ของคำถาม (Query Vector) และเวกเตอร์ของเอกสาร (Document Vector) โดยค่าที่เข้าใกล้ 1 จะแสดงถึงระดับความคล้ายคลึงทางความหมายที่สูง การค้นคืนในขั้นตอนนี้มักทำงานผ่านสถาปัตยกรรมแบบ Bi-Encoder ซึ่งแปลงเวกเตอร์แยกกัน ทำให้ระบบสามารถสืบค้นเอกสารจำนวนมหาศาลได้อย่างรวดเร็วและสามารถดึงเนื้อหาที่สอดคล้องกันได้แม้จะไม่ใช้คำศัพท์เดียวกันโดยตรง

2.14.2 การจัดอันดับใหม่ (Reranking ด้วย Cross-Encoder)

แม้กระบวนการ Vector Search เริ่มต้นจะสามารถสืบค้นข้อมูลได้อย่างมีประสิทธิภาพแต่ยังมีข้อจำกัดในการพิจารณาความสัมพันธ์เชิงลึก ส่งผลให้เนื้อหาบางส่วนที่มีความ

คล้ายคลึงเพียงระดับผิวเผินถูกดึงเข้ามาปะปนในผลลัพธ์ด้วย เพื่อลดข้อจำกัดดังกล่าว ระบบค้นคืนจึงมักถูกเสริมด้วยขั้นตอนการจัดอันดับใหม่ (Reranking) โดยใช้โมเดลแบบ Cross-Encoder [17] ซึ่งจะทำหน้าที่รับข้อมูลคำถามและเอกสารที่ค้นพบมาประมวลผลร่วมกันภายในโมเดลเดียว เพื่อประเมินระดับความเกี่ยวข้องอย่างละเอียดและให้คะแนน (Relevance Score) ใหม่ กลไกนี้ช่วยคัดกรองสัญญาณรบกวน (Noise) และผลักดันเนื้อหาที่สมบูรณ์ที่สุดขึ้นมาสู่อันดับต้น ๆ (Top-K) ก่อนส่งเป็นบริบทให้ LLM

2.14.3 ปรากฏการณ์ความย้อนแย้งของการค้นคืน (The Retrieval Paradox)

ในการประเมินประสิทธิภาพของการค้นคืนข้อมูล ระบบจะถูกวัดผลผ่านเมตริกชี้วัดเชิงการจัดอันดับ เช่น อัตราการค้นพบ (Hit Rate@K), ค่าเฉลี่ยอันดับส่วนกลับ (MRR), และค่าคุณภาพการจัดอันดับ (NDCG) จากการศึกษาวิจัยการค้นคืนในสภาพแวดล้อมข้อมูลจริง โดยเฉพาะเมื่อต้องเผชิญกับเอกสารภาษาไทยที่ผ่านการแปลงภาพเป็นข้อความ (OCR) ซึ่งมีสัญญาณรบกวนแฝงอยู่ ทำให้พบปรากฏการณ์ที่เรียกว่า "ความย้อนแย้งของการค้นคืน" (Retrieval Paradox) ปรากฏการณ์นี้อธิบายสภาวะที่โมเดลการค้นคืนสามารถดึงชิ้นส่วนข้อมูลที่เกี่ยวข้องเข้ามารวมในผลลัพธ์ได้เป็นจำนวนมาก (มีความหนาแน่นของการค้นคืนสูง) แต่สัญญาณรบกวนเหล่านั้นกลับทำลายการจับคู่ความหมายเชิงลึก (Semantic Matching) ส่งผลให้คุณภาพการจัดอันดับความถูกต้อง (MRR และ NDCG) ลดต่ำลง ท้ายที่สุดข้อมูลที่ถูกต้องสมบูรณ์จึงถูกฉุดให้ตกไปอยู่ในอันดับท้าย ๆ และทำให้ LLM สังเคราะห์คำตอบที่ผิดพลาด

2.15 Retrieval-Augmented Generation (RAG)

เป็นสถาปัตยกรรมที่ผสานกระบวนการค้นคืนข้อมูล (Retrieval) เข้ากับกระบวนการสร้างข้อความ (Generation) เพื่อเพิ่มความถูกต้องและความน่าเชื่อถือของคำตอบที่สร้างโดยโมเดลภาษาขนาดใหญ่ (Large Language Models: LLMs) แนวคิดหลักของ RAG คือการให้โมเดลภาษาอ้างอิงข้อมูลจากแหล่งความรู้ภายนอกที่กำหนดไว้ แทนการอาศัยเพียงข้อมูลที่ถูกฝึกมาในตัวโมเดลเท่านั้น ซึ่งช่วยลดปัญหาการสร้างข้อมูลที่ไม่เป็นจริงหรือไม่สามารถตรวจสอบได้ (Hallucination) และช่วยให้ระบบสามารถตอบคำถามโดยใช้ข้อมูลที่มีความทันสมัย โดยไม่จำเป็นต้องฝึกโมเดลใหม่เมื่อข้อมูลมีการเปลี่ยนแปลง

กระบวนการทำงานของ RAG ประกอบด้วย 2 ขั้นตอนหลัก ได้แก่ Retrieval และ Augmentation and Generation ในขั้นตอน Retrieval ระบบจะทำการรวบรวมเอกสารจากแหล่งความรู้ภายนอก เช่น เอกสารกฎหมาย ระเบียบ หรือฐานข้อมูลองค์กร แล้วแปลงเอกสารเหล่านั้นให้อยู่ในรูปของเวกเตอร์เชิงความหมายก่อนจัดเก็บในฐานข้อมูลเวกเตอร์ เมื่อผู้ใช้ส่งคำถามเข้ามา ระบบ

จะแปลงคำถามเป็นเวกเตอร์ในลักษณะเดียวกัน และทำการค้นคืนเอกสารที่มีความคล้ายคลึงเชิงความหมายสูงสุดจากฐานข้อมูลเวกเตอร์เพื่อใช้เป็นข้อมูลอ้างอิง จากนั้นในขั้นตอน Augmentation and Generation ระบบจะนำคำถามของผู้ใช้ร่วมกับข้อมูลที่ดึงมาได้จากขั้นตอน Retrieval มาประกอบเป็นบริบท ใหม่ที่สมบูรณ์ และส่งต่อให้โมเดลภาษาใช้ในการสังเคราะห์คำตอบที่มีความถูกต้อง สอดคล้องกับแหล่งข้อมูล

2.16 Vector Database

การเก็บข้อมูลในรูปแบบ Vector โดยอาศัยระบบฐานข้อมูล PostgreSQL [18] ร่วมกับส่วนขยาย pgvector [19] ซึ่งรองรับการจัดเก็บเวกเตอร์เชิงความหมาย การสร้างดัชนี (Indexing) และการค้นคืนข้อมูลเวกเตอร์ได้อย่างมีประสิทธิภาพ เวกเตอร์ของเอกสารและคำถามที่ได้จากกระบวนการ Embedding จะถูกจัดเก็บในฐานข้อมูลเดียวกัน ทำให้สามารถผสานการจัดการข้อมูลเชิงโครงสร้างและข้อมูลเชิงความหมายเข้าด้วยกันในสภาพแวดล้อมเดียว ซึ่งเหมาะสมกับการพัฒนาระบบ Retrieval-Augmented Generation (RAG) ที่ต้องการความยืดหยุ่นและความสามารถในการขยายระบบ

กระบวนการค้นคืนข้อมูลเวกเตอร์ (Vector Search) ใช้หลักการวัดความคล้ายคลึงเชิงความหมายด้วย Cosine Similarity ซึ่งเป็นการคำนวณค่ามุมระหว่างเวกเตอร์ของคำถาม (Query Vector) และเวกเตอร์ของเอกสาร (Document Vector) โดยค่าที่ได้จะอยู่ในช่วงตั้งแต่ -1 ถึง 1 และค่าที่เข้าใกล้ 1 แสดงถึงระดับความคล้ายคลึงเชิงความหมายที่สูง วิธีการนี้ช่วยให้ระบบสามารถค้นคืนเอกสารที่มีความหมายสอดคล้องกับคำถามได้อย่างแม่นยำ แม้ว่าจะไม่ได้ใช้คำศัพท์เดียวกันโดยตรง และเป็นกลไกพื้นฐานที่สนับสนุนประสิทธิภาพของขั้นตอน Retrieval ก่อนนำผลลัพธ์ไปผ่านกระบวนการจัดอันดับใหม่และการสร้างคำตอบในระบบ RAG

2.17 Optical Character Recognition

เทคโนโลยีที่ใช้สำหรับแปลงข้อมูลจากรูปภาพให้อยู่ในรูปของข้อความ ซึ่งมีบทบาทสำคัญในการประมวลผลเอกสารองค์กร เนื่องจากเอกสารส่วนใหญ่มักจัดเก็บอยู่ในรูปแบบไฟล์ PDF โดยลักษณะของไฟล์ PDF สามารถแบ่งออกได้เป็นสองประเภทหลัก ได้แก่ Text-based PDF ซึ่งสามารถสกัดข้อความออกมาได้โดยตรง และ Image-based PDF ซึ่งเป็นไฟล์ที่เกิดจากการสแกนเอกสารและจำเป็นต้องใช้เทคโนโลยี OCR ในการแปลงภาพเป็นข้อความก่อนนำไปใช้

การทำ OCR กับภาษาไทยยังคงมีข้อจำกัดสำคัญ เนื่องจากโครงสร้างของภาษาไทยประกอบด้วยพยัญชนะ สระ และวรรณยุกต์ที่มีการจัดวางในหลายตำแหน่ง ส่งผลให้กระบวนการ OCR มักเกิดสิ่งรบกวนของข้อมูล (Noise) เช่น วรรณยุกต์ลอย สระซ้อน หรือการอ่านตัวอักษร

ผิดพลาด ความคลาดเคลื่อน ส่งผลกระทบโดยตรงต่อกระบวนการแปลงข้อความเป็นเวกเตอร์เชิงความหมาย (Vector Embedding) เนื่องจากคำที่ถูกอ่านผิดจะถูกแปลงเป็นเวกเตอร์ที่แตกต่างจากคำต้นฉบับอย่างสิ้นเชิง ตัวอย่างเช่น คำว่า “บริษัท” หากถูก OCR อ่านผิดเป็น “บวิษัท” เวกเตอร์ที่ได้จะมีตำแหน่งในพื้นที่เวกเตอร์ (Vector Space) ที่ห่างจากคำที่ถูกต้องอย่างมีนัยสำคัญ เมื่อวัดความคล้ายคลึงด้วย Cosine Similarity ระยะห่างระหว่างเวกเตอร์ของคำถามและเอกสารจะเพิ่มขึ้น ส่งผลให้ระบบไม่สามารถค้นคืนเอกสารที่เกี่ยวข้องได้อย่างถูกต้อง ปัญหาด้านคุณภาพของข้อความจาก OCR จึงเป็นสมมติฐานหลักในการศึกษาผลกระทบของตัวแปรด้าน คุณภาพแหล่งข้อมูล ต่อประสิทธิภาพของระบบ Retrieval-Augmented Generation

2.18 Evaluation

การพัฒนาระบบค้นหาข้อมูลและระบบตอบคำถามอัตโนมัติจำเป็นต้องมีการประเมินประสิทธิภาพของระบบ ให้มีความสอดคล้องกับการนำไปใช้งาน การเลือกใช้เมตริกที่ไม่เหมาะสมในการประเมินไม่เหมาะสมอาจทำให้ผลการประเมินคลาดเคลื่อน และนำไปสู่ข้อสรุปที่ไม่ถูกต้องเมื่อนำระบบไปใช้งานจริงกลุ่มของการค้นคืน การประเมินโดยแบ่งเมตริกออกเป็น 2 กลุ่มหลัก ดังนี้

2.18.1 Hit Rate@K วัดสัดส่วนของคำถามทั้งหมดที่ระบบสามารถค้นพบ อย่างน้อย 1 Chunk ที่มาจากเอกสารคำตอบอ้างอิง (gold document) ภายในผลลัพธ์ Top-K (Chunk list) โดยมีสมการดังนี้

$$HitRate@K = \frac{Number\ of\ queries\ with\ at\ least\ one\ relevant\ result\ in\ top\ K}{Total\ number\ of\ queries} \quad (1)$$

เมตริกนี้สะท้อนว่า ระบบ ‘พลาด’ หรือ ‘ไม่พลาด’ การค้นคืน อย่างน้อย 1 Chunk ของ gold document ภายใน Top-K หากค่า Hit Rate ต่ำ แสดงว่าระบบมีความเสี่ยงสูงที่จะไม่สามารถส่งบริบทที่ถูกต้องให้ LLM ได้ แม้ในขั้นตอนการสร้างคำตอบจะมีความสามารถสูงก็ตาม

2.18.2 Mean Reciprocal Rank (MRR) เป็นตัววัดตำแหน่งเฉลี่ยของ Chunk ที่เกี่ยวข้อง (มาจาก gold document) รายการแรก ในผลลัพธ์การค้นคืน โดยให้ความสำคัญกับการที่ผลลัพธ์ที่เกี่ยวข้องถูกจัดอยู่ในอันดับต้น ๆ โดยมีสมการดังนี้

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \quad (2)$$

โดยที่ $rank_i$ คืออันดับของ Chunk ที่ relevant รายการแรกสำหรับคำถาม i

ค่า MRR สูงหมายความว่า Chunk ที่เกี่ยวข้องจาก gold document มักถูกจัดอยู่ในอันดับต้น ๆ ซึ่งเป็นคุณสมบัติสำคัญมากสำหรับระบบ RAG เนื่องจาก LLM มีข้อจำกัดด้านความยาวของบริบท และมักใช้เพียง Top-K Chunks แรก เท่านั้น

2.18.3 Normalized Discounted Cumulative Gain (nDCG) ใช้สำหรับวัดคุณภาพของการจัดลำดับผลลัพธ์ โดยให้ค่าน้ำหนักกับ Chunk ที่เกี่ยวข้องตามตำแหน่งอันดับ โดย Chunk ที่อยู่ลำดับต้นมีความสำคัญมากกว่า ในงานวิจัยนี้คำนวณ nDCG โดยกำหนด relevance แบบไบนารีต่อแต่ละ Chunk (relevant = 1 เมื่อ Chunk มาจาก gold document, และไม่ relevant = 0)

nDCG ไม่เพียงพิจารณาว่า “เจอหรือไม่เจอ” แต่พิจารณาว่า จัดเรียงดีแค่ไหน จึงเหมาะสำหรับประเมินระบบที่มีขั้นตอน Reranking และต้องการบริบทคุณภาพสูงที่สุดส่งให้ LLM โดยค่าที่ได้จะอยู่ในช่วง $0 \le nDCG@K \le 1$

