การวิจัยและพัฒนาระบบสนทนาอัตโนมัติในข้อมูลป้องกันประเทศ

ชนาธิป ชื่นมนัส [1*] วรากร เลื่องลือวุฒิ [1] กิตตากร วิริยะศาสตร์ [1, 2]
ผกามาศ วงค์สาย [1] อุบล ธงสถาพรวัฒนา [1] วิชัย แผ้วเกษม [1] และ พันธุ์เทพ แก้วมงคล [1]

วันที่รับ 24 เมษายน 2567 วันที่แก้ไข 24 มิถุนายน 2567 วันตอบรับ 24 กรกฎาคม 2567

บทคัดย่อ

จักรกลสนทนา (Chatbot) เป็นปัญญาประดิษฐ์ (AI) ประเภทหนึ่งซึ่งอยู่ในสาขาการประมวลภาษาธรรมชาติ (NLP) เป็นเทคโนโลยีส่วนการเรียนรู้ของเครื่องที่ช่วยให้คอมพิวเตอร์สามารถตีความ จัดการ และทำความเข้าใจภาษามนุษย์ได้ องค์กรในปัจจุบันมีข้อมูลเสียงและข้อความจำนวนมากจากช่องทางการสื่อสารต่าง ๆ เช่น อีเมล ข้อความ ฟีดข่าวโซเชียลมีเดีย วิดีโอ เสียง และอื่น ๆ โดยใช้ซอฟต์แวร์ NLP เพื่อประมวลผลข้อมูลนี้โดยอัตโนมัติ ในปัจจุบันโมเดลภาษาขนาดใหญ่ (Large Language Model: LLM) ซึ่งเป็นเทคโนโลยีที่ถูกฝึกด้วยข้อมูลขนาดใหญ่เพื่อนำไปใช้ในการสร้างข้อความคล้ายกับที่มนุษย์เขียนหรือพูดได้ สามารถสร้างข้อความที่มีความสมเหตุสมผลและเป็นธรรมชาติเหมือนที่มนุษย์สร้าง แต่เนื่องจาก LLM ถูกเทรนมาจากข้อมูลที่มีอยู่จำนวนมาก การตอบคำถามที่อยู่บนพื้นฐานข้อมูลที่เฉพาะเจาะจงและเป็นปัจจุบันอาจไม่ถูกต้องแม่นยำ ซึ่งเรียกว่าเกิดการ “Hallucination” การเกิด Hallucination นี้เป็นความท้าทายสำคัญในการพัฒนาและปรับปรุงโมเดลภาษาขนาดใหญ่ เพราะอาจทำให้ผู้ใช้ได้รับข้อมูลที่ผิดพลาดและเกิดความเข้าใจผิด วัตถุประสงค์ในงานวิจัยนี้เพื่อทดลองระบบจักรกลสนทนากับข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศ ว่าสามารถให้ความรู้และตอบคำถามแก่บุคคลภายนอกเกี่ยวกับข้อมูลของเทคโนโลยีป้องกันประเทศได้แม่นยำเพียงใด โดยใช้เทคโนโลยีการดึงข้อมูลเชิงเพิ่มประสิทธิภาพ (Retrieval-Augmented Generation: RAG) ที่มีอยู่แล้วนำมาประยุกต์ใช้เพื่อแก้ปัญหา Hallucination ใน LLM ผลการทดลองพบว่า ระบบจักรกลสนทนาที่ใช้เทคโนโลยี RAG มีประสิทธิภาพในการตอบคำถามได้อย่างถูกต้องถึง 93% ซึ่งสามารถตอบคำถามที่มีข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศได้ถูกต้องดีมาก

คำสำคัญ : จักรกลสนทนา, ปัญญาประดิษฐ์, การประมวลภาษาธรรมชาติ, โมเดลภาษาขนาดใหญ่, การดึงข้อมูลเชิงเพิ่มประสิทธิภาพ

---

[1] ส่วนงานวิศวกรรมการสื่อสารข้อมูลทางอิเล็กทรอนิกส์และเครือข่ายคอมพิวเตอร์, สถาบันเทคโนโลยีป้องกันประเทศ
[2] ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยธรรมศาสตร์
* ผู้แต่ง, อีเมล: chanatip.c@dti.or.th

Research and Development of Chatbot in National Defense Information

Chanatip Chuenmanus [1*] Warakorn Luangluewut [1] Kittakorn Viriyasatr [1, 2]
Pakamaj Wongsai [1] Ubon Thongsatapornwatana [1] Wichai Pawgasame [1] and Pantape Kaewmongkol [1]

Received 24 April 2024, Revised 24 June 2024, Accepted 24 July 2024

Abstract

A chatbot is a type of Artificial Intelligence (AI) that falls within the field of Natural Language Processing (NLP). It is a machine learning technology that enables computers to interpret, manage, and understand human language. Nowadays, organizations have large amounts of data from various communication channels such as emails, text messages, social media news feeds, videos, audio, and more. They use NLP software to automatically process this data. Currently, Large Language Models (LLMs), which are technologies trained with large datasets to generate text similar to what humans write or speak, can create reasonable and natural - sounding text like that produced by humans. However, because LLMs are trained on vast amounts of existing data, their responses to specific, current information may not always be accurate, a phenomenon known as “Hallucination”. This hallucination is a significant challenge in the development and improvement of LLMs, as it may lead to users receiving incorrect information and developing misunderstandings. The purpose of this research is to experiment with a chatbot using information from the DTech (Thailand Defence Technology Journal) to determine its accuracy in providing knowledge and answering questions about defense technology information to external individuals. The chatbot applies Retrieval-Augmented Generation (RAG) technology to address the issue of hallucinations in large language models (LLMs). The experimental results showed that the chatbot utilizing RAG technology was highly effective, accurately answering questions 93% of the time, demonstrating a high level of correctness in responding to questions based on the DTech articles.

Keywords : Chatbot, Artificial intelligence, Natural language processing, Large language model, Retrieval-augmented generation

[1] Data Communication Division, Defence Technology Institute
[2] Department of Electrical and Computer Engineering, Faculty of Engineering, Thammasat University
[*] Corresponding author, E-mail: chanatip.c@dti.or.th

1. บทนำ

ในปัจจุบันเทคโนโลยีในด้านปัญญาประดิษฐ์ หรือ Artificial Intelligence (AI) มีบทบาทในโลกของเราอย่างมากมาย ซึ่งหนึ่งในนั้นที่กำลังเป็นที่นิยมอยู่ คือ จักรกลสนทนา (Chatbot) ซึ่งเป็น AI ประเภทหนึ่งในสาขาการประมวลภาษาธรรมชาติ หรือ Natural Language Processing (NLP) เป็นเทคโนโลยี Machine Learning ที่ช่วยให้คอมพิวเตอร์สามารถตีความ จัดการ และทำความเข้าใจภาษาของมนุษย์ได้ องค์กรในปัจจุบันมีข้อมูลเสียงและข้อความจำนวนมากจากช่องทางการสื่อสารต่าง ๆ เช่น อีเมล ข้อความ ฟีดข่าวโซเชียลมีเดีย วิดีโอ เสียง และอื่น ๆ โดยใช้ซอฟต์แวร์ NLP เพื่อประมวลผลข้อมูลนี้โดยอัตโนมัติ เพื่อวิเคราะห์เจตนาหรือความเชื่อมั่นในข้อความ และตอบสนองการสื่อสารของมนุษย์

โมเดลภาษาขนาดใหญ่ หรือ Large Language Model (LLM) เป็นโมเดลปัญญาประดิษฐ์ประเภทหนึ่งที่ได้รับการฝึกฝนบนชุดข้อมูลข้อความขนาดใหญ่ โมเดลเหล่านี้ สามารถเรียนรู้รูปแบบและความสัมพันธ์ระหว่างคำ วลี และประโยค ซึ่งทำให้สามารถทำงานประมวลผลภาษาธรรมชาติได้หลากหลาย โดยงานหนึ่งที่ LLM สามารถทำได้ดี คือ การตอบคำถาม ซึ่งเหมาะกับจุดประสงค์ในงานวิจัยนี้ แต่ในปัจจุบัน LLM ยังมีข้อจำกัดและปัญหาสำคัญคือ

1. การเกิดอาการหลอน (Hallucination)

หมายถึง การที่ LLM สร้างข้อความที่ไม่ถูกต้องเท็จ หรือไร้เหตุผล แม้ว่าจะดูเหมือนสมเหตุสมผลก็ตาม จะทำให้มนุษย์เกิดการเข้าใจผิดว่าเป็นข้อมูลที่ถูกต้อง แล้วหากขาดการตรวจสอบ สาเหตุหลัก ๆ ที่ทำให้เกิด Hallucination อาจเกิดจากข้อมูลที่ใช้เรียนรู้มีข้อมูลที่ผิดพลาด เป็นเท็จ มีอคติ หรือไม่สมบูรณ์ ขาดความเข้าใจบริบทของข้อความ มีความคลุมเครือของภาษา

2. ปัญหาข้อมูลที่ LLM เรียนรู้ไม่เป็นปัจจุบัน

การที่ข้อมูลที่ LLM เรียนรู้ไม่เป็นปัจจุบัน อาจส่งผลต่อประสิทธิภาพและความถูกต้องของโมเดล สาเหตุหลักที่ข้อมูลไม่เป็นปัจจุบัน เพราะข้อมูลมีการเปลี่ยนแปลงอยู่เสมอ ข้อมูลใหม่ ๆ ถูกสร้างขึ้น ข้อมูลเก่าถูกลบ ข้อมูลที่มีอยู่ถูกแก้ไข LLM ที่ได้รับการฝึกฝนเกี่ยวกับข้อมูลที่เก่าอาจไม่สามารถสร้างผลลัพธ์ที่ถูกต้องหรือเป็นประโยชน์

จากปัญหาดังกล่าวข้างต้นจึงมีการคิดค้นเทคนิคที่ช่วยเพิ่มศักยภาพของ LLM ที่มีชื่อว่า Retrieval - Augmented Generation (RAG) โดยการเพิ่มคลังความรู้ที่เฉพาะเจาะจงและน่าเชื่อถือให้แก่ LLM ให้กลายเป็นผู้เชี่ยวชาญที่ครอบคลุมความรู้ในสายงานนั้น ๆ ตามข้อมูลที่เพิ่ม ส่งผลให้ LLM ทำงานได้อย่างมีประสิทธิภาพดีขึ้น

1. แก้ปัญหาการเกิด Hallucination

RAG ช่วยให้ LLM ตรวจสอบข้อเท็จจริงของข้อความที่สร้างขึ้น โดยเปรียบเทียบข้อความกับข้อมูลในแหล่งความรู้ภายนอก LLM สามารถระบุและแก้ไขข้อผิดพลาดในข้อความหรือปฏิเสธข้อความที่ไม่ถูกต้อง RAG ช่วยให้ LLM เข้าใจบริบทของข้อความ โดยดึงข้อมูลเพิ่มเติมจากแหล่งความรู้ภายนอก LLM สามารถสร้างข้อความที่สอดคล้องกับบริบทของการสนทนาและมีความหมายมากขึ้น

