บทความวิจัย (Research Article)

การพัฒนาแชทบอทบริการลูกค้าที่เข้าใจบริบทและเลือกใช้เครื่องมือโดยอัตโนมัติด้วยแบบจำลองภาษาขนาดใหญ่และ LangChain

Development of a Context-Aware Customer Service Chatbot with Automated Tool Selection Using Large Language Models and LangChain

ประทีป หมื่นทิศ และ สุขุมาล กิติสิน*

ภาควิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยเกษตรศาสตร์
* ผู้รับผิดชอบบทความ
sukumal.i@ku.th

Received: 10 Apr 2025
Revised: 6 Aug 2025
Accepted: 22 Aug 2025

บทคัดย่อ

ในการให้บริการลูกค้า องค์กรจำนวนมากเผชิญความท้าทายจากการที่เจ้าหน้าที่ต้องสืบค้นข้อมูลจากหลายแหล่งเพื่อตอบคำถามและแก้ไขปัญหาอย่างถูกต้องและรวดเร็ว แหล่งข้อมูลเหล่านี้มักกระจายในหลายระบบและมีรูปแบบแตกต่างกัน ส่งผลให้การเข้าถึงข้อมูลล่าช้า ขาดความต่อเนื่อง และลดประสิทธิภาพการปฏิบัติงาน นอกจากนี้ การสูญเสียบุคลากรที่เชี่ยวชาญด้านการค้นคืนและสังเคราะห์ข้อมูลยังส่งผลให้เกิดช่องว่างในการถ่ายทอดองค์ความรู้เชิงปฏิบัติและขาดความต่อเนื่องขององค์ความรู้ การศึกษานี้นำเสนอการพัฒนาระบบแชทบอทอัจฉริยะที่สามารถวิเคราะห์เจตนาและบริบทของคำถาม และเลือกใช้เครื่องมือการค้นคืนข้อมูลที่เหมาะสมโดยอัตโนมัติ ระบบใช้การประสานงานระหว่างแบบจำลองภาษาขนาดใหญ่ (Large Language Model: LLM) และเครื่องมือเฉพาะทาง ระบบได้รับการออกแบบบนเฟรมเวิร์ก LangChain ซึ่งอำนวยความสะดวกในการจัดการตัวแทนอัจฉริยะ (Agent Orchestration) และการผสานรวมเครื่องมือแบบแยกส่วน (Modular Integration) เครื่องมือหลักประกอบด้วยสองประเภท คือ เครื่องมือการค้นคืนข้อมูลแบบผสมผสาน (Hybrid Retrieval Tool) ที่ทำหน้าที่สืบค้นข้อมูลจากฐานความรู้ภายในองค์กร และเครื่องมือการเชื่อมต่อระบบภายนอก (API Retrieval Tool) ที่เข้าถึงข้อมูลจากแหล่งข้อมูลภายนอก ผลการประเมินประสิทธิภาพระบบแสดงให้เห็นว่าระบบมีความแม่นยำในการเลือกใช้เครื่องมือ (Tool Call Accuracy) เฉลี่ย 0.88 ความถูกต้องของการตอบคำถาม (Answer Correctness) เฉลี่ย 0.80 เวลาในการตอบสนองเฉลี่ย (Average Latency) 7,782 มิลลิวินาที และระดับความพึงพอใจของผู้ใช้งาน (User Satisfaction) ผู้ใช้บริการและพนักงานอยู่ที่ 85.67% และ 86.16% ตามลำดับ ผลการศึกษาแสดงให้เห็นถึงศักยภาพของระบบในการเสริมสร้างประสิทธิภาพการให้บริการลูกค้าในองค์กรที่มีข้อมูลหลากหลายและซับซ้อน

คำสำคัญ: แชทบอท ตัวแทนปัญญาประดิษฐ์ โมเดลภาษาขนาดใหญ่ การสร้างข้อความแบบเสริมด้วยการค้นคืนข้อมูล ระบบบริการลูกค้าอัตโนมัติ

Abstract

In providing customer service, many organizations face challenges due to staff needing to retrieve information from multiple sources to answer questions and resolve issues accurately and quickly. These data sources are often distributed across various systems and exist in different formats, resulting in delayed access, discontinuity, and reduced operational efficiency. Furthermore, the loss of personnel skilled in information retrieval and synthesis leads to gaps in the transfer of

practical knowledge and discontinuity in organizational know-how. This study presents the development of an intelligent chatbot system capable of analyzing the intent and context of questions and automatically selecting appropriate information retrieval tools. The system utilizes coordination between a Large Language Model (LLM) and specialized tools. It is designed based on the LangChain framework, which facilitates agent orchestration and modular tool integration. The primary tools consist of two types: a Hybrid Retrieval Tool that retrieves information from internal organizational knowledge bases, and an API Retrieval Tool that accesses data from external sources. Performance evaluation results show an average tool call accuracy of 0.88, an average answer correctness of 0.80, an average response time of 7,782 milliseconds, and user satisfaction levels of 85.67% and 86.16% for customers and staff, respectively. The study's findings demonstrate the system’s potential to enhance customer service efficiency in organizations with diverse and complex information environments.

Keywords: Chatbot, Artificial Intelligence Agent, Large Language Model, Retrieval-Augmented Generation, Automated Customer Service System

1. บทนำ

องค์กรที่ให้บริการลูกค้าในปัจจุบันจำเป็นต้องพัฒนาระบบบริการที่สามารถตอบสนองความต้องการของผู้ใช้บริการได้อย่างรวดเร็ว แม่นยำ และยืดหยุ่น เพื่อสร้างความได้เปรียบในการแข่งขัน ปัญหาหลักที่พบคือการกระจายตัวของข้อมูลในระบบงานหลากหลาย เช่น ฐานข้อมูลสินค้า ระบบจัดการคำสั่งซื้อ และระบบจัดการเอกสารองค์กร ทำให้การค้นหาข้อมูลเป็นไปได้ยาก ประกอบกับอัตราการเปลี่ยนแปลงบุคลากรที่สูง ส่งผลให้เกิดการสูญเสียองค์ความรู้และต้องใช้เวลาในการฝึกอบรมบุคลากรใหม่อย่างต่อเนื่อง

เพื่อแก้ไขปัญหาดังกล่าว องค์กรจำนวนมากได้นำระบบแชทบอทมาใช้ในการตอบคำถามเบื้องต้น อย่างไรก็ตาม แชทบอทแบบดั้งเดิมที่อาศัยชุดคำตอบที่กำหนดไว้ล่วงหน้าไม่สามารถรองรับคำถามที่ซับซ้อนหรือปรับตัวตามบริบทได้อย่างเหมาะสม [1] ดังนั้น การประยุกต์ใช้ Large Language Models (LLMs) ร่วมกับเทคนิค Retrieval-Augmented Generation (RAG) จึงได้รับความสนใจในการยกระดับความแม่นยำของระบบ [2] แต่การใช้เทคโนโลยีเหล่านี้แบบเดี่ยวยังมีข้อจำกัดในการจัดการข้อมูลจากหลายแหล่งและการเชื่อมต่อกับระบบภายนอก

การศึกษาเกี่ยวกับระบบ RAG ที่ใช้ agent ในการเลือกเครื่องมือและแหล่งข้อมูลแสดงศักยภาพในการแก้ไขข้อจำกัดดังกล่าว โดยสามารถปรับปรุงคุณภาพการตอบคำถามได้ดีขึ้น [3] อย่างไรก็ตาม การนำไปประยุกต์ใช้ในงานบริการลูกค้ายังต้องการการศึกษาเพิ่มเติมเกี่ยวกับการออกแบบระบบที่สามารถขยายขนาดได้และการประเมินประสิทธิภาพในบริบทการใช้งานที่หลากหลาย

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแชทบอทที่ใช้ AI agent ซึ่งสามารถเลือกและใช้งานเครื่องมือสำหรับค้นหาข้อมูลจากแหล่งต่างๆ เพื่อตอบคำถามลูกค้าได้อย่างเหมาะสม ระบบนี้มีกลไกการวิเคราะห์เจตนาของผู้ใช้และการเลือกเครื่องมือตามบริบท โดยสามารถเข้าถึงฐานข้อมูลภายในองค์กรและเชื่อมต่อกับระบบภายนอกได้ เป้าหมายของการศึกษาคือการออกแบบและประเมินประสิทธิภาพของแชทบอทต้นแบบดังกล่าว เพื่อยกระดับคุณภาพของระบบบริการลูกค้า ลดภาระงานของเจ้าหน้าที่ และเพิ่มประสิทธิภาพในการตอบสนองคำถามที่หลากหลาย

2. การทบทวนวรรณกรรม

แชทบอทได้รับการพัฒนาและนำมาใช้ในหลากหลายสาขา โดยเฉพาะงานบริการลูกค้า ซึ่งต้องการการตอบคำถามที่รวดเร็วและแม่นยำ อย่างไรก็ตาม แชทบอทแบบดั้งเดิมที่ใช้คำตอบสำเร็จรูปยังมีข้อจำกัดในการรองรับคำถามที่ซับซ้อนหรือเปลี่ยนแปลงตาม

บริบท จึงเกิดการพัฒนาแชทบอทที่สามารถตีความเจตนาและดึงข้อมูลตามบริบทเพื่อยกระดับคุณภาพของคำตอบ