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{3}$$

DCG จากคะแนนความเกี่ยวข้องและตำแหน่ง

$$DCG@K = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)} \tag{4}$$

IDCG@K (Ideal DCG) คือค่า DCG สูงสุดที่เป็นไปได้ โดยคำนวณจากการจัดลำดับ Chunk ที่เกี่ยวข้อง ให้อยู่ในตำแหน่งที่ดีที่สุด

$$IDCG@K = \sum_{i=1}^{k} \frac{(rel_i^{ideal})}{log_2(i+1)} \tag{5}$$

เมตริกสำหรับขั้นตอนการสร้างคำตอบ เมตริกที่ใช้ประเมินคุณภาพของคำตอบสุดท้ายที่โมเดลภาษา (SLM) สร้างขึ้น โดยเปรียบเทียบกับคำตอบอ้างอิง

2.18.4 Accuracy การวัด สัดส่วนของคำตอบที่ระบบตอบ “ถูกต้อง” เมื่อเทียบกับจำนวนคำถามทั้งหมด เหมาะสำหรับการประเมินแบบจัดประเภทผลลัพธ์เป็น “ถูก/ไม่ถูก” ตามเกณฑ์ที่กำหนด เช่น Strict (≥0.8) หรือ Lax (≥0.6) โดยมีสมการดังนี้

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

2.18.5 Precision วัดสัดส่วนของข้อมูลที่ระบบสร้างขึ้นซึ่งถูกต้อง เมื่อเทียบกับข้อมูลทั้งหมดที่ระบบตอบออกมา Precision สูง ระบบไม่สร้างข้อมูลเกินจริงหรือข้อมูลผิด (ลด Hallucination) ซึ่งเป็นคุณสมบัติสำคัญของระบบถาม-ตอบเชิงข้อเท็จจริง โดยมีสมการดังนี้

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

2.18.6 Recall วัดสัดส่วนของข้อมูลที่ถูกต้องทั้งหมดในคำตอบอ้างอิง ที่ระบบสามารถตอบได้ครบถ้วน Recall สูงแสดงว่าระบบสามารถดึงสาระสำคัญออกมาได้ครบ แต่หาก Precision ต่ำ อาจหมายถึงตอบยืดยาวหรือมีข้อมูลเกินจำเป็น โดยมีสมการดังนี้

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

2.18.7 F1-Score เป็นค่าเฉลี่ยเชิงฮาร์มอนิก (Harmonic Mean) ของ Precision และ Recallใช้ประเมินสมดุลระหว่างความแม่นยำและความครบถ้วนของคำตอบ ประเมินคุณภาพโดยรวมของคำตอบในระบบ RAG เพราะสะท้อนทั้ง ความถูกต้อง (Precision) ความครบถ้วน (Recall) โดยมีสมการดังนี้

$$F1 - Score = \frac{2\,x\,Precision\ \times\ Recall}{Precision\ + Recall} \tag{9}$$

เมตริกแต่ละตัวถูกเลือกให้สะท้อนคุณลักษณะของระบบ RAG ในมิติที่แตกต่างกัน โดย HitRate@K ใช้ตรวจสอบเงื่อนไขขั้นต่ำของการค้นคืนข้อมูลที่ถูกต้อง ขณะที่ MRR@K และ nDCG@K ใช้วัดคุณภาพของลำดับผลลัพธ์ซึ่งมีผลโดยตรงต่อคุณภาพของบริบทที่ส่งเข้าสู่โมเดลภาษา ส่วนเมตริกในขั้นตอน Generation ได้แก่ Accuracy, Precision, Recall และ F1-Score ถูกออกแบบให้สะท้อนทั้งความถูกต้องและความครอบคลุมของคำตอบภายใต้เกณฑ์ Strict/Lax

สำหรับการค้นคืนข้อมูลด้วยเวกเตอร์

$$Cosine\ similarity(E_a, E_b) = \frac{E_a \cdot E_b}{\|E_a\|\|E_b\|} \tag{10}$$

คำอธิบายตัวแปร

Ea แทน เวกเตอร์ของคำถาม (Query Vector)

Eb แทน เวกเตอร์ของเอกสาร (Document Vector)

2.18.8 Similarity การวัดความคล้ายคลึงของคำตอบที่ระบบสร้างขึ้นกับคำตอบอ้างอิง โดยใช้เกณฑ์ความคล้ายคลึง แม้รูปแบบถ้อยคำจะแตกต่างจากคำตอบอ้างอิง

$$Cosine\ similarity(E_a, E_b) = \frac{E_a \cdot E_b}{\|E_a\|\|E_b\|} = \frac{\sum_{i=1}^{n} E_{a,i} E_{b,i}}{\sqrt{\sum_{i=1}^{n} E_{a,i}^2} \sqrt{\sum_{i=1}^{n} E_{b,i}^2}} \quad (11)$$

คำอธิบายตัวแปร

Ea แทน เวกเตอร์ (Embedding) ของคำตอบที่โมเดลระบบสร้างขึ้น

Eb แทน เวกเตอร์ (Embedding) ของคำตอบที่ถูกต้องหรือคำตอบอ้างอิง

n แทน จำนวนมิติของเวกเตอร์

ค่าที่ได้จะอยู่ในช่วง $[-1,1]$ โดยค่าที่เข้าใกล้ 1 แสดงถึงระดับความคล้ายคลึงเชิงความหมายที่สูงมาก

เมตริกเหล่านี้ช่วยสะท้อนทั้งจุดแข็งและข้อจำกัดของระบบในแต่ละขั้นตอนของ RAG ทำให้สามารถวิเคราะห์และปรับปรุงประสิทธิภาพของระบบได้อย่างเป็นระบบ

ตารางที่ 2.1 แสดงงานวิจัยที่เกี่ยวข้อง

ลำดับ	ปี	วัตถุประสงค์	เทคนิค	ชุดข้อมูล	ผลลัพธ์
1	Zhu et al. [20]	เพื่อเพิ่มคุณภาพการดึงข้อมูลและการสร้างคำตอบของระบบ RAG โดยใช้โครงสร้างความรู้	Knowledge Graph-Guided RAG	HotpotQA, Wikipedia	ปรับปรุงความสอดคล้องของบริบทและเพิ่มความแม่นยำของคำตอบเมื่อเทียบกับ RAG แบบดั้งเดิม
2	Chang et al. [21]	เพื่อเพิ่มความแม่นยำของการคัดเลือกบริบทก่อนการสร้างคำตอบในระบบ RAG	Multi-Agent Filtering RAG (MAIN-RAG)	QA Benchmark หลายชุด	Retrieval Precision และ Answer Accuracy สูงขึ้นอย่างมีนัยสำคัญ
3	Chen et al. [22]	เพื่อนำ RAG มาประยุกต์ใช้กับ LLM เฉพาะโดเมนอุตสาหกรรม	Domain-Specific RAG Framework	ข้อมูลเอกสารอุตสาหกรรมภายในองค์กร	ลด Hallucination และเพิ่มความสอดคล้องของคำตอบกับเอกสารต้นทาง
4	Pradeesh et al. [23]	เพื่อสร้างคำถามแบบปรนัย (MCQ) จากเอกสารอัตโนมัติ	RAG-based Question Generation	เอกสาร PDF ทางการศึกษา	คุณภาพคำถามและความถูกต้องของคำตอบดีขึ้นกว่าวิธีไม่ใช้ Retrieval
5	Arya et al. [24]	เพื่อทบทวนและสรุปความก้าวหน้าของเทคนิค RAG	Survey & Comparative Analysis	งานวิจัย RAG หลายโดเมน	แสดงแนวโน้มและข้อจำกัดของ RAG โดยเฉพาะด้าน Retrieval และ Context Quality

จากงานวิจัยที่เกี่ยวข้องในตารางที่ 2.1 สามารถจำแนกงานวิจัยออกเป็น 3 กลุ่มหลัก ดังนี้

กลุ่มที่มุ่งเน้นความซับซ้อนของสถาปัตยกรรม งานวิจัยของ Zhu et al. [20] เสนอการผสาน Knowledge Graph เพื่อเพิ่มความแม่นยำของคำตอบ และ Chang et al. [21] พัฒนาแนวทาง multi-Agent เพื่อคัดกรองบริบทก่อนการสร้างคำตอบ งานในกลุ่มนี้มุ่งปรับปรุงกระบวนการในช่วง หลังการค้นคืน หรือ การจัดการบริบทก่อนการสร้างคำตอบ อย่างไรก็ตาม ยังขาดการศึกษาผลกระทบของ “คุณภาพข้อมูลดิบ” ที่มีสัญญาณรบกวน (noise) ตั้งแต่ขั้นตอนการนำเข้าข้อมูล โดยเฉพาะความคลาดเคลื่อนจาก OCR ซึ่งอาจทำให้ความหมายของข้อความผิดเพี้ยน ส่งผลต่อคุณภาพการฝังความหมาย (embedding) และประสิทธิภาพการค้นคืนในลำดับถัดไป

กลุ่มที่มุ่งเน้นการประยุกต์ใช้เฉพาะโดเมน งานวิจัยของ Chen et al. [22] ที่มุ่งเน้นในการลดอาการหลอน (hallucination) ในบริบทข้อมูลอุตสาหกรรม และ Pradeesh et al. [23] ที่ประยุกต์ RAG เพื่อสร้างข้อสอบจากเอกสาร PDF แม้งานกลุ่มนี้จะใช้เอกสารจริงในการทดสอบ แต่ยังขาดการออกแบบการทดลองแบบควบคุมตัวแปร เพื่อเปรียบเทียบเชิงปริมาณว่ารูปแบบแหล่งข้อมูลต้นฉบับ (เช่น clean text, PDF text และ PDF ที่ต้องผ่าน OCR) ส่งผลต่อคุณภาพการค้นคืนและผลลัพธ์ปลายทางของระบบแตกต่างกันมากน้อยเพียงใด

กลุ่มงานสำรวจและทบทวนวรรณกรรม งานวิจัยของ Arya et al. [24] ซึ่งสะท้อนภาพรวมความก้าวหน้าของเทคนิค RAG และชี้ให้เห็นความสำคัญของคุณภาพบริบท อย่างไรก็ตามจากการสำรวจพบว่ายังมีงานวิจัยเชิงประจักษ์ จำนวนน้อยที่ศึกษาความสัมพันธ์ระหว่าง “คุณภาพข้อมูลต้นทาง” และ “กลยุทธ์การแบ่งส่วนข้อมูล” อย่างเป็นระบบ โดยเฉพาะการวิเคราะห์ผลกระทบเชิงปริมาณภายใต้การควบคุมตัวแปรที่ชัดเจน

จากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่าแม้งานวิจัยก่อนหน้าจะมีการพัฒนาเทคนิคในแต่ละองค์ประกอบของระบบ Retrieval-Augmented Generation (RAG) อย่างต่อเนื่อง ไม่ว่าจะเป็นการพัฒนาโมเดลสำหรับการค้นคืนข้อมูล การแบ่งส่วนข้อความ หรือการจัดอันดับเอกสาร แต่ยังขาดการศึกษาที่พิจารณา “ผลกระทบร่วม” ของปัจจัยสำคัญหลายด้านภายใต้การควบคุมตัวแปรอย่างเป็นระบบ

โดยเฉพาะอย่างยิ่ง ผลกระทบร่วมระหว่างคุณภาพของแหล่งข้อมูลต้นทาง (Data Source Quality) และกลยุทธ์การแบ่งข้อมูล (Chunking Strategy) ซึ่งถือเป็นปัจจัยพื้นฐานที่ส่งผลโดยตรงต่อทั้งกระบวนการค้นคืนข้อมูลและการสร้างคำตอบ

นอกจากนี้ งานวิจัยส่วนใหญ่ยังมุ่งเน้นไปที่ภาษาอังกฤษเป็นหลัก ซึ่งมีโครงสร้างภาษาที่แตกต่างจากภาษาไทยอย่างมีนัยสำคัญ เช่น การเว้นวรรคและการตัดคำ ส่งผลให้ผลลัพธ์จากงานวิจัยต่างประเทศไม่สามารถนำมาประยุกต์ใช้กับภาษาไทยได้โดยตรง

ดังนั้น งานวิจัยนี้จึงมุ่งเติมเต็มช่องว่างดังกล่าว โดยการศึกษาเชิงระบบในบริบทของเอกสารภาษาไทย พร้อมทั้งควบคุมตัวแปรสำคัญให้สอดคล้องกันในทุกการทดลอง

ตารางที่ 2.2 การวิเคราะห์ช่องว่างของงานวิจัย

ประเด็นการศึกษา	งานวิจัยก่อนหน้า	จุดเด่น	ข้อจำกัด
Retrieval-Augmented Generation	งานวิจัยด้าน RAG หลายงาน	พัฒนาโมเดล Retrieval และ Generation เพื่อเพิ่มคุณภาพคำตอบ	ไม่ได้วิเคราะห์ผลกระทบของคุณภาพแหล่งข้อมูลต้นทางต่อระบบโดยตรง
Text Chunking	งานวิจัยด้านการแบ่งข้อความ	ศึกษาผลของขนาด Chunk ต่อประสิทธิภาพการค้นคืนข้อมูล	ไม่ได้ทดสอบร่วมกับระบบ RAG แบบครบกระบวนการ
Document Reranking	งานวิจัยด้าน Reranker	ปรับปรุงการจัดอันดับเอกสารให้มีความแม่นยำสูงขึ้น	ไม่ได้ประเมินผลกระทบต่อคุณภาพคำตอบของโมเดลภาษา
งานวิจัยนี้	การศึกษา RAG สำหรับเอกสารภาษาไทย	เปรียบเทียบแหล่งข้อมูล (Text, PDF, OCR) ศึกษาผลของ Chunk Size และใช้ Reranker ในการจัดอันดับใหม่ก่อนส่งไปขั้นตอนการสร้างคำตอบ (Generation)	มุ่งเน้นการทดลองกับชุดข้อมูลภายในองค์กร

