International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Finding Greenhouse Gas Emission Factor Based on Large Language Models

Suchanan Krabuanyutha[1], Yachai Limpiyakorn[2]
[1, 2]Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand
Email address: [1]6670319521@student.chula.ac.th, [2]yachai.l@chula.ac.th

***Abstract*—*This research has been conducted to explore large language models (LLMs) and various techniques that enhance their capabilities, including prompt engineering and domain-specific knowledge expansion. The study involves the design and development of AI agents to return the greenhouse gas emission factor based on the database maintained by the Thailand Greenhouse Gas Management Organization (Public Organization). The Retrieval Augmented Generation is used to locate the stored chunks that have high similarity to the input query. Subsequently, the AI agent will infer the single answer of emission factor value associated to the query activity. The performance evaluation of the selected five medium and large scale LLMs, is promising.*

***Keywords*—** *Large language model: prompt engineering: retrieval augmented generation: greenhouse gas emission factor: sustainability.*

I. Introduction

Nowadays, the world is facing increasingly severe and unpredictable weather conditions. The extreme weather significantly impacts on ecosystems, economy, agriculture, and human health. These challenges have heightened global awareness regarding the importance of sustainable development. In particular, the concept of Environmental, Social and Governance (ESG) has emerged as a critical framework promoting responsible business practices. ESG has gained substantial attention from governments, businesses, and investors worldwide as a key mechanism for addressing the intensifying climate crisis.

In Thailand, the government has formulated a national strategic plan aiming to achieve carbon neutrality between 2030 and 2050, and to reach net-zero greenhouse gas emissions between 2050 and 2065, in accordance with the Paris Agreement.

In alignment with these national goals, entrepreneurs in Thailand have begun to place greater emphasis on the assessment of both the Carbon Footprint for Organization (CFO) and the Carbon Footprint of Product (CFP). These assessments not only facilitate smoother export processes to countries with stringent environmental regulations but also serve as essential tools for decarbonization or identifying direct and indirect means of greenhouse gas emission reduction. Furthermore, efforts to reduce carbon footprints significantly contribute to long-term cost reduction in production processes.

The process of preparing carbon footprint data is resource consuming, inherently complex and requires a high degree of domain-specific knowledge. Practitioners are required to identify and collect data pertaining to all carbon-emitting activities, possess a comprehensive understanding of carbon emission classifications in accordance with the Greenhouse Gas Protocol or other standardized methodologies for the quantification and reporting of greenhouse gas emissions, accurately calculate the volume of emissions, and regularly update relevant greenhouse gas emission factors.

In order to alleviate the burden and reduce the costs due to manually searching for greenhouse gas emission factors associated with reported activity data, this study proposes the development of AI Agents leveraging Large Language Model technology. The primary function is to infer the emission factor matched with the user input query from the database maintained by the Thailand Greenhouse Gas Management Organization (Public Organization), hereafter referred to as TGO.

II. Background

A. Greenhouse Gases Emission Factor

According to the Kyoto Protocol adopted in 1997, the recognized greenhouse gases (GHGs) include Carbon Dioxide (CO_2), Methane (CH_4), Nitrous Oxide (N_2O), Hydrofluorocarbons (HFCs), Perfluorocarbons (PFCs), Sulphur Hexafluoride (SF_6), and Nitrogen Trifluoride (NF_3).

The Global Warming Potential (GWP) is a measurement that compares the quantity of heat absorbed by a particular greenhouse gas to the amount absorbed by carbon dioxide over a certain period of time, usually 100 years. Each greenhouse gas has a different GWP value. For example, 1 kg of methane has a GWP of 28, meaning that it has a global warming effect equivalent to 28 kg of CO_2 over 100 years [1].

Greenhouse gas emissions are expressed in terms of kilograms of carbon dioxide equivalent ($kgCO_2e$), a standard metric referred to as carbon emissions.

The formula for calculating carbon emissions is as follow.

$$Carbon\ Emissions = Activity\ Data \times Emission\ Factor$$

where emission factor (EF) is a coefficient that describes the rate at which a given activity releases greenhouse gases into the atmosphere. Activities may include fuel combustion, transportation, or indirect emissions from energy consumption.

B. Carbon Footprint Reporting

Carbon footprint reporting in Thailand is overseen by TGO. The organization is responsible for establishing guidelines, standards, and verifying the carbon footprint calculation data. There are two types of reporting: 1) CFP, and 2) CFO.

