LARGE LANGUAGE MODEL ASSISTED DATA QUERY FOR PINEAPPLE PRODUCTION

Badril Abu Bakar
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: badril@mardi.gov.my

Siti Noor Aliah Baharom
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: aliah@mardi.gov.my

Mohd Taufik Ahmad
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: taufik@mardi.gov.my

Mohd Aufa Md Bookeri1
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: aufa@mardi.gov.my

Mohd Nizam Zubir
Horticulture Research Centre
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: znizam@mardi.gov.my

Mohd Zamri Khairi Abdullah
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: mzamri@mardi.gov.my

Muhammad Hariz Musa
Engineering Research Centre,
Malaysian Agricultural Research and Development Institute (MARDI),
43400 Serdang, Selangor, Malaysia
Email: mhariz@mardi.gov.my

ABSTRACT

Generative artificial intelligence is showing unprecented performance in various fields and taks. Large Language Models in particular have sped up the development of machine learning applications. This work presents a large language model based technique to query data collected during MD2 pineapple crop production. Retrieval Augmented Generation was used to feed structured and unstructured data to a large language model in order to train and fine tune the model. The performance of the model was then measured using actual and predicted question-answer pairs. Results showed that the model had a 86% and 75% correct answer rate respectively for structured and unstructured data. However, results showed that the model had a 63% correct answer rate when an answer to a question needed to refer to both structured and unstructured data.

Keywords: pineapple cultivation; MD2; large language model; retrieval augmented generation; generative artificial intelligence.

INTRODUCTION

Pineapple (Ananas comosus) is a tropical fruit highly sought after for its sweet taste [1]. The market value of world pineapple production stood at USD 27.08 billion in 2022. The largest producers in the world are Indonesia (3.20 Mt), followed by Philippines (2.91 Mt) and Costa Rica (2.90 Mt) [2]. Malaysia is the 25th largest producer with a total production of 287,799 tonnes. Although some mechanization and automation systems exist, pineapples are largely planted manually worldwide [3]. This is also true in Malaysia [4].

The Malaysian government is pushing to modernize the agricultural industry under the National Agrofood Policy 2.0 [5]. Agricultural players are urged to embrace the fourth industrial revolution[6]. Heeding the call, the Malaysian Agricultural Research and Development Institute (MARDI) is currently developing a smart MD2 pineapple production system.

Central to any smart production system, is the ability to make management decisions based on the data being presented to it [7]. The system then has to somehow gain insight from the data. This can be done using machine learning algorithms such as support vector machines or neural networks [8 - 10]. Data is usually presented in a tabular database or structured format. A new approach is to make use unstructured data to gain insight about the process being monitored.

A subset of generative artificial intelligence called large language models (LLM) are able to process unstructured data such as natural language and output a response [11], [12]. Models GPT-4 by OpenAI and LLAMA2 by Meta use the deep learning neural network algorithm to predict a series of words one after the other by taking natural language as input context. These words then form coherent sentences and paragraphs [13], [14].

LLMs have shown exceptional performance in answering questions regarding subjects that it has been trained on. This includes philosophy, mathematics, computer coding and many more [15]. It has shown that it has the ability to solve high school level mathematics by reasoning. However, since LLMs are trained with datasets limited to a certain cut-off date, it doesn't do well when asked with questions on information that might have happened after the cut-off date. Another issue with LLMs is processing structured data [16], [17], [18]. A way to address this issue was described in [19]. The author proposed a model that learned to reduce the input context using an on-policy reinforcement learning algorithm.

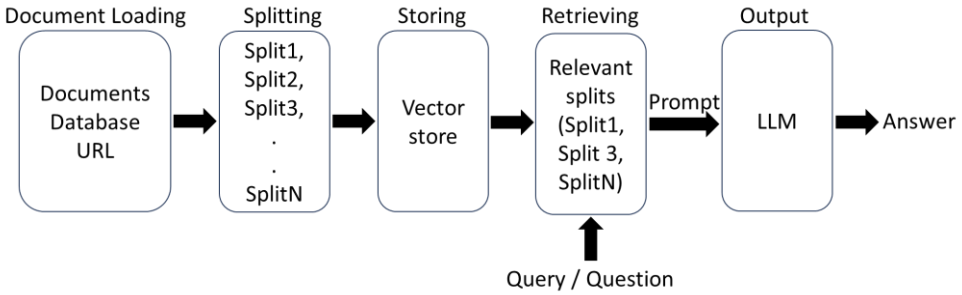
An interesting use case would be if LLMs can be trained on propriety datasets such as confidential company documents, or scientific experiments. One can then prompt question about the data. In order to do this however, whole documents would have to be fed in as part of the prompt. For large documents and datasets, this could be a challenge.