จากการวิเคราะห์ข้างต้น สามารถสรุปได้ว่างานวิจัยก่อนหน้ายังมีข้อจำกัดในลักษณะของการศึกษาแบบแยกส่วน พิจารณาเฉพาะองค์ประกอบใดองค์ประกอบหนึ่งของระบบ RAG เช่น Retrieval, Chunking หรือ Reranking โดยไม่ได้วิเคราะห์ผลกระทบของปัจจัยร่วมกันแบบครบกระบวนการ (End-to-End) ดังนั้น งานวิจัยนี้จึงมีเป้าหมายเพื่อศึกษาผลกระทบร่วมของ คุณภาพแหล่งข้อมูลต้นทาง ขนาดของการแบ่งข้อมูล ต่อประสิทธิภาพของระบบ RAG สำหรับเอกสารภาษาไทย ภายใต้การควบคุมคุณภาพการค้นคืนด้วยโมเดลจัดอันดับใหม่ (Reranker) ในการพิจารณาข้อจำกัดของระบบเมื่อเจอกับสัญญาณรบกวน (Noise) ในสภาวะการใช้งานจริง ซึ่งเป็นบริบทที่ยังมีการศึกษาอย่างจำกัด ก่อนส่งไปขั้นตอนการสร้างคำตอบ (Generation)

บทที่ 3
วิธีการดำเนินงานวิจัย

บทนี้นำเสนอวิธีการดำเนินงานวิจัย เพื่ออธิบายขั้นตอนการทดลอง การควบคุมตัวแปร สถาปัตยกรรมของระบบ และการประเมินผลให้ชัดเจนและสามารถทำซ้ำได้ การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อพัฒนาและประเมินประสิทธิภาพของระบบ Retrieval-Augmented Generation (RAG) สำหรับการตอบคำถามจากเอกสารภายในองค์กร โดยระบบถูกออกแบบให้สามารถค้นคืนข้อมูลที่เกี่ยวข้องจากเอกสารต้นทาง และนำข้อมูลดังกล่าวมาใช้ร่วมกับโมเดลภาษาเพื่อสร้างคำตอบที่ถูกต้องและสอดคล้องกับบริบท

3.1 กรอบแนวคิดและภาพรวมของงานวิจัย

ระบบ RAG ที่ใช้ในการวิจัยนี้ถูกออกแบบให้ทำงานแบบประมวลผลภายในเครื่อง (Local Environment) เพื่อรักษาความปลอดภัยของข้อมูล โดยมีองค์ประกอบและขั้นตอนการทำงาน ดังนี้

3.1.1 การเตรียมข้อมูลเอกสาร (Data Preparation) รวบรวมเอกสารประกาศและคำสั่งภายในองค์กร

3.1.2 การสกัดข้อความจากเอกสาร (Text Extraction) ดึงข้อความจากเอกสารในรูปแบบ Text, สกัดจาก PDF โดยตรง และผ่านกระบวนการ OCR

3.1.3 การแบ่งส่วนข้อมูล (Chunking) หั่นข้อความออกเป็นส่วนย่อยตามขนาดที่กำหนด

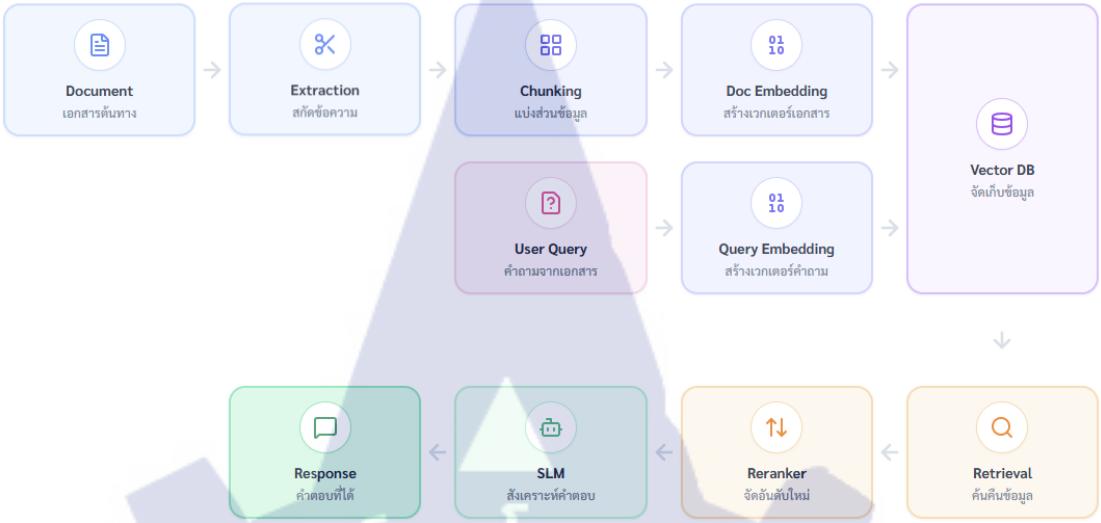
3.1.4 การสร้างเวกเตอร์ (Embedding Generation) แปลงชิ้นส่วนข้อความเป็นเวกเตอร์เชิงความหมายด้วย Embedding Model

3.1.5 การจัดเก็บข้อมูล (Vector Database) นำเวกเตอร์ไปจัดเก็บในฐานข้อมูลเพื่อรอการสืบค้น

3.1.6 การค้นคืนข้อมูล (Retrieval) เมื่อผู้ใช้ถามคำถาม ระบบจะดึงข้อมูลบริบทที่เกี่ยวข้องมากที่สุดเบื้องต้นออกมา

3.1.7 การจัดอันดับข้อมูล (Reranking) ใช้โมเดลจัดอันดับซ้ำเพื่อคัดกรองเอกสารที่เกี่ยวข้องกันเชิงลึกที่สุดขึ้นสู่อันดับแรก

3.1.8 การสร้างคำตอบ (Answer Generation) ส่งบริบทที่เกี่ยวข้องสูงสุดไปให้โมเดลภาษาขนาดเล็กสังเคราะห์เป็นคำตอบ



รูปที่ 3.1 สถาปัตยกรรมของระบบ Retrieval-Augmented Generation

รูปที่ 3.1 แสดงโครงสร้างของระบบ RAG ซึ่งเริ่มจากการเตรียมข้อมูลและการแบ่งส่วนข้อความเป็นหน่วยย่อย (Chunking) จากนั้นสร้างเวกเตอร์ฝังความหมาย (embedding) และจัดเก็บในฐานข้อมูลเวกเตอร์ เมื่อรับคำถาม ระบบจะทำการค้นคืนผลลัพธ์เบื้องต้น (Top-N) จากนั้นจัดอันดับซ้ำด้วย reranker เพื่อคัดเลือกผลลัพธ์ Top-K และส่งบริบทที่เกี่ยวข้องเข้าสู่โมเดลภาษา (LLM) เพื่อสร้างคำตอบ

3.2 การเตรียมข้อมูลและกลยุทธ์การแบ่งส่วนข้อมูล

ในการทดลองครั้งนี้ใช้เอกสารภายในองค์กรจำนวน 197 ฉบับ ซึ่งอยู่ในรูปแบบไฟล์ PDF, OCR และ TXT โดยจากเอกสารต้นทางเดียวกัน ได้สร้างแหล่งข้อมูลต้นทาง 3 ประเภท ดังตารางที่ 3.1

ตารางที่ 3.1 ประเภทเอกสารที่ใช้ในการทดลอง

แหล่งข้อมูล	วิธีการได้มา	ลักษณะข้อมูล
Text	คัดลอกและตรวจสอบด้วยมือจาก PDF ออกมาเป็นข้อความ Text	ข้อมูลสะอาด มีสัญญาณรบกวนน้อย
PDF Text	สกัดข้อความจาก PDF โดยตรง	อาจมีความคลาดเคลื่อนจากโครงสร้างเอกสาร/สัญลักษณ์พิเศษ
OCR	แปลง PDF เป็นภาพและสกัดข้อความด้วย OCR	มี noise จากการอ่านอักขระผิด/การจัดวางหน้าเอกสาร

3.2.1 การสร้างชุดคำถามและคำตอบอ้างอิง

สำหรับการประเมินระบบ ได้จัดเตรียมชุดคำถามจำนวน 50 คำถาม พร้อมเฉลยใช้ในการทดสอบประสิทธิภาพ

การสร้างชุดคำถามและคำตอบอ้างอิง (Ground Truth) เป็นขั้นตอนสำคัญในการประเมินประสิทธิภาพของระบบ โดยในงานวิจัยนี้ ผู้วิจัยได้ดำเนินการสร้างชุดคำถามจากเอกสารต้นทางประเภทประกาศและระเบียบปฏิบัติขององค์กร

กระบวนการเริ่มจากการคัดเลือกเอกสารที่มีความสำคัญและมีเนื้อหาหลากหลาย จากนั้นจึงออกแบบคำถามให้ครอบคลุมทั้งในระดับข้อเท็จจริง (Factoid Questions) และคำถามเชิงตีความ (Interpretive Questions) เพื่อสะท้อนลักษณะการใช้งานจริง

คำตอบอ้างอิงถูกกำหนดโดยอิงจากข้อความในเอกสารต้นทางโดยตรง พร้อมทั้งมีการตรวจสอบความถูกต้องโดยผู้วิจัย เพื่อให้มั่นใจว่าคำตอบมีความสอดคล้องกับบริบทและไม่มีความคลาดเคลื่อน

นอกจากนี้ ได้มีการกำหนดเกณฑ์การประเมินแบบยืดหยุ่น (Lax Matching) เพื่อรองรับความแตกต่างของรูปแบบภาษาในการตอบของโมเดลภาษา ซึ่งช่วยให้การประเมินสะท้อนคุณภาพของคำตอบได้เหมาะสมยิ่งขึ้น

เอกสารจากแต่ละแหล่งข้อมูลถูกแบ่งเป็นหน่วยย่อย (Chunks) โดยกำหนดขนาด Chunk เป็นหน่วย tokens จำนวน 3 ระดับ ได้แก่ 250, 500 และ 1,000 tokens ทั้งนี้การนับความยาวของข้อความใช้ tokenizer ของโมเดลฝังความหมาย BAAI/bge-m3 เพื่อให้ขนาด Chunk สอดคล้องกับการประมวลผลในขั้นตอน embedding และ retrieval

เพื่อรักษาความต่อเนื่องของบริบทระหว่างรอยต่อของแต่ละ Chunk งานวิจัยกำหนด Chunk overlap = 20% ของขนาด Chunk ซึ่งเท่ากับ 50, 100 และ 200 tokens สำหรับขนาด 250, 500 และ 1,000 tokens ตามลำดับ ส่งผลให้เกิดชุดทดลองทั้งหมด 9 รูปแบบ (3 แหล่งข้อมูล × 3 ขนาด Chunk)

ตารางที่ 3.2 แสดงชุดการทดลองตาม Source Quality และ Chunk Size

Source Quality	Chunk Size (tokens)	Chunk Overlap (tokens)	รหัสชุดข้อมูล
Text	250	50	T-250
Text	500	100	T-500
Text	1000	200	T-1000
PDF Text	250	50	P-250
PDF Text	500	100	P-500
PDF Text	1000	200	P-1000
OCR	250	50	O-250
OCR	500	100	O-500
OCR	1000	200	O-1000

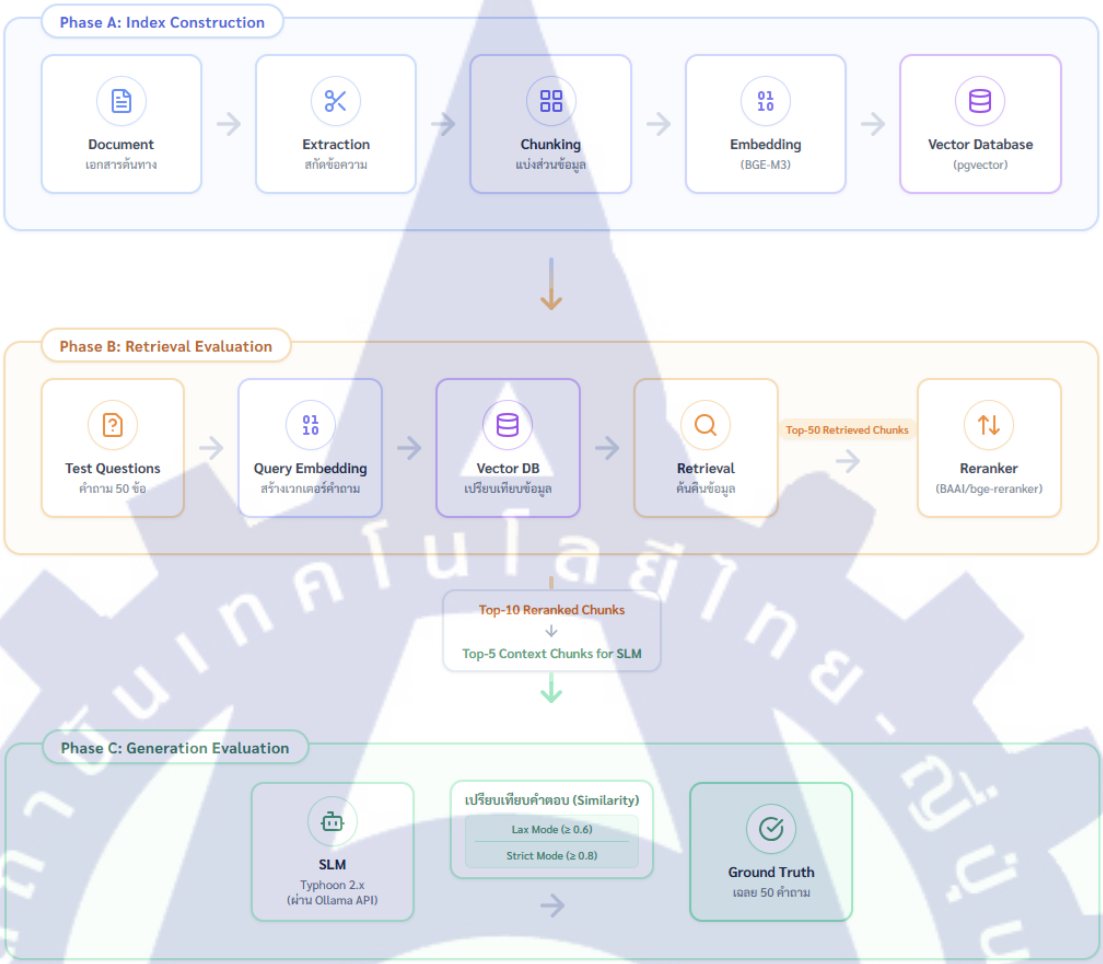
3.3 การออกแบบการทดลองและตัวแปรที่ศึกษา

งานวิจัยนี้แบ่งการทดลองออกเป็น 3 ขั้นตอนหลัก (Phases) เพื่อให้สอดคล้องกับการทำงานของระบบ RAG ได้แก่