Ezenkwu [4] ได้เสนอระบบที่ผสานการใช้ ChatGPT กับเทคนิค prompt engineering เพื่อสร้างคำตอบแบบ few-shot ในรูปแบบภาษาธรรมชาติ โดยฝังองค์ความรู้จากผู้เชี่ยวชาญไว้ใน prompt ระบบสามารถสร้างคำตอบที่มีเหตุผลได้อย่างยืดหยุ่นจากข้อความของผู้ใช้ โดยไม่ต้องแปลงคำถามให้ตรงกับชุดคำตอบที่กำหนดไว้ล่วงหน้าเหมือนระบบผู้เชี่ยวชาญแบบดั้งเดิม อย่างไรก็ตาม ระบบมีข้อจำกัดในการจัดการกับคำถามที่อยู่นอกขอบเขตของ prompt หรือกรณีที่ prompt ซับซ้อนมาก ซึ่งอาจนำไปสู่การให้คำตอบที่ไม่สอดคล้องกับบริบท

Lee และคณะ [2] ได้พัฒนาแชทบอทโดยใช้แนวทาง Retrieval-Augmented Generation (RAG) ร่วมกับโมเดล LLaMA 2 เพื่อให้การสนับสนุนด้านเทคนิค โดยอ้างอิงข้อมูลจากเอกสารภายในองค์กร ระบบสามารถลดปัญหาเรื่องการสร้างข้อมูลเท็จ (Hallucination) และเพิ่มความแม่นยำในการตอบคำถาม อย่างไรก็ตาม การรวมเอกสารหลายประเภทไว้ในคลังข้อมูลเดียวกัน อาจส่งผลกระทบต่อประสิทธิภาพของระบบการค้นคืนข้อมูล

Guettala และคณะ [3] ได้นำเสนอระบบถาม–ตอบตามแนวทาง Retrieval-Augmented Generation (RAG) โดยใช้เฟรมเวิร์ก LangChain ระบบดังกล่าวได้รับการออกแบบให้มีตัวแทนอัจฉริยะ (Agent) ที่สามารถเลือกใช้เครื่องมือในการค้นคืนข้อมูลให้เหมาะสมกับคำถาม เช่น vector retriever หรือ API ทั้งนี้ Agent ยังสามารถควบคุมกระบวนการทั้งหมด ตั้งแต่การค้นคืนไปจนถึงการสร้างคำตอบได้อย่างมีประสิทธิภาพ กลไกการทำงานในลักษณะนี้ช่วยลดความคลาดเคลื่อนของข้อมูล และเพิ่มความน่าเชื่อถือของคำตอบ โดยเฉพาะในบริบทที่ต้องการความยืดหยุ่นในการจัดการข้อมูล

งานวิจัยที่ผ่านมาพบว่า การบูรณาการเทคโนโลยี LLM กับ RAG เป็นแนวโน้มสำคัญในการพัฒนาแชทบอทที่มีความแม่นยำและน่าเชื่อถือ อย่างไรก็ตาม ในบริบทงานบริการลูกค้ายังมีความท้าทายด้านการเข้าถึงข้อมูลจากแหล่งที่หลากหลายเพื่อสร้างข้อมูลอ้างอิงที่สอดคล้องกับบริบทผู้ใช้ ดังนั้น การประยุกต์ใช้ระบบ Agent จึงเป็นแนวทางที่มีประสิทธิภาพสำหรับแชทบอทบริการลูกค้า เนื่องจากสามารถวิเคราะห์เจตนาและบริบทของผู้ใช้ได้แม่นยำ พร้อมเลือกเครื่องมือค้นคืนข้อมูลที่เหมาะสมโดยอัตโนมัติ

3. ทฤษฎีที่เกี่ยวข้อง

3.1 แชทบอท

แชทบอท เป็นโปรแกรมคอมพิวเตอร์ที่ออกแบบมาเพื่อโต้ตอบกับผู้ใช้งานผ่านข้อความหรือเสียง โดยมีวัตถุประสงค์เพื่อให้ข้อมูลตอบคำถาม หรือให้บริการในรูปแบบอัตโนมัติ ซึ่งช่วยเพิ่มประสิทธิภาพในการให้บริการและลดภาระงานที่เกิดซ้ำ [1] ในปัจจุบัน แชทบอทสมัยใหม่สามารถเข้าใจภาษาธรรมชาติและสนทนากับผู้ใช้ได้อย่างเป็นธรรมชาติมากขึ้น

3.2 Large Language Models (LLMs)

แบบจำลองภาษาขนาดใหญ่ (Large Language Models: LLMs) คือระบบปัญญาประดิษฐ์ที่เรียนรู้จากข้อมูลภาษาขนาดใหญ่ ทำให้สามารถเข้าใจและสร้างข้อความได้ใกล้เคียงกับมนุษย์ จุดเด่นของ LLMs คือความสามารถในการประยุกต์ใช้ความรู้ไปยังบริบทใหม่โดยไม่ต้องฝึกเฉพาะทาง อย่างไรก็ตาม LLMs ยังเผชิญข้อจำกัดสำคัญ ได้แก่ การสร้างข้อมูลผิดพลาด (Hallucination) อคติจากข้อมูลฝึก และความไม่โปร่งใสในการอธิบายเหตุผล [5]

3.3 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) เป็นแนวทางที่ผสานกระบวนการสืบค้นข้อมูลเข้ากับการสร้างข้อความโดยโมเดลภาษาขนาดใหญ่ (LLM) เพื่อเพิ่มความแม่นยำในการตอบคำถาม และลดความเสี่ยงในการนำเสนอข้อมูลที่คลาดเคลื่อนหรือขาดแหล่งอ้างอิง โมเดลจะสืบค้นข้อมูลที่เกี่ยวข้องจากฐานความรู้หรือจาก API จากนั้นจึงประมวลผลข้อมูลร่วมกับคำถาม เพื่อสร้างคำตอบที่มีความสอดคล้องกับบริบท [2-3]

3.4 Hybrid Search

Hybrid Search เป็นแนวทางการสืบค้นที่ผสานการค้นหาเชิงคำ (Lexical search) ซึ่งอาศัยการจับคู่คำที่ตรงกันระหว่างคำค้นกับเนื้อหา และการค้นหาเชิงความหมาย (Semantic search) ซึ่งใช้เวกเตอร์ฝังความหมายของข้อความเพื่อประเมินความใกล้เคียงทางบริบท [6] ผลลัพธ์จากทั้งสองวิธีจะถูกรวบรวมและจัดลำดับใหม่ผ่านเทคนิคการรวมอันดับ เช่น Reciprocal Rank Fusion

(RRF) ซึ่งพิจารณาอันดับของเอกสารจากแต่ละแหล่งสืบค้น แล้วนำมาคำนวณคะแนนรวมเพื่อให้ได้ผลลัพธ์ที่มีความเกี่ยวข้องสูงในทั้งสองมิติอย่างสมดุล [7]

3.5 Reciprocal Rank Fusion (RRF)

Reciprocal Rank Fusion (RRF) เป็นเทคนิคการรวมผลลัพธ์จากหลายกระบวนการสืบค้น โดยพิจารณาอันดับของเอกสารจากแต่ละแหล่งค้นหา แล้วคำนวณคะแนนรวมใหม่ เทคนิคนี้ช่วยให้สามารถรวมผลลัพธ์จากการค้นหาเชิงคำและเชิงความหมายได้ RRF จึงเป็นองค์ประกอบสำคัญที่สนับสนุนการทำงานของ Hybrid Search [7]

3.6 Vector Embedding

Vector Embedding เป็นเทคนิคที่แปลงข้อความให้อยู่ในรูปเวกเตอร์เชิงตัวเลข เพื่อสะท้อนความสัมพันธ์เชิงความหมายระหว่างข้อความ โดยเวกเตอร์ที่มีความใกล้เคียงกันจะแสดงความหมายที่คล้ายคลึงกัน เทคนิคนี้มีความสำคัญในการเพิ่มความสามารถของ LLMs ในบริบทของระบบ RAG โดยช่วยให้โมเดลสามารถสืบค้นข้อมูลที่เกี่ยวข้องได้อย่างเหมาะสม ในการดำเนินงานวิจัยนี้ได้เลือกใช้โมเดล text-embedding-3-small ซึ่งพัฒนาโดย OpenAI [2,3,8]

3.7 Artificial Intelligence Agent (AI Agent)

Artificial Intelligence Agent (AI Agent) หมายถึงระบบอัจฉริยะที่สามารถดำเนินงานโดยอัตโนมัติ โดยไม่จำเป็นต้องอาศัยการควบคุมจากมนุษย์ AI Agent มีความสามารถในการรับรู้สภาพแวดล้อม ประมวลผลข้อมูล ตัดสินใจ และตอบสนองต่อสถานการณ์ตามเป้าหมายที่กำหนดไว้ ภายใต้ข้อจำกัดที่กำหนดไว้ล่วงหน้า

ในปัจจุบัน แนวโน้มสำคัญในการพัฒนา AI agent คือการใช้ LLMs เป็นแกนกลางของกระบวนการประมวลผลและตัดสินใจ แนวทางนี้นำไปสู่การพัฒนา LLM-based agent ที่มีความสามารถหลากหลาย [9]

LLM-based Agent ประกอบด้วยองค์ประกอบหลัก 4 ด้านที่ทำงานร่วมกันอย่างเป็นระบบ ได้แก่

- Perception: การรับและแปลงข้อมูลจากสิ่งแวดล้อม
- Planning: การวิเคราะห์และวางแผนการดำเนินการ
- Memory: การจัดเก็บข้อมูลการโต้ตอบและบริบท
- Action: การตอบสนองต่อผู้ใช้หรือระบบ