Based on the Life Cycle Assessment (LCA) methodology, CFP or Carbon Footprint of Product assesses the total greenhouse gas emissions released throughout the life cycle of a product consisting five main stages: raw material procurement, production, transportation, use, and waste disposal. Whereas CFO or Carbon Footprint for Organization evaluates the total greenhouse gas emissions released by an organization over a specific period, such as annually. It includes direct emissions (Scope 1), indirect emissions from energy use (Scope 2), and other indirect emissions (Scope 3).

C. Large Language Model (LLM)

LLM is a large-scale language model that uses deep learning to process and generate new information, allowing it to understand and predict human language with high accuracy [2]. LLM is the foundation of Generative AI, used in various tasks such as automated conversations, data summarization, and content generation.

In the context of LLMs, an instruct model is built on top of a base LLM and is further trained on instruction-specific data. Instruct models are designed to take user instructions and generate the appropriate output based on those prompts. For example, if we asked a base LLM to "explain the concept of gravity", it might generate a general explanation. An instruct model, however, would be more likely to provide a detailed and specific explanation based on the instruction, potentially including formulas, diagrams, or analogies.

Instruct models and Reasoning models differ in approach to problem-solving. While instruct models are trained to execute specific tasks by following instructions precisely and generate outputs based on those instructions, reasoning models are designed to break down problems into smaller steps and solve them through logical reasoning. However, reasoning may be inefficient or expensive for simpler tasks, sometimes it may overthink and produce unnecessary reasoning steps [3].

D. Retrieval Augmented Generation (RAG)

The concept of RAG gained recognition in 2020 [4]. It is a technique commonly applied in conjunction with LLMs. The RAG architecture combines the strengths of pretrained language models with an information retrieval system, enhancing the language model's capabilities by accessing and integrating knowledge from external data sources.

E. Prompt Engineering

Prompt Engineering is a new field focused on developing and fine-tuning prompts to use language models effectively for various applications and research topics. Prompt engineering skills help users better understand the capabilities and limitations of LLMs.

Researchers use prompt engineering techniques to improve the capabilities of LLMs in a variety of general and complex tasks, such as question answering and mathematical reasoning. Developers use prompt engineering techniques to design effective and robust prompting methods that connect LLMs with other tools.

Techniques used in designing prompts to enhance model performance include: Zero-shot, One-shot, Few-shot, Chain of Thought, Active Prompting, Directional Stimulus, ReAct, Self-Consistency, Knowledge Generation, Automatic Prompt Engineering, Multimodal CoT, and Graph Prompting [5].

The method of Generated Knowledge Prompting is used in this study, which begins by asking the LLM to produce pertinent information before responding to a question. This preliminary knowledge serves as additional contextual information, enabling the LLM to provide more accurate and meaningful responses, while also reducing the occurrence of hallucinations in the model’s output.

III. RELATED WORK

Huo et al. [6] presented an approach to text simplification, aiming at enhancing the readability and comprehensibility of textual content. Chinese language was chosen due to its complex grammatical structures and vocabulary. The study employed LLMs, incorporating both prompt engineering and fine-tuning techniques to optimize the simplification process. The experimental results reported that prompt engineering, especially when incorporating grammatical rules yields higher performance scores compared to models that are fine-tuned directly.

Chaubey et al. [7] conducted a comparative analysis of two chatbot development methodologies: RAG Fine-Tuning and Prompt Engineering, both of which are extensively utilized in contemporary chatbot systems. The study involved the design and implementation of two distinct chatbot prototypes, each employing one of the techniques, with the goal of evaluating their respective performance across several dimensions consisting of accuracy, operational flexibility, computational resource requirements, and development time. The experimental results demonstrated that RAG Fine-Tuning yielded superior accuracy and it particularly well-suited for applications requiring domain-specific knowledge. Whereas Prompt Engineering provided notable advantages in terms of implementation flexibility, faster development cycles, and lower resource consumption. Based on these findings, the authors proposed a hybrid approach that integrates the strengths of both techniques, with the aim of optimizing the efficiency and effectiveness of chatbot development in future applications.

Beri and Srivastava [8] aimed to explore advanced methodologies, techniques, and strategic frameworks for the design and development of prompts tailored for the effective utilization of LLMs, such as GPT. The research emphasized a comprehensive analysis of the critical components, processes, and design paradigms that influence the quality, coherence, and task-alignment of model-generated responses.