An LLMs business model is usually to charge a certain amount of money for a prompt or query based on the number of characters or words that is being fed to the LLM. A group of words or characters are called tokens. The more tokens are passed in a query or prompt, the more expensive it would be. Feeding a whole document such as the MD2 crop production protocol that has over 20,000 words is equivalent to about roughly 27,000 tokens. Prompting a query with this amount of tokens to GPT-4 model for example would cost USD 0.90 each for each prompt. This would get very expensive as the number of prompts increase.

Retrieval augmented generation (RAG) is a way to preprocess a query or a prompt on our own document or data locally before feeding the prompt to an LLM [20]. This greatly reduces the amount of tokens that need to be fed to the LLM.

RAG works by breaking down the input document into smaller chunks or splits. The contents of a split are then numerically represented in a series of vectors called embeddings and kept in a vector store. When a query is made, the query itself is represented numerically and compared to the embeddings in the vector store. A number of similar embeddings are chosen and fed together with the original query to an LLM. The LLM then responds to the query based on the embeddings fed with it. Fig. 1 shows the overall workflow of RAG.

Figure 1. Overview of retrieval augmented generation.



The implementation of RAG has the potential to assist crop production of MD2 pineapples by querying data collected from the field and comparing it to a crop production guide in the form of a document by feeding the generated prompt from the RAG technique to an LLM. Therefore, the first objective of this work is to evaluate the performance of different RAG variations in retrieving relevant information from the input. The second objective of this work is to evaluate the performance of two LLMs (GPT-4 and LLAMA2) in answering the queries when applying the RAG technique on structured and unstructured data.

METHODOLOGY

Software Platform

The Python 3.11.3 programming language was used to prompt two large language models through their application programming interface (API). The two LLMs chosen were GPT-4 and LLAMA2 [21], [22]. These two represent models represent the state-of-the-art in generative artificial intelligence.

MD2 Pineapple Cultivation Data Input Sources

The data input to be queried came from two sources. The first source was the MD2 pineapple variety crop production protocol which entailed how the MD2 pineapple variety should be planted[23]. It contained the steps to be taken from preparing land before planting all the way through harvesting of the fruits. This represented data that was unstructured in the form of written natural language. The document was 60 pages long.

The second source of input came from data collected in the field. Crop parameters, work log, environmental, scouting as well as production data were gathered at regular intervals through sensors and human input. Table 1 shows the complete data fields

that were collected. In all 40 parameters were included in a tabular structure format. A total of 5,500 data points were collected spanning a period of 14 months during the 2023 – 2024 plaNting season.

Table 1. Data parameters collected throughout the planting season. Four data types were collected. They are; production data, crop data, environmental data and work log data.

Data type	Parameter
Production	plot name, plot size, planting date, harvesting date plant count, induction date, seedling size, seedling origin, yield, total input use
Crop	plant height, leaf length, leaf width, leaf colour, leaf count, fruit size, crown size
Environment	temperature, relative humidity, pressure, solar radiatoin, wind speed, precipitation, soil PH, soil EC, soil salinity, soil nutrient level
Work log	task name, task date, task location, task time, task report, operator name, equipment use, input use, task image

Retrieval Augmented Generation

In the document loading stage, the MD2 pineapple variety crop production protocol was loaded as a file in the portable document format (PDF). The field data was loaded as a tabular file in the comma separated values (CSV) format.

In the splitting stage, a total of 198 splits were generated for the PDF file and a total of 5,500 splits were generated for the CSV file. These splits were then saved in the vector store.

In the retrieving stage, A total of 50 query-answer pairs were generated manually to serve as a control. Five methods of retrieving were implemented. They were; similarity search (SS), maximum marginal relevance (MMR), self query (SQ), compression (Comp) and a combination of MMR and Comp (MMR+Comp).