ความสามารถในการบูรณาการกระบวนการรับรู้ วางแผน จดจำ และปฏิบัติการของ LLM-based Agent ภายใต้โครงสร้างเดียวกัน นับเป็นจุดเปลี่ยนสำคัญที่ทำให้ AI Agent ก้าวสู่การเป็นระบบอัจฉริยะที่สามารถทำงานร่วมกับมนุษย์ได้อย่างมีประสิทธิภาพในบริบทที่หลากหลาย

3.8 LangChain Framework

LangChain เป็นเครื่องมือโอเพนซอร์สที่ออกแบบมาเพื่อสนับสนุนการพัฒนาแอปพลิเคชันที่ใช้โมเดลภาษาขนาดใหญ่ (LLMs) ซึ่งช่วยให้สามารถสร้างระบบโต้ตอบที่เข้าใจบริบทและเชื่อมโยงกับข้อมูลภายนอกได้ง่ายขึ้น [3]

3.8.1 Tools

ใน LangChain คำว่า "Tools" หมายถึงอินเตอร์เฟซหรือกลไกที่ช่วยให้โมเดลสามารถเรียกใช้งานฟังก์ชันเฉพาะทาง และสามารถเรียกใช้งานผ่าน agent หรือ chain เพื่อเพิ่มขีดความสามารถของโมเดลในการตอบสนองต่อคำถามหรือคำสั่งที่ซับซ้อนมากขึ้น [10]

3.8.2 Agents

Agents ใน LangChain ทำหน้าที่เป็นตัวกลางที่ช่วยให้โมเดลภาษาตัดสินใจเลือกเครื่องมือหรือแหล่งข้อมูลที่เหมาะสมกับคำถามแต่ละประเภท Agents ใช้โมเดลเป็นกลไกการให้เหตุผลช่วยให้ระบบตอบสนองได้อย่างยืดหยุ่น [3]

ตารางที่ 1 โครงสร้างหมวดหมู่และจำนวน chunks ของฐานความรู้

หมวด (Category)	เอกสารย่อย (Sub-documents)	จำนวน chunks
Case Management	• FAQ • Credit Card Promotion • Transportation Policy • Payment Condition • Refund Policy • Product Warranty	7,520
Search Product	• Product Detail	4,855
รวม		12,375

3.9 RAGAS Framework

RAGAS เป็นเฟรมเวิร์กในรูปแบบไลบรารีของภาษา Python สำหรับประเมินคุณภาพของระบบ RAG โดยใช้ตัวชี้วัด เช่น ความถูกต้องของคำตอบ (Answer Correctness) ความแม่นยำในการดึงข้อมูล (Retrieval Precision) ความครอบคลุมของบริบท (Context Recall) และความถูกต้องในการเรียกใช้เครื่องมือภายนอก (Tool Call Accuracy) ซึ่งครอบคลุมทั้งด้านเนื้อหาและประสิทธิภาพของระบบ [11]

4. วิธีดำเนินการวิจัย

4.1 การเก็บรวบรวมข้อมูล (Data Collection)

ในการดำเนินงานระบบนี้ ข้อมูลที่นำมาใช้สามารถจำแนกออกได้เป็น 2 ประเภท ได้แก่ ข้อมูลฐานความรู้ (Knowledge Base) และข้อมูลจากระบบภายนอก (API Data)

4.1.1 ข้อมูลฐานความรู้ (Knowledge Base)

เป็นข้อมูลที่ใช้โดยฝ่ายบริการลูกค้าในการสนับสนุนและตอบคำถามของลูกค้า โดยประกอบด้วยเอกสารหลัก 7 ประเภท และได้มีการจัดเตรียมให้อยู่ในรูปหน่วยข้อมูล (Chunks) เพื่อให้เหมาะสมกับการค้นคืนแบบไฮบริด ทั้งเชิงคำ (Lexical) และเชิงความหมาย (Semantic) รายละเอียดของโครงสร้างและจำนวนข้อมูลแสดงในตารางที่ 1 กระบวนการแบ่งและการเตรียมข้อมูลจะอธิบายในหัวข้อ 4.2.1

4.1.2 ข้อมูลจากระบบภายนอก (API Data)

เป็นข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลาและจำเป็นสำหรับการให้บริการแบบเรียลไทม์ โดยใช้การเชื่อมต่อกับระบบภายนอกผ่าน RESTful API โดยมี Order Status API: ใช้เพื่อตรวจสอบสถานะและข้อมูลของคำสั่งซื้อโดยอ้างอิงจาก orderId

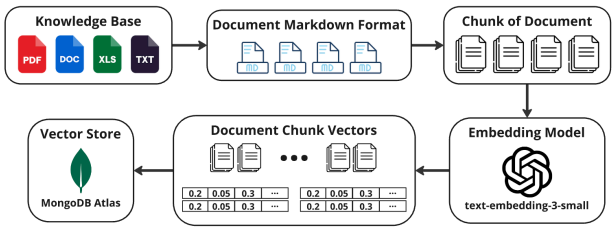
4.2 การค้นคืนข้อมูลแบบไฮบริด (Hybrid Retrieval)

การค้นคืนข้อมูลแบบไฮบริด (Hybrid Retrieval) เป็นแนวทางที่ใช้ในการดึงข้อมูลจากฐานความรู้ (Knowledge Base) ซึ่งมีรายละเอียดของข้อมูลอธิบายไว้ในหัวข้อ 4.1.1 โดยข้อมูลดังกล่าวมีลักษณะเป็นข้อมูลกึ่งโครงสร้างหรือไม่มีโครงสร้าง การค้นคืนข้อมูลในลักษณะนี้จึงต้องผสานการค้นหาตามคำ (Lexical Search) เข้ากับการค้นหาตามความหมาย (Semantic Search) เพื่อเพิ่มความครอบคลุมและความแม่นยำในการดึงเอกสารที่เกี่ยวข้อง

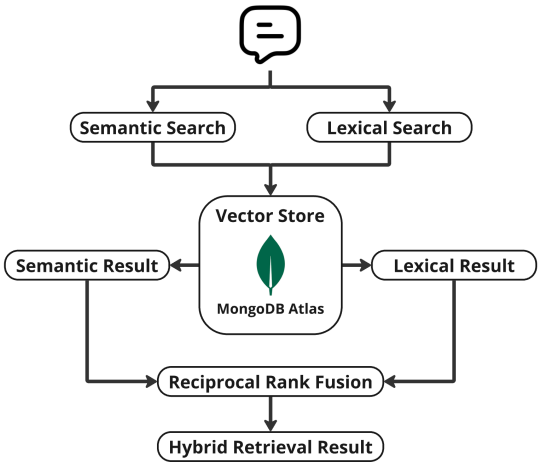
4.2.1 การเตรียมข้อมูลและการฝังเวกเตอร์ (Preprocessing and Embedding)

การค้นคืนข้อมูลแบบไฮบริดต้องอาศัยฐานข้อมูลที่ผ่านการเตรียมให้พร้อมสำหรับการค้นหา ทั้งในเชิงคำและเชิงความหมาย ดังแสดงในรูปที่ 1 โดยมีกระบวนการดังนี้

- การจัดโครงสร้างข้อมูล: แปลงข้อมูลให้อยู่ในรูปแบบ Markdown เพื่อระบุหัวข้อและโครงสร้างข้อความให้ชัดเจน
- การแบ่งข้อมูล (Chunking): แบ่งเนื้อหาออกเป็นหน่วยข้อมูลย่อยที่เรียกว่า chunk ตามหัวข้อหลักและหัวข้อย่อย
- การฝังเวกเตอร์ (Embedding): แปลงข้อมูลแต่ละ chunk เป็นเวกเตอร์เชิงความหมายใช้โมเดล text-embedding-3-small
- การจัดเก็บและสร้างดัชนี: จัดเก็บ chunk และเวกเตอร์ใน MongoDB Atlas พร้อมสร้าง Atlas Vector Index สำหรับการค้นหาความคล้ายเชิงความหมาย และ Atlas Search Index สำหรับการค้นหาข้อความแบบเต็ม



รูปที่ 1 Preprocessing and Embedding



รูปที่ 2 Hybrid Retrieval Process

4.2.2 กระบวนการค้นคืนข้อมูลแบบไฮบริด (Hybrid Retrieval Process)

เมื่อได้รับคำถาม กระบวนการค้นคืนข้อมูลแบบไฮบริดจะเริ่มทำงานโดยใช้ Semantic Search และ Lexical Search ควบคู่กัน เพื่อให้ได้ผลลัพธ์ที่ครอบคลุมทั้งเชิงความหมายและเชิงคำศัพท์ดังแสดงในรูปที่ 2

Semantic Search ทำงานโดย แปลงคำถามเป็นเวกเตอร์เชิงความหมายด้วยโมเดล text-embedding-3-small จากนั้นใช้ Atlas Vector Index ค้นหา chunk ที่มีความหมายใกล้เคียง โดยคำนวณค่าความคล้ายด้วย cosine similarity เพื่อจัดอันดับ chunk ตามความคล้ายทางความหมาย

Lexical Search ทำงานโดย ใช้ Atlas Search Index และอัลกอริทึม BM25 ในการค้นหาเอกสารที่มีคำหรือวลีตรงกับคำถาม และจัดอันดับเอกสารตามความตรงเชิงคำศัพท์

เมื่อได้ผลลัพธ์จากทั้งสองวิธี ระบบจะรวมผลลัพธ์ด้วยเทคนิค Reciprocal Rank Fusion (RRF) โดยคำนวณคะแนนรวมจากลำดับเอกสารของแต่ละวิธี เอกสารที่ติดอันดับสูงจากทั้งสองวิธีจะได้รับคะแนนรวมสูงขึ้นและถูกจัดไว้ในผลลัพธ์ลำดับต้น สุดท้ายเอกสารในลำดับ Top-k จะถูกเลือกเป็นบริบทสำหรับโมเดล RAG เพื่อนำไปสร้างคำตอบที่เหมาะสม

