International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 11, Issue 4, April-2026, http://ijmec.com/, ISSN: 2456-4265

Enhancing Agricultural Decision-Making with Combined Language Models

Mr.CH Gopi[1], Sripada Naga Venkata Sudheer[2], Nenavath venkatesh[3], Syed Asif Ali[4]
[1]Assistant professor; Department of Information Technology Guru Nanak Institutions Technical campus, Hyderabad, India.
[2,3,4]B.Tech Students; Department of Information Technology Guru Nanak Institutions Technical campus, Hyderabad, India.
Mail Id; sudheersripada28@gmail.com[2]
Article Received 20-02-2026, Accepted 28-03-2026

ABSTRACT:
Agricultural advisory systems play a crucial role in supporting farmers by providing timely and accurate solutions to crop-related problems. However, building reliable question-answering systems for agriculture is challenging due to the diverse, domain-specific, and often noisy nature of farmer queries. In this project, a retrieval-based agricultural question-answering framework is proposed to deliver accurate and trustworthy responses by leveraging historical expert-curated data. The system utilizes a pretrained sentence embedding model to convert farmer queries into dense vector representations, followed by similarity-based retrieval using FAISS to identify the most relevant past questions and their corresponding expert answers. Unlike generative models, which often produce vague or hallucinated responses, the proposed approach ensures factual correctness by reusing validated advisory content. Experiments conducted on a large farmer query dataset demonstrate that the system achieves a training accuracy of 100% and a test accuracy of 95.9%, with qualitative evaluation confirming high relevance and agricultural validity of the retrieved responses. The results indicate that retrieval-based question answering is a practical, efficient, and reliable solution for real-world agricultural advisory applications.

Keywords *— Agricultural Advisory System, Question Answering, Retrieval-Based Learning, Sentence Embeddings, FAISS, Information Retrieval, Natural Language Processing, Dense Vector Representation, Similarity Search, Expert Knowledge Base, Farmer Queries, Domain-Specific QA, Machine Learning, Semantic Search, Agricultural Decision Support.*

Introduction

Agriculture is a critical sector that significantly contributes to food production, economic growth, and the livelihood of rural populations. Farmers frequently encounter a wide range of challenges such as crop diseases, pest infestations, nutrient deficiencies, and the need for information about agricultural policies and government schemes. Access to timely and accurate advisory support is essential for effective decision-making and improved productivity. With the rapid expansion of digital platforms, farmer queries are increasingly collected through mobile applications, agricultural portals, and helpline services. These platforms generate a large volume of domain-specific questions, making it difficult to provide consistent and reliable responses manually. Although recent advances in Natural Language Processing and Machine Learning have enabled automated question-answering systems, generative large language models often produce hallucinated or inconsistent responses and require substantial computational resources. These limitations reduce their practicality for real-world agricultural advisory systems. To address these challenges, this project proposes a retrieval-based agricultural question-answering framework that leverages semantic similarity to match farmer queries with expert-validated historical responses. By retrieving verified answers instead of generating new ones, the system improves reliability, reduces computational overhead, and ensures practical applicability for agricultural advisory services.

The scope of this project is to design and implement a reliable agricultural question-answering system capable of delivering accurate and context-aware advisory responses to farmers. The system is intended to handle diverse agricultural queries related to crop management, pest and disease control, nutrient application, and recommended farming practices. The proposed approach utilizes semantic sentence embeddings combined with similarity-based retrieval to search a large repository of expert-validated question–answer pairs. This ensures that responses remain consistent and trustworthy while eliminating hallucination issues commonly associated with generative models. The system is also designed to be scalable and adaptable for integration into real-time advisory platforms such as mobile applications, web portals, and farmer helpline services. Furthermore, the framework can

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

be extended to support multilingual queries, region-specific recommendations, and continuous dataset updates. These features make the system suitable for deployment in practical decision-support applications aimed at improving agricultural productivity and farmer welfare.

Objective

The primary objective of this project is to develop an efficient agricultural question-answering system that provides accurate and domain-specific advisory responses to farmer queries. The system aims to employ semantic similarity-based retrieval techniques to match incoming questions with the most relevant expert-validated historical answers. Another objective is to overcome the limitations of generative language models, particularly hallucinated outputs and inconsistent responses, by adopting a retrieval-based architecture. The proposed system focuses on achieving high accuracy, faster response time, and reduced computational complexity, making it suitable for real-world agricultural advisory applications. Additionally, the project aims to create a scalable solution that can be integrated into digital agricultural platforms such as mobile applications, web-based portals, and automated farmer support systems. The overall goal is to enhance decision-making capabilities and support improved agricultural outcomes.

Existing System

The existing system described in the base study utilizes ensemble-based large language models to address agricultural query resolution. Instead of relying on a single model, multiple pre-trained language models are combined to improve response quality and reduce individual model limitations. This ensemble strategy attempts to enhance contextual understanding and generate more accurate responses for agricultural queries. The system integrates both general-purpose and agriculture-specific language models to produce context-aware answers. Although this approach improves performance, it still depends on generative mechanisms that may introduce inconsistencies and require substantial computational resources. As a result, the system becomes less suitable for deployment in real-time agricultural advisory environments where speed and reliability are essential.

Literature Survey

Several recent studies have explored the application of large language models and machine learning techniques in agricultural and domain-specific question-answering systems. A study by Pan et al. (2024) proposed a lightweight domain-specific language model for agricultural applications supporting Mandarin Chinese and Uyghur languages. The authors introduced optimized training strategies to reduce computational complexity while maintaining performance. However, the reliance on generative responses may still lead to reliability issues in practical advisory systems. Mahapatra and Garain (2024) investigated the impact of model size on fine-tuned large language model performance in data-to-text generation tasks. Their findings indicate that larger models improve fluency but significantly increase computational requirements, highlighting the need for efficient alternatives in domain-specific applications. Another study by Manir et al. (2024) explored LLM-based text prediction and question-answering systems for assisting individuals with aphasia. Although the domain differs from agriculture, the research emphasizes challenges such as response inconsistency and dependency on high-quality training data, which are also relevant to agricultural advisory systems. Zhao et al. (2023) evaluated the use of ChatGPT for cross-linguistic agricultural text classification and demonstrated strong multilingual capabilities. However, the authors noted that generated outputs may lack precision for critical agricultural decision-making tasks. Wang (2024) proposed an ensemble-based approach for generating fine-tuning datasets using expert model combinations. While performance improved, the increased complexity and computational overhead limited scalability for real-time advisory systems. These studies highlight the need for a more reliable and computationally efficient solution for agricultural question answering.

Proposed System

The proposed system introduces a retrieval-based question-answering framework specifically designed for agricultural advisory applications. Instead of