For each query, the relevant splits were compared to the manually generated answer to evaluate its similarity in terms of context. A score of 1 was given if the split was relevant to the query context. A score of 0 was given if the split was irrelevant to the context. The score was aggregated and divided by the total number of queries. This gave a score of how well the retrieval method performed. The method with the highest score was chosen as the retriever.

In the output stage, the query together with all relevant splits were fed to the LLMs in a prompt. The answer to the prompt was then evaluated.

Evaluation of model performance

The evaluation of the LLMs with RAG was done by comparing the predicted answers from the LLMs to the answers generated manually which was the control. The number of correct answers to queries was aggregated and divided by the total number of answers to queries. This was the score given to the LLM. The queries were divided into three categories. The first category was questions that required answers taken from unstructured input data. The second category was questions that required answers taken from structured input data. The last category was questions that required answers taken from both unstructured and structured input data.

RESULTS AND DISCUSSION

Vector Store Retrieval

Retrieval is the centerpiece of the retrieval augmented generation (RAG). Table 2 shows the performance of different variations of the retriever.

Table 2. Performance of different retriever variations. They are similarity search (SS), maximum marginal relevance (MMR), self query (SQ), compression (Conp) and a combination of MMR and Comp (MMR+Comp).

Variation	GPT-4	LLAMA 2
SS	0.72	0.63
MMR	0.85	0.79
SQ	0.83	0.80
Comp	0.73	0.66
MMR+Comp	0.91	0.85

The combination of MMR and compression yielded the best retrieval result among all other variations. This is true for both GTP-4 and LLAMA2. However, GPT-4 performed better against LLAMA2 using all variations of retrieval. Similarity search retrieval had the lowest score among all variations. This is because the similarity search method tries to find the most similar responses among the dataset, which may ignore relevant but diverse responses. The method of MMR on the other hand strives to achieve both relevance to the query and diversity among the responses. The Comp method on the other hand addresses a different issue. Information most relevant to a query may be buried in a document with a lot of irrelevant text. Passing the full document into an

LLM is more expensive and can yield poorer responses. The Comp method tries to compress the response by discarding these irrelevant text. The MMR+Comp method uses the advantages of each individual component

Model Performance

The performance of the model was measured by calculating the number of correct predicted answers to questions over the total number of predicted answers. Table 3 shows the results for three type of queries; namely questions that refers to structured data, questions that refer to unstructured data, and lastly questions that refer to a combination of both structured and unstructured data.

Table 3. Model performance for queries on structured data (St), unstructured data (Un) and structured+unstructured data (St+Un)

Data type	GPT-4	LLAMA 2
Un	0.87	0.78
St	0.79	0.75
St+Un	0.68	0.61

GPT-4 achieved a better score than LLAMA2 on all queries. Both models did best when the query referred to unstructured data for an answer. On structured data, the models fared slightly worse. This is due to the fact that LLMs are better at making sense of unstructured data such as natural language input. They do well at finding information from disparate sources, and group data for efficient analysis. Unstructured data on the other hand have structural dependencies that are different from natural langue input. Synthesizing data from both structured and unstructured data presents a challenge for state-of-the-art LLMs.

The results from this work showed that there is a potential to use LLMs to query propriety data that are not available publicly and would not be available for LLMs to be trained on. These could be data collected in scientific experiments as well as confidential documents in an organization. Furthermore, there is a potential to use LLMs to diagnose crop conditions during production and derive insight into why the crop is in the condition that it is in by comparing data collected from the field to a known standard protocol.

CONCLUSIONS

A technique to query MD2 pineapple crop cultivation data was presented. The results showed that although the LLMs (GPT-4 and LLAMA2) did well on unstructured and structured data individually, the performance of the LLMs decreased when both structured and unstructured data needed to be referred to. Nevertheless, the results in this work is encouraging and it is hypothesized that with additional fine tuning techniques, the model would be able to improve its performance.

Future work will concentrate on extending the capability of the model to be able to diagnose the state of plant cultivation.