2. แก้ปัญหาข้อมูลที่ LLM เรียนรู้ไม่เป็นปัจจุบัน

RAG ช่วยให้ LLM เข้าถึงข้อมูลที่เป็นปัจจุบันจากแหล่งความรู้ภายนอก เช่น ฐานข้อมูล เอกสาร เว็บไซต์ และ API ข้อมูลเหล่านี้ช่วยให้ LLM เรียนรู้ข้อมูลใหม่ ๆ และอัปเดตข้อมูลที่มีอยู่ให้ทันสมัย

ในงานวิจัยนี้มีวัตุประสงค์เพื่อทดลองระบบจักรกลสนทนาที่ทำการเรียนรู้ข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศ หรือ Thailand Defence Technology Journal (DTech) ที่จัดทำโดยสถาบันเทคโนโลยีป้องกันประเทศ โดยทำการสร้าง RAG จากข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศจำนวน 5 เล่ม แต่ละเล่มจะคัดบทความทั้งบทที่เกี่ยวข้องกับเทคโนโลยีป้องกันประเทศรวมแล้วได้ 22 บทความ จากนั้นจึงทำการทดสอบกับโมเดลแบบต่าง ๆ ที่มีการกำหนดพารามิเตอร์ต่างกัน เลือกโมเดลที่มีประสิทธิภาพในการตอบคำถามได้ดีที่สุดเพื่อจะนำไปสร้าง Chatbot ในการตอบคำถามของบุคคลทั่วไปที่ถามเกี่ยวกับข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศได้

2. ทฤษฎีที่เกี่ยวข้อง

2.1 Large Language Model (LLM)

การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) เป็นสาขาย่อยของปัญญาประดิษฐ์ (AI) ที่มุ่งเน้นไปที่ปฏิสัมพันธ์ระหว่างคอมพิวเตอร์กับภาษาของมนุษย์ โดยเฉพาะอย่างยิ่งโมเดล NLP ช่วยให้คอมพิวเตอร์เข้าใจ ตีความ และสร้างข้อความหรือคำพูดที่เหมือนมนุษย์ได้โมเดลภาษาขนาดใหญ่ (Large Language Model: LLM) เป็นโมเดล NLP ขั้นสูงภายในหมวดหมู่ของโมเดลภาษาที่ได้รับการฝึกล่วงหน้า (Pre-trained Language Models: PLM) ประสบความสำเร็จในด้านการปรับขนาดโมเดล คลังข้อมูลที่ถูกการฝึกล่วงหน้า และทรัพยากรการคำนวณ [1]

Large Language Model (LLM) ประมวลผลข้อมูลและรูปแบบบทสนทนาได้คล้ายกับภาษาของมนุษย์ เป็นโมเดลที่มีขนาดใหญ่มากที่ผ่านการฝึกฝนและวิเคราะห์ข้อมูลที่เกี่ยวกับข้อความเป็นจำนวนมากจากแหล่งข้อมูลทั้งเอกสาร หนังสือ บทความ รวมถึงข้อมูลบนอินเทอร์เน็ต LLM สามารถเรียนรู้และเชื่อมต่อคำได้คล้ายกับมนุษย์ อีกทั้งยังสามารถตีความจากข้อความในประโยคที่ไม่ชัดเจน โดยอาศัยการคาดเดาและการวิเคราะห์จากรูปแบบประโยค ทำให้การสื่อสารเป็นไปได้อย่างธรรมชาติ นอกจากนี้ LLM ยังสามารถขยายขนาด ปรับตัว และมีความยืดหยุ่นสูง โดยใช้การตั้งค่าเพียงเล็กน้อยหรือแทบไม่ต้องปรับการตั้งค่าเลย [2]

LLM ได้รับการพัฒนาผ่านวิธีการเรียนรู้เชิงลึก โดยเฉพาะอย่างยิ่งการใช้สถาปัตยกรรม Transformer โดยตัว Transformer นี้เป็นเทคนิคที่ถูกพัฒนาขึ้นมาโดย Google ในปี 2017 [3]

ตัวอย่างของ LLM เช่น ChatGPT [4], Codex [5], PaLM [6] ที่ก่อให้เกิดความสามารถในการประมวลผลภาษาธรรมชาติที่น่าสนใจ รวมถึงความสามารถในการปรับขนาดที่นำไปสู่การยอมรับ LLM ในงาน NLP ที่หลากหลาย อย่างไรก็ตาม LLM ส่วนใหญ่สามารถเข้าถึงได้ในรูปแบบกล่องดำ (Black Box) เท่านั้น ซึ่งเป็นเพราะมันไม่ใช่ซอฟต์แวร์โอเพนซอร์สและเก็บไว้เป็นส่วนตัว เช่น ChatGPT เป็นต้น นอกจากนี้ ระดับพารามิเตอร์ขนาดใหญ่ต้องใช้ทรัพยากรในการคำนวณมาก ซึ่งผู้ใช้ไม่อาจจะหาซื้อทรัพยากรเหล่านี้ได้เสมอไป แนวทางในการสร้าง LLM มี 2 แนวทางหลัก ดังนี้

1. Pre-trained Model เหมาะสำหรับการเริ่มต้นใช้ LLM เนื่องจากเป็นวิธีที่ง่ายและประหยัดเวลา อย่างไรก็ตาม ควรพิจารณาปัจจัยต่าง ๆ เช่น ขนาดของโมเดลและทรัพยากรระบบที่ต้องการใช้งาน ค่าใช้จ่ายที่เกิดขึ้นจากการใช้โมเดล ทั้งค่าใช้จ่ายด้าน Computing Resource และค่าใช้จ่ายสำหรับโมเดลที่เป็นสินค้าพาณิชย์ ประเด็นด้านความเป็นส่วนตัวและความปลอดภัยของข้อมูล เมื่อข้อมูลถูกส่งไปยัง API ของโมเดลภายนอกองค์กร ปัญหาเกี่ยวกับ “Hallucinations” ซึ่งหมายถึงโมเดลสร้างข้อมูลหรือคำตอบที่ไม่สะท้อนความจริงหรือไม่เกี่ยวข้องกับข้อมูลที่ได้รับ

2. การถ่ายโอนการเรียนรู้ (Transfer Learning) เป็นการนำข้อมูลในส่วนของ Pre-trained Model มาปรับแต่ง (Fine-tuning) ด้วยข้อมูลที่เฉพาะเจาะจง เพื่อลดปัญหาตัวอย่างเช่น Hallucinations และปัญหาด้านความเป็นส่วนตัวของข้อมูล เป็นต้น การเรียนรู้การถ่ายโอน ช่วยให้โมเดลตอบสนองต่อคำถามหรือคำขอที่เฉพาะเจาะจงได้ดีขึ้น เช่น การนำโมเดล Llama ที่เป็น Pre-trained Model มาใช้โดยตรงอาจไม่ให้ผลลัพธ์ที่ดีในภาษาไทย แต่ถ้า Transfer Learning ด้วยข้อมูลภาษาไทยที่เฉพาะเจาะจง โมเดลสามารถทำนายผลลัพธ์ที่เกี่ยวข้องมากขึ้น นอกจากนี้ ควรคำนึงถึงการเตรียมข้อมูลสำหรับ Transfer Learning ให้มีโครงสร้างที่เหมาะสมกับ Pre-trained Model และปรับให้เข้ากับภาษาหรือบริบทที่ต้องการด้วย

การจัดเตรียมข้อมูลสำหรับ Fine-tuning ใน LLM ที่ใช้สถาปัตยกรรม Transformer จะมี “Attention” เป็นกลไกหลักที่ช่วยให้โมเดลสามารถเน้นและตีความความสำคัญของโทเคนต่าง ๆ ในข้อความได้ กลไกนี้ช่วยให้โมเดลสามารถประมวลผลข้อมูลที่ซับซ้อนและเข้าใจบริบทของข้อความได้ดีขึ้น

การ Fine-tuning เป็นกระบวนการที่เราปรับแต่ง Pre-trained Model ให้เข้ากับงานหรือบริบทเฉพาะ ซึ่งในกระบวนการนี้ความสามารถของโมเดลในการใช้กลไก Attention จะถูกปรับให้เข้ากับข้อมูลใหม่ด้วยการ Fine-tuning เพื่อให้โมเดลเรียนรู้วิธีการให้น้ำหนักและตีความข้อความ และความสำคัญของโทเคนในบริบทที่เฉพาะเจาะจงมากขึ้น การจัดเตรียมข้อมูลสำหรับ Fine-tuning ใน LLM ต้องดำเนินการอย่างรอบคอบเพื่อรักษาความหมาย (Semantic Meaning)

ข้อควรระวังหลักคือ เทคนิคที่ใช้ใน NLP แบบดั้งเดิม เช่น Stop-words Removal, Stemming, Truncation อาจทำให้ข้อความสูญเสียความหมายที่แท้จริงได้ ตัวอย่างเช่น การตัดคำหรือประโยค (Truncation) จาก “เมื่อเช้านี้ฉันไปโรงเรียน และพบว่าลืมหนังสือไว้ที่บ้าน” เป็น “เมื่อเช้านี้ฉันไปโรงเรียน” ทำให้สูญเสียประเด็นหลักของเรื่องราว เป็นต้น

จากการศึกษาที่มีอยู่ได้พิสูจน์แล้วว่าความสามารถของ LLM สามารถใช้ประโยชน์ได้ดีขึ้นด้วยวิธีการโต้ตอบที่ออกแบบมาอย่างพิถีพิถัน โดยมีตัวอย่างดังนี้

1. GenRead [7] ใช้ Prompt กับ LLM เพื่อสร้างบริบทแทนที่การปรับใช้ตัวดึงข้อมูล (Retriever) ซึ่งแสดงให้เห็นว่า LLM สามารถดึงความรู้ภายในโดยการใช้ Prompt โดยที่ Prompt คือข้อความที่ส่งให้กับ LLM โดยคาดหวังให้ LLM ตอบกลับหรือทำอะไรบางอย่างตามที่เราต้องการ โดย Prompt อาจจะประกอบไปด้วยคำสั่ง คำถาม หรือแม้แต่การบอกเล่าด้วยข้อมูล

2. ReAct [8] และ Self-Ask [9] ผสมผสาน Chain-of-Thought (CoT) ซึ่งเป็นเทคนิคที่จะแบ่งคำถามที่ซับซ้อนออกเป็นส่วนย่อย ๆ ที่เป็นตรรกะ ซึ่งเลียนแบบขบวนความคิดที่เป็นลำดับและการโต้ตอบ (Interactions) ด้วยเว็บ API นอกจากนี้ ReAct อาศัยเพียงการสร้าง Prompt เท่านั้น ดังนั้น ReAct เป็นพื้นฐานใหม่สำหรับงานเชิงโต้ตอบ