Test A: การสร้างดัชนีข้อมูล (Indexing Process)

Test B: การประเมินการค้นคืนข้อมูล (Retrieval Evaluation)

Test C: การประเมินการสร้างคำตอบ (Generation Evaluation)



รูปที่ 3.2 ขั้นตอนการดำเนินการทดลองของระบบ RAG

โดยมีรายละเอียดของตัวแปรที่ใช้ในการศึกษา ดังนี้

1. ตัวแปรอิสระ (Independent Variables)

คุณภาพของแหล่งข้อมูลต้นทาง ประกอบด้วย Text, PDF Text, OCR

ขนาดของหน่วยข้อมูล ประกอบด้วย 250, 500, 1000 tokens

2. ตัวแปรตาม (Dependent Variables)

ประสิทธิภาพของการสร้างดัชนี (Indexing Performance - Test A): วัดจาก จำนวน Chunks ทั้งหมด (Total Chunks) และระยะเวลาในการสร้าง Embedding และจัดเก็บดัชนี

คุณภาพของการค้นคืนข้อมูล (Retrieval Quality - Test B): วัดความสามารถของระบบในการค้นหาเอกสารที่เกี่ยวข้อง ผ่านตัวชี้วัด อัตราการค้นพบ (Hit Rate@K), ค่าเฉลี่ยอันดับส่วนกลับ (MRR@10) และค่าคุณภาพการจัดอันดับ (nDCG@10)

คุณภาพของการสร้างคำตอบ (Generation Quality - Test C): ประเมินความถูกต้องของคำตอบจากโมเดลภาษาผ่านค่า F1-Score ภายใต้เกณฑ์ 2 ระดับ คือ Lax Condition (Threshold ≥ 0.6) และ Strict Condition (Threshold ≥ 0.8)

3. ตัวแปรควบคุม (Controlled Variables)

เพื่อให้ผลการทดลองมีความน่าเชื่อถือ ตัวแปรต่อไปนี้ถูกควบคุมให้คงที่ในทุกการทดลอง

ชุดเอกสารจำนวน 197 เอกสาร และชุดคำถามจำนวน 50 ข้อ

Embedding Model: BAAI/bge-m3

Reranker Model: BAAI/bge-reranker-v2-m3

Small Language Model (SLM) 3 รุ่น ได้แก่ Typhoon2.1-Gemma3-4B, Llama3.2-Typhoon2-3B และ Typhoon2.5-Qwen3-4B

โดยในการทดลองนี้ ได้กำหนดให้ Reranker เป็นองค์ประกอบมาตรฐานที่ใช้ในทุกเงื่อนไขการทดลอง เพื่อควบคุมตัวแปรด้านคุณภาพของการจัดอันดับเอกสาร และมุ่งเน้นศึกษาผลกระทบของคุณภาพข้อมูลต้นทางและขนาด Chunk เป็นหลัก

3.4 วิธีการประเมินผล

3.4.1 การวัดผลด้านการค้นคืนข้อมูล (Retrieval Metrics - Test B)

1. อัตราการค้นพบ (Hit Rate@K)

$$HitRate@K = \frac{Number\ of\ queries\ with\ at\ least\ one\ relevant\ result\ in\ top\ K}{Total\ number\ of\ queries} \quad (3.1)$$

2. ค่าเฉลี่ยอันดับส่วนกลับ (Mean Reciprocal Rank: MRR)

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \quad (3.2)$$

3. ค่าคุณภาพการจัดอันดับแบบปรับมาตรฐาน (Normalized Discounted Cumulative Gain: NDCG@K)

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{3.3}$$

4. ค่าคุณภาพการจัดอันดับ (Discounted Cumulative Gain: DCG@K)

$$DCG@K = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)} \tag{3.4}$$

5. ค่าการจัดอันดับแบบอุดมคติ (Ideal Discounted Cumulative Gain: IDCG@K)

$$IDCG@K = \sum_{i=1}^{k} \frac{\left(rel_i^{ideal}\right)}{log_2(i+1)} \tag{3.5}$$

3.4.2 การวัดผลด้านการสร้างคำตอบ (Generation Metrics - Test C)

ประเมินความคล้ายคลึงระหว่างคำตอบของโมเดลและเฉลยอ้างอิงผ่าน Cosine Similarity

$$Cosine\ similarity(E_a, E_b) = \frac{E_a \cdot E_b}{\|E_a\|\|E_b\|} = \frac{\sum_{i=1}^{n} E_{a,i}E_{b,i}}{\sqrt{\sum_{i=1}^{n} E_{a,i}^2}\sqrt{\sum_{i=1}^{n} E_{b,i}^2}} \tag{3.6}$$

การตัดสินความถูกต้องของคำตอบ (Accuracy และ F1-Score) อิงตามเกณฑ์ (Threshold) 2 ระดับ ได้แก่

1. เกณฑ์แบบยืดหยุ่น (Lax Mode) กำหนด Threshold ≥ 0.6 เพื่อยอมรับการเรียบเรียงประโยคใหม่ หากจับใจความสำคัญได้ตรงกัน

2. เกณฑ์แบบเข้มงวด (Strict Mode) กำหนด Threshold ≥ 0.8 เพื่อวัดความแม่นยำเชิงข้อเท็จจริง

3.5 สภาพแวดล้อมในการทดลอง

3.5.1 Hardware

CPU intel Core i5-14400

GPU NVIDIA GeForce RTX4060 (8 GB)

RAM 16 GB

Storage 500 GB

OS Windows 11

3.5.2 Software

Python

Visual Studio Code (VS Code) สำหรับการเขียนโค้ดโปรแกรม

Ollama ใช้ประมวลผล Small Language Model แบบ local

PyMuPDF (fitz) ใช้แปลงหน้า PDF เป็นภาพก่อนทำ OCR

Tesseract-OCR ใช้สกัดข้อความจากภาพ (OCR)

บทที่ 4
ผลการทดลองและการวิเคราะห์ผล

จากการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อนำมาเป็นพื้นฐานในการพัฒนาระบบ Retrieval-Augmented Generation (RAG) สำหรับการตอบคำถามจากเอกสารภายในองค์กร ในการวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการทดลองระบบ โดยมีวัตถุประสงค์เพื่อศึกษาผลกระทบของคุณภาพแหล่งข้อมูลต้นทาง (Source Quality) และกลยุทธ์การแบ่งส่วนข้อมูล (Chunking Strategy) ต่อประสิทธิภาพของระบบ RAG

การทดลองนี้ใช้เอกสารประกาศและคำสั่งของสถาบันเทคโนโลยีไทย-ญี่ปุ่น จำนวน 197 ฉบับ และชุดคำถามทดสอบจำนวน 50 คำถาม โดยผลการทดลองถูกนำมาวิเคราะห์และเปรียบเทียบประสิทธิภาพใน 3 ด้านหลัก ได้แก่ ต้นทุนการจัดทำดัชนี (Indexing Cost), การค้นคืนข้อมูล (Retrieval Quality) และการสร้างคำตอบ (Generation Quality) โดยมีรายละเอียดของผลการทดลองนำเสนอในหัวข้อดังต่อไปนี้

4.1 ผลการประเมินต้นทุนการจัดทำดัชนี (Test A Indexing Cost)

การทดลองในส่วนนี้มีวัตถุประสงค์เพื่อประเมินต้นทุนทรัพยากรที่ใช้ในการเตรียมข้อมูล (Preprocessing) และการแปลงข้อความเป็นเวกเตอร์ (Embedding) โดยพิจารณาจากตัวชี้วัดหลัก ได้แก่ จำนวนชิ้นส่วนข้อมูลทั้งหมด (Total Chunks) และเวลาที่ใช้ในการประมวลผล (Elapsed Seconds) ซึ่งสะท้อนถึงขนาดพื้นที่ที่ต้องใช้ในฐานข้อมูลเวกเตอร์และภาระการประมวลผลของระบบ ผลการทดลองแสดงดังตารางที่ 4.1

ตารางที่ 4.1 ผลการเปรียบเทียบต้นทุนการจัดทำดัชนีจำแนกตามคุณภาพแหล่งข้อมูลและขนาดการแบ่งส่วนข้อมูล (Chunk Size)

แหล่งข้อมูลต้นทาง	ขนาด Chunk (Tokens)	จำนวนเอกสาร (ฉบับ)	จำนวน Chunk รวม	ค่าเฉลี่ย Chunk ต่อเอกสาร	เวลาประมวลผลรวม(วินาที)	เวลาเฉลี่ยต่อ Chunk (วินาที)
Text	250	197	681	3.460	284.040	0.417
	500	197	358	1.820	260.040	0.726
	1000	197	235	1.200	239.790	1.020
PDF Text	250	197	3,568	18.110	393.430	0.110
	500	197	1,939	9.840	413.770	0.213
	1000	197	877	4.450	379.820	0.433
OCR Text	250	197	3,782	19.200	704.660	0.186
	500	197	2,002	10.160	567.440	0.283
	1000	197	973	4.940	477.120	0.490

4.1.1 การวิเคราะห์และอภิปรายผลการทดลอง Test A

จากตารางที่ 4.1 สามารถวิเคราะห์ผลกระทบของคุณภาพแหล่งข้อมูลและกลยุทธ์การแบ่งส่วนข้อมูลที่มีต่อต้นทุนการจัดทำดัชนีได้ดังนี้

1. ผลกระทบของคุณภาพข้อมูล (Source Quality) ต่อขนาดฐานข้อมูล

พบว่าคุณภาพของข้อมูลมีความสัมพันธ์โดยตรงกับปริมาณชิ้นส่วนข้อมูล (Total Chunks) ที่เกิดขึ้น ข้อมูลประเภท Text ซึ่งผ่านการทำความสะอาดมาแล้ว มีความกระชับและไม่มีสัญลักษณ์ขยะปะปน ทำให้เกิดจำนวน Chunk รวมน้อยที่สุด (เพียง 681 Chunks ที่ขนาด 250 tokens) ในทางตรงกันข้าม ข้อมูลประเภท OCR ซึ่งมีสัญญาณรบกวน (Noise) เช่น การตัดคำที่ผิดพลาด การเว้นวรรคที่ไม่เป็นระเบียบ หรือตัวอักษรขยะ ส่งผลให้กระบวนการ Chunking ต้องตัดแบ่งข้อความบ่อยขึ้น ทำให้ฐานข้อมูลมีขนาดขยายตัวเพิ่มขึ้นอย่างมาก โดยสร้างจำนวน Chunk สูงถึง 3,782 Chunks (ที่ขนาด 250 tokens) ซึ่งมากกว่า Text ถึง 5.5 เท่า

2. ผลกระทบของคุณภาพข้อมูลต่อเวลาประมวลผล (Embedding Time)

ระยะเวลาที่ใช้ในการประมวลผลแปรผันตามจำนวน Chunk ที่ระบบต้องนำไปแปลงเป็นเวกเตอร์ แหล่งข้อมูล OCR (ขนาด 250 tokens) ใช้เวลาประมวลผลรวมสูงสุดถึง 704.660 วินาที ในขณะที่แหล่งข้อมูล Text (ขนาด 250 tokens) ใช้เวลาเพียง 284.040 วินาที อย่างไรก็ตาม เมื่อพิจารณาค่าเฉลี่ยเวลาต่อ 1 Chunk จะพบว่า Text ใช้เวลาเฉลี่ยสูงกว่า OCR

เล็กน้อย ซึ่งเป็นผลมาจากความหนาแน่นของตัวอักษรที่อัดแน่นและสมบูรณ์ใน 1 Chunk ของ Text เมื่อเทียบกับ OCR ที่มีชิ้นส่วนข้อมูลขนาดสั้น (Short Chunks) ปะปนอยู่เป็นจำนวนมาก

3. ประสิทธิภาพของกลยุทธ์การแบ่งขนาด (Chunking Strategy)

เมื่อพิจารณาในทุกกลุ่มแหล่งข้อมูล การขยายขนาด Chunk Size จาก 250 tokens เป็น 1,000 tokens ช่วยลดภาระทรัพยากร (Overhead) ได้อย่างชัดเจน ตัวอย่างเช่น ในชุดข้อมูล OCR การปรับขนาดเป็น 1,000 tokens สามารถลดจำนวน Chunk รวมลงจาก 3,782 เหลือเพียง 973 Chunks และลดเวลาประมวลผลรวมจาก 704.660 วินาที เหลือ 477.120 วินาที ในมุมมองของการนำไปใช้งานจริงกับทรัพยากรขององค์กร การใช้ข้อมูลที่สะอาด (Text) หรือการใช้ Chunk ขนาดใหญ่ (1,000 tokens) เป็นกลยุทธ์ที่มีประสิทธิภาพสูงสุดในการควบคุมขนาดของฐานข้อมูลเวกเตอร์ (Vector Database) และช่วยลดเวลาในการจัดทำดัชนีได้อย่างมีนัยสำคัญ

4.2 ผลการประเมินคุณภาพการค้นคืน (Test B Retrieval Quality)