4.3 การค้นคืนข้อมูลผ่าน API (API Retrieval)

การค้นคืนข้อมูลผ่าน API (API Retrieval) เป็นแนวทางสำหรับดึงข้อมูลจากระบบภายนอก (API Data) ซึ่งมีรายละเอียดอธิบายไว้ในหัวข้อ 4.1.2 โดยเป็น RESTful API ซึ่งมีลักษณะเป็นโครงสร้างชัดเจน การทำงานเริ่มจากการรับ parameter ที่กำหนดตามข้อกำหนดของ API จากนั้นส่ง HTTP Request ไปยัง API Endpoint พร้อม parameter ดังกล่าว เมื่อได้รับผลลัพธ์ข้อมูลที่จะถูกส่งต่อพร้อมคำอธิบายของแต่ละฟิลด์ให้ระบบ RAG เพื่อตีความและสร้างคำตอบที่สอดคล้องกับข้อมูลที่ได้รับ

4.4 การพัฒนาโมดูลเครื่องมือสำหรับ AI Agent (Tool Development for AI Agent)

เมื่อการพัฒนากระบวนการ Hybrid Retrieval และ API Retrieval ตามหัวข้อ 4.2 และ 4.3 เสร็จสมบูรณ์ ขั้นตอนต่อไปคือการแปลงกระบวนการเหล่านี้ให้กลายเป็น เครื่องมือ (Tools) สำหรับ AI Agent เครื่องมือเหล่านี้ถูกสร้างขึ้นบนเฟรมเวิร์ก LangChain โดยใช้ชนิด Tool Function ซึ่งทำให้ Agent สามารถเลือกเรียกใช้งาน Retrieval ที่เหมาะสมตามประเภทคำถามได้อย่างอัตโนมัติ

แนวคิดสำคัญคือการ ห่อ (Wrap) ฟังก์ชันการค้นคืน (Retrieval Functions) ให้อยู่ในรูปของ Tool ซึ่งประกอบด้วย 4 ส่วนหลัก ได้แก่

- ฟังก์ชันการทำงาน: ส่วนประมวลผลที่เรียกใช้ฟังก์ชันการค้นคืน และส่งผลลัพธ์กลับไปยัง Agent
- schema: กำหนดโครงสร้างและอธิบายความหมายของอินพุต เพื่อให้ AI agent ส่งข้อมูลเข้ามาได้อย่างถูกต้อง
- name: ชื่อเฉพาะของเครื่องมือที่ใช้ในการระบุและอ้างอิงภายในระบบ เพื่อให้ AI agent เข้าใจบทบาทของเครื่องมือนั้น
- description: คำอธิบายบทบาทของเครื่องมือ ช่วยให้ Agent เลือกใช้อย่างเหมาะสม

เมื่อองค์ประกอบครบถ้วน เครื่องมือสามารถสร้างได้ด้วยฟังก์ชัน tool() ของ LangChain และพร้อมเชื่อมต่อกับ Agent โดยมีตัวอย่างใน รูปที่ 3

```
const caseManagementTool = tool(
  async (input): Promise<string> => {
    const result = await findCase(input.question);
    return result;
  },
  {
    name: 'case_management',
    description: 'ใช้สำหรับค้นหาแนวทางในการจัดการเคส',
    schema: caseSchema
  }
);
```

รูปที่ 3 การสร้าง Tool Function

4.4.1 Hybrid Retrieval Tool

Hybrid Retrieval Tool จำนวน 2 รายการ ถูกสร้างแยกตามหมวดของเอกสารในหัวข้อ 4.1.1 โดยฟังก์ชันการทำงานใช้กระบวนการ Hybrid Retrieval ตามข้อ 4.2 เพื่อค้นหาข้อมูลที่เกี่ยวข้องและส่งผลลัพธ์กลับไปยัง Agent

อินพุตของ Tool กำหนดเป็น query ซึ่งเป็นคำถามที่ Agent ส่งมาหลังจากการวิเคราะห์ข้อความของผู้ใช้

Tools ทั้ง 2 รายการประกอบด้วย (1) Case Management Tool (2) Product Too

4.4.2 API Retrieval Tool

API Retrieval Tool จำนวน 1 รายการ ถูกสร้างจากข้อมูลในหัวข้อ 4.1.2 โดยฟังก์ชันการทำงานคือ API Retrieval ตามข้อ 4.3 เพื่อดึงข้อมูลจากระบบภายนอกและส่งผลลัพธ์กลับไปยัง Agent

อินพุตของ Tool จะขึ้นอยู่กับรูปแบบของแต่ละ API และถูกกำหนดให้สอดคล้องกับข้อมูลที่ต้องการดึงมาใช้งาน

Tools ที่สร้างขึ้นประกอบด้วย (1) Order Status Tool

```
const prompt = ChatPromptTemplate.fromMessages([
  ['system', 'You are a helpful AI assistant.'],
  ['placeholder', 'chat_history'],
  ['human', '{input}'],
  ['placeholder', 'agent_scratchpad']
]);
```

รูปที่ 4 ตัวอย่างการสร้าง prompt

```
const tools = [hybrid_retrieval_tool, api_retrieval_tool];
const llm = new ChatOpenAI({ model: "gpt-4o", temperature: 0 });
const agent = createOpenAIToolsAgent(llm, tools, prompt);
const agentExecutor = new AgentExecutor({ agent, tools });
```

รูปที่ 5 ตัวอย่างการสร้าง Agent และ AgentExecutor

```
const agentWithChatHistory = new RunnableWithMessageHistory({
  runnable: agentExecutor,
  getMessageHistory,
  inputMessagesKey: 'input',
  historyMessagesKey: 'chat_history'
});
```

รูปที่ 6 ตัวอย่างการเชื่อม AgentExecutor เข้ากับ Chat History

```
const answer = await agentWithChatHistory.invoke(
  { input: data.query },
  { configurable: { sessionId: chatSessionId } }
);
```

รูปที่ 7 ตัวอย่างการเรียกใช้งาน Agent

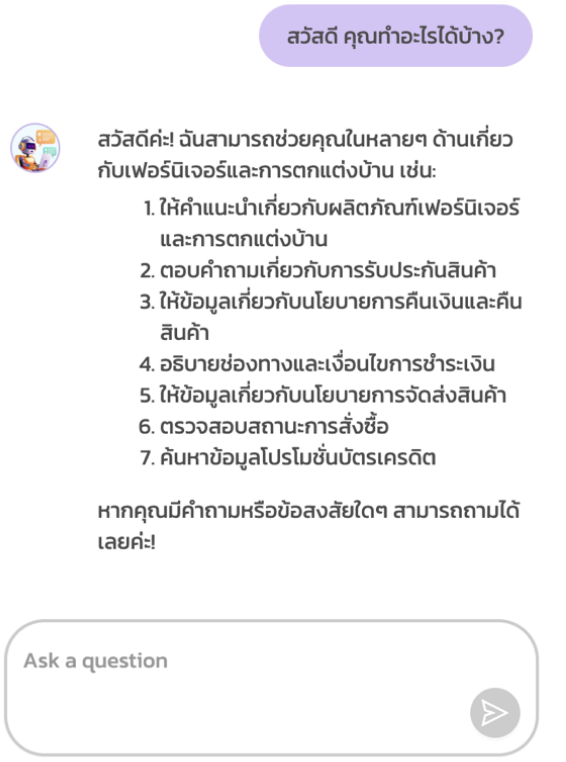
4.5 การพัฒนา Agent (Agent Development)

เมื่อพัฒนาโมดูลเครื่องมือจากข้อ 4.4 เสร็จแล้ว ขั้นตอนถัดมาคือการพัฒนา Agent เพื่อเลือกใช้เครื่องมืออย่างเหมาะสมและตอบคำถามผู้ใช้งานได้อย่างมีประสิทธิภาพ โดยมีขั้นตอนดังนี้

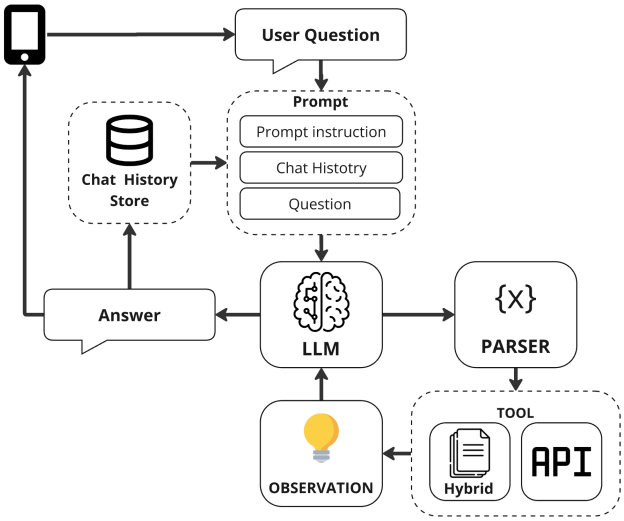
- การสร้าง prompt: เพื่อควบคุมพฤติกรรมและบริบทของ Agent ดังแสดงในรูปที่ 4
- การเชื่อม LLM เข้ากับ Tools: ขั้นตอนนี้คือการสร้าง Agent โดยเชื่อม LLM เข้ากับ Tools และ Prompt ที่เตรียมไว้ จากนั้นนำ Agent ที่ได้มาใช้งานผ่าน AgentExecutor ซึ่งทำหน้าที่เป็นตัวควบคุมการทำงาน ดังแสดงในรูปที่ 5
- การจัดการ Chat History: ทำให้ Agent จำบทสนทนาที่ผ่านมาได้ โดยใช้ RunnableWithMessageHistory ร่วมกับ agentExecutor ดังแสดงในรูปที่ 6
- การเรียกใช้งาน Agent: เมื่อได้ Agent ที่เชื่อมต่อกับ Tools และจัดการประวัติการสนทนาแล้ว สามารถเรียกใช้งานได้โดยส่งคำถามจากผู้ใช้ (Input) และกำหนด sessionId เพื่ออ้างอิงประวัติการสนทนา ดังแสดงในรูปที่ 7