3. Demonstrate–Search–Predict (DSP) [10] กำหนดไปป์ไลน์ (Pipeline) ที่ซับซ้อนระหว่าง LLM และ Retriever ซึ่งต่างจาก ReAct ตรงที่ DSP ผสมผสาน Prompts สำหรับการสุ่มกลุ่มตัวอย่าง (Demonstration Bootstrap) นอกเหนือจากการ Multihop Breakdown และ Retrieval (Multihop Breakdown และ Retrieval เป็นกระบวนการรับรู้ใน AI NLP ที่ข้อมูลจะถูกรวบรวมและสังเคราะห์ข้ามบริบทหรือคลังความรู้ โดยเกี่ยวข้องกับการเชื่อมโยงขั้นตอนการอนุมานต่าง ๆ เพื่อให้ได้คำตอบที่ครอบคลุม ช่วยเพิ่มความเข้าใจและการแก้ปัญหา)

แม้จะมีความคาดหวังในการตั้งค่า LLM ให้เป็นศูนย์หรือปรับตั้งค่าเพียงเล็กน้อย แต่ในบางครั้งพฤติกรรมของ LLM ก็จำเป็นต้องปรับเปลี่ยนมาก ดังนั้นแนวทางแก้ไขที่เป็นไปได้คือการต่อท้ายด้วยโมเดลขนาดเล็กที่สามารถฝึกได้ไว้ด้านหน้าหรือด้านหลัง LLM โดยโมเดลขนาดเล็กนี้เป็นส่วนหนึ่งของพารามิเตอร์ของระบบที่สามารถปรับแต่งอย่างละเอียด (Fine-tuned) เพื่อการปรับให้เหมาะสม (Optimization) ได้

4. RePlug [11] ถูกเสนอเพื่อ Fine-tune ตัว Retriever แบบ Dense สำหรับ Frozen LLM หรือไม่ต้องการปรับค่า LLM ใน Retrieve-then-read Pipeline ตัว Retriever ได้รับการฝึกภายใต้การดูแลของ LLM เพื่อรับเอกสารที่เหมาะสมสำหรับ LLM ด้วยจุดประสงค์เดียวกัน Directional Stimulus Prompting [12] ใช้แบบจำลองขนาดเล็กเพื่อกระตุ้น LLM เช่น คำสำคัญ (Keywords) สำหรับการสรุป หรือการดำเนินการสนทนาสำหรับการสร้างการตอบสนอง ซึ่งได้รับการอัปเดตตามคะแนนของ LLM เป็นตัวกำหนด

2.2 Retrieval Augmented Generation (RAG)

เป็นเทคนิคที่ใช้ปรับปรุงประสิทธิภาพของ LLM ให้ดีขึ้น โดยสร้างข้อความที่เสริมด้วยการเรียกค้นข้อมูล ซึ่งเป็นวิธีการที่ใช้ฐานข้อมูลภายนอกเพื่อช่วยให้โมเดลภาษาขนาดใหญ่สามารถสร้างข้อความที่มีความถูกต้องและมีความหลากหลายได้ โดยปกติโมเดลที่ใช้ในการสร้างข้อความจะพึ่งพาข้อมูลที่ได้รับการฝึกฝนมาเพื่อสร้างข้อความใหม่ แต่ RAG ยกระดับการทำงานนี้

ขึ้นไปอีกขั้น โดยการค้นหาข้อมูลที่เกี่ยวข้องจากฐานข้อมูลขนาดใหญ่ก่อน แล้วจึงใช้ข้อมูลนั้นเป็นพื้นฐานในการสร้างข้อความ

RAG มีการพัฒนาอย่างรวดเร็วหลังจากปี 2020 เมื่อมีการปล่อย GPT-3 และ ChatGPT ออกมา ซึ่งเป็นโมเดลภาษาขนาดใหญ่ที่มีความสามารถสูงมีแนวคิดหลักสามประเภท คือ Naive RAG, Advanced RAG และ Modular RAG ซึ่งแตกต่างกันตามวิธีการเรียกค้นข้อมูล วิธีการสร้างข้อความและวิธีการปรับปรุงโมเดล ซึ่งมีการวิจัยเรื่องนี้เรื่อยมาจนถึงปัจจุบัน โดย RAG มีส่วนประกอบหลัก 3 ส่วน คือ 1. Retriever 2. Generator และ 3. Augmentation Method ซึ่งมีเทคโนโลยีหลายอย่างที่ใช้ในแต่ละส่วน เช่น Embedding, Re-ranking, Compression, Alignment, Adapter, Validation ฯลฯ

การทำงานของ RAG สามารถแบ่งออกเป็น 2 ขั้นตอนหลัก ๆ ได้แก่

1. ขั้นตอนการค้นคืนข้อมูล (Retrieval Step)

โมเดลจะค้นหาข้อมูลที่เกี่ยวข้องจากฐานความรู้ภายนอก เช่น ฐานข้อมูลข้อความออนไลน์ เอกสาร หรือเว็บไซต์ต่าง ๆ เพื่อใช้เป็นข้อมูลป้อนเข้าให้กับโมเดลสร้างข้อความจากฐานข้อมูลหรือเอกสารที่มีอยู่ โดยอาศัยคำหรือประโยคที่ได้รับเป็นคำแนะนำ

2. ขั้นตอนการสร้างข้อความ (Generation Step)

ข้อมูลที่ได้จากขั้นตอนการค้นคืนข้อมูลจะถูกนำมาเป็นอินพุตในการสร้างข้อความใหม่ โดยอาศัยโมเดลที่มีความสามารถในการเรียนรู้เชิงลึก (Deep Learning) และเทคนิค NLP (Natural Language Processing) เพื่อผลิตข้อความที่ทั้งเกี่ยวข้องและเป็นธรรมชาติ เช่น Transformer หรือ GPT (Generative Pre-trained Transformer) โดยโมเดลเหล่านี้จะถูกฝึกให้เข้าใจและสร้างข้อความได้ดีในหลาย ๆ ประเภท เช่น แปลภาษา สร้างคำบรรยาย และการตอบคำถาม เป็นต้น

เมื่อมีการรวมการสร้างข้อความและการค้นคืนข้อมูลเหล่านี้เข้าด้วยกัน โมเดลที่ได้จะมีความสามารถในการสร้างเนื้อหาที่มีความเชื่อถือได้มากขึ้น โดยมีความรู้และข้อมูลจากฐานความรู้ภายนอกเป็นส่วนหนึ่งของเนื้อหาที่สร้างขึ้น เช่น การตอบคำถามโดยใช้ข้อมูลจากฐานความรู้เพื่อเสริมสร้างคำตอบที่มีความเชื่อถือมากขึ้น หรือการสร้างบทความโดยใช้ข้อมูลที่ได้จากการค้นหาข้อมูลจากฐานข้อมูลออนไลน์

RAG สามารถประยุกต์งานในหลายแอปพลิเคชัน เช่น การตอบคำถามอัตโนมัติ การสร้างเนื้อหาตามคำขอ รวมถึงการสร้างบทความหรือรายงาน ซึ่งข้อดีของ RAG คือการที่สามารถผลิตข้อความที่แม่นยำและรวดเร็วขึ้น โดยอาศัยข้อมูลที่มีความเกี่ยวข้องสูง ทำให้เป็นเทคนิคที่น่าสนใจสำหรับการพัฒนาโมเดล AI ในอนาคต

การเปรียบเทียบระหว่าง RAG กับ Fine-tuning

RAG (Retrieval Augmented Generation) และ Fine-tuning เป็นเทคนิคสองแบบที่ใช้ปรับปรุงประสิทธิภาพของ LLM แต่มีวิธีการทำงานที่แตกต่างกัน โดยที่ RAG คล้ายกับการให้โมเดลเรียนหนังสือ ทำให้สามารถเรียกค้นข้อมูลตามคำถามที่เฉพาะเจาะจงได้ วิธีการนี้เหมาะสำหรับสถานการณ์ที่โมเดลต้องการตอบคำถามที่เฉพาะเจาะจงหรือแก้ไขงานเกี่ยวกับการเรียกค้นข้อมูล โดยโมเดลจะดึงข้อมูลที่เกี่ยวข้องจากแหล่งข้อมูลภายนอกและจะใช้ข้อมูลที่ดึงมาเพื่อสร้างข้อความ

ส่วน Fine-tuning เป็นการปรับโมเดล LLM ที่มีอยู่โดยใช้ชุดข้อมูลเฉพาะโดเมน โมเดลจะเรียนรู้จากตัวอย่างในชุดข้อมูลที่เหมาะสำหรับงานเฉพาะ เช่น โมเดล LLM ที่ผ่านการ Fine-tuning บนชุดข้อมูลบทความข่าว สามารถเขียนบทความข่าวได้ดีกว่า Fine-tuning ที่ทำการปรับแต่งโมเดลที่มีอยู่โดยใช้ชุดข้อมูลที่เฉพาะเจาะจงกับงานที่ต้องการ เพื่อปรับพารามิเตอร์ของโมเดลให้เหมาะสมกับงานนั้น ๆ มุ่งเน้นการปรับแต่งโมเดลให้มีประสิทธิภาพสูงสุดในงานที่ต้องการ โดยไม่จำเป็นต้องใช้ข้อมูลจากแหล่งฐานความรู้ภายนอก ดังนั้น RAG และ Fine-tuning สรุปได้ดังนี้

Fine-tune เมื่อต้องการข้อมูลเฉพาะเจาะจงโมเดล (ความรู้ในด้านเฉพาะ สไตล์ของโมเดลความคาดหวัง

ในรูปแบบของผลลัพธ์ เป็นต้น)

RAG เมื่อต้องการให้คำตอบมีความเป็นจริง (คำตอบต้องถูกต้อง ข้อมูลเปลี่ยนแปลงบ่อย คำตอบต้องมีการอ้างอิง มีขอบเขตกว้างเกินไปสำหรับความรู้ที่มีพารามิเตอร์ เป็นต้น) ที่สำคัญคือ ถ้าชุดข้อมูลมีขนาดใหญ่ ทรัพยากรที่จำเป็นสำหรับการฝึกหรือทำการ Fine-tune โมเดลอาจมีขนาดใหญ่ เช่นกัน RAG และ Fine-tuning สามารถเสริมกันได้ ช่วยเพิ่มความสามารถของโมเดลในระดับต่าง ๆ ในบางสถานการณ์การผสมผสานเทคนิคสองอย่างนี้สามารถทำให้ได้ผลลัพธ์ที่ดีที่สุด กระบวนการที่จะปรับปรุงด้วย RAG และ Fine-tuning อาจต้องทำซ้ำหลายรอบเพื่อให้ได้ผลลัพธ์ที่น่าพอใจ