Findings from the study underscored the pivotal role of prompt engineering in shaping the outcomes produced by LLMs. Several fundamental prompting techniques have emerged as standard practices within the field, including: Zero-shot Prompting, Few-shot Prompting, Chain-of-Thought Prompting, Self-consistency and Instruction Tuning.

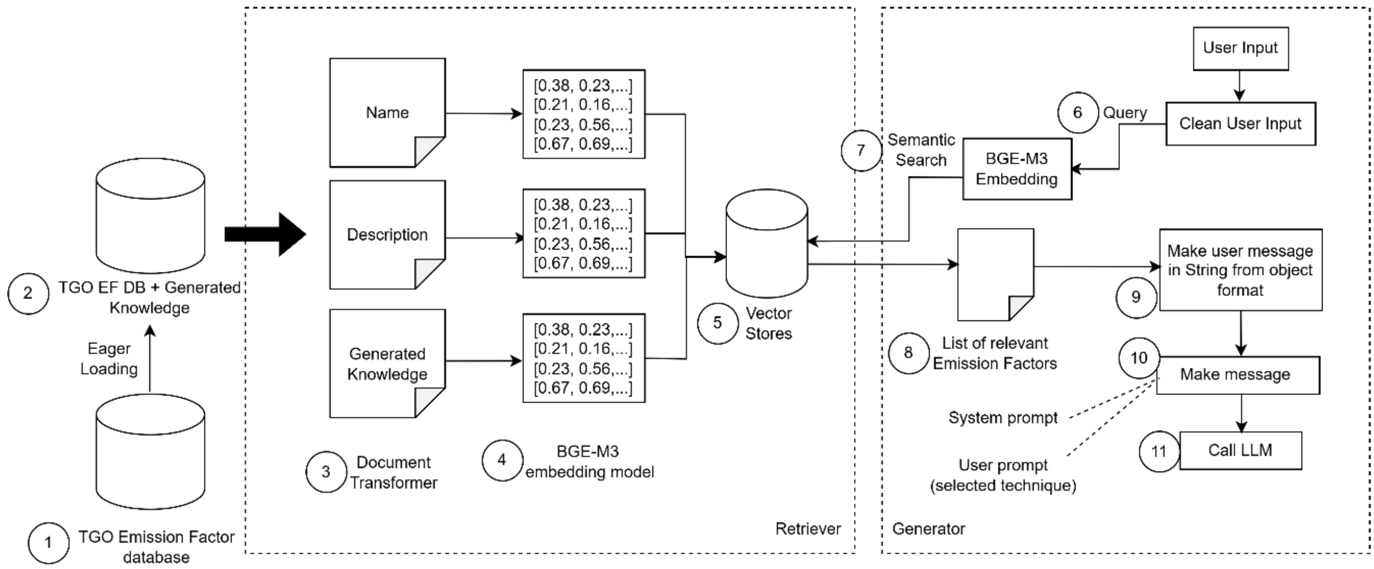

Fig. 1. Overview of research methodology.

IV. METHODOLOGY

Fig. 1. illustrates an overview of the research methodology mainly separated into two parts: Retriever and Generator. In part of Retriever, RAG plays an important role for semantic search to acquire knowledge. The technique of Generated Knowledge Prompting is applied to generate relevant knowledge added to the initial TGO emission factor database. This would help the models understand the context better and provide more accurate and nuanced answers. In part of Generator, the two AI Agents are built with two distinct user prompts, using Name, and Name + details (Generated Knowledge). The system prompt is designed to effectively direct the AI to perform the intended tasks.

The methodology encompasses the following key steps.

A. Emission Factor Database Construction

The EF transactions are collected from the TGO website [9] and stored in a database. Next, all entries in the database are utilized to generate knowledge through the Eager Loading method, thereby creating a database that not only contains the activity emission factors but also provides enhanced details.

B. Document Transformation

The contents contained in the EF database, including Activity Name, Description, and Details or Generated Knowledge, are transformed into vectors based on their meanings using BGE-M3 and stored in a Vector Store database.

C. User Input

The user enters the name of the activity querying for the associated EF value. The entered text is cleaned to get rid of extraneous spaces, special characters, and grammatical mistakes.

The user interface is developed using LangChain, example as illustrated in Fig. 2. This interface is designed to integrate LLMs and supports the input text data.