การทดลองในส่วนนี้มีวัตถุประสงค์เพื่อประเมินประสิทธิภาพของระบบในการค้นคืนชิ้นส่วนข้อมูล (Chunks) ที่เกี่ยวข้องกับคำถาม และผ่านการจัดอันดับใหม่ (Reranking) ด้วยโมเดล BAAI/bge-reranker-v2-m3 โดยพิจารณาผลลัพธ์ใน 10 อันดับแรก (Top-10) ตัวชี้วัดที่ใช้ ได้แก่ อัตราการค้นพบ (Hit Rate@10) และคุณภาพการจัดอันดับ (MRR@10 และ NDCG@10) ผลการทดลองแสดงดังตารางที่ 4.2

ตารางที่ 4.2 ผลการเปรียบเทียบประสิทธิภาพการค้นคืนข้อมูล (Retrieval Quality) ในระดับ Top-10

แหล่งข้อมูลต้นทาง	ขนาด Chunk	HitRate@10	MRR@10	NDCG@10
Text (Clean)	250	0.940	0.862	0.844
	500	0.940	0.853	0.857
	1000	0.940	0.837	0.899
PDF Text	250	0.820	0.621	0.669
	500	0.900	0.710	0.747
	1000	0.940	0.755	0.783
OCR Text	250	0.880	0.738	0.752
	500	0.920	0.773	0.778
	1000	0.940	0.827	0.811

แม้ว่าการเพิ่มขนาดของ Chunk จะช่วยรักษาบริบทของข้อความได้ดียิ่งขึ้น แต่ผลการทดลองในตารางที่ 4.2 พบว่า ค่า Mean Reciprocal Rank (MRR) ของข้อมูลประเภท Text กลับมีแนวโน้มลดลงเมื่อใช้ Chunk ขนาดใหญ่ขึ้น

ปรากฏการณ์นี้สามารถอธิบายได้จากลักษณะของการจัดอันดับผลลัพธ์ โดย MRR ให้ความสำคัญกับตำแหน่งของเอกสารที่ถูกต้องลำดับแรก เมื่อ Chunk มีขนาดใหญ่ขึ้น แต่ละหน่วยข้อมูลจะครอบคลุมเนื้อหาที่กว้างขึ้น ส่งผลให้ความเฉพาะเจาะจงของเนื้อหาลดลง (Reduced Granularity)

ดังนั้น แม้ระบบจะยังสามารถค้นพบข้อมูลที่ถูกต้องอยู่ในผลลัพธ์ แต่ตำแหน่งอันดับแรกอาจถูกแทนที่ด้วย Chunk อื่นที่มีเนื้อหาครอบคลุมแต่ไม่ตรงประเด็นที่สุด ส่งผลให้ค่า MRR ลดลง

ข้อค้นพบนี้สะท้อนให้เห็นถึง trade-off ระหว่าง “ความครบถ้วนของบริบท” และ “ความแม่นยำในการจัดอันดับลำดับต้น” ซึ่งเป็นปัจจัยสำคัญในการออกแบบระบบ RAG

4.2.1 การวิเคราะห์และอภิปรายผลการทดลอง Test B

จากตารางที่ 4.2 สามารถวิเคราะห์แนวโน้มและผลกระทบของตัวแปรที่มีต่อคุณภาพการค้นคืนได้ดังนี้

1. ประสิทธิภาพของข้อมูลคุณภาพสูง (Text)

แหล่งข้อมูลประเภท Text (Clean Data) มีประสิทธิภาพเชิงการจัดอันดับสูงที่สุดในการทดลอง โดยเฉพาะที่ขนาด 1,000 tokens ซึ่งให้ค่า NDCG@10 สูงถึง 0.899 และมีค่า MRR@10 สูงถึง 0.837 ตัวเลขนี้สะท้อนให้เห็นว่า ข้อมูลที่ไม่มีสัญญาณรบกวน (Noise-free) จะช่วยให้โมเดลสามารถเข้าใจบริบทเชิงความหมาย (Semantic) ได้อย่างแม่นยำ ส่งผลให้ระบบสามารถดึงชิ้นส่วนข้อมูลที่ถูกต้องที่สุดขึ้นมาจัดไว้ในอันดับที่ 1 ได้อย่างสม่ำเสมอ ซึ่งเป็นปัจจัยวิกฤตที่ทำให้โมเดลภาษาได้รับข้อมูลที่ถูกต้องไปใช้ในการสร้างคำตอบ

2. ความย้อนแย้งของการค้นคืนในข้อมูล OCR (The Retrieval Paradox)

ผลการทดลองเผยให้เห็นปรากฏการณ์ที่น่าสนใจในชุดข้อมูล OCR ขนาด Chunk 250 tokens เมื่อพิจารณาคุณภาพการจัดอันดับกลับพบว่า MRR@10 (0.738) และ NDCG@10 (0.752) ต่ำกว่าข้อมูลประเภท Text อย่างชัดเจน ปรากฏการณ์นี้เรียกว่า ความย้อนแย้งของการค้นคืน กล่าวคือ โมเดลสามารถกวาดข้อมูลที่มีคำคล้ายคลึงกันจากกระบวนการ OCR เข้ามาใน Top-10 ได้จำนวนมาก แต่ข้อมูลเหล่านั้นมักมีสัญญาณรบกวน เช่น ตัวอักษรเพี้ยน หรือการเว้นวรรคผิด ปะปนอยู่ ส่งผลให้ชิ้นส่วนข้อมูลที่ถูกต้องและสมบูรณ์ที่สุดถูกฉุดให้ตกไปอยู่ในอันดับรองลงมา

3. ผลกระทบของขนาดการแบ่งส่วนข้อมูล (Chunk Size Impact)

เมื่อพิจารณาค่า Hit Rate@10 ซึ่งเป็นด่านแรกที่วัดว่าระบบหาข้อมูลเจอหรือไม่ พบว่าการใช้ขนาด Chunk ที่ 1,000 tokens ส่งผลให้ระบบมีค่า Hit Rate@10 สูงสุดที่ระดับ 0.940 (94%) เท่ากันในทุกแหล่งข้อมูล (ทั้ง Text, PDF และ OCR) ผลลัพธ์นี้สอดคล้องกับโครงสร้างของเอกสารคำสั่งและประกาศสถาบันที่เป็นภาษาไทย ซึ่งมักมีการเกริ่นนำหรือมีบริบทที่ต้องอ่านต่อเนื่องกันยาว การแบ่ง Chunk ขนาดเล็กเกินไป (250 tokens) ทำให้บริบทถูกตัดขาด (Context Fragmentation) และสูญเสียความหมายดั้งเดิม ส่งผลให้ค่า Hit Rate และ NDCG ในกลุ่ม 250 tokens ต่ำกว่า 1,000 tokens ในทุกรูปแบบข้อมูล

สรุปผลด้านการค้นคืน (Test B) เพื่อเพิ่มประสิทธิภาพการค้นคืนในระบบ RAG สำหรับภาษาไทย การใช้ข้อมูลที่สะอาด (Text) ร่วมกับการใช้ขนาด Chunk 1,000 tokens เป็นรูปแบบที่ให้คุณภาพการจัดอันดับ (Ranking Quality) เสถียรและแม่นยำที่สุด อย่างไรก็ตาม ในกรณีที่ต้องปฏิบัติงานกับข้อมูลไฟล์สแกน (OCR) อย่างหลีกเลี่ยงไม่ได้ การขยายขนาด Chunk เป็น 1,000 tokens จะช่วยยกระดับ MRR@10 ให้กลับมาสูงถึง 0.827 ซึ่งช่วยบรรเทาผลกระทบจากสัญญาณรบกวน และเพิ่มโอกาสให้ LLM ได้รับข้อมูลที่ถูกต้องเพื่อนำไปสร้างคำตอบในขั้นตอนต่อไป

4.3 ผลการประเมินคุณภาพการสร้างคำตอบ (Test C Generation Quality)

ในการประเมินผลการสร้างคำตอบของโมเดล TEST C งานวิจัยนี้ได้กำหนดเกณฑ์การวัดความคล้ายคลึงเชิงความหมาย (Semantic Similarity) โดยใช้ค่า cosine similarity ระหว่างคำตอบที่โมเดลสร้างขึ้นและคำตอบอ้างอิง เพื่อใช้เป็นตัวชี้วัดความถูกต้องของคำตอบในระดับความหมาย

โดยกำหนด threshold ออกเป็น 2 ระดับ ได้แก่

- Lax Threshold (Similarity ≥ 0.6) สำหรับพิจารณาความครอบคลุมของใจความสำคัญ
- Strict Threshold (Similarity ≥ 0.8) สำหรับพิจารณาความถูกต้องเชิงข้อเท็จจริง

เกณฑ์ดังกล่าวถูกนำมาใช้ในการคำนวณค่า Accuracy, Precision, Recall และ F1-Score ที่นำเสนอในผลการทดลอง

การทดลองในขั้นตอนนี้เป็นการประเมินประสิทธิภาพปลายทาง (End-to-End Performance) ของระบบ RAG โดยนำชิ้นส่วนข้อมูลที่ผ่านการจัดอันดับใหม่ (Top-5 Context Chunks) จาก Test B มาเป็นบริบทตั้งต้นให้โมเดลภาษาขนาดเล็ก (SLMs) สังเคราะห์คำตอบ การประเมินผลใช้ชุดคำถาม 50 ข้อเทียบกับคำตอบอ้างอิง (Ground Truth) ภายใต้เกณฑ์ความคล้ายคลึง 2 ระดับ ได้แก่ เกณฑ์แบบยืดหยุ่น (Lax Threshold ≥ 0.6) ซึ่งวัดความครอบคลุมของใจความสำคัญ

และ เกณฑ์แบบเข้มงวด (Strict Threshold ≥ 0.8) ซึ่งคาดหวังความถูกต้องแม่นยำสูงเชิงข้อเท็จจริง โดยทดสอบกับโมเดลตระกูล Typhoon 2.x จำนวน 3 รุ่น ผลการทดลองแสดงดังตารางต่อไปนี้

4.3.1 ผลการประเมินโมเดล Typhoon2.1-Gemma3-4B

โมเดลรุ่นนี้แสดงประสิทธิภาพสูงที่สุดในการทดลอง โดยสามารถรักษารูปแบบประโยคและการจับใจความสำคัญของภาษาไทยได้ดี ผลลัพธ์แสดงดังตารางที่ 4.3

ตารางที่ 4.3 ผลการประเมินคุณภาพการสร้างคำตอบของโมเดล Typhoon2.1-Gemma3-4B

แหล่งข้อมูลต้นทาง	ขนาด Chunk	Accuracy (Lax)	F1-Score (Lax)	Accuracy (Strict)	F1-Score (Strict)
Text (Clean)	250	0.780	0.876	0.580	0.734
	500	0.840	0.913	0.640	0.781
	1000	0.860	0.925	0.660	0.795
PDF Text	250	0.620	0.765	0.480	0.649
	500	0.560	0.718	0.420	0.592
	1000	0.700	0.824	0.520	0.684
OCR Text	250	0.700	0.824	0.480	0.649
	500	0.700	0.824	0.460	0.630
	1000	0.580	0.734	0.360	0.529

4.3.2 ผลการประเมินโมเดล Llama3.2-Typhoon2-3B-Instruct

โมเดลรุ่นนี้มีความโดดเด่นในด้านความเสถียรและสามารถจัดการกับข้อมูลที่มีสัญญาณรบกวน (Noise) ได้ในระดับที่น่าพอใจ ผลลัพธ์แสดงดังตารางที่ 4.4

ตารางที่ 4.4 ผลการประเมินคุณภาพการสร้างคำตอบของโมเดล Llama3.2-Typhoon2-3B-Instruct

แหล่งข้อมูลต้นทาง	**ขนาด Chunk**	**Accuracy (Lax)**	**F1-Score (Lax)**	**Accuracy (Strict)**	**F1-Score (Strict)**
Text (Clean)	250	0.720	0.837	0.500	0.667
	500	0.680	0.810	0.480	0.649
	1000	0.700	0.824	0.500	0.668
PDF Text	250	0.500	0.667	0.280	0.438
	500	0.500	0.667	0.320	0.485
	1000	0.540	0.701	0.360	0.529
OCR Text	250	0.600	0.750	0.340	0.508
	500	0.560	0.718	0.340	0.508
	1000	0.520	0.684	0.260	0.413

4.3.3 ผลการประเมินโมเดล Typhoon2.5-Qwen3-4B

โมเดลรุ่นนี้มีแนวโน้มที่จะเรียบเรียงประโยคใหม่ (Paraphrasing) ค่อนข้างมาก ซึ่งส่งผลให้คะแนนความคล้ายคลึงแบบ Strict ลดลงอย่างมีนัยสำคัญ ผลลัพธ์แสดงดังตารางที่ 4.5

ตารางที่ 4.5 ผลการประเมินคุณภาพการสร้างคำตอบของโมเดล Typhoon2.5-Qwen3-4B

แหล่งข้อมูลต้นทาง	**ขนาด Chunk**	**Accuracy (Lax)**	**F1-Score (Lax)**	**Accuracy (Strict)**	**F1-Score (Strict)**
Text (Clean)	250	0.480	0.649	0.240	0.387
	500	0.460	0.630	0.220	0.361
	1000	0.520	0.684	0.340	0.508
PDF Text	250	0.300	0.462	0.180	0.305
	500	0.300	0.462	0.180	0.305
	1000	0.320	0.485	0.220	0.361
OCR Text	250	0.320	0.485	0.140	0.246
	500	0.360	0.529	0.140	0.246
	1000	0.360	0.529	0.180	0.305

4.3.4 การวิเคราะห์และอภิปรายผลการทดลอง Test C

จากตารางผลการประเมินของโมเดลทั้ง 3 รุ่น สามารถวิเคราะห์ความสัมพันธ์เชิงสาเหตุระหว่างคุณภาพข้อมูลต้นทาง ขนาดการแบ่งข้อมูล และความแม่นยำของคำตอบ ได้ดังนี้

1. คุณภาพข้อมูลคือปัจจัยชี้ขาด (Source Quality Impact)