RAG Framework รูปแบบการวิจัย Naive RAG แสดงถึงวิธีการแรกเริ่มที่ได้รับความนิยมอย่างกว้างขวาง หลังจากการใช้ ChatGPT อย่างแพร่หลาย Naive RAG ปฏิบัติตามกระบวนการแบบดั้งเดิมที่ประกอบด้วย การจัดทำดัชนี การเรียกค้น และการสร้าง ถือว่าเป็น “Retrieve-Read” Framework กระบวนการทำงานของ RAG ประกอบด้วยขั้นตอนหลัก ๆ ที่แบ่งออกเป็น 3 ขั้นตอน ได้แก่

1. Indexing (การจัดสรรดัชนี)

เนื้อหาที่ใช้ในการสร้าง (Generation) จะถูกจัดเก็บไว้ในรูปแบบของดัชนี เพื่อให้สามารถทำการสืบค้นได้อย่างมีประสิทธิภาพ ข้อมูลที่จัดเก็บในดัชนีสามารถเป็นข้อความอย่างเดียวหรือรวมข้อมูลภาพหรือเสียงก็ได้ เพื่อให้สามารถใช้ในกระบวนการ Generation ได้อย่างหลากหลาย โดยทำความสะอาดและแยกข้อมูลออกเป็นข้อความธรรมดาที่ไม่มีโครงสร้าง และแบ่งข้อมูลออกเป็นส่วน ๆ ที่อาจจะมีประโยชน์สำหรับโมเดลในการใช้งาน จากนั้นใช้ตัวฝังข้อมูลที่เลือกเพื่อแปลงข้อมูลเหล่านั้นเป็นรูปแบบเวกเตอร์ เก็บไว้ในรูปแบบดัชนีข้อมูล

2. Retrieval (การสืบค้น)

เป็นขั้นตอนที่ระบบจะทำการสืบค้นข้อมูลที่เก็บไว้ในดัชนีตามคำค้นหาหรือเงื่อนไขที่กำหนด เพื่อให้ได้เนื้อหาที่เกี่ยวข้องมากที่สุด การสืบค้นนี้สามารถใช้วิธีการต่าง ๆ ได้ เช่น การใช้คำค้นหาเป็นข้อความ การใช้ภาพเป็นข้อมูลเข้าช่วยในการค้นหา หรือแม้กระทั่งการใช้รายการเนื้อหาที่มีเนื้อความคล้ายคลึงกัน เป็นต้น Retrieval จะใช้ข้อมูลเดียวกันกับขั้นตอน Indexing เพื่อฟังคำถามของผู้ใช้และเรียกค้นเอกสารที่คล้ายกันที่สุดตามเกณฑ์ความคล้ายคลึงที่กำหนดไว้ บางเกณฑ์ที่นิยมใช้คือ Cosine Similarity, EuclideanDistance, DotProduct ซึ่งแต่ละเกณฑ์มีข้อดีในรูปแบบของตัวเอง

3. Generation (การสร้าง)

เมื่อข้อมูลที่เกี่ยวข้องถูกสืบค้นและเลือกมาแล้ว ขั้นตอนต่อไปคือการสร้างเนื้อหาใหม่จากข้อมูลที่ถูกสืบค้นเหล่านั้น การสร้างนี้อาจใช้โมเดลการเรียนรู้ของเครื่อง (Machine Learning) เพื่อสร้างเนื้อหาที่มีความสมเหตุสมผลและมีคุณภาพ โดยใช้ข้อมูลที่ได้จากขั้นตอน Retrieval เป็นข้อมูลอ้างอิง

ในขั้นตอนการเลือกเอกสารที่จะส่งให้กับ LLM สามารถทำได้โดยพิจารณาดังต่อไปนี้

1. ความสำคัญ (Importance)

เอกสารที่มีความสำคัญสูงสุดสำหรับคำถามหรืองานที่กำลังดำเนินการ ความสำคัญของเอกสารสามารถวัดได้จากคะแนนการค้นคืน (Retrieval Score) ความสัมพันธ์กับคำถามว่ามีความสอดคล้องเพียงใดกับคำถามนั้น ๆ

2. ความหลากหลาย (Diversity)

เอกสารที่มีความหลากหลายในเนื้อหา ไม่ให้มีข้อมูลที่ซ้ำซ้อนกันมากเกินไป การใช้เอกสารที่มีความหลากหลายจะช่วยให้ LLM ได้รับมุมมองที่หลากหลาย

3. ความยาก (Difficulty)

เลือกเอกสารที่มีระดับความยากที่เหมาะสมกับ LLM หากเอกสารยากเกินไป อาจทำให้ LLM ไม่สามารถตอบคำถามได้ตามที่ต้องการ

4. ความยาว (Length)

พิจารณาความยาวของเอกสาร หากเอกสารยาวมาก อาจต้องตัดเล็กน้อยเพื่อให้พอดีกับ Context Window ของ LLM โดยการเลือกเอกสารที่เหมาะสมสำหรับ LLM ต้องพิจารณาความสำคัญ ความหลากหลาย

ความยาก และความยาวให้ดี เพื่อให้ LLM สามารถสร้างข้อความที่มีความถูกต้องและครอบคลุมได้

2.3 การออกแบบคำสั่งที่ใช้ในการสื่อสารกับโมเดล (Prompt Engineering)

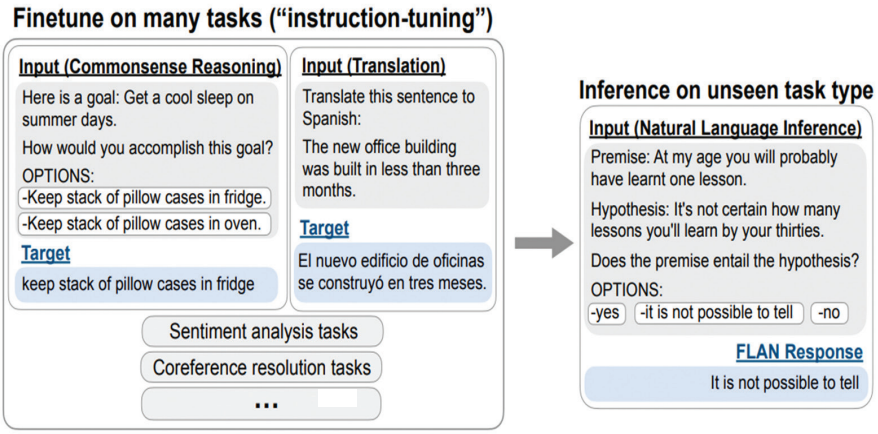
ในระบบ Artificial Intelligence (AI) ที่เน้นในสาขา Natural Language Processing (NLP) เป็นการพัฒนาและปรับปรุงขั้นตอนการสร้าง Prompt เพื่อให้เหมาะสมสำหรับการสั่งการโมเดล LLM ให้มีประสิทธิภาพมากขึ้น โดย Prompt Engineering เกี่ยวข้องกับการเพิ่มทักษะและเทคนิคในการทำความเข้าใจ การจำกัดความสามารถ และการเพิ่มประสิทธิภาพของโมเดล LLM การเขียนคำอธิบายกับสิ่งที่เราต้องการ เพื่อสั่งการให้กับ AI โดยรายละเอียดทั้งหมดจะถูกรวมอยู่ใน Prompt ทั้งคำสั่ง บริบท ตัวอย่าง วิธีการ รูปแบบของผลลัพธ์ และข้อความ Input ในรูปแบบคำถาม

ส่งให้ AI ฝึง LLM เป็นผู้ตอบคำถามโดยทั่วไป Prompt Engineering จะเป็นการแปลงรายละเอียดของงานต่าง ๆ ให้เป็น ชุดข้อมูล Prompt เตรียมไว้สำหรับใช้ในการเทรนโมเดลทางภาษาให้มีความสามารถเฉพาะด้านเพิ่มขึ้นตามต้องการ เรียกว่า “Prompt-Based Learning” หรือ “Prompt Learning” โดยหัวข้อ Prompt Engineering ที่น่าสนใจ ได้แก่

1) **Zero-shot** เป็นเทคนิคหนึ่งในขั้นตอน Prompt Engineering ที่ใช้ในการสร้าง Prompt ซึ่ง LLM เรียนรู้และเข้าใจคำถามหรือคำสั่งที่ไม่เคยเห็นมาก่อนโดยไม่ต้องมีข้อมูลการเรียนรู้เพิ่มเติมในกระบวนการเทรน ในสถานการณ์ Zero-shot โมเดลจะต้องทำการเรียนรู้และเข้าใจจากคำถามหรือคำสั่งที่เข้ามาในขณะที่มีข้อมูลอยู่แค่ในรูปแบบของ Prompt เท่านั้น

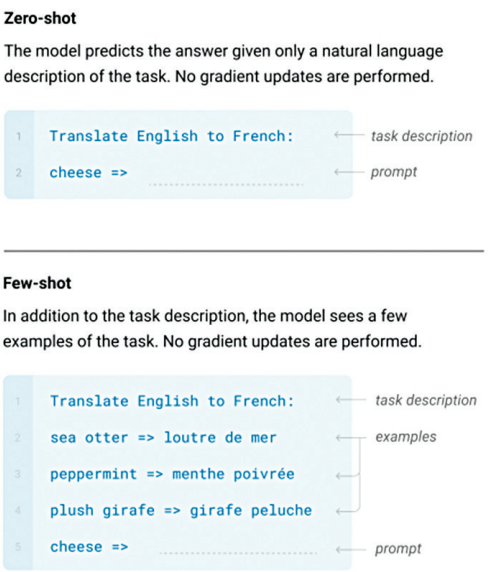
โมเดลจะพยายามอธิบายหรือตอบโดยใช้ความเข้าใจจาก Prompt เพียงอย่างเดียว โดยไม่มีข้อมูลการเรียนรู้เพิ่มเติมที่เข้ามาช่วยในกระบวนการเรียนรู้หรือเทรน (Train) [13] ตัวอย่างเช่น กรณีของ Sentiment Analysis สามารถใช้หลาย ๆ ย่อหน้า (Paragraphs) ที่มีความคิดเห็นต่างกัน และ Label ในแต่ละ Classes จากนั้น Train Model (เช่น RNN "Recurrent Neural Network" กับข้อมูลข้อความเหล่านี้) เพื่อรับข้อความ Input ให้ผลลัพธ์การ Classify เป็น Output ซึ่งจะพบว่าโมเดลดังกล่าวไม่สามารถปรับเปลี่ยนได้ หากเราเพิ่ม Class ใหม่ ซึ่งกรณีนี้โมเดลจะต้องถูกเทรนใหม่ โดยตัวอย่างการปรับคำสั่งของ Zero-shot แสดงดังรูปที่ 1