REFERENCES

[1] R. M. Q. De Ramos and E. B. Taboada, “Cradle-to-gate life cycle assessment of fresh and processed pineapple in the Philippines,” *Nature Environment and Pollution Technology*, vol. 17, no. 3, pp. 783–790, 2018.
[2] FAO, “Agriculture production data, License: CC BY-NC-SA 3.0 IGO,” Mar. 2022, Accessed: Mar. 25, 2024. [Online]. Available: https://www.fao.org/faostat/en/#data/QCL
[3] S. Cotabato, “A Study on the Production Methods of Conventionally-grown Pineapples in the,” *Magsasaka at Siyetipiko para sa Pag-unlad ng Agrikultura*, no. February, pp. 1–25, 2015.
[4] B. Abu Bakar *et al.*, “A Review of Mechanization and Automation in Malaysia’s Pineapple Production,” *Advances in Agricultural and Food Research Journal*, May 2021, doi: 10.36877/aafrj.a0000206.
[5] MAFI, “Dasar Agromakanan Negara 2.0 2021 - 2030,” Putrajaya, 2021.
[6] A. S. Bujang and B. H. Abu Bakar, “Agriculture 4.0: Data-Driven Approach to Galvanize Malaysia’s Agro-Food Sector Development.”
[7] A. Knierim, M. Kernecker, K. Erdle, T. Kraus, F. Borges, and A. Wurbs, “Smart farming technology innovations – Insights and reflections from the German Smart-AKIS hub,” *NJAS - Wageningen Journal of Life Sciences*, vol. 90–91, Dec. 2019, doi: 10.1016/j.njas.2019.100314.
[8] D. F. Nettleton, D. Katsantonis, A. Kalaitzidis, N. Sarafijanovic-Djukic, P. Puigdollers, and R. Confalonieri, “Predicting rice blast disease: Machine learning versus process-based models,” *BMC Bioinformatics*, vol. 20, no. 1, Oct. 2019, doi: 10.1186/s12859-019-3065-1.
[9] C. Y. N. Norasma, A. R. M. Shariff, E. Jahanshiri, M. Amin, S. Khairunniza-Bejo, and A. R. Mahmud, “SCIENCE & TECHNOLOGY Web-Based Decision Support System for Paddy Planting Management,” *Pertanika J. Sci. & Technol*, vol. 21, no. 2, pp. 343–364, 2013, [Online]. Available: http://www.pertanika.upm.edu.my/
[10] Y. Kawakami *et al.*, “Rice Cultivation Support System Equipped with Water-level Sensor System,” *IFAC-PapersOnLine*, vol. 49, no. 16, pp. 143–148, 2016, doi: 10.1016/j.ifacol.2016.10.027.
[11] W. X. Zhao *et al.*, “A Survey of Large Language Models,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.18223
[12] A. Q. Jiang *et al.*, “Mixtral of Experts,” Jan. 2024, [Online]. Available: http://arxiv.org/abs/2401.04088
[13] H. Touvron *et al.*, “Llama 2: Open Foundation and Fine-Tuned Chat Models,” 2023, [Online]. Available: http://arxiv.org/abs/2307.09288
[14] OpenAI *et al.*, “GPT-4 Technical Report,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.08774

[15] C. Ziems, W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang, "Can Large Language Models Transform Computational Social Science? under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license," *Computational Linguistics*, vol. 50, no. 1, 2024, doi: 10.1162/coli.

[16] S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag, "TabLLM: Few-shot Classification of Tabular Data with Large Language Models," 2023.

[17] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Wang, "HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data," *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, pp. 1026–1036, 2020, doi: 10.18653/V1/2020.FINDINGS-EMNLP.91.

[18] A. Kalyanpur *et al.*, "Structured data and inference in DeepQA," *IBM J Res Dev*, vol. 56, no. 3.4, pp. 10:1-10:14, 2012, doi: 10.1147/JRD.2012.2188737.

[19] Y. Lee, S. Kim, T. Yu, R. A. Rossi, and X. Chen, "Learning to Reduce: Optimal Representations of Structured Data in Prompting Large Language Models," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.14195

[20] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," Dec. 2023, [Online]. Available: http://arxiv.org/abs/2312.10997

[21] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.09288

[22] OpenAI *et al.*, "GPT-4 Technical Report," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.08774

[23] PIP, "Crop production protocol; Production Guide for the Production of the Pineapple Variety MD2 (a handbook for farm managers and technicians)," vol. 2, no. December, p. 60, 2011, [Online]. Available: www.coleacp.org/pip