4.6 การออกแบบและพัฒนาส่วนติดต่อผู้ใช้ของระบบแชทบอท

ส่วนติดต่อผู้ใช้ของแชทบอทถูกออกแบบให้เรียบง่าย ใช้งานสะดวก และสื่อสารชัดเจน ด้วยโทนสีอ่อน กล่องข้อความมุมมน และการจัดวางองค์ประกอบอย่างเป็นระเบียบ โดยรองรับการใช้งานบนอุปกรณ์พกพา ดังแสดงในรูปที่ 8



รูปที่ 8 หน้าจอส่วนติดต่อผู้ใช้ของระบบแชทบอท



รูปที่ 9 ภาพรวมของกระบวนการทำงานของระบบ

4.7 กระบวนการทำงานของระบบ (System Workflow)

ระบบแชทบอทถูกออกแบบให้ทำงานอย่างเป็นลำดับขั้นตอน โดยเชื่อมโยงข้อมูลจากทั้งเอกสารภายในและบริการภายนอก เพื่อสร้างคำตอบที่แม่นยำและสอดคล้องกับบริบท ภาพรวมของกระบวนการทำงานแสดงในรูปที่ 9 และสามารถอธิบายเป็นขั้นตอนดังนี้

- รับคำถาม (User Question): ผู้ใช้งานป้อนคำถามผ่านส่วนติดต่อผู้ใช้
- สร้างพรอมป์ (Prompt Creation): พรอมป์ประกอบด้วย 3 ส่วนคือ Prompt Instruction เพื่อกำหนดทิศทางการตอบ Chat History จาก Chat History Store เพื่อรักษาบริบท และ Question ซึ่งเป็นคำถามหลักของผู้ใช้ ก่อนรวมทั้งหมดเพื่อสร้างพรอมป์ที่สมบูรณ์ส่งให้โมเดลภาษา (LLM)
- ประมวลผลโดย LLM (LLM Processing): ใช้ LLM วิเคราะห์คำถามและบริบท เพื่อตัดสินใจว่าจะสร้างคำตอบจากข้อมูลที่มีอยู่ หรือเรียกใช้เครื่องมือที่เหมาะสมแล้วส่งคำสั่งต่อไปยัง Parser
- ประมวลผลคำสั่ง (Parser): ประมวลคำสั่งจาก LLM เพื่อเตรียมเรียกใช้เครื่องมือที่ LLM ระบุ พร้อมจัดเตรียมพารามิเตอร์สำหรับการเรียกใช้งาน
- เรียกใช้เครื่องมือ (Tool Invocation): ระบบดำเนินการเรียกใช้เครื่องมือที่ Parser เตรียมพารามิเตอร์ไว้ตามคำสั่งของ LLM
- การรับผลลัพธ์ (Observation): ระบบรับผลลัพธ์จากเครื่องมือและผสานเข้ากับบริบทการสนทนา ก่อนส่งต่อให้ LLM เพื่อนำไปวิเคราะห์ต่อ
- การสร้างคำตอบ (Answer Generation): LLM ประมวลผลข้อมูลทั้งหมดเพื่อสร้างคำตอบสุดท้ายและบันทึกใน Chat History Store
- การส่งคำตอบ (Answer): ระบบส่งคำตอบกลับไปยังผู้ใช้งาน

4.8 การประเมินระบบแชทบอท (Chatbot Evaluation)

การประเมินระบบแชทบอทมีเป้าหมายเพื่อวิเคราะห์ประสิทธิภาพของแชทบอทที่ขับเคลื่อนด้วย AI Agent โดยครอบคลุม 4 ด้าน ได้แก่ ความแม่นยำในการเรียกใช้เครื่องมือ (Tool Call Accuracy) ความถูกต้องของคำตอบ (Answer Correctness) ความหน่วงเฉลี่ยในการตอบสนอง (Average Latency) และ ความพึงพอใจของผู้ใช้งาน (User Satisfaction) ข้อมูลจากการประเมินทั้งสี่ด้านจะถูกนำไปใช้ปรับปรุงความแม่นยำ ความรวดเร็ว และความสามารถในการตอบสนองของระบบ

สำหรับการประเมินในด้าน Tool Call Accuracy, Answer Correctness และ Average Latency ได้ใช้ชุดคำถามจำนวน 650 ข้อ ที่ครอบคลุม 8 กลุ่มคำถามซึ่งออกแบบมาจากข้อมูลฐานความรู้และ API ได้แก่ FAQ, Credit Card Promotion, Transportation Policy, Payment Condition, Refund Policy, Product Warranty, Product Detail และ Order Status เพื่อให้การทดสอบมีความหลากหลายและครอบคลุมการใช้งานจริง

ส่วนการประเมิน User Satisfaction ใช้กลุ่มตัวอย่างผู้ใช้งาน 2 กลุ่ม ได้แก่ ลูกค้า 50 คน และพนักงานภายในองค์กร 15 คน โดยใช้แบบสอบถาม Likert Scale 5 ระดับ เพื่อสะท้อนประสบการณ์จริงในการใช้งานแชทบอทในด้านความสะดวก ความถูกต้อง และความเหมาะสมกับบริบทงานบริการ

4.8.1 ความแม่นยำในการเรียกใช้เครื่องมือ (Tool Call Accuracy)

Tool Call Accuracy เป็นเมตริกที่ใช้ประเมินความสามารถของโมเดลภาษา (LLM) ในการเลือกและเรียกใช้เครื่องมือได้ตรงกับคำตอบอ้างอิง reference_tool_calls ทั้งในแง่ของชื่อเครื่องมือ ลำดับ และพารามิเตอร์ โดยให้คะแนนระหว่าง 0 ถึง 1 หาก

เรียกผิดลำดับหรือพารามิเตอร์ไม่ตรง ระบบจะให้ค่าคะแนนเป็น 0 เพื่อสะท้อนข้อผิดพลาดในการดำเนินงาน

ในการประเมินนี้ ได้ใช้ Semantic Similarity แทนการจับคู่พารามิเตอร์แบบตรงตัว เพื่อรองรับความคลาดเคลื่อนเชิงความหมาย โดยกำหนดผ่าน arg_comparison_metric ของ ToolCallAccuracy ในไลบรารี RAGAS ตัวอย่างการใช้งานแสดงไว้ในรูปที่ 10

```
user_input = [
    HumanMessage(content="สินค้านี้รับประกันนานเท่าไร"),
    AIMessage(content="", tool_calls=[
        ToolCall(
            name="warranty-info",
            args={"question": "สินค้านี้รับประกันนานเท่าไร"}
        )
    ])
]
reference_tool_calls=[
    ToolCall(
        name="warranty-info",
        args={"question": "สินค้านี้รับประกันนานเท่าไร"}
    )
]

sample = MultiTurnSample(
    user_input=user_input,
    reference_tool_calls=reference_tool_calls
)

metric = ToolCallAccuracy()
metric.arg_comparison_metric = SemanticSimilarity(
    embeddings=generator_embeddings)
score = await metric.multi_turn_ascore(sample)
print(score) #0.9999999999999984
```

รูปที่ 10 รหัสตัวอย่างสำหรับการประเมิน Tool Call Accuracy

4.8.2 ความถูกต้องของคำตอบ (Answer Correctness)

Answer Correctness เป็นเมตริกที่ใช้ประเมินความแม่นยำของคำตอบที่ระบบสร้าง โดยอิงจากการเปรียบเทียบกับคำตอบอ้างอิง ground truth ให้ค่าคะแนนในช่วง 0 ถึง 1 โดยค่าที่สูงกว่าสะท้อนถึงความสอดคล้องของคำตอบกับความเป็นจริงมากยิ่งขึ้น เมตริกนี้พิจารณาสององค์ประกอบหลัก ได้แก่ Semantic Similarity ซึ่งวัดความใกล้เคียงทางความหมาย และ Factual Correctness ซึ่งประเมินความตรงกันของข้อเท็จจริงผ่านแนวคิด True Positive (TP) False Positive (FP) และ False Negative (FN) คะแนนสุดท้ายได้จากการถ่วงน้ำหนักของสององค์ประกอบนี้

4.8.3 ความหน่วงเฉลี่ยในการตอบสนอง (Average Latency)

การประเมินความหน่วงเฉลี่ยในการตอบสนองมีวัตถุประสงค์เพื่อวัดระยะเวลาที่ระบบใช้ในการประมวลผลคำถามและส่งกลับคำตอบ โดยอาศัยชุดคำถามเดียวกับที่ใช้ในการประเมินด้านอื่น เพื่อความสอดคล้องในการเปรียบเทียบผล ระบบจะบันทึกระยะเวลาในการตอบกลับแต่ละคำถามโดยอัตโนมัติ นับตั้งแต่เวลาที่คำถามถูกป้อนเข้าสู่ระบบจนถึงเวลาที่ข้อความตอบกลับถูกส่งสำเร็จ การทดลองดำเนินการภายใต้สภาพแวดล้อมที่ควบคุม เพื่อให้ผลลัพธ์สะท้อนสภาพการใช้งานจริงมากที่สุด ข้อมูลที่ได้จะถูกนำไปคำนวณค่าเฉลี่ยทั้งในภาพรวมและแยกตามประเภทของคำถาม เพื่อวิเคราะห์ประสิทธิภาพของระบบในบริบทที่แตกต่างกัน