2) **Few-shot** หมายถึง การฝึกโมเดลหรือระบบด้วยข้อมูลที่มีจำนวนตัวอย่างน้อยมาก ๆ เพื่อทดสอบความสามารถของโมเดลในการเรียนรู้และสร้างความเข้าใจใหม่ต่อข้อมูลใหม่ของการฝึกโมเดลภาษา [14] ตัวอย่างเช่น GPT-3 หรือ FLAN การฝึกแบบ Few-shot จะใช้ข้อมูลตัวอย่างเพียงไม่กี่ตัวอย่างเท่านั้นในแต่ละกลุ่มงานหรือชุดข้อมูล ซึ่งอาจเป็นเพียงหลายตัวอย่าง



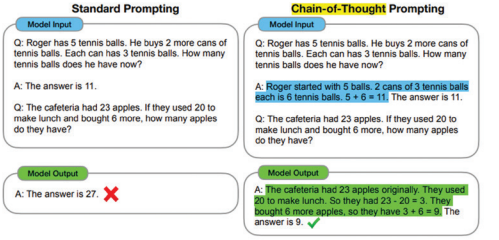
รูปที่ 1 *การปรับคำสั่งโดยใช้ FLAN Zero-shot [13]*

ถึงสองตัวอย่างต่อกลุ่มงาน โดยปกติแล้วข้อมูลตัวอย่างที่มีจำนวนน้อยนี้อาจไม่เพียงพอสำหรับโมเดลที่ฝึกในการเรียนรู้แบบจำลองทั่วไป แต่การฝึกโมเดลแบบ Few-shot ช่วยให้โมเดลเรียนรู้ได้โดยความสามารถในการทำงานบางอย่าง ใช้ข้อมูลน้อยกว่าการฝึกแบบปกติโดยส่วนมาก ซึ่งจะต่างจาก Zero-shot ตรงที่ Zero-shot จะไม่มีข้อมูลตัวอย่างแต่ Few-shot จะมีข้อมูลตัวอย่างมากกว่า Zero-shot ดังตัวอย่างรูปที่ 2



รูปที่ 2 *การทำงานของ Few-shot เมื่อเทียบกับ Zero-shot [14]*

3) Chain-of-Thought (CoT) เป็นวิธีที่ให้โมเดลนั้นเข้าใจในภาพรวมของวัตถุประสงค์ในคำถามมากขึ้น ก่อนที่จะไปถึงผลลัพธ์ที่ต้องการ โดยที่วิธี CoT จะทำการแตกงานเป็นขั้นย่อย ๆ แต่ละขั้นจะทำต่อจากขั้นตอนก่อนหน้า ในการที่จะช่วยให้โมเดลเข้าใจภาพรวมวัตถุประสงค์ของงานและเชื่อมโยงความเข้าใจระหว่างข้อมูลหลายข้อมูลในลำดับของคำถามจนสามารถทำนายหรือตอบคำถามต่อไปตามลำดับของเหตุการณ์ที่เกิดขึ้นได้ด้วยการให้โมเดลมีเวลาประมวลผลมากขึ้น และการใช้ CoT Prompt สามารถช่วยให้โมเดลเข้าใจงานได้ดีขึ้น และสร้างคำตอบที่แม่นยำ ตรงประเด็นมากขึ้น ช่วยเพิ่มประสิทธิภาพโดยรวมของ LLM ทำให้เหมาะสำหรับการนำไปประยุกต์ใช้ในงานที่หลากหลาย [15] โดยหลักการทำงานของ CoT Prompt ดังแสดงในรูปที่ 3



รูปที่ 3 *การทำงานของ CoT Prompt เมื่อเทียบกับวิธี Prompt โดยทั่วไป [15]*

3. วิธีการดำเนินการ

3.1 Data Collection

Data Collection คือ กระบวนการเก็บรวบรวมข้อมูลจากแหล่งต่าง ๆ เพื่อนำมาใช้ในการวิเคราะห์ประมวลผล หรือจัดเก็บเพื่อใช้ในการตัดสินใจ การเก็บข้อมูลสามารถทำได้หลายวิธี เช่น การสัมภาษณ์ผู้เกี่ยวข้อง การสำรวจแบบสอบถาม การเก็บข้อมูลจากฐานข้อมูลออนไลน์ หรือการเก็บข้อมูลจากการวิเคราะห์ข้อมูลที่มีอยู่แล้ว การเก็บข้อมูลที่ถูกต้องและเพียงพอเป็นสิ่งสำคัญในการใช้ข้อมูลให้เกิดประโยชน์สูงสุดในการทำงานต่อไป สำหรับงานวิจัยนี้ได้ทำการเก็บรวบรวมข้อมูลบทความในวารสารเทคโนโลยีป้องกันประเทศ หรือ DTech (Thailand Defence Technology Journal) จำนวน 5 เล่ม แต่ละเล่มจะคัดทั้งบทความที่เกี่ยวข้องกับเทคโนโลยีป้องกันประเทศ รวมแล้วได้ 22 บทความ ซึ่งข้อมูลบทความอยู่ในรูปแบบไฟล์ pdf เพื่อนำมาแปลงเป็นข้อมูลที่อยู่ในรูปแบบของ Plain Text ก่อนที่จะนำไปใช้ในขั้นตอนต่อไป

โดย Plain Text คือ ความหมายทางวิทยาการคอมพิวเตอร์ เป็นข้อความที่มีลักษณะของตัวอักษรธรรมดา ตัวอักษรแบบปกติทั่วไปที่ไม่มีการจัดรูปแบบใด ๆ เช่น ตัวเอียง ตัวหนา การขีดเส้นใต้ หรือการจัดหน้าแบบพิเศษ เป็นต้น

3.2 Data Preprocessing

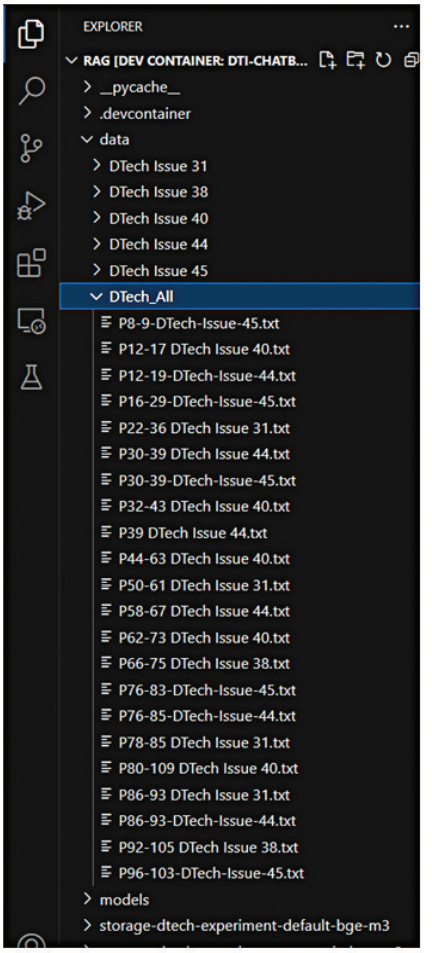
ก่อนที่จะนำข้อมูล Plain Text ที่ได้ไปใช้งาน จะต้องดำเนินการในขั้นตอนการเตรียมข้อมูล (Data Preprocessing) เสียก่อน ซึ่งการเตรียมข้อมูลหมายถึง การทำความเข้าใจและตรวจสอบคุณภาพของข้อมูล ทำข้อมูลให้อยู่ในรูปแบบที่ถูกต้อง หรือปรับเปลี่ยนให้อยู่ในรูปแบบที่เหมาะสม เพื่อให้ข้อมูลอยู่ในลักษณะหรือรูปแบบที่สามารถนำไปใช้งานต่อในขั้นตอนอื่นอย่างมีประสิทธิภาพ หากเตรียมข้อมูลไม่ดีพอจะส่งผลให้ผลการวิเคราะห์หรือการตีความจากการนำข้อมูลไปใช้ผิดไปจากที่ควรจะเป็น ซึ่งในการวิจัยนี้ได้ดำเนินการเตรียมข้อมูลตามขั้นตอนดังนี้

1. ตรวจสอบความถูกต้องของข้อมูล และการสะกดคำ

2. ตรวจสอบและทำความสะอาดข้อความ โดยการลบเครื่องหมาย หรือสัญลักษณ์ต่าง ๆ เช่น เครื่องหมายคำสั่งที่ใช้ในการเขียนภาษาโปรแกรมมิงต่าง ๆ หรือการแสดงผล HTML เป็นต้น

3. จะต้องไม่มีการแบ่งบรรทัดระหว่างข้อความที่อยู่ในย่อหน้า (Paragraph) เดียวกัน

4. ระหว่างย่อหน้าให้คั่นด้วยการขึ้นบรรทัดใหม่ โดยตัวอย่างการเตรียมข้อมูลดังแสดงในรูปที่ 4 เมื่อเตรียมข้อมูลของบทความทั้งหมดที่คัดเลือกเรียบร้อยแล้ว ข้อมูลส่วนนี้จะกลายเป็นคลังข้อมูลเอกสารสำหรับการสร้างคำตอบด้วยเทคนิค RAG ต่อไป



รูปที่ 4 *ตำแหน่งที่อยู่ของโฟลเดอร์ (Folder Path) ที่จัดเก็บไฟล์ Plain Text ในการทำ Data Preprocessing*

การสร้าง Prompts

Prompts เป็นข้อความที่ถูกส่งให้กับโมเดล โดยคาดหวังให้โมเดลตอบกลับหรือทำอะไรบางอย่างตามที่เราต้องการ ซึ่ง Prompt อาจจะประกอบไปด้วยคำสั่ง คำถาม หรือแม้แต่การบอกเล่าด้วยข้อมูล เป็น Prompts จากวารสารเกี่ยวกับเทคโนโลยีป้องกันประเทศที่จัดทำโดยสถาบันเทคโนโลยีป้องกันประเทศ (สทป.) แบ่งลักษณะของ Prompts ออกเป็น 5 ลักษณะ ดังนี้

1. คำถามมีคำตอบระบุในเอกสาร (Extraction Test) ตัวอย่างคำถาม เช่น “หุ่นยนต์ตัวแรกของประเทศไทยสร้างขึ้นในปี พ.ศ. อะไร และมีชื่อว่าอะไร โดยบรรยายคุณลักษณะของหุ่นยนต์ตัวดังกล่าวด้วย” “ยกตัวอย่างภารกิจทางการทหารของระบบยานไร้คนขับ”

2. คำถามมีคำตอบระบุในเอกสารแต่ต้องใช้ข้อมูลหลาย ๆ ส่วนในเอกสารมาช่วยตอบ (Logic Test) ตัวอย่างคำถาม เช่น “หุ่นยนต์แสนดีสามารถบรรทุกสิ่งของที่หนัก 81 กิโลกรัม ได้หรือไม่ และสามารถทำงานต่อเนื่องนาน 11 ชั่วโมง ได้หรือไม่” “ฉากฝึกแบบรู้ระยะสามารถสร้างฉากให้เป็นสภาพแวดล้อมของป่าที่มีศัตรู (ฉากสนามฝึกสนามรบเสมือนจริง) ได้หรือไม่”