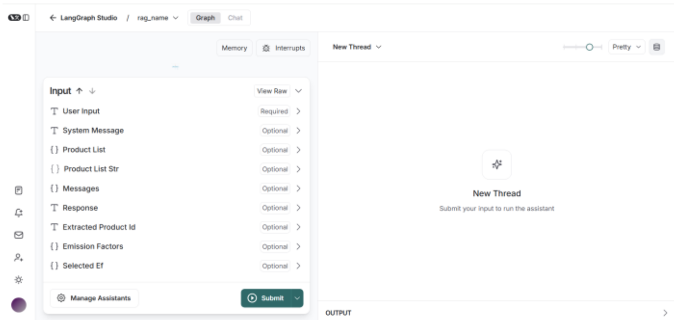

Fig. 2. Langchain user interface.

D. Semantic Search

The input is then converted into vector representation to enable semantic search. Using vector embeddings generated by BGE-M3, the retriever, RAG, fetches the top 15 most relevant data chunks from the EF database. Example is shown in Fig. 3.

E. Prepare Message

The text in Python Object format is converted into a String format.

F. Make Message

The two AI agents are developed using different knowledge. Each agent has a system prompt providing guidelines and a user prompt. The user prompt of the first agent uses the information of Name, while the second agent uses the information of Name + additional details. The Instruct model and the Reasoning model are used to fine-tune both agents.

G. Calling the LLM

The LLM is called to select the most relevant activity from the retrieved 15 candidates, and returns a single answer of the EF associated with the inferred entry in the TGO database.

Suchanan Krabuanyutha and Yachai Limpiyakorn, "Finding Greenhouse Gas Emission Factor Based on Large Language Models," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 12, pp. 62-67, 2025.

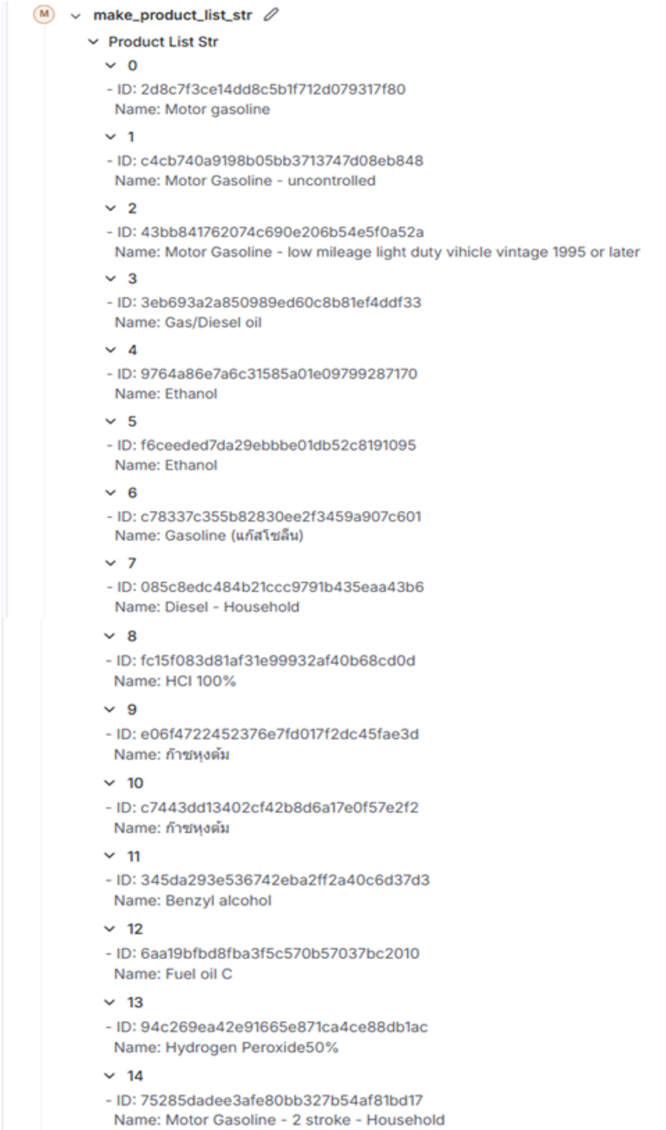

Fig. 3. Example retrieval of the top 15 relevant candidates.