ผลการทดลองแสดงให้เห็นชัดเจนว่า ชุดข้อมูล Text (Clean Data) ให้คะแนนความถูกต้องสูงที่สุดในทุกโมเดล โดยเฉพาะเมื่อใช้ร่วมกับโมเดล Typhoon2.1-Gemma3-4B ที่ขนาด Chunk 1,000 tokens ซึ่งทำคะแนน Accuracy (Lax) ได้สูงสุดถึง 0.860 (86%) และ F1-Score (Lax) ที่ 0.925 ในทางกลับกัน เมื่อพิจารณาผลลัพธ์ของชุดข้อมูล OCR แม้ใน Test B จะพบว่า OCR สามารถดึงชิ้นส่วนข้อมูลที่เกี่ยวข้องขึ้นมาได้จำนวนมาก แต่เมื่อเข้าสู่กระบวนการสร้างคำตอบ คะแนน Accuracy กลับลดลงอย่างมีนัยสำคัญ (เช่น Gemma3 ลดลงเหลือ 0.580 ในกลุ่ม OCR c1000) ข้อมูลเชิงประจักษ์นี้ช่วยยืนยันการค้นพบเรื่อง "ความย้อนแย้งของการค้นคืน (Retrieval Paradox)" ซึ่งชี้ให้เห็นว่า ความหนาแน่นในการค้นคืนไม่สามารถชดเชยคุณภาพของข้อมูลต้นทางที่เต็มไปด้วยสัญญาณรบกวนได้

2. ความอ่อนไหวต่อสัญญาณรบกวนของโมเดลภาษา (LLM Noise Sensitivity)

เมื่อบริบทที่ส่งให้ LLM ประมวลผลมีสัญญาณรบกวนปะปนอยู่ เช่น ความคลาดเคลื่อนของตัวเลข (Numeric Distortion) จากการทำ OCR หรือตัวอักษรที่ผิดเพี้ยน โมเดลภาษาจะมีแนวโน้มที่จะเกิดอาการตอบมั่วด้วยความมั่นใจผิด ๆ (False Confidence) โมเดลบางรุ่นอย่าง Qwen3 ได้รับผลกระทบจาก Noise อย่างรุนแรงจนคะแนน Accuracy (Strict) ในชุดข้อมูล OCR ตกไปอยู่ระดับต่ำสุดที่ 0.140 สะท้อนให้เห็นว่าในงานเอกสารสถาบันที่ต้องการความถูกต้องของข้อเท็จจริง ตัวเลข และวันที่ ความสะอาดของข้อมูลเป็นเงื่อนไขที่หลีกเลี่ยงไม่ได้

3. ผลกระทบของขนาดการแบ่งส่วนข้อมูล (Chunk Size Effect in Generation)

จากผลลัพธ์ของโมเดล Gemma3 พบว่า ขนาด Chunk 1,000 tokens เป็นขนาดที่เหมาะสมที่สุดสำหรับการใช้ข้อมูล Text เนื่องจากโครงสร้างเอกสารภาษาไทยมักมีความสัมพันธ์ของประโยคที่ต่อเนื่องกัน การมอบบริบทที่กว้างและครบถ้วน (Large Context Window) ช่วยให้โมเดลเข้าใจความหมายได้สมบูรณ์กว่าการตัดบริบทให้สั้นลง (250 tokens) อย่างไรก็ตาม พบปรากฏการณ์ที่แตกต่างออกไปในชุดข้อมูล OCR โดย Chunk ขนาดเล็ก (250 และ 500 tokens) กลับให้ผลลัพธ์การสร้างคำตอบที่ดีกว่า Chunk 1,000 tokens (Gemma3 F1-Score เพิ่มจาก 0.734 เป็น 0.824) สาเหตุเนื่องมาจากในข้อมูลที่มีความสกปรกสูง การส่ง Chunk ขนาดใหญ่เข้าสู่โมเดลจะเป็นการส่งต่อ "สัญญาณรบกวน" จำนวนมากเข้าไปพร้อมกัน (Noise Dilution) ทำให้โมเดลเกิดความสับสน การซอยข้อมูลให้เล็กลงจึงเปรียบเสมือนการจำกัดขอบเขตของความผิดพลาดไม่ให้รบกวนบริบทส่วนอื่นมากเกินไป

ผลของการสร้างคำตอบ (Test C) การสร้างระบบ RAG สำหรับภาษาไทยที่มีความน่าเชื่อถือสูง ควรเลือกใช้ข้อมูล Text ที่ผ่านการทำความสะอาด ร่วมกับการแบ่ง Chunk ขนาดใหญ่ (1,000 tokens) และปฏิบัติงานบนโมเดลที่มีความเสถียรด้านรูปแบบประโยคอย่าง Typhoon2.1-Gemma3-4B หากมีความจำเป็นต้องใช้ข้อมูลจากการทำ OCR ควรเพิ่มกระบวนการคัดกรองชิ้นส่วนข้อมูลอย่างเข้มงวด เพื่อป้องกันความผิดพลาดของคำตอบก่อนส่งถึงผู้ใช้งาน

นอกจากนี้ ผลการทดลองยังสะท้อนให้เห็นว่า ค่า Similarity Threshold มีผลโดยตรงต่อการประเมินผลลัพธ์ของโมเดล โดยเกณฑ์แบบ Lax (≥ 0.6) ช่วยสะท้อนความสามารถของโมเดลในการจับใจความสำคัญ ขณะที่เกณฑ์แบบ Strict (≥ 0.8) ให้ค่าความแม่นยำที่เข้มงวดและสะท้อนความถูกต้องเชิงข้อเท็จจริง แม้ว่าจะมีการเรียบเรียงข้อความใหม่ (Paraphrasing) ซึ่งพบได้ในโมเดลบางรุ่น เช่น Typhoon2.5-Qwen3-4B

4.4 สรุปผลการทดลอง

จากผลการประเมินประสิทธิภาพในทุกขั้นตอน (Test A, Test B และ Test C) สามารถสรุปข้อค้นพบที่สำคัญได้ดังนี้

4.4.1 คุณภาพของข้อมูลต้นทาง เป็นปัจจัยสำคัญที่สุด: ข้อมูลประเภทข้อความที่ผ่านการทำความสะอาด (Clean Text) ให้ผลลัพธ์ดีที่สุดในทุกมิติ ทั้งต้นทุนการทำดัชนีที่ต่ำที่สุด ในการจัดอันดับที่แม่นยำที่สุด และคุณภาพการสร้างคำตอบที่ถูกต้องที่สุดเมื่อเทียบกับกระบวนการ OCR

4.4.2 อิทธิพลของขนาด Chunk Size การใช้ขนาด Chunk ที่เหมาะสมสัมพันธ์กับความสะอาดของข้อมูล สำหรับข้อมูลที่สะอาด การใช้ Chunk ขนาดใหญ่ (1,000 tokens) ช่วยรักษารูปแบบความหมายของภาษาไทยได้ดีที่สุด แต่สำหรับข้อมูลที่มีสัญญาณรบกวน (OCR) การใช้ Chunk ขนาดเล็ก (250 tokens) จะช่วยจำกัดกระจายข้อมูลไม่ให้กระทบต่อการสร้างคำตอบของโมเดล

4.4.3 ปรากฏการณ์ความย้อนแย้งของการค้นคืน การดึงข้อมูลที่มีคำคล้ายคลึงกันขึ้นมาได้ในปริมาณมากไม่ได้การันตีความถูกต้องของผลลัพธ์ หากข้อมูลนั้นเต็มไปด้วยอักขระขยะจาก OCR ซึ่งจะส่งผลให้ประสิทธิภาพการจัดอันดับลดลง

4.4.4 ความอ่อนไหวของโมเดลภาษา คุณภาพในขั้นตอนการค้นคืน (Retrieval) มีผลโดยตรงต่อคุณภาพคำตอบ โมเดลภาษาทำหน้าที่เป็นเพียงผู้สังเคราะห์ข้อความ หากได้รับข้อมูลตั้งต้นที่ผิดพลาด โมเดลก็จะสร้างคำตอบที่ผิดเพี้ยนตามไปด้วยอย่างหลีกเลี่ยงไม่ได้

4.4.5 เกณฑ์ความคล้ายคลึง (Similarity Threshold) การกำหนดค่า similarity threshold ที่เหมาะสม เช่น 0.6 สำหรับความยืดหยุ่นของคำตอบ และ 0.8 สำหรับความถูกต้องเชิงข้อเท็จจริง มีบทบาทสำคัญในการประเมินผลระบบ RAG โดยช่วยให้สามารถสะท้อนความสามารถของโมเดลได้ทั้งในเชิงความแม่นยำและความเข้าใจเชิงความหมาย

บทที่ 5
สรุปผล อภิปรายผล และข้อเสนอแนะ

การศึกษาวิจัยเรื่อง การเพิ่มประสิทธิภาพและประเมินผล Retrieval-Augmented Generation (RAG) สำหรับเอกสารภาษาไทย มีวัตถุประสงค์เพื่อศึกษาผลกระทบเชิงปริมาณของคุณภาพแหล่งข้อมูลต้นทาง (Source Quality) และกลยุทธ์การแบ่งส่วนข้อมูล (Chunking Strategy) ที่มีต่อประสิทธิภาพของระบบ RAG โดยจำลองสภาพแวดล้อมการทำงานจริงด้วยเอกสารประกาศและคำสั่งของสถาบันเทคโนโลยีไทย-ญี่ปุ่น จำนวน 197 ฉบับ และประเมินผลผ่าน 3 ขั้นตอนหลัก ได้แก่ ต้นทุนการจัดทำดัชนี (Test A) คุณภาพการค้นคืน (Test B) และคุณภาพการสร้างคำตอบ (Test C) ซึ่งสามารถสรุปผล อภิปรายผล และเสนอแนะได้ดังต่อไปนี้

5.1 สรุปผลการวิจัย

จากการทดลองแบบควบคุมตัวแปรระหว่างคุณภาพแหล่งข้อมูล (Text, PDF Text, OCR) และขนาดการแบ่งส่วนข้อมูล (250, 500, 1,000 tokens) สามารถสรุปผลได้ดังนี้

5.1.1 สรุปผลด้านต้นทุนการจัดทำดัชนี (Test A: Indexing Cost)

ผลการประเมินต้นทุนการจัดทำดัชนีสะท้อนให้เห็นว่าคุณภาพของแหล่งข้อมูลต้นทางและกลยุทธ์การกำหนดขนาดข้อมูล (Chunk Size) มีอิทธิพลโดยตรงต่อภาระการประมวลผลและการใช้พื้นที่ของฐานข้อมูลเวกเตอร์อย่างมีนัยสำคัญ

เมื่อพิจารณาในภาพรวม ข้อมูลประเภทข้อความที่ผ่านการทำความสะอาด (Clean Text) ร่วมกับการกำหนดขนาด Chunk ที่ 1,000 โทเค็น ถือเป็นรูปแบบที่ใช้ทรัพยากรได้อย่างมีประสิทธิภาพสูงสุด เนื่องจากสามารถควบคุมปริมาณชิ้นส่วนข้อมูลได้ขนาดเล็กที่สุดเพียง 235 ชิ้น และใช้เวลาประมวลผลน้อยที่สุด ผลลัพธ์นี้แสดงให้เห็นว่า โครงสร้างข้อความที่สมบูรณ์และไม่มีอักขระขยะเจือปน ช่วยให้อัลกอริทึมแบ่งส่วนข้อมูลทำงานได้อย่างพอดีและไม่ต้องตัดแบ่งข้อความบ่อยครั้ง

ในทางกลับกัน เมื่อระบบต้องประมวลผลข้อมูลที่ได้จากกระบวนการแปลงภาพเป็นข้อความ (OCR) ร่วมกับขนาด Chunk เล็กที่ 250 โทเค็น ภาระการทำงานของระบบกลับเพิ่มสูงขึ้นอย่างก้าวกระโดด ปริมาณชิ้นส่วนข้อมูลขยายตัวเพิ่มมากถึง 3,782 ชิ้น ซึ่งสูงกว่ากรณีของข้อมูล Text ถึง 5.5 เท่า ส่งผลให้ฐานข้อมูลเวกเตอร์มีขนาดใหญ่ขึ้นและใช้เวลาประมวลผลยาวนานที่สุด และถ้าหากพิจารณาตัวเลขที่แตกต่างกันอย่างชัดเจนนี้ ไม่ได้บ่งชี้เพียงแค่ประเด็นเรื่องความสิ้นเปลือง

พื้นที่จัดเก็บหรือความล่าช้าของระบบเท่านั้น แต่ยังมีนัยสำคัญที่สามารถนำไปอภิปรายต่อได้ว่า สัญญาณรบกวน (Noise) ที่เกิดจาก OCR เช่น การตัดคำที่ผิดพลาด การเว้นวรรคที่ไม่เป็นระเบียบ หรือตัวอักษรขยะ เป็นตัวแปรสำคัญที่ทำให้กระบวนการ Chunking ต้องตัดย่อยข้อความถี่เกินความจำเป็น การกระจายของเนื้อหาออกเป็นชิ้นส่วนย่อย ๆ จำนวนมากนี้ อาจทำให้ความหมายที่ควรจะต่อเนื่องกันของประโยคภาษาไทยถูกตัดขาดออกจากกัน ซึ่งเป็นประเด็นสำคัญที่จะนำไปสู่การตั้งข้อสังเกตในขั้นตอนการค้นคืนข้อมูล (Retrieval) ต่อไปว่า ชิ้นส่วนข้อมูลจำนวนมหาศาลที่เต็มไปด้วยสัญญาณรบกวนเหล่านี้ จะส่งผลกระทบลูกโซ่ไปลดทอนความแม่นยำในการจัดอันดับ และลดทอนประสิทธิภาพในการสร้างคำตอบของโมเดลภาษาในท้ายที่สุดมากน้อยเพียงใด

5.1.2 สรุปผลด้านคุณภาพการค้นคืนและปรากฏการณ์ความย้อนแย้ง (Test B: Retrieval Quality)

ผลการประเมินประสิทธิภาพในขั้นตอนการค้นคืนข้อมูล (Retrieval) ช่วยย้ำให้เห็นถึงอิทธิพลของการทำสะอาดของข้อมูลที่มีต่อความแม่นยำในการจัดอันดับอย่างชัดเจน เมื่อพิจารณาแหล่งข้อมูลประเภทข้อความ (Clean Text) ร่วมกับการกำหนดขนาด Chunk ที่ 1,000 โทเค็น พบว่าเป็นรูปแบบที่มีประสิทธิภาพสูงสุด โดยสามารถทำคะแนนคุณภาพการจัดอันดับ (NDCG@10) ได้สูงถึง 0.899 และมีอัตราการค้นพบ (HitRate@10) ที่ระดับ 0.940 ตัวเลขเหล่านี้ยืนยันว่า เมื่อข้อมูลต้นทางปราศจากสัญญาณรบกวน โมเดลการฝังความหมาย (Embedding) จะสามารถเรียนรู้และจับคู่ความหมายเชิงบริบท (Semantic Matching) ได้อย่างสมบูรณ์แบบ ทำให้ข้อมูลที่เกี่ยวข้องที่สุดถูกดึงขึ้นมาจัดอยู่ในอันดับแรกได้อย่างสม่ำเสมอ