3. คำถามที่ไม่มีคำตอบระบุอยู่ในเอกสาร (Hallucination Test) ตัวอย่างคำถามเช่น “ระบบในการจ่ายไฟผ่านรางสำหรับหุ่นยนต์น้องแมงมุม หรือ Sevy Bot มีหลักการอย่างไร” “5 อันดับประเทศที่มีจำนวนและขีดความสามารถของดาวเทียมมากที่สุดในโลก”

4. คำถามที่ต้องการคำตอบแบบเดิมแต่เปลี่ยนรูปประโยคคำถาม (Robustness Test) ตัวอย่างคำถาม เช่น “หุ่นยนต์ตัวแรกของประเทศไทยถูกสร้างขึ้นในปีใดและมีชื่อเรียกว่าอะไร พร้อมทั้งอธิบายลักษณะเด่นของหุ่นยนต์ตัวนั้นด้วย” “สทป. ได้พัฒนาและสร้างต้นแบบหุ่นยนต์ที่ใช้สำหรับงานกู้ภัยและกู้วัตถุระเบิดขึ้นมาทั้งหมดกี่รุ่น รุ่นใดบ้างที่ได้รับการพัฒนา”

5. คำถามที่มีคำตอบระบุในเอกสารแต่ต้องการคำตอบตามคำแนะนำที่กำหนด เช่น ให้ตอบเป็นภาษาอังกฤษ ให้หาค่าเฉลี่ยของคำตอบ (Instruction Test) ตัวอย่างคำถาม เช่น “แนะนำสิ่งประดิษฐ์ที่สามารถนำไปใช้ในการสำรวจแนวท่อเดินน้ำมันและก๊าซธรรมชาติที่วางทอดเป็นแนวยาวใต้ท้องทะเลได้” “กรุณาเลือกคุณลักษณะเด่นของหุ่นยนต์ตรวจการณ์ขนาดเล็กรุ่น NOONAR Version 4 มา 3 ข้อ”

3.3 Data Testing

ในวิธีการทดสอบผลของโมเดลนั้นจะมีการวัดผลดังนี้

1. หากมีการตอบคำถามถูกทั้งหมดจะได้คะแนน 1 คะแนน ต่อ 1 คำถาม

2. หากมีการตอบคำถามผิดบางส่วนหรือตอบคำถามไม่ครบถ้วนจะได้คะแนน 0.5 คะแนน ต่อ 1 คำถาม

3. หากมีการตอบคำถามผิดทั้งหมดจะได้คะแนน 0 คะแนน ต่อ 1 คำถาม

***หมายเหตุ** คำถามประเภทไม่มีคำตอบระบุอยู่ในเอกสาร (Hallucination Test) หากมีการตอบในเชิงความหมายว่า ไม่พบคำตอบในเอกสาร จะได้ 1 คะแนน หากตอบนอกเหนือจากนี้ จะได้ 0 คะแนน ส่วนคะแนน 0.5 จะไม่เกิดในคำถามประเภทนี้

สมการ Accuracy เขียนได้ดังนี้

$$\%Accuracy = 100 \times \frac{TS}{AQ} \quad (1)$$

เมื่อ

Total score of answers (TS) = ผลคะแนนรวมของคำตอบ

All questions (AQ) = คำถามทั้งหมด

จากนั้นการคำนวน %Accuracy จะคำนวณดังสมการที่ (1) คือนำคะแนนในแต่ละข้อมารวมกันเทียบกับคำถามทั้งหมดโดยโมเดลที่ใช้เป็นโมเดลที่ใช้กำหนด Chunking เป็น Default และ Semantic

3.4 Model Testing

การทดสอบจะใช้ LlamaIndex ซึ่งเป็น Data Framework สำหรับสร้าง LLM แอปพลิเคชัน Llama Index มีเครื่องมือสำหรับการนำเข้าข้อมูล การทำดัชนีและการสืบค้นสำหรับ LLM รวมถึง RAG โดยทำการติดตั้ง LlamaIndex เป็นไลบรารีใน Python และในการทดสอบจะมีการกำหนดพารามิเตอร์ ดังนี้

1. Chunking

เป็นกระบวนการแบ่งข้อมูลของข้อความขนาดใหญ่หรือเอกสารออกเป็นส่วนย่อย ๆ ที่เรียกว่า "Chunks" เพื่อให้สามารถจัดการและประมวลผลได้ง่ายขึ้น ในขั้นตอนการค้นคืนข้อมูล (Retrieval) ระบบจะค้นหา Chunk ที่เกี่ยวข้องจากฐานข้อมูลเพื่อใช้เป็นบริบทในการสร้างข้อความและในขั้นตอนการสร้างข้อความ (Generation) ระบบจะใช้ข้อมูลจาก Chunk ที่ค้นคืนมาเพื่อช่วยในการสร้างข้อความที่สมบูรณ์และสอดคล้องกับข้อมูลที่มีอยู่ ขนาดของ Chunk กำหนดโดยจำนวนคำหรือโทเคนที่อยู่ในแต่ละส่วนย่อยของข้อมูล เช่น การแบ่งบทความเป็นหลาย ๆ ส่วนย่อยที่มีขนาด 100 คำต่อส่วน

ในการทดสอบนี้จะสามารถกำหนด Chunking ได้ 2 แบบ คือ 1) Default หรือ Fixed Size แบ่ง Chunk ให้มีขนาดตัวอักษรที่แน่นอน 2) Semantic แบ่ง Chunk โดยพิจารณาจากความหมายหรือบริบทของข้อความ

เพื่อให้แต่ละ Chunk มีเนื้อหาที่มีความหมายสัมพันธ์กันหรือมีบริบทที่ชัดเจน

2. Embedding Model

เป็นโมเดลที่ใช้ในการแปลงข้อความหรือข้อมูลเป็นเวกเตอร์เชิงความหมาย (Semantic Vectors) เพื่อให้สามารถประมวลผลและใช้งานได้ง่ายขึ้นในการค้นคืนข้อมูล (Retrieval) และการสร้างข้อความ (Generation)

ในการทดสอบนี้จะสามารถกำหนด Embed Model ได้ 2 แบบ คือ

1) bge - m3 [16] เป็น Embedding Model ที่พัฒนาโดย BAAI ซึ่งมีความโดดเด่นใน 3 ด้านหลัก (M3) คือ Multi-Linguality (รองรับหลายภาษา) Multi-Functionality (ทำงานได้หลากหลาย) Multi-Granularity (รองรับขนาดข้อมูล)

2) Text-embedding-3-large [17] เป็น Embedding Model พัฒนาโดย OpenAI มีจุดเด่น คือ มีประสิทธิภาพสูง มีขนาดเวกเตอร์ใหญ่

3. Generator Model

เป็นโมเดลภาษาทำหน้าที่สร้างข้อความเพื่อเป็นคำตอบของคำถามโดยอาศัยข้อมูลที่ดึงมาจากฐานข้อมูลหรือ Knowledge Base ในการทดสอบนี้คือข้อมูลจากวารสารเทคโนโลยีป้องกันประเทศ ประกอบกับ Prompt ที่สร้างจากคำถามและข้อมูลที่ดึงมา

ในการทดสอบนี้จะสามารถกำหนด Generator ได้ 2 แบบ คือ 1) claude-3-sonnet-20240229 [18] เป็นโมเดลภาษาขนาดกลางจากตระกูล Claude-3 พัฒนาโดยบริษัท Anthropic มีจุดเด่น คือ รองรับข้อมูลภาษาไทย มีขนาดกะทัดรัด ใช้งานบนอุปกรณ์ที่มีประสิทธิภาพจำกัดได้ 2) gpt-4-turbo-preview เป็นโมเดลภาษาขนาดใหญ่ พัฒนาโดย OpenAI มีจุดเด่น คือ รองรับข้อมูลภาษาไทย มีประสิทธิภาพสูง

ในการทดสอบจะแบ่งเป็น 4 โมเดล โดยแต่ละโมเดลจะกำหนดพารามิเตอร์ที่ใช้ในการทดสอบดังต่อไปนี้

แบบที่ 1: Chunking เป็น Default, Embedding Model เป็น bge - m3 และ Generator Model เป็น claude-3-sonnet-20240229

แบบที่ 2: Chunking เป็น Semantic, Embedding Model เป็น bge - m3 และ Generator Model เป็น claude-3-sonnet-20240229

แบบที่ 3: Chunking เป็น Default, Embedding Model เป็น Text - embedding-3-large และ Generator Model เป็น gpt-4-turbo-preview

แบบที่ 4: Chunking เป็น Semantic, Embedding Model เป็น bge - m3 และ Generator Model เป็น gpt-4-turbo-preview

4. ผลการทดลอง

ผลลัพธ์ที่ได้จากการทดลองทั้ง 4 โมเดล ดังแสดงในตารางที่ 2 มีดังนี้

การตอบคำถามประเภทที่มีคำตอบระบุในเอกสาร (Extraction Test) แสดงให้เห็นว่าโมเดลที่ 1, 2 และ 4 สามารถตอบคำถามได้ถูกต้องทั้งหมด ส่วนโมเดลที่ 3 สามารถตอบคำถามได้ถูกต้อง 92%

คำถามมีคำตอบระบุในเอกสารแต่ต้องใช้ข้อมูลหลาย ๆ ส่วนในเอกสารมาช่วยตอบ (Logic Test) แสดงให้เห็นว่าโมเดลที่ 1, 2 และ 4 สามารถตอบคำถามได้ถูกต้องเท่ากันที่ 90% ส่วนโมเดลที่ 3 สามารถตอบคำถามได้ถูกต้อง 80%

คำถามที่ไม่มีคำตอบระบุอยู่ในเอกสาร (Hallucination Test) แสดงให้เห็นว่าโมเดลที่ 2 มีประสิทธิภาพในการตอบคำถามดีที่สุด ซึ่งสามารถตอบคำถามได้ถูกต้องทั้งหมด ส่วนโมเดลที่ 1 และ 4 สามารถตอบคำถามได้ถูกต้องเท่ากันที่ 80% ส่วนโมเดลที่ 3 สามารถตอบคำถามได้ถูกต้อง 60%

คำถามที่ต้องการคำตอบแบบเดิมแต่เปลี่ยนรูปประโยคคำถาม (Robustness Test) แสดงให้เห็นว่าโมเดลที่ 1 มีประสิทธิภาพในการตอบคำถามดีที่สุด ซึ่งสามารถตอบคำถามได้ถูกต้องทั้งหมด ส่วนโมเดลที่ 4, 2 และ 3 สามารถตอบคำถามได้ถูกต้อง 91%, 86% และ 82% ตามลำดับ

คำถามที่มีคำตอบระบุในเอกสารแต่ต้องการคำตอบตามคำแนะนำที่กำหนด (Instruction Test) โมเดลที่ 1 มีประสิทธิภาพในการตอบคำถามดีที่สุด