V. DEMONSTRATION

Refer to the GHG Protocol [10], greenhouse gas emissions are classified into three categories:

- Scope 1 Direct Emissions: The reporting firm owns or has control over the sources of these direct GHG emissions.
- Scope 2 Indirect Emissions from purchased Electricity: These are indirect GHG emissions from the generation of purchased electricity consumed by the reporting company.
- Scope 3 Other Indirect Emissions: These are all other indirect GHG emissions that occur as a result of the activities of the reporting company, but from sources not owned or controlled by the reporting company.

Two queries of emission factors of the activities categorized in Scope 1 and Scope 3 are demonstrated.

The first query asking for the EF of Gasohol 95 fossil fuel combustion that produces GHG emissions.

Fig. 4. The greenhouse gas emission factor obtained by LLMs.

Fig. 4 reports the LLM's judgement of selecting EF of แก๊สโซลีน (Gasoline) as the answer for the user input "Gasohol 95". Since the TGO EF database does not contain the item named Gasohol 95, the LLM infers the most relevant entry based on the fact that Gasohol 95 is the mixture of 90% gasoline and 10% ethanol. Therefore, the EF of Gasoline is selected as shown in Fig. 5. For Scope 1, the TGO database will report all greenhouse gases emissions for the activity, in addition to the total GHG in the unit of $kgCO_2e$.

The second query asking for the EF of กระดาษ A4 (paper A4) categorized in Scope 3 indirect emissions as purchasing office supplies promotes carbon emissions indirectly caused by the manufacturing process of product. The LLM returns the valid emission factor of กระดาษพิมพ์เขียนแบบไม่เคลือบผิว (Uncoated Paper) as shown in Fig. 6. Note that the TGO database does not report the constituents of GHG total in case of Scope 3.

It is evident that user searches are generally expressed in everyday language, whereas entries within the TGO database are characterized by highly specialized language, often comprising detailed descriptions of the attributes of each item rather than the terms commonly used in everyday communication. This disparity necessitates that individuals conducting carbon footprint assessments possess specialized knowledge regarding the components or classifications of the items in question.

VI. EXPERIMENTAL RESULT

The two AI Agents are developed to assess two kinds of user prompts containing the information of

- Name
- Name + Details (description and generated knowledge)

Totally 7 models are used as following:

- Llama-70B (Instruct) = Llama-3.3-70B-Instruct-Turbo
- Qwen-72B (Instruct) = qwen-2.5-72b-instruct
- Llama-70B (Reasoning) = DeepSeek-R1-Distill-Llama-70B
- Qwen-32B (Reasoning) = DeepSeek-R1-Distill-Qwen-32b
- ChatGPT = gpt-4o-mini
- Deepseek-V3 = deepseek-chat-v3-0324
- Gemini = gemini-2.5-pro-exp