4.8.4 ความพึงพอใจของผู้ใช้งานระบบแชทบอท (User Satisfaction)

การประเมินความพึงพอใจของผู้ใช้งานมีวัตถุประสงค์เพื่อสะท้อนประสบการณ์จริงจากการใช้ระบบแชทบอทในด้านความสะดวก ความแม่นยำ และความเหมาะสมกับบริบทการใช้งาน โดยใช้แบบสอบถามในรูปแบบ Likert Scale 5 ระดับ โดยกำหนดคะแนนเป็น 1 = น้อยที่สุด 2 = น้อย 3 = ปานกลาง 4 = มาก และ 5 = มากที่สุด

ในการประเมิน แบ่งกลุ่มผู้ตอบแบบสอบถามออกเป็น ลูกค้า และ พนักงานภายในองค์กร ซึ่งมีลักษณะการใช้งานที่แตกต่างกัน แบบสอบถามแต่ละชุดจึงถูกออกแบบให้เหมาะสมกับบริบทของแต่ละกลุ่ม โดยกลุ่มพนักงานมีขอบเขตการประเมินที่ครอบคลุมมากกว่า เพื่อสะท้อนการใช้งานในเชิงปฏิบัติงานจริง

สำหรับกลุ่มลูกค้า แบบสอบถามประกอบด้วย 11 ข้อ ครอบคลุม 2 ด้านหลัก ได้แก่ ประสบการณ์การใช้งานระบบแชทบอท และ ประสิทธิภาพและคุณภาพของระบบ รายละเอียดแสดงไว้ในตารางที่ 4

สำหรับกลุ่มพนักงาน แบบสอบถามประกอบด้วย 14 ข้อ ครอบคลุม 3 ด้านหลัก ได้แก่ ประสบการณ์การใช้งาน ประสิทธิภาพและคุณภาพของระบบ และความเหมาะสมในการใช้งานในบริบทของงานบริการจริง โดยรายละเอียดของคำถามแสดงไว้ในตารางที่ 5

5. ผลการวิจัย

ผลการประเมินระบบแชทบอทที่พัฒนาขึ้นในงานวิจัยนี้สะท้อนให้เห็นถึงความสามารถของ AI Agent ในการเลือกใช้เครื่องมือให้สอดคล้องกับเจตนาและบริบทของคำถามผู้ใช้งาน รวมถึงการสร้างคำตอบที่มีความถูกต้องและได้รับการยอมรับจากกลุ่มผู้ใช้งานจริง ดังรายละเอียดต่อไปนี้

5.1 ความแม่นยำในการเรียกใช้เครื่องมือ (Tool Call Accuracy)

ผลการประเมินระบบแชทบอทพบว่า ระบบสามารถเลือกใช้เครื่องมือได้อย่างแม่นยำ โดยมีค่า Tool Call Accuracy เฉลี่ยเท่ากับ 0.88 จากจำนวนคำถามทั้งหมด 650 ข้อ

เครื่องมือที่มีค่า Tool Call Accuracy สูงที่สุดคือ Order Status Tool ที่ระดับ 0.95 โดยเป็นผลจากคำถามที่เกี่ยวข้องกับเครื่องมือนี้มักมีโครงสร้างชัดเจนและเจตนาเฉพาะ เช่น การระบุ “หมายเลขคำสั่งซื้อ” ที่ถูกกำหนดไว้เป็นพารามิเตอร์ใน schema ของเครื่องมืออย่างชัดเจน ส่งผลให้ระบบสามารถจับเจตนาและเลือกเครื่องมือได้อย่างแม่นยำและสม่ำเสมอ

Case Management Tool และ Product Tool มีค่าความแม่นยำอยู่ในระดับใกล้เคียงกันที่ 0.85 และ 0.83 ตามลำดับ เนื่องจากทั้งสองเครื่องมือรองรับคำถามที่มีลักษณะไม่เป็นโครงสร้าง เช่น การสอบถามข้อมูลทั่วไปหรือเชิงสถานการณ์ ซึ่งทำให้ระบบต้องอาศัยการตีความเจตนาในเชิงบริบทมากกว่าการจับข้อมูลจากรูปแบบตายตัว อย่างไรก็ตาม ผลลัพธ์สะท้อนว่าโมเดลสามารถจัดการกับข้อมูลประเภทนี้ได้ในระดับที่น่าพึงพอใจ ทั้งนี้ รายละเอียดค่าความแม่นยำในการเรียกใช้เครื่องมือ แสดงไว้ในตารางที่ 2

ตารางที่ 2 Tool Call Accuracy

เครื่องมือ (Tool)	จำนวนคำถาม (ข้อ)	Tool Call Accuracy (0–1)
Case Management Tool	480	0.85
Product Tool	110	0.83
Order Status Tool	60	0.95
รวม / ค่าเฉลี่ยรวม	650	0.88

ตารางที่ 3 Answer Correctness และ Average Latency

เครื่องมือ (Tool)	คำถาม (ข้อ)	Answer Correctness (0–1)	Average Latency (ms)
FAQ	95	0.91	4,842
Credit Card Promotion	70	0.75	5,224
Transportation Policy	65	0.78	6,082
Payment Condition	85	0.81	8,123
Refund Policy	90	0.76	5,885
Product Warranty	75	0.75	11,359
Product Detail	110	0.71	15,734
Order Status	60	0.93	5,011
รวม / ค่าเฉลี่ยรวม	650	0.8	7782.50

5.2 ความถูกต้องของคำตอบ (Answer Correctness)

ผลการประเมินด้าน Answer Correctness พบว่า ระบบสามารถสร้างคำตอบที่สอดคล้องกับข้อมูลจริงและเจตนาของผู้ใช้งานได้อย่างมีประสิทธิภาพ โดยมีค่าเฉลี่ยรวมเท่ากับ 0.80 จากคำถามทั้งหมด 650 ข้อ

กลุ่มคำถาม Order Status มีค่า Answer Correctness สูงที่สุดที่ 0.93 เนื่องจากคำถามในกลุ่มนี้มักมีอินพุตที่ชัดเจน เช่น การระบุหมายเลขคำสั่งซื้อ ทำให้ระบบสามารถนำข้อมูลดังกล่าวไปใช้ในการค้นคืนได้อย่างแม่นยำ อีกทั้งผลลัพธ์ที่ได้จาก API มักอยู่ในรูปแบบโครงสร้างที่เป็นระบบ จึงเอื้อต่อการสร้างคำตอบที่ถูกต้องและสม่ำเสมอ

กลุ่ม FAQ มีค่า Answer Correctness เป็นอันดับรองลงมาที่ 0.91 สะท้อนว่าระบบมีความสามารถในการเข้าถึงข้อมูลพื้นฐานในฐานความรู้และสร้างคำตอบได้อย่างแม่นยำ เนื่องจากคำถามในกลุ่มนี้มักอยู่ในรูปแบบที่เป็นประโยคถาม-ตอบชัดเจน ทำให้การจับเจตนาและการดึงข้อมูลมีความตรงประเด็นและมีความคลาดเคลื่อนต่ำ

ในทางกลับกัน กลุ่มคำถามที่มีลักษณะข้อมูลซับซ้อน เช่น Product Detail, Product Warranty, และ Credit Card Promotion มีค่า Answer Correctness ต่ำกว่ากลุ่มอื่น อยู่ในช่วง 0.71 ถึง 0.75 โดยลักษณะคำถามในกลุ่มนี้มักต้องอาศัยการตีความบริบทในระดับลึกหรือมีความหลากหลายของเนื้อหา

ส่งผลให้ระบบมีโอกาสประมวลผลคลาดเคลื่อนได้มากกว่า สำหรับกลุ่มคำถามอื่น ๆ เช่น Refund Policy, Payment Condition, และ Transportation Policy แม้จะมีระดับความถูกต้องอยู่ในระดับกลาง แต่ยังจัดอยู่ในกลุ่มที่มีลักษณะข้อมูลซับซ้อนเช่นกัน ซึ่งอาจส่งผลต่อความแม่นยำของระบบในลักษณะเดียวกัน

รายละเอียดของผลการประเมินจำแนกตามกลุ่มคำถามแสดงไว้ในตารางที่ 3

5.3 ความหน่วงเฉลี่ยในการตอบสนอง (Average Latency)

ผลการประเมิน Average Latency พบว่าค่าเฉลี่ยรวมอยู่ที่ 7,782.50 มิลลิวินาที โดยสามารถจำแนกกลุ่มคำถามตามช่วงเวลาในการตอบกลับได้ 3 กลุ่มหลัก ดังนี้

- กลุ่มที่มี Latency ต่ำกว่า 6,000 มิลลิวินาที ได้แก่ FAQ, Order Status และ Credit Card Promotion โดยกลุ่มนี้มีแนวโน้มว่าจะใช้ปริมาณข้อมูลในการประมวลผลไม่มากนักและมีความชัดเจนในตัวสูง
- กลุ่มที่มี Latency อยู่ในช่วง 6,000 ถึง 9,000 มิลลิวินาที ได้แก่ Transportation Policy, Refund Policy และ Payment Condition โดยกลุ่มนี้มีแนวโน้มว่าจะต้องใช้ข้อมูลในระดับปานกลาง และมักมีรายละเอียดหรือเงื่อนไขที่ซับซ้อนมากขึ้น
- กลุ่มที่มี Latency สูงกว่า 10,000 มิลลิวินาที ได้แก่ Product Warranty และ Product Detail โดยกลุ่มนี้มีแนวโน้มว่าจะต้องอาศัยข้อมูลจำนวนมากหรือข้อมูลที่กระจายอยู่หลายหมวดหมู่