ซึ่งสามารถตอบคำถามได้ถูกต้อง 85% โมเดลที่ 2 สามารถตอบคำถามได้ถูกต้อง 80% ส่วนโมเดลที่ 3 และ 4 สามารถตอบคำถามได้ถูกต้องเท่ากันที่ 70%

ตัวอย่างคำตอบที่ได้ 1 คะแนน กับคำตอบที่ได้ 0 คะแนน

คำถาม: หุ่นยนต์ตัวใดบ้างที่ได้รับรางวัลรองชนะเลิศอันดับ 2 ในการแข่งขันหุ่นยนต์นวัตกรรมทางการแพทย์ TCELS IMEDBOT 2021 Innovation Contest หรือการแข่งขัน “The Next Normal of Healthcare Robotics” ในปี 2564

คำตอบที่คาดหวัง: หุ่นยนต์ขนส่งอาหารและยาสำหรับปฏิบัติงานในโรงพยาบาล CARVER พัฒนาโดยสถาบันวิทยาการหุ่นยนต์ภาคสนาม (FIBO) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

คำตอบที่ถูก (ได้ 1 คะแนน): จากข้อมูลที่ให้มาระบุว่าหุ่นยนต์ขนส่งอาหารและยาสำหรับปฏิบัติงานในโรงพยาบาล CARVER ที่พัฒนาโดยสถาบันวิทยาการหุ่นยนต์ภาคสนาม (FIBO) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ได้รับรางวัลรองชนะเลิศอันดับ 2 ในการแข่งขัน TCELS IMEDBOT 2021 Innovation Contest 2021 - The Next Normal of Healthcare Robotics ซึ่งเป็นการแข่งขันหุ่นยนต์เพื่อสร้างนวัตกรรมทางการแพทย์ในปี พ.ศ. 2564

คำตอบที่ผิด (ได้ 0 คะแนน): หุ่นยนต์ที่ได้รางวัลรองชนะเลิศอันดับ 2 ในการแข่งขันหุ่นยนต์นวัตกรรมทางการแพทย์ TCELS IMEDBOT 2021 Innovation Contest หรือการแข่งขัน “The Next Normal of Healthcare Robotics” ในปี 2564 คือ หุ่นยนต์เดิน 2 ขาคล้ายมนุษย์ พัฒนาโดยสถาบันวิทยาการหุ่นยนต์ภาคสนาม (FIBO) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ตัวอย่างคำตอบที่ได้ 0.5 คะแนน

คำถาม: กรุณาเลือกคุณลักษณะเด่นของหุ่นยนต์ตรวจการณ์ขนาดเล็ก รุ่น NOONAR version 4 มา 3 ข้อ

คำตอบที่คาดหวัง: มีคำตอบ 3 ข้อ จาก 11 ข้อต่อไปนี้

1. ประเภท (ไม่รวม Payload) 7.6 กิโลกรัม
2. โหมดการทำงาน Remote Controlled
3. ความเร็วสูงสุด 1.3 เมตร/วินาที
4. ระยะปฏิบัติการ (Operation Range) (LOS) 300 เมตร
5. ระยะเวลาปฏิบัติการ (Operation Time) 2 ชั่วโมง 30 นาที
6. กล้อง 3 กล้อง
7. ระบบสื่อสาร สามารถเพิ่มระยะได้โดยใช้ชุดขยายสัญญา (Repeater) 4.9 - 6.2 GHz (สามารถเลือกใช้งานได้ในย่านความถี่ที่กำหนด)
8. การป้องน้ำและฝุ่น IP54
9. อุปกรณ์บรรทุก Track Wheel, PTZ (Option: Disruptor, Track Extension)
10. ความสามารถในการรับน้ำหนัก 10 กิโลกรัม
11. การซ่อมบำรุง ดำเนินการได้เองภายในประเทศ

คำตอบที่ได้ 0.5 คะแนน: จากข้อมูลที่ให้มาสามารถสรุปคุณลักษณะเด่นของหุ่นยนต์ตรวจการณ์ขนาดเล็ก รุ่น NOONAR Version 4 ได้ดังนี้

1. มีขนาดเล็ก สะดวกต่อการพกพา มีความคล่องตัวและทนทานสูง เนื่องจากผลิตด้วยวัสดุคอมโพสิตที่รองรับแรงกระแทกจากการตกได้ดี
2. สามารถควบคุมและสั่งการแบบไร้สายระยะไกล 300 เมตร และใช้งานได้อย่างต่อเนื่อง 2 ชั่วโมง 30 นาที
3. ผ่านการรับรองมาตรฐานจากกระทรวงกลาโหมว่าด้วยระบบยานภาคพื้นไร้คนขับสำหรับภารกิจตรวจการณ์ และ/หรือ การเก็บกู้/ทำลายวัตถุระเบิด

เหตุผลการให้คะแนน: เพราะมีคำตอบที่ถูกเพียง 1 ข้อ จากคำตอบที่คาดหวัง 11 ข้อ คือ คำตอบข้อ 2 ส่วนคำตอบข้อ 1 และ 3 ไม่ได้อยู่ในคุณลักษณะเด่นของหุ่นยนต์ตรวจการณ์ขนาดเล็ก รุ่น NOONAR Version 4 ที่ระบุในเอกสาร คำตอบนี้จึงได้คะแนน 0.5 คะแนน

ตัวอย่างคำตอบที่ได้ 1 คะแนนจากคำถามประเภทไม่มีคำตอบระบุอยู่ในเอกสาร (Hallucination Test)

คำถาม: ชื่อหุ่นยนต์ที่ได้รับรางวัลชนะเลิศในการแข่งขัน TCELS IMEDBOT 2021 Innovation Contest 2021 - The Next Normal of Healthcare Robotics การแข่งขันหุ่นยนต์เพื่อสร้างนวัตกรรมทางการแพทย์ เมื่อปี พ.ศ. 2564 คือ

คำตอบที่คาดหวัง: ไม่มีคำตอบในเอกสาร คำตอบที่ถูก (ได้ 1 คะแนน): ขออภัยครับ จากข้อมูลที่ให้มานั้นไม่มีรายละเอียดเกี่ยวกับหุ่นยนต์ที่ได้รับรางวัลชนะเลิศในการแข่งขัน TCELS IMEDBOT 2021 Innovation Contest 2021

5. สรุปและอภิปรายผล

สรุปผลการทดลองระบบจักรกลสนทนากับข้อมูลบทความในวารสารป้องกันประเทศ โมเดลแบบที่ 1 มีประสิทธิภาพดีที่สุดในการตอบคำถามได้อย่างถูกต้องถึง 93% ถ้าพิจารณาถึงความแม่นยำในการตอบคำถามแต่ละประเภท พบว่า คำถามที่มีคำตอบระบุในเอกสาร โมเดลที่ใช้ทดสอบทั้ง 4 แบบ มีความแม่นยำในการตอบดีมากที่ 92-100% ส่วนคำถามที่ต้องการคำแนะนำในการตอบมีความแม่นยำในการตอบน้อยสุดในทั้ง 5 ประเภทคำถาม โดยมีความแม่นยำที่ 70-85% อาจเป็นเพราะคำแนะนำที่ใส่ไปในคำถามทำให้คำถามยาวและซับซ้อนเพิ่มมากขึ้นกว่าคำถามประเภทอื่น

ส่วนการกำหนดค่าพารามิเตอร์ที่ต่างกันสามารถสรุปได้ดังนี้

1. Chunking

จากผลการทดลอง โมเดลแบบที่ 1 ที่กำหนด Chunking เป็น Fixed Size เทียบกับโมเดลแบบที่ 2 ที่กำหนด Chunking เป็น Semantic ส่วนพารามิเตอร์ค่าอื่นของโมเดลทั้ง 2 เหมือนกัน พบว่า การกำหนด Chunking เป็น Fixed Size ให้คำตอบที่มีความแม่นยำมากกว่าที่กำหนด Chunking เป็น Semantic อาจเป็นเพราะสาเหตุดังนี้

ภาษาไทยมีความซับซ้อนทางไวยากรณ์ประโยคเดียวอาจตีความได้หลายแบบ ขึ้นอยู่กับโครงสร้างประโยคและบริบท Semantic Chunking อาจมีปัญหาในการระบุกลุ่มคำที่ถูกต้องตามไวยากรณ์โดยเฉพาะอย่างยิ่งในประโยคที่ซับซ้อนหรือคลุมเครือ

Semantic Chunking จำเป็นต้องได้รับการฝึกฝนเกี่ยวกับข้อมูลตัวอย่างจำนวนมากที่มีกลุ่มคำที่ถูกต้องตามไวยากรณ์และความหมาย ถ้าข้อมูลที่ให้เรียนรู้ไม่มากพอ อาจจะไม่สามารถแบ่งกลุ่มคำได้ถูกต้อง

สรุปได้ว่าขนาดของ Chunking มีผลต่อประสิทธิภาพของการค้นคืนข้อมูลเพราะ Chunking ที่เล็กเกินไป อาจไม่สามารถให้บริบทที่เพียงพอ แต่ Chunking ที่ใหญ่เกินไปอาจทำให้ระบบประมวลผลช้าลงและใช้ทรัพยากรมากขึ้น จึงควรเลือกให้เหมาะสมกับประเภทของข้อมูลและภารกิจที่ต้องการ Chunk Size ที่เล็กลงหมายความว่าการ Embeddings จะมีความแม่นยำมากขึ้น ในขณะที่ขนาด Chunk Size ที่ใหญ่ขึ้นหมายความว่าการ Embeddings อาจจะกว้างกว่าแต่อาจพลาดรายละเอียดเล็กน้อยได้

2. Embedding Model

จากผลการทดลอง โมเดลแบบที่ 3 ที่กำหนด Embedding เป็น Text-embedding-3-large เทียบกับโมเดลแบบที่ 4 ที่กำหนด Embedding เป็น bge-m3 พบว่า การกำหนด Embedding เป็น bge-m3 ให้คำตอบที่มีความแม่นยำมากกว่าที่กำหนด Embedding เป็น Text-embedding-3-large เพราะ bge-m3 รองรับภาษาได้หลากหลายภาษามากกว่าและมีประสิทธิภาพดีกว่าในงานคำถามคำตอบ ส่วน Text-embedding-3-large มีประสิทธิภาพดีสำหรับงานบางประเภท เช่น การแปลภาษาอัตโนมัติ

สรุปได้ว่าการเลือก Embedding Model ให้เหมาะสมกับภาษาและงานที่ใช้จะมีผลให้ประสิทธิภาพของผลลัพธ์ดีขึ้น

3. Generator Model

จากผลการทดลอง โมเดลแบบที่ 1 และ 2 ที่กำหนด Generator เป็น Claude-3-sonnet-20240229