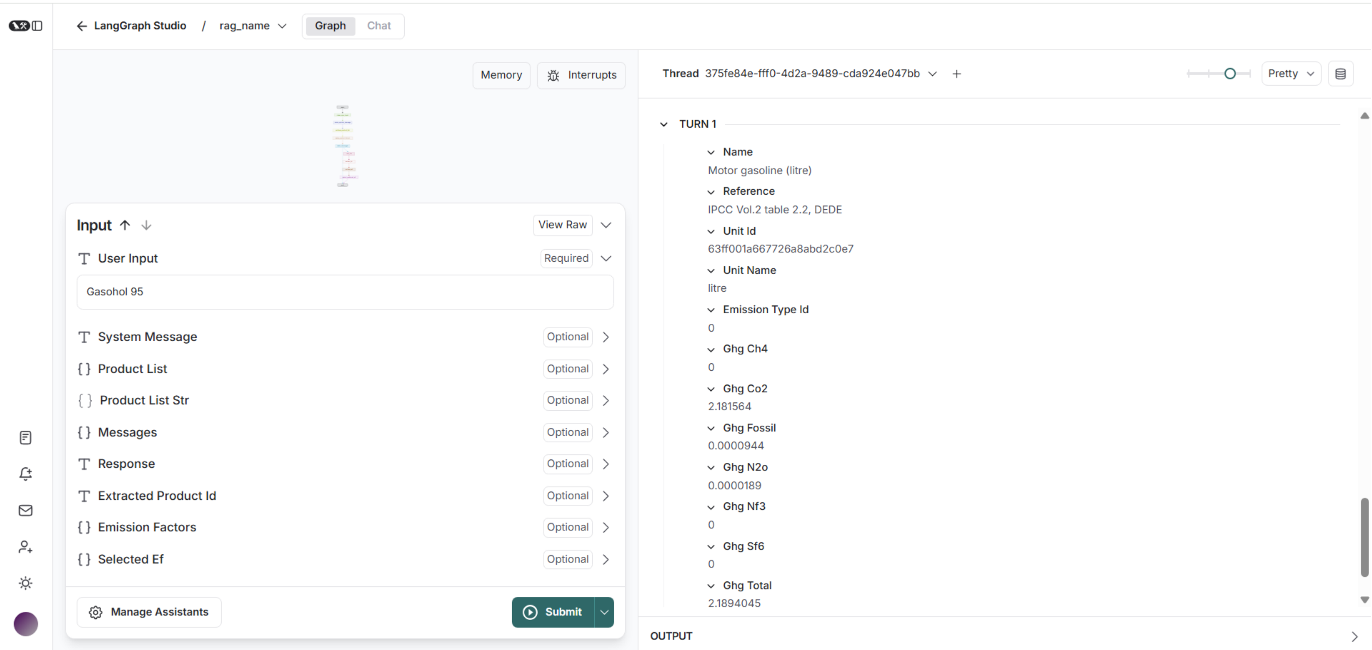

Fig. 5. Emission factors reported for Gasohol 95.

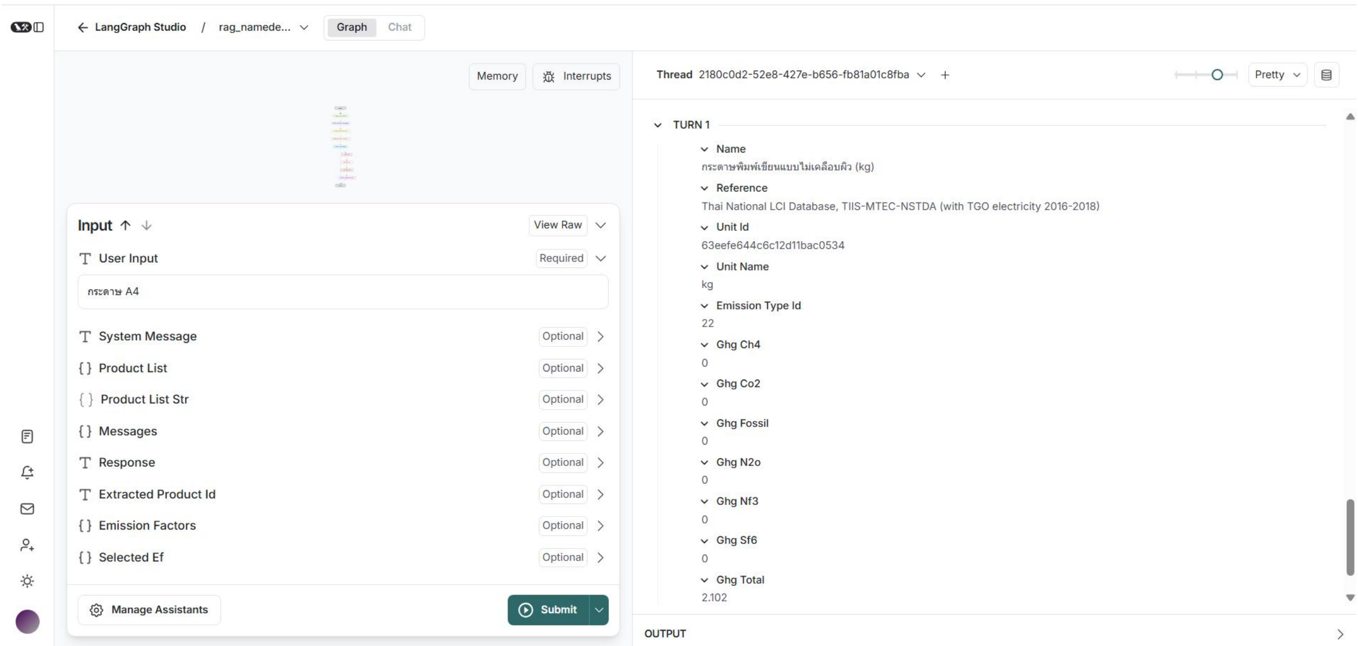

Fig. 6. Emission factors reported for กระดาษ A4 (A4 paper).