จากการจัดกลุ่มข้างต้น จะเห็นได้ว่าค่า Latency มีแนวโน้มสัมพันธ์กับปริมาณและความหลากหลายของข้อมูลที่จำเป็นต่อการประมวลผล โดยกลุ่มที่ใช้ข้อมูลเฉพาะเจาะจงและมีจุดเข้าถึงชัดเจน มักใช้เวลาตอบกลับน้อยกว่า

รายละเอียดค่า Average Latency จำแนกตามเครื่องมือแสดงไว้ในตารางที่ 3

5.4 ความพึงพอใจของผู้ใช้งานระบบแชทบอท (User Satisfaction)

ผลการประเมินความพึงพอใจของผู้ใช้งานระบบแชทบอทได้มาจากการสำรวจโดยใช้แบบสอบถามที่พัฒนาขึ้นสำหรับกลุ่มลูกค้าและพนักงาน ตามรายละเอียดในหัวข้อในหัวข้อ 4.8.4 การวิเคราะห์ข้อมูลใช้สถิติพื้นฐานในรูปแบบคะแนนเฉลี่ยและร้อยละเพื่อแสดงระดับความคิดเห็นในแต่ละประเด็นอย่างเป็นระบบ ผลการศึกษาแบ่งการนำเสนอตามกลุ่มผู้ใช้งานเพื่อเปรียบเทียบความแตกต่างของความพึงพอใจตามบริบทการใช้งาน ดังนี้

ตารางที่ 4 ผลการประเมินความพึงพอใจกลุ่มลูกค้า

คำถาม	ผลการประเมิน	
	ค่าเฉลี่ย	ร้อยละ
ด้านประสบการณ์การใช้งานระบบแชทบอท	**4.36**	**87.28**
1. ระบบแชทบอทช่วยให้คุณค้นหาข้อมูลที่ต้องการได้สะดวกหรือไม่	4.22	84.40
2. คุณรู้สึกว่าการใช้งานระบบแชทบอทง่ายและรวดเร็วกว่าการติดต่อเจ้าหน้าที่หรือไม่	4.36	87.20
3. ระบบใช้ถ้อยคำที่สุภาพและเหมาะสมกับการสื่อสาร	4.32	86.40
4. โดยรวมแล้ว คุณพึงพอใจต่อประสบการณ์ใช้งานระบบแชทบอท	4.48	89.60
5. คุณมีแนวโน้มที่จะกลับมาใช้ระบบแชทบอทนี้อีกในอนาคต	4.44	88.80
ด้านประสิทธิภาพและคุณภาพของระบบ	**4.22**	**84.33**
6. คำตอบจากระบบแชทบอทมีความถูกต้องและเชื่อถือได้	4.30	86.00
7. คำตอบที่ได้รับจากระบบเข้าใจง่าย ไม่ซับซ้อน	4.42	88.40
8. ระบบตอบกลับคำถามของคุณได้รวดเร็วโดยไม่ต้องรอนาน	4.02	80.40
9. ระบบสามารถเชื่อมโยงบทสนทนาและคำถามก่อนหน้าได้อย่างต่อเนื่อง	4.14	82.80
10. ระบบสามารถเข้าใจและตอบคำถามได้ แม้ในกรณีที่พิมพ์ผิดหรือใช้ภาษาที่ไม่ชัดเจน	4.20	84.00
11. คุณพึงพอใจในคุณภาพของคำตอบที่ได้รับจากแชทบอท	4.22	84.40
รวมผลคะแนน	**4.28**	**85.67**

5.4.1 ผลการประเมินความพึงพอใจกลุ่มลูกค้า

จากการประเมินผ่านแบบสอบถามของกลุ่มลูกค้าจำนวน 50 คน พบว่าคะแนนเฉลี่ยโดยรวมของการประเมินระบบแชทบอทอยู่ที่ 4.28 คะแนน คิดเป็นร้อยละ 85.67 ซึ่งสะท้อนถึงระดับความพึง

พอใจที่สูงต่อระบบแชทบอททั้งในภาพรวม ทั้งในมิติประสบการณ์การใช้งานและประสิทธิภาพของระบบ

มิติที่ได้รับการประเมินสูงสุดคือประสบการณ์การใช้งานระบบแชทบอท ซึ่งมีคะแนนเฉลี่ย 4.36 คะแนน คิดเป็นร้อยละ 87.28 โดยเฉพาะในประเด็นความพึงพอใจโดยรวมและแนวโน้มการใช้งานต่อเนื่องในอนาคต ซึ่งสื่อถึงทัศนคติเชิงบวกของลูกค้าต่อความสะดวกสบายและความเหมาะสมของระบบแชทบอทในการใช้งานจริง

สำหรับมิติประสิทธิภาพและคุณภาพของระบบมีคะแนนเฉลี่ย 4.22 คะแนน คิดเป็นร้อยละ 84.33 ทั้งนี้ ตัวชี้วัดที่ได้รับคะแนนต่ำสุดคือความเร็วในการตอบสนองคำถาม ซึ่งมีคะแนนเฉลี่ย 4.02 คะแนน คิดเป็นร้อยละ 80.40 แม้ว่าคะแนนดังกล่าวจะอยู่ในระดับที่น่าพึงพอใจ แต่ถือเป็นประเด็นที่สมควรได้รับการพิจารณาเพื่อการปรับปรุงและพัฒนาต่อไป รายละเอียดของผลการประเมินแสดงในตารางที่ 4

5.4.2 ผลการประเมินความพึงพอใจกลุ่มพนักงาน

จากการประเมินผ่านแบบสอบถามของกลุ่มพนักงานทั้ง 15 คน พบว่าค่าเฉลี่ยโดยรวมของระบบแชทบอทมีค่าเท่ากับ 4.31 คะแนน คิดเป็นร้อยละ 86.16 ซึ่งจัดอยู่ในระดับสูง ผลการประเมินนี้สะท้อนให้เห็นว่าระบบแชทบอทสามารถสนับสนุนการปฏิบัติงานของพนักงานได้อย่างมีประสิทธิภาพ โดยเฉพาะในด้านการลดภาระงานซ้ำซ้อนและการเพิ่มความสะดวกในการให้บริการแก่ผู้ใช้งาน

ในด้านประสบการณ์การใช้งาน ระบบได้รับคะแนนเฉลี่ยสูงสุดที่ 4.43 คะแนน คิดเป็นร้อยละ 88.50 โดยเฉพาะในหัวข้อความสุภาพของภาษาและความพึงพอใจโดยรวม ซึ่งชี้ให้เห็นว่าระบบสามารถตอบสนองต่อการใช้งานในบริบทงานบริการได้อย่างเหมาะสม

สำหรับด้านความเหมาะสมในการใช้งาน ระบบได้รับคะแนนเฉลี่ย 4.38 คะแนน คิดเป็นร้อยละ 87.60 โดยประเด็นการเรียนรู้และใช้งานได้ง่ายโดยไม่ต้องใช้เวลาอบรมนานได้รับคะแนนเฉลี่ยสูงสุดในหมวดนี้ที่ 4.61 คะแนน คิดเป็นร้อยละ 92.20 ซึ่งแสดงถึงศักยภาพของระบบในการปรับใช้กับงานจริงโดยไม่สร้างภาระเพิ่มเติมแก่พนักงาน

อย่างไรก็ตาม ประเด็นความเร็วในการตอบกลับของระบบได้รับคะแนนเฉลี่ยต่ำสุดที่ 3.96 คะแนน คิดเป็นร้อยละ 79.20 จึงควรได้รับการพัฒนาเพิ่มเติมเพื่อเสริมสร้างความคล่องตัวในการใช้งานระหว่างการปฏิบัติงาน รายละเอียดของผลการประเมินแสดงในตารางที่ 5

6. สรุปผลการวิจัย

การศึกษานี้มีวัตถุประสงค์เพื่อพัฒนาระบบแชทบอท บริการลูกค้าที่มีความสามารถในการวิเคราะห์เจตนาและบริบทของคำถามผู้ใช้งาน พร้อมทั้งเลือกใช้เครื่องมือที่เหมาะสมโดยอัตโนมัติ ระบบบูรณาการการทำงานระหว่างโมเดลภาษาขนาดใหญ่ (LLM) และเครื่องมือสองประเภท ได้แก่ เครื่องมือการค้นคืนแบบผสม (Hybrid Retrieval Tool) สำหรับการสืบค้นข้อมูลจากฐานความรู้ และเครื่องมือการเรียกใช้ API (API Retrieval Tool) สำหรับการดึงข้อมูลจากระบบภายนอกแบบเรียลไทม์ โดยใช้เฟรมเวิร์ก LangChain เป็นกลไกหลักในการควบคุม agent และจัดการการเรียกใช้งานเครื่องมือ

ผลการประเมินประสิทธิภาพแสดงให้เห็นว่าระบบมีค่าเฉลี่ย Tool Call Accuracy อยู่ในระดับสูง โดย Order Status Tool ซึ่งเป็น API Retrieval Tool แสดงความแม่นยำสูงสุด เนื่องจากโครงสร้างคำถามและพารามิเตอร์ที่ชัดเจน ส่วน Case Management Tool และ Product Tool ซึ่งเป็น Hybrid Retrieval Tool ที่รองรับคำถามไม่มีโครงสร้างคงที่และต้องตีความบริบท ยังคงสามารถเลือกใช้ได้ถูกต้องแม้จะต้องประมวลผลข้อมูลที่หลากหลายและไม่แน่นอน