สำหรับข้อค้นพบที่น่าสนใจและถือเป็นแกนหลักของการวิจัยชิ้นนี้ คือสภาวะที่เกิดขึ้นกับแหล่งข้อมูลประเภท OCR ผลการทดลองในกลุ่ม OCR ขนาด 250 โทเค็น ได้ชี้ให้เห็นถึงความย้อนแย้งของการค้นคืน (Retrieval) อย่างเป็นรูปธรรม คือ ระบบดูเหมือนจะทำงานได้ดีในการดึงชิ้นส่วนข้อมูลที่เกี่ยวข้องขึ้นมาไว้ในผลลัพธ์ได้เป็นจำนวนมาก (มีความหนาแน่นของการค้นคืนเฉลี่ยสูงถึง 4.18 ชิ้นต่อคำถาม) แต่ในความเป็นจริงแล้ว คุณภาพของการจัดอันดับกลับสวนทางกัน โดยค่าเฉลี่ยอันดับส่วนกลับ (MRR@10) ลดต่ำลงเหลือเพียง 0.738

สาเหตุความย้อนแย้งดังกล่าวอาจเกิดจากกระบวนการ OCR ที่มักสร้างคำศัพท์สะกดผิดหรือมีอักขระขยะปะปนอยู่ สัญญาณรบกวนเหล่านี้ไปบิดเบือนตำแหน่งเวกเตอร์ในระบบ ทำให้โมเดลค้นคืนดึงเอาชิ้นส่วนข้อมูลที่มี คำคล้ายคลึงกันแต่ผิดบริบท เข้ามาแทรกใน 10 อันดับแรกเป็นจำนวนมาก ผลที่ตามมาคือ ข้อมูลอ้างอิงที่ถูกต้องและสมบูรณ์ที่สุดกลับถูกดึงให้ตกไปอยู่ในอันดับรองลงมา

ข้อค้นพบประเด็นนี้ถือเป็นจุดเปราะบางที่สำคัญที่สุดของ RAG ซึ่งต้องนำไปขยายผลต่อ เพราะข้อเท็จจริงนี้ได้ชี้ให้เห็นแล้วว่า การค้นหาข้อมูลเจอในปริมาณมาก ไม่ได้การันตีว่าจะได้ข้อมูลที่มีคุณภาพเสมอไป หากชิ้นส่วนข้อมูลที่ถูกดึงขึ้นมาเต็มไปด้วยสัญญาณรบกวนและไม่ได้ถูกจัดให้อยู่ในอันดับต้น ๆ ย่อมหลีกเลี่ยงไม่ได้ที่โมเดลภาษา ในขั้นตอนสุดท้ายจะได้รับบริบทที่สร้างความสับสน สภาพการณ์นี้จึงเป็นบทพิสูจน์ที่สำคัญว่า ในขั้นตอนการสร้างคำตอบ (Test C) โมเดลภาษาแต่ละรุ่นจะมีความสามารถในการต้านทานและคัดกรอง ขยะสารสนเทศ เหล่านี้ได้ดีเพียงใด ก่อนที่จะสังเคราะห์ออกมาเป็นคำตอบสุดท้ายให้กับผู้ใช้งาน

5.1.3 สรุปผลด้านคุณภาพการสร้างคำตอบและขีดจำกัดของโมเดลภาษา (Test C: Generation Quality)

ผลการประเมินในขั้นตอนสุดท้ายซึ่งเป็นหัวใจสำคัญของระบบอย่างคุณภาพการสร้างคำตอบ ซึ่งคุณภาพของข้อมูลต้นทาง คือตัวแปรสำคัญในการชี้ขาดความถูกต้องของผลลัพธ์ท้ายสุด ข้อมูลผลลัพธ์ที่เกิดขึ้นแสดงให้เห็นว่า โมเดลภาษาที่ทำงานได้อย่างมีประสิทธิภาพสูงสุดคือ Typhoon2.1-Gemma3-4B เมื่อได้รับบริบทจากแหล่งข้อมูลประเภทข้อความที่สะอาด (Clean Text) ร่วมกับขนาด Chunk 1,000 โทเค็น โดยสามารถทำคะแนนความแม่นยำ (Accuracy) ภายใต้เกณฑ์การประเมินแบบยืดหยุ่น (Lax) ได้สูงถึงร้อยละ 86 ผลลัพธ์ที่เกิดขึ้นนี้สะท้อนว่า หากระบบสามารถส่งบริบทที่ถูกต้อง สมบูรณ์ และมีเนื้อหาที่กว้างเพียงพอ โมเดลภาษาก็จะสามารถสังเคราะห์คำตอบที่มีความน่าเชื่อถือสูงออกมาได้อย่างมีประสิทธิภาพ

อย่างไรก็ตาม ความท้าทายอย่างยิ่งคือ เมื่อทดสอบกับชุดข้อมูลที่ผ่านกระบวนการ OCR ประสิทธิภาพของโมเดลภาษาทุกรุ่นในการตอบคำถามลดต่ำลงอย่างมีนัยสำคัญ โดยเฉพาะอย่างยิ่งเมื่อถูกประเมินภายใต้เกณฑ์แบบเข้มงวด (Strict) ที่เน้นความถูกต้องสมบูรณ์ของข้อเท็จจริง สาเหตุหลักในส่วนนี้อาจจะมาจากความอ่อนไหว ของโมเดลภาษาต่อความผิดพลาดประเภท ตัวเลขคลาดเคลื่อน (Numeric Distortion) ซึ่งเป็นผลพวงที่หลีกเลี่ยงไม่ได้จากข้อบกพร่องของตัวอักษรหรือสัญลักษณ์ที่เกิดจาก OCR

ข้อค้นพบประเด็นนี้ถือเป็นจุดที่สำคัญของงานวิจัย เพราะจะไปประกอบเข้ากับความย้อนแย้งของการค้นคืน ในขั้นตอนก่อนหน้าได้อย่างลงตัว และนำไปสู่การอภิปรายผลในประเด็นเรื่องหลักการ สัญญาณรบกวนขาเข้า ย่อมส่งผลเป็นสัญญาณรบกวนขาออก ในสถาปัตยกรรม RAG ข้อมูลในส่วนนี้ชี้ให้เห็นชัดเจนว่า ลำพังเพียงความอัจฉริยะของโมเดลภาษานั้น ไม่สามารถชดเชยหรือแก้ไขข้อมูลต้นทางที่เต็มไปด้วยขยะสารสนเทศได้ การที่โมเดลถูกชี้นำด้วยตัวเลขหรือข้อความที่ผิดเพี้ยนจนสร้างคำตอบที่ผิดพลาด ถือเป็นข้อควรระวังอย่างยิ่งสำหรับองค์กรที่คิดจะนำระบบ RAG ไปใช้กับเอกสารสำคัญทางราชการหรือเอกสารที่ต้องการความแม่นยำ

5.2 การอภิปรายผลการวิจัย

ผลการวิจัยนำมาสู่ข้อค้นพบที่สามารถอภิปรายเชื่อมโยงกับทฤษฎีและตอบคำถามวิจัยได้ 4 ประเด็นหลัก ดังนี้

5.2.1 อิทธิพลของคุณภาพข้อมูลต่อความน่าเชื่อถือของ RAG (RQ1)

ผลการทดลองยืนยันหลักการ “สัญญาณรบกวนขาเข้า ย่อมส่งผลเป็นสัญญาณรบกวนขาออก” ในระบบ RAG อย่างชัดเจน โดยพบว่าแม้ระบบจะขับเคลื่อนด้วยโมเดลภาษาที่มีประสิทธิภาพสูงและมีกลไกการจัดอันดับซ้ำ (Reranker) ที่ทรงพลัง แต่เมื่อข้อมูลต้นทางมีสัญญาณรบกวนจากกระบวนการ OCR ปะปนอยู่ (เช่น อักขระผิดเพี้ยน วรรณยุกต์ลอย หรือข้อมูลตัวเลขคลาดเคลื่อน) ระบบจะสร้างคำตอบที่ผิดพลาดออกมาพร้อมกับความมั่นใจที่ผิดปกติ (False Confidence)

ผลการทดลองที่สำคัญคือ ค่าความถูกต้อง (Accuracy) ของโมเดล Qwen3 ลดลงต่ำสุดเหลือเพียงร้อยละ 36 (0.360) ซึ่งสะท้อนให้เห็นถึงข้อจำกัดที่รุนแรงของการใช้ข้อมูล OCR ในบริบทของเอกสารทางการไทยที่ต้องการความถูกต้องเชิงข้อเท็จจริง ข้อค้นพบนี้สอดคล้องกับงานวิจัยของ Chang et al. [21] ที่ชี้ให้เห็นว่า ระบบ RAG มักประสบปัญหาเมื่อต้องจัดการกับเอกสารที่ถูกค้นคืนมาแต่มีสัญญาณรบกวน (Noisy documents) ซึ่งปัจจัยดังกล่าวจะไปลดทอนประสิทธิภาพของโมเดล เพิ่มภาระการประมวลผล และทำลายความน่าเชื่อถือของคำตอบ (Response reliability) อย่างรุนแรง ดังนั้น ข้อค้นพบนี้จึงสามารถใช้ตอบคำถามวิจัยข้อที่ 1 (RQ1) ได้อย่างชัดเจนว่า “คุณภาพของแหล่งข้อมูลต้นทาง เป็นปัจจัยชี้ขาดต่อประสิทธิภาพและความน่าเชื่อถือของระบบ”

5.2.2 ความย้อนแย้งของการค้นคืน (Retrieval Paradox) (RQ3)

งานวิจัยนี้ได้แสดงให้เห็นถึงปรากฏการณ์ที่เรียกว่า “ความย้อนแย้งของการค้นคืน (Retrieval Paradox)” กล่าวคือ การเพิ่มปริมาณและความหนาแน่นของข้อมูลที่ค้นคืนได้ ไม่ได้ส่งผลให้คำตอบมีความถูกต้องมากขึ้นเสมอไป โดยเฉพาะในกรณีที่ข้อมูลมีสัญญาณรบกวน (Noise) สูง

กลไกเบื้องหลังของปรากฏการณ์นี้เกิดจากการที่สัญญาณรบกวนได้เข้าไปทำให้กระบวนการจับคู่ความหมายเชิงลึก (Semantic Matching) ของ Embedding มีความคลาดเคลื่อน ส่งผลให้ระบบดึงเอกสารที่มีคำคล้ายคลึงกันแต่ผิดบริบทเข้ามาปะปนในผลลัพธ์ Top-10 เป็นจำนวนมาก (High Retrieval Density) เมื่อโมเดลภาษาได้รับก้อนบริบทที่ผสมปนเปกันระหว่างข้อมูลที่ถูกต้องและขยะสารสนเทศ จะเกิดความสับสนในการตีความเนื้อหา และนำไปสู่การลดลงของความแม่นยำในการสร้างคำตอบ ปรากฏการณ์นี้สอดคล้องกับข้อสังเกตของ Zhu et al. [20] ที่ระบุว่าการค้นคืนที่อาศัยเพียงความหมาย (Semantic-based) มักดึงชิ้นส่วนข้อมูลที่ถูกแยกส่วนและขาด

ความสัมพันธ์ที่ถูกต้องมาปะปนกัน และสอดคล้องกับงานวิจัยของ Arya et al. [24] ที่เน้นย้ำว่าปัญหาด้านคุณภาพของการค้นคืน (Retrieval quality) คือความท้าทายหลักของระบบ ข้อค้นพบนี้สามารถใช้ตอบคำถามวิจัยข้อที่ 3 (RQ3) ได้ว่า “ในสถาปัตยกรรม RAG คุณภาพและความสะอาดของข้อมูลมีความสำคัญมากกว่าปริมาณความหนาแน่นของการค้นคืน”

5.2.3 บริบทภาษาไทยและกลยุทธ์การแบ่งข้อมูล (RQ2)

ผลการทดลองแสดงให้เห็นว่า กลยุทธ์ขนาดของการแบ่งส่วนข้อมูล (Chunk Size) มีความสัมพันธ์กับระดับคุณภาพของข้อมูลและลักษณะทางโครงสร้างของภาษาไทยอย่างมีนัยสำคัญ สำหรับข้อมูลที่มีความสะอาด (Text / PDF Text) การใช้ Chunk ขนาดใหญ่ (1,000 tokens) ให้ผลลัพธ์การสร้างคำตอบที่ดีที่สุด เนื่องจากโครงสร้างของภาษาไทยในเอกสารทางการมักมีลักษณะประโยคที่ยาวและเชื่อมโยงเนื้อหาข้ามย่อหน้า การเปิดหน้าต่างบริบทให้กว้างจึงช่วยให้โมเดลเข้าใจเนื้อหาได้ครบถ้วนสมบูรณ์

ในทางตรงกันข้าม สำหรับข้อมูล OCR ที่มีสัญญาณรบกวนสูง การใช้ Chunk ขนาดเล็ก (250 tokens) กลับช่วยเพิ่มความแม่นยำได้ดีกว่า เนื่องจากทำหน้าที่เสมือน “การจำกัดขอบเขตของสัญญาณรบกวน (Noise Isolation)” ป้องกันไม่ให้ขยะสารสนเทศเจือจางบริบทสำคัญในหน้าต่างการประมวลผล นอกจากนี้ ยังพบข้อสังเกตที่สำคัญคือ ค่า MRR ของข้อมูล Text มีแนวโน้มลดลงเมื่อใช้ Chunk ขนาดใหญ่ขึ้น (จาก 0.862 ที่ขนาด 250 tokens ลดลงเหลือ 0.837 ที่ขนาด 1,000 tokens) ซึ่งอธิบายได้ว่า Chunk ที่มีขนาดใหญ่ทำให้ความเฉพาะเจาะจง (Specificity) ของเนื้อหาลดลง ข้อค้นพบเชิงพฤติกรรมนี้สอดคล้องกับงานวิจัยของ Chen et al. [22] ที่นำระบบ RAG ไปประยุกต์ใช้กับเอกสารเฉพาะทางอุตสาหกรรม โดยชี้ว่ากลยุทธ์การจัดการและประมวลผลข้อมูล (Data processing) มีผลโดยตรงต่อการทำความเข้าใจบริบทเฉพาะทาง และใช้ตอบคำถามวิจัยข้อที่ 2 (RQ2) ได้ว่า “กลยุทธ์ Chunk Size ที่เหมาะสมนั้นต้องแปรผันตามระดับความสะอาดของข้อมูลต้นทาง”