The selected LLMs can be classified into medium-scale and large-scale. Llama and Qwen are considered medium models, whereas ChatGPT, DeepSeek, and Gemini are regarded as large models.

The system prompt of each AI Agent is instructed by interaction with the Instruct model as guidelines. The performance is then evaluated. Further distilling reasoning capability with the chosen Reasoning models is conducted. The performance is evaluated compared to that of the Instruct model.

The experiments involved generating 100 user search queries fed into each model listed above. The results are summarized in Table 1 reporting the number of correctly returned entries. The instruct model of Llama-70B prompting with the information of Name outperforms the rest, followed by Qwen-72B and ChatGPT, respectively. The performance of those medium-scale models when adding description and generated knowledge to prompt tends to decline, while the large-scale model, ChatGPT, yields the highest score. When distilling reasoning capabilities, the performance of those medium-scale models decreases.

The fact that the size of TGO database is not large containing 762 records with blank, identical or similar description. Adding more details may be not useful.

TABLE I. Comparisons between models and prompting.

Prompt	**Medium**				**Large**		
	Instruct		**Reasoning**				
	Llama-70B	**Qwen-72B**	**Llama-70B**	**Qwen-32B**	**ChatGPT**	**DeepSeek-V3**	**Gemini**
Name	**79**	78	67	65	75	64	72
Name + Details	72	64	65	64	**74**	64	72

Here, the Reasoning model exhibits lower performance compared to the Instruct model. Since Reasoning models suit for logical reasoning to solve complex problems, at this stage of work, distilling reasoning capabilities may exceed the requirements and results in reduced accuracy.

The experimental results of medium-scale LLMs trained with Instruct model seem promising. Using large LLMs would consume resources beyond what is necessary. In contrast, performance tends to decline when using a Reasoning model, as such models are designed for complex tasks. However, for the future full-blown system, when the input will cover more informative contents extracted from documents such as bills, receipts, and statements. Reasoning capabilities would be required for complex problem solving. Moreover, the EF database should be improved and expanded to cover more entries from other reliable regional, international sources.

VII. Conclusion

This research was conducted to develop a prototype AI agent and identify a model that is both efficient and cost-effective. The goal is to support entrepreneurs by reducing the burden, costs, and expenses associated with preparing carbon footprint reports. It aims to encourage businesses, especially small and medium-sized enterprises (SMEs), to analyze their carbon emissions. This effort represents a critical first step toward achieving long-term sustainability.

References

[1] GHG Emission Factors Hub, by EPA, 2024. Available: https://www.epa.gov/climateleadership/ghg-emission-factors-hub

[2] What is Large Language Model (LLM)?, by BUA Labs, 2024. Available: https://www.bualabs.com/archives/4402/what-is-large-language-model-llm/

[3] What Are Reasoning Models and Why You Should Care, 2025. Available: https://www.bigdatawire.com/2025/02/04/what-are-reasoning-models-and-why-you-should-care/

[4] B. Tural, Z. Örpek and Z. Destan, "Retrieval-Augmented Generation (RAG) and LLM Integration," *2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, İstanbul, Turkiye, pp. 1-5, 2024.

[5] Prompt Engineering Guide, DAIR.AI, 2024. Available: https://aigencorp.com/what-is-prompt-engineering/

[6] H. Huo, Z. Fu, H. Yu, S. Yang and G. Tang, "Chinese Text Simplification Based on Large Language Models," *2024 International Conference on Computational Linguistics and Natural Language Processing (CLNLP)*, Yinchuan, China, pp. 52-56, 2024.

[7] H. K. Chaubey, G. Tripathi, R. Ranjan and S. k. Gopalaiyengar, "Comparative Analysis of RAG, Fine-Tuning, and Prompt Engineering in Chatbot Development," *2024 International Conference on Future Technologies for Smart Society (ICFTSS)*, Kuala Lumpur, Malaysia, pp. 169-172, 2024.

[8] G. Beri and V. Srivastava, "Advanced Techniques in Prompt Engineering for Large Language Models: A Comprehensive Study," *2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, pp. 1-4, 2024.

[9] Carbon footprint for organization, by TGO, 2024. Available: https://thaicarbonlabel.tgo.or.th/index.php?lang=TH&mod=YjNKbllXNXBlbUYwYVc5dVgyVnRhWE56YVc5dQ

[10] The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard (revised edition), World Resource Institute. Available: https://ghgprotocol.org/