ในด้านความถูกต้องของคำตอบ (Answer Correctness) กลุ่มคำถาม Order Status และ FAQ ซึ่งมีข้อมูลชัดเจนและมีโครงสร้าง ได้ผลลัพธ์ถูกต้องในระดับสูง ขณะที่กลุ่ม Product Detail, Product Warranty และ Credit Card Promotion ที่มีบริบทซับซ้อนและหลากหลายมีระดับความถูกต้องต่ำกว่าอย่างไรก็ตาม ค่าเฉลี่ยโดยรวมยังอยู่ในระดับที่น่าพึงพอใจและสะท้อนศักยภาพของระบบในการประมวลผลคำถามที่หลากหลายภายใต้ข้อจำกัดทางบริบทและรูปแบบข้อมูล

สำหรับประสิทธิภาพเชิงเวลา ระบบมีค่าเฉลี่ย Average Latency 7,782 มิลลิวินาที โดยคำถามที่มีข้อมูลจำเพาะใช้เวลา

ประมวลผลน้อยกว่าคำถามที่ต้องใช้ข้อมูลหลายแหล่งหรือมีความซับซ้อน โดยค่า Average Latency ต่ำสุดอยู่ที่ 4,842 มิลลิวินาทีในกลุ่มคำถาม FAQ ซึ่งมีรูปแบบข้อมูลคงที่และโครงสร้างชัดเจน ขณะที่ค่า Average Latency สูงสุดอยู่ที่ 15,734 มิลลิวินาทีในกลุ่ม Product Detail ซึ่งต้องดึงข้อมูลปริมาณมากและมีความซับซ้อนสูง แนวโน้มนี้สอดคล้องกับผลประเมินความพึงพอใจที่พบว่า "ความเร็วในการตอบกลับ" ได้คะแนนต่ำสุดทั้งในกลุ่มลูกค้าที่ให้คะแนน 4.02 คิดเป็น 80.40% และกลุ่มพนักงานที่ให้คะแนน 3.96 คิดเป็น 79.20% แม้จะยังอยู่ในเกณฑ์ยอมรับได้ แต่เป็นจุดที่ควรปรับปรุง

จากการประเมินโดยใช้ค่าเฉลี่ยเวลาในการตอบคำถาม (Average Latency) ที่ 7.782 มิลลิวินาทีต่อคำถาม และพิจารณาว่าในสถานการณ์ใช้งานจริงจะมีช่วงเวลาที่ไม่มีคำถามเข้ามาอย่างต่อเนื่อง ระบบจึงอาจใช้เวลาประมวลผลจริงได้ประมาณ 60–70% ของเวลาทั้งหมด โดยระบบแชทบอทสามารถตอบคำถามได้ประมาณ 6,000–7,000 คำถามต่อวัน ซึ่งสะท้อนศักยภาพของระบบในการรองรับปริมาณคำถามในระดับสูง

ผลการประเมินความพึงพอใจของผู้ใช้งานทั้งสองกลุ่มอยู่ในระดับสูง โดยลูกค้าให้คะแนนเฉลี่ย 85.67% และพนักงาน 86.16% โดยเฉพาะในมิติประสบการณ์การใช้งาน ความสะดวกในการโต้ตอบ และความเข้าใจง่ายของคำตอบ ทั้งนี้ ความสอดคล้องในการรับรู้คุณภาพของระบบแสดงให้เห็นถึงประสิทธิภาพของระบบจากมุมมองของผู้ใช้งานที่แตกต่างกัน

ระบบแชทบอทที่พัฒนาขึ้นมีศักยภาพในการประยุกต์ใช้งานจริง อย่างไรก็ตาม ยังมีข้อจำกัดในการรองรับฟังก์ชันครบถ้วนของระบบ CRM การพัฒนาต่อยอดควรเน้นการปรับปรุงประสิทธิภาพเชิงเวลาและขยายความสามารถให้ครอบคลุมฟังก์ชัน CRM ให้มากขึ้น

ตารางที่ 5 ผลการประเมินความพึงพอใจกลุ่มพนักงาน

คำถาม	ผลการประเมิน	
	ค่าเฉลี่ย	ร้อยละ
ด้านประสบการณ์การใช้งานระบบแชทบอท	**4.43**	**88.50**
1. คุณรู้สึกว่าการใช้งานระบบแชทบอทง่ายและช่วยให้ทำงานได้รวดเร็วกว่าการใช้ช่องทางเดิม	4.22	84.40
2. ระบบใช้ถ้อยคำที่สุภาพและเหมาะสมกับการสื่อสารในงานบริการ	4.57	91.40
3. โดยรวมแล้ว คุณพึงพอใจต่อประสบการณ์ใช้งานระบบแชทบอทในการทำงาน	4.48	89.60
4. คุณมีแนวโน้มที่จะเลือกใช้ระบบแชทบอทนี้ต่อไปในอนาคต	4.43	88.60
ด้านประสิทธิภาพและคุณภาพของระบบ	**4.18**	**83.63**
5. ระบบแชทบอทให้ข้อมูลที่ถูกต้องและเชื่อถือได้	4.22	84.40
6. ข้อมูลหรือคำตอบจากระบบเข้าใจง่ายและนำไปใช้งานได้ทันที	4.39	87.80
7. ระบบตอบสนองต่อคำถามของคุณได้อย่างรวดเร็วโดยไม่ล่าช้า	3.96	79.20
8. ระบบสามารถเชื่อมโยงข้อมูลหรือบทสนทนาที่เกี่ยวข้องในงานบริการได้อย่างต่อเนื่อง	4.22	84.40
9. ระบบสามารถเข้าใจและจัดการกับคำถามที่คลุมเครือ หรือข้อมูลที่ไม่ชัดเจนได้ดี	4.13	82.60
10. คุณพึงพอใจในคุณภาพของข้อมูลที่ได้รับจากระบบในการปฏิบัติงาน	4.17	83.40
ด้านความเหมาะสมในการใช้งาน	**4.38**	**87.60**
11. ระบบสามารถบูรณาการเข้ากับงานประจำวันของคุณได้อย่างราบรื่น	4.39	87.80
12. ระบบเรียนรู้และใช้งานได้ง่ายโดยไม่ต้องอบรมนาน	4.61	92.20
13. ระบบช่วยให้คุณทำงานได้สะดวกกว่ารูปแบบเดิมที่เคยใช้	4.52	90.40
14. ฟีเจอร์ที่ระบบมีเพียงพอกับความต้องการในงานของคุณ	4.00	80.00
รวมผลคะแนน	**4.31**	**86.16**

7. เอกสารอ้างอิง

[1] C. C. Chang, W. S. Cheng, and S. Hsiao, "Customer Service Chatbot Enhanced with Conversational Language Understanding and Knowledge Base". *IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE). 28-30 October. Yunlin, Taiwan* : pp. 231-234, 2022. doi: 10.1109/ECICE55674.2022.10042940.

[2] H. C. Lee, K. Hung, G. M. T. Man, R. Ho, and M. Leung, "Development of an RAG-Based LLM Chatbot for Enhancing Technical Support Service". *IEEE Region 10 Conference (TENCON). 1-4 December. Singapore* : pp. 1080-1083, 2024. doi: 10.1109/TENCON61640.2024.10902801

[3] M. Guettala, S. Bourekkache, O. Kazar, and S. Harous, "Building Advanced RAG Q&A with Multiple Data Sources Using Langchain: A Multi-Search Agent RAG Application in Ubiquitous Learning". *International Conference on Computing and Data Analytics (ICCDA). 12-13 November. Shinas, Oman* : pp. 1-7, 2024. doi: 10.1109/ICCDA64887.2024.10867361

[4] C. P. Ezenkwu, "Towards Expert Systems for Improved Customer Services Using ChatGPT as an Inference Engine". *International Conference on Digital Applications, Transformation & Economy (ICDATE). 14-16 July, Miri, Sarawak, Malaysia* : pp. 1-5, 2023. doi: 10.1109/ICDATE58146.2023.10248647

[5] T. Zheng et al. (2 May 2025). *From Automation to Autonomy: A Survey on Large Language Models in Scientific Discovery*. [Online] Available : http://arxiv.org/abs/2505.13259

[6] MongoDB Inc. (14 July 2025). *How to Perform Hybrid Search*. [Online] Available : https://www.mongodb.com/docs/atlas/atlas-vector-search/hybrid-search

[7] MongoDB Inc. (14 July 2025). *Perform Hybrid Search with Atlas Vector Search and Atlas Search*. [Online] Available : https://www.mongodb.com/docs/atlas/atlas-vector-search/hybrid-search/vector-search-with-full-text-search

[8] OpenAi. (10 April 2025). *New Embedding Models and API Updates*. [Online] Available : https://openai.com/index/new-embedding-models-and-api-updates

[9] X. Dong, X. Zhang, W. Bu, D. Zhang, and F. Cao, "A Survey of LLM-based Agents: Theories, Technologies, Applications and Suggestions". *International Conference on Artificial Intelligence, Internet of Things and Cloud Computing Technology (AIoTC). 13-15 September. Wuhan, China* : pp. 407-413, 2024. doi: 10.1109/AIoTC63215.2024.10748304

[10] LangChain. (14 July 2025). *Tools*. [Online] Available : https://js.langchain.com/docs/concepts/tools

[11] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert. (2 May 2025). *RAGAS: Automated Evaluation of Retrieval Augmented Generation*. [Online] Available : http://arxiv.org/abs/2309.15217