เทียบกับโมเดลแบบที่ 3 และ 4 ที่กำหนด Generator เป็น gpt-4-turbo-preview พบว่า การกำหนด Generator เป็น Claude-3-sonnet-20240229 ให้คำตอบที่มีความแม่นยำมากกว่าที่กำหนด Generator เป็น gpt-4-turbo-preview เพราะ claude-3-sonnet มีประสิทธิภาพดีสำหรับการตอบคำถาม แต่ gpt-4-turbo-preview มีประสิทธิภาพดีสำหรับงานบางประเภท เช่น การสร้างข้อความ

สรุปได้ว่าการเลือก Generator Model ให้เหมาะสมกับภาษาและงานที่ใช้จะมีผลให้ประสิทธิภาพของผลลัพธ์ดีขึ้น

โดยจักรกลสนทนา (Chatbot) มีส่วนช่วยนำเทคโนโลยีป้องกันประเทศมาใช้ให้เกิดประโยชน์ได้หลายวิธีในการพัฒนาต่อยอดในอนาคต ตัวอย่างเช่น

1. การสนับสนุนด้านการฝึกอบรม จักรกลสนทนาสามารถช่วยให้ทหารเข้าถึงข้อมูลการฝึกอบรมหรือคู่มือการใช้งานอุปกรณ์ต่าง ๆ ได้อย่างรวดเร็วและง่ายดาย

2. การให้คำแนะนำและคำตอบ จักรกลสนทนาสามารถตอบคำถามเกี่ยวกับขั้นตอนการปฏิบัติในสถานการณ์ฉุกเฉินหรือการใช้งานอุปกรณ์ทางทหาร

3. การให้บริการลูกค้า สำหรับหน่วยงานที่เกี่ยวข้องกับการขายหรือการจัดซื้ออาวุธ จักรกลสนทนาสามารถตอบคำถามและให้ข้อมูลแก่ลูกค้าอย่างมีประสิทธิภาพ

การนำเทคโนโลยี LLM และ RAG มาใช้ในด้านความมั่นคงสามารถช่วยเพิ่มประสิทธิภาพในด้านต่าง ๆ ดังนี้

1. การวิเคราะห์และประมวลผลข้อมูลข่าวกรอง: LLM สามารถใช้ในการวิเคราะห์ข้อมูลข่าวกรองจากแหล่งข้อมูลต่าง ๆ เช่น รายงานข่าว บทความ หรือข้อมูลโซเชียลมีเดีย เพื่อสกัดข้อมูลที่สำคัญและเกี่ยวข้อง ด้วยความสามารถของ RAG ในการดึงข้อมูลที่เกี่ยวข้องจากฐานข้อมูลขนาดใหญ่ และสรุปข้อมูล LLM สามารถสร้างรายงานที่สั้นกระชับ และมีความหมาย

2. การตรวจจับและป้องกันภัยคุกคามทางไซเบอร์: LLM สามารถใช้ในการวิเคราะห์ Log Files และข้อมูลการใช้งานระบบเพื่อตรวจจับกิจกรรมที่ผิดปกติหรือมีความเสี่ยง วิเคราะห์ตัวอย่างมัลแวร์และสร้างรายงานเกี่ยวกับพฤติกรรมและลักษณะของมัลแวร์

3. การสนับสนุนการตัดสินใจในสถานการณ์ฉุกเฉิน: LLM และ RAG สามารถใช้ในการรวบรวมและวิเคราะห์ข้อมูลจากแหล่งต่าง ๆ ในเวลาจริง เพื่อให้ข้อมูลที่เป็นประโยชน์แก่ผู้ตัดสินใจ สามารถใช้ LLM ในการสร้างแบบจำลองและจำลองสถานการณ์เพื่อประเมินผลกระทบและความเสี่ยงจากการตัดสินใจต่าง ๆ

4. การฝึกอบรมและพัฒนาทักษะ: LLM สามารถสร้างเนื้อหาการฝึกอบรมที่มีคุณภาพสูงและครอบคลุมหลายด้าน เช่น การฝึกอบรมด้านความมั่นคงทางไซเบอร์ การป้องกันและตอบโต้ภัยคุกคาม RAG สามารถดึงข้อมูลที่เกี่ยวข้อง และ LLM สามารถตอบคำถามและให้คำปรึกษาแก่เจ้าหน้าที่ในสถานการณ์ต่าง ๆ ได้

5. การสื่อสารและการจัดการข้อมูลภายในองค์กร: LLM สามารถใช้ในการจัดการและจัดระเบียบเอกสารสำคัญ รวมถึงการค้นหาและดึงข้อมูลที่ต้องการได้อย่างรวดเร็วช่วยสร้างข้อความและรายงานที่ชัดเจนและถูกต้อง เพื่อใช้ในการสื่อสารภายในองค์กร

การนำเทคโนโลยี LLM และ RAG มาใช้ในด้านความมั่นคงมีศักยภาพสูงในการเพิ่มประสิทธิภาพ ลดข้อผิดพลาด และเสริมสร้างความมั่นคงให้กับองค์กร การประยุกต์ใช้เทคโนโลยีนี้

ควรคำนึงถึงความเป็นส่วนตัวและความปลอดภัยของข้อมูล เพื่อให้การใช้งานเป็นไปอย่างปลอดภัยและมีประสิทธิภาพสูงสุด LLM และ RAG สามารถประมวลผลและวิเคราะห์ข้อมูลจากแหล่งข้อมูลหลายแหล่งในเวลาอันสั้น ซึ่งช่วยให้เจ้าหน้าที่สามารถตอบสนองต่อภัยคุกคามได้รวดเร็วและมีประสิทธิภาพมากขึ้น RAG ช่วยให้สามารถดึงข้อมูลที่เกี่ยวข้องและทันสมัยจากฐานข้อมูลขนาดใหญ่ ซึ่งช่วยเสริมสร้างการตัดสินใจที่มีประสิทธิภาพมากขึ้น

การนำ LLM และ RAG มาใช้ต้องคำนึงถึงความเป็นส่วนตัวของข้อมูลส่วนบุคคล เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต ต้องมีการใช้เทคโนโลยีและวิธีการที่ช่วยปกป้องข้อมูลส่วนบุคคล เช่น การเข้ารหัสข้อมูลและการควบคุมการเข้าถึงข้อมูล ต้องมีการรักษาความปลอดภัยของข้อมูลอย่างเข้มงวด เพื่อป้องกันการโจมตีทางไซเบอร์และการรั่วไหลของข้อมูล ต้องมีการทดสอบและประเมินความปลอดภัยของระบบอย่างสม่ำเสมอ เพื่อให้แน่ใจว่าข้อมูลและระบบขององค์กรมีความปลอดภัยสูงสุด

ตารางที่ 1 จำนวนข้อที่ตอบถูกทั้งหมด ตอบผิดทั้งหมด และตอบผิดบางส่วนของระบบจักรกลสนทนา โดยใช้โมเดลในการทดลอง 4 โมเดล

รายการผลลัพธ์	ประเภทของโมเดล			
	Model 1	Model 2	Model 3	Model 4
จำนวนข้อที่ตอบถูกทั้งหมด	45	44	39	43
จำนวนข้อที่ตอบผิดทั้งหมด	3	4	10	6
จำนวนข้อที่ตอบผิดบางส่วน	1	1	0	0

ตารางที่ 2 % Accuracy ของการตอบคำถามของระบบจักรกลสนทนา โดยใช้โมเดลในการทดลอง 4 โมเดล แบ่งตามชุดคำถามแต่ละประเภท

ประเภทของคำถาม	ประเภทของโมเดล			
	Model 1	Model 2	Model 3	Model 4
Extraction Test 13 ข้อ	100 %	100 %	92 %	100 %
Logic Test 10 ข้อ	90 %	90 %	80 %	90 %
Hallucination Test 5 ข้อ	80 %	100 %	60 %	80 %
Robustness Test 11 ข้อ	100 %	86 %	82 %	91 %
Instruction Test 10 ข้อ	85 %	80 %	70 %	70 %
รวมทั้งหมด 49 ข้อ	93 %	91 %	80 %	88 %

6. เอกสารอ้างอิง

[1] L. Ouyang *et al*., "Training Language Models to Follow Instructions with Human Feedback," 2022, arXiv:2203.02155.

[2] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query Rewriting for Retrieval -augmented Large Language Models," 2023, arXiv:2305.14283.

[3] A. Vaswani *et al*., "Attention is all you need," in *31st Conf. Neural Inf. Process. Syst. (NIPS 2017)*, Long Beach, CA, USA, 2017, pp. 6000-6010.

[4] C. C. H. Chan, Y. - C. Lin, and P. - C. Shih, "Natural Language Processing for Supply Chain with Chat GPT," in *2023 IEEE 5th Int. Conf. Architecture, Construction, Environ. and Hydraul. (ICACEH)*, Taichung, Taiwan, 2023, pp. 28 - 31, doi: 10.1109/ICACEH59552.2023.10452639.

[5] M. Chen *et al*., "Evaluating Large Language Models Trained on Code," 2021, arXiv:2107.03374.

[6] A. Chowdhery *et al*., "PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res*., vol. 24, pp. 1 – 113, 2023.

[7] W. Yu *et al*., "Generate rather than Retrieve: Large Language Models are Strong Context Generators," 2022, arXiv:2209.10063.

[8] S. Yao *et al*., "ReAct: Synergizing Reasoning and Acting in Language Models," 2023, arXiv:2210.03629.

[9] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and Narrowing the Compositionality Gap in Language Models," 2022, arXiv:2210.03350.

[10] O. Khattab *et al*., "Demonstrate-Search -Predict: Composing Retrieval and Language Models for Knowledge -intensive NLP," 2022, arXiv:2212.14024.

[11] W. Shi *et al*., "RePlug: Retrieval -augmented Black-box Language Models," 2023, arXiv:2301.12652.

[12] Z. Li, B. Peng, P. He, M. Galley, J. Gao, and X. Yan, "Guiding Large Language Models via Directional Stimulus Prompting," in *37th Conf. Neural Inf. Process. Syst. (NeurIPS 2023)*, New Orleans, Louisiana, USA, 2003, pp. 1 - 27.

[13] J. Wei *et al*., "Finetuned Language Models are Zero-shot Learners," 2022, arXiv:2109.01652.

[14] T. B. Brown *et al*., "Language Models are Few-shot Learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, vol. 33, pp. 1877 - 1901.

[15] J. Wei *et al*., "Chain-of-thought Prompting Elicits Reasoning in Large Language Models," in *Adv. Neural Inf. Process. Syst.*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, vol. 35, pp. 24824 - 24837.

[16] J. Chen *et al*., "M3-Embedding: Multi-lingual, Multi-functionality, Multi-granularity Text Embeddings through Self-knowledge Distillation," 2024, arXiv:2402.03216.

[17] L. Merrick, D. Xu, G. Nuti, and D. Campos, "D. Arctic-Embed: Scalable, Efficient, and Accurate Text Embedding Models," 2024, arXiv:2405.05374.

[18] M. Enis and M. Hopkins, "From LLM to NMT: Advancing Low-Resource Machine Translation with Claude," 2024, arXiv:2404.13813.