5.2.4 ความสอดคล้องของสถาปัตยกรรม RAG (RQ4)

ผลการทดลองแสดงให้เห็นว่า สถาปัตยกรรมที่ใช้แบบจำลองภาษาขนาดเล็ก (Small Language Model: SLM) ร่วมกับการจัดอันดับใหม่ (Reranking) สามารถสร้างคำตอบที่มีความแม่นยำในระดับที่นำไปใช้งานได้จริง โดยทำคะแนนความถูกต้อง (Accuracy) ได้สูงสุดถึงร้อยละ 86 (0.860)

อย่างไรก็ตาม ประสิทธิภาพดังกล่าวอยู่ภายใต้เงื่อนไขที่ต้องอิงกับคุณภาพของข้อมูลต้นทางเป็นหลัก แม้กลไก Reranker จะช่วยจัดอันดับเอกสารและคัดกรองบริบทได้ดีเพียงใด

แต่ก็ไม่สามารถแก้ไขข้อมูลที่ผิดพลาดเสียหายจากกระบวนการ OCR ได้ ซึ่งถือเป็นข้อจำกัดเชิงโครงสร้างของระบบ RAG ที่สอดคล้องกับรากฐานแนวคิดดั้งเดิมของ Lewis et al. [2] และงานวิจัยของ Pradeesh et al. [23] ที่ชี้ให้เห็นว่าระบบสามารถสร้างคำตอบที่ถูกต้องได้ดีเมื่อใช้ข้อมูลที่มีคุณภาพสูงจากการจัดเตรียมเอกสารต้นฉบับ (PDF source documents) ดังนั้น ข้อค้นพบนี้จึงสามารถตอบคำถามวิจัยข้อที่ 4 (RQ4) ได้อย่างสมบูรณ์ว่า “ระบบ RAG ที่ใช้ SLM ร่วมกับ Reranker สามารถประยุกต์ใช้งานในองค์กรได้จริงและมีประสิทธิภาพสูง แต่จำเป็นต้องอาศัยการจัดเตรียมข้อมูลต้นทางให้มีความสะอาดสูงจึงจะได้ผลลัพธ์ที่น่าเชื่อถือ”

5.3 ข้อจำกัดของการวิจัย

งานวิจัยนี้มีข้อจำกัดบางประการที่ควรพิจารณาเมื่อนำผลลัพธ์ไปประยุกต์ใช้

5.3.1 ขอบเขตชุดคำถามและโดเมนของข้อมูล ชุดคำถามประเมินผล 50 ข้อถูกออกแบบมาสำหรับการทดสอบเชิงข้อเท็จจริง จากเอกสารประกาศและคำสั่งของสถาบันเทคโนโลยีไทย-ญี่ปุ่นเท่านั้น ผลลัพธ์ที่ได้อาจมีความแตกต่างหากนำไปใช้กับเอกสารเชิงพรรณนา หรืองานที่ต้องใช้การให้เหตุผลเชิงซ้อน

5.3.2 ข้อจำกัดด้านโครงสร้างตาราง เอกสารที่มีตารางซับซ้อน เช่น ตารางงบประมาณ เมื่อถูกแปลงเป็นข้อความ อาจสูญเสียความสัมพันธ์แบบข้ามคอลัมน์ ทำให้ความแม่นยำในการตอบคำถามเชิงตารางลดลง

5.3.3 ข้อจำกัดด้านตัวแปรโมเดล งานวิจัยอ้างอิงพฤติกรรมการสร้างคำตอบจากโมเดลตระกูล Typhoon 2.x แบบ Local เป็นหลัก หากใช้งานบนโมเดลเชิงพาณิชย์ขนาดใหญ่ (เช่น GPT-4 หรือ Claude) ผลกระทบจาก Noise อาจแตกต่างออกไป

5.3.4 ข้อจำกัดด้านการทดสอบทางสถิติ แม้ว่างานวิจัยนี้จะมีการเปรียบเทียบผลลัพธ์เชิงปริมาณระหว่างวิธีการต่าง ๆ อย่างชัดเจน แต่ยังไม่ได้มีการนำวิธีการทดสอบทางสถิติ (Statistical Significance Testing) มาใช้ในการยืนยันความแตกต่างของผลลัพธ์อย่างเป็นทางการ ดังนั้น ความแตกต่างของค่าตัวชี้วัด เช่น Accuracy, MRR และ NDCG ที่ปรากฏ อาจยังไม่สามารถสรุปได้อย่างมีนัยสำคัญทางสถิติ ในการศึกษาครั้งต่อไป ควรมีการนำเทคนิค เช่น Paired t-test หรือ Bootstrap Sampling มาใช้ เพื่อเพิ่มความน่าเชื่อถือของผลการเปรียบเทียบ และลดความเสี่ยงจากความแปรปรวนของข้อมูลตัวอย่าง

5.3.5 ระบบมีข้อจำกัดด้านขนาด token ของโมเดล ซึ่งส่งผลต่อปริมาณบริบทที่สามารถนำเข้าได้ โดยการทดลองนี้ใช้ chunk size 250, 500 และ 1000 token ซึ่งยังอยู่ในขอบเขตที่โมเดลสามารถประมวลผลได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม หากใช้ขนาดที่ใหญ่เกินไป (เช่น ถ้ามากกว่า 1500 token) อาจทำให้เกิด truncation หรือประสิทธิภาพลดลง

5.4 ข้อเสนอแนะ

5.4.1 ข้อเสนอแนะสำหรับการนำไปประยุกต์ใช้งานในองค์กร

จากผลการทดลอง ระบบต้นแบบนี้พร้อมเป็นแนวทางเบื้องต้นสำหรับหน่วยงานสารสนเทศ หรือสายสนับสนุนของสถาบันเทคโนโลยีไทย-ญี่ปุ่น ในการนำไปสร้างแชตบอตสืบค้นระเบียบปฏิบัติ โดยมีข้อแนะนำดังนี้

การตั้งค่าพารามิเตอร์ (Parameter Tuning) สำหรับข้อมูลประกาศภาษาไทย แนะนำให้ตั้งค่า Chunk Size ที่ 1,000 tokens และ Overlap ที่ 200 tokens ซึ่งเป็นจุดสมดุลที่สุดระหว่างต้นทุนและประสิทธิภาพ

การป้องกันอาการหลอน (Hallucination Prevention) ควรประยุกต์ใช้เทคนิคการคัดกรองข้อมูลก่อนสร้างดัชนี (Noise-aware Chunk Validation) และใช้กลไกบังคับการอ้างอิงหลักฐาน (Citation Guardrails) ใน Prompt เสมอ เพื่อลดความเสี่ยงของการให้คำตอบที่ผิดพลาดแก่บุคลากร

5.4.2 ข้อเสนอแนะสำหรับการวิจัยในอนาคต (Future Work)

การใช้เทคนิค Semantic Chunking ควรมีการศึกษาเปรียบเทียบการแบ่งส่วนข้อมูลตามหลักไวยากรณ์หรือโครงสร้างความหมาย (Semantic Chunking) แทนการแบ่งด้วยจำนวน Token คงที่ เพื่อรักษาบริบทของภาษาไทยให้สมบูรณ์ยิ่งขึ้น

การปรับปรุงคุณภาพของข้อมูล OCR เช่น การเลือกใช้โมเดล OCR ที่มีความแม่นยำสูงขึ้น การทำ post-processing เพื่อลด noise หรือการใช้เทคนิค error correction เพื่อเพิ่มความถูกต้องของข้อความก่อนนำเข้าสู่ระบบ RAG

การกู้คืนคุณภาพ OCR (Post-OCR Correction) ควรมีการวิจัยต่อยอดโดยนำ Small Language Model (SLM) มาทำหน้าที่เป็นตัวแก้ไขคำผิดและจัดรูปแบบข้อความจาก OCR อัตโนมัติ (Data Cleaning Pipeline) ก่อนส่งเข้าสู่กระบวนการ Vector Embedding

การประเมินผลกับตารางและรูปภาพ (Multimodal RAG) ควรขยายขอบเขตการทดลองไปยังสถาปัตยกรรม Multimodal ที่สามารถทำความเข้าใจโครงสร้างตาราง (Tables) และแผนผัง (Diagrams) ในเอกสารได้โดยตรงโดยไม่ต้องแปลงเป็นข้อความ

บรรณานุกรม

บรรณานุกรม

[1] สถาบันเทคโนโลยีไทย-ญี่ปุ่น, "*Announcement Systems*," [Online]. Available: https://announcement.tni.ac.th [Accessed: December 3, 2025].

[2] Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," *in Advances in Neural Information Processing Systems, NeurIPS 2020*, Vancouver, Canada, December 6-12, 2020, pp. 9459–9474, [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf [Accessed: December 3, 2025].

[3] AGPL, "*PyMuPDF*," [Online]. Available: https://pymupdf.readthedocs.io [Accessed: December 3, 2025].

[4] GitHub, " *Tesseract-OCR*," [Online]. Available: https://github.com/tesseract-ocr [Accessed: December 3, 2025].

[5] Pipatanakul et al., "*Typhoon 2: A Family of Open Text and Multimodal Thai Large Language Models*," arXiv, 2024, [Online]. Available: https://arxiv.org/abs/2412.13702 [Accessed: December 3, 2025].

[6] Vaswani et al., "Attention is all you need," *in 31st Conference on Neural Information Processing Systems,* NIPS 2017, Long Beach, CA, USA., December 4 - 9, 2017, pp.1-15, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html [Accessed: March 17, 2025].

[7] OpenAI, "*ChatGPT*," [Online]. Available: https://openai.com [Accessed: January 12, 2026].

[8] Qwen, "*Qwen*," [Online]. Available: https://qwen.ai/home [Accessed: January 12, 2026].

[9] Google, "*Gemini*," [Online]. Available: https://deepmind.google/technologies/gemini [Accessed: March 17, 2025].

[10] Facebook, " *Llama*," [Online]. Available: https://www.llama.com [Accessed: January 12, 2026].

[11] OpenThaiGPT, “*OpenThaiGPT*,” [Online]. Available: https://github.com/OpenThaiGPT [Accessed: March 17, 2025].

[12] SCB 10X Co. Ltd., “*Typhoon*,” [Online]. Available: https://opentyphoon.ai [Accessed: December 3, 2025].

[13] Ling et al., “Automatic question-answer pairs generation using pre-trained large language models in higher education,” *Computers and Education: Artificial Intelligence*, vol. 6, no. 1, pp. 100252, June 2024, [Online]. Available: https://doi.org/10.1016/j.caeai.2024.100252 [Accessed: March 17, 2025].

[14] Raheem et al., “E-commerce fake reviews detection using LSTM with word2 vec embedding,” *CIT Journal of Computing and Information Technology*, vol. 32, no. 2, pp. 65–80, September 2024, [Online]. Available: https://hrcak.srce.hr/en/clanak/464256 [Accessed: March 17, 2025].

[15] Si et al., “Sub-character tokenization for Chinese pretrained language models,” *Transactions of the Association for Computational Linguistics*, vol. 11, no. 1, pp. 469–487, 2023, [Online]. Available: https://doi.org/10.1162/tacl_a_00560 [Accessed: March 17, 2025].

[16] Chen et al., “BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,” *in Findings of the Association for Computational Linguistics,* ACL 2024, Bangkok, Thailand, August 2024, pp. 2318–2335, [Online]. Available: https://aclanthology.org/2024.findings-acl.137 [Accessed: March 17, 2025].

[17] BAAI,“ *Bge-reranker-V2-M3*,” [Online]. Available: https://huggingface.co/BAAI/bge-reranker-v2-m3 [Accessed: January 29, 2026].

[18] PostgreSQL Global Development Group, “*PostgreSQL*,” [Online]. Available: https://www.postgresql.org [Accessed: January 28, 2026].

[19] Docker Hub, “ *Pgvector*,” [Online]. Available: https://hub.docker.com/r/pgvector/pgvector [Accessed: January 28, 2026].

[20] Zhu et al., “Knowledge graph-guided retrieval augmented generation,” *in 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* NAACL-HLT 2025, Albuquerque, New Mexico, USA., April 29 - May 4, 2025, pp. 8912-8924, [Online]. Available: https://aclanthology.org/2025.naacl-long.449 [Accessed: March 17, 2025].

[21] Chang et al., “MAIN-RAG: Multi-agent filtering retrieval-augmented generation,” *in 63rd Annual Meeting of the Association for Computational Linguistics,* ACL 2025, Vienna, Austria., July 27 - August 1, 2025, pp. 2607-2622, [Online]. Available: https://aclanthology.org/2025.acl-long.131 [Accessed: March 17, 2025].

[22] Chen et al., “Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model,” *Computer Standards & Interfaces*, vol. 94, no. 103995, pp. 1–13, August 2025. [Online]. Available: https://doi.org/10.1016/j.csi.2025.103995 [Accessed: March 17, 2025].

[23] Pradeesh et al., “Retrieval-augmented generation for multiple-choice questions and answers generation,” *Procedia Computer Science*, vol. 259, no. 1, pp. 504–511, January 2025. [Online]. Available: https://doi.org/10.1016/j.procs.2025.03.352 [Accessed: March 17, 2025].

[24] Arya et al., “Advances in retrieval-augmented generation (RAG) and related frameworks,” *International Journal on Science and Technology (IJSAT)*, vol. 16, no. 3, pp. 2229-7677, July–September 2025, [Online]. Available: https://www.ijsat.org/research-paper.php?id=7595 [Accessed: March 17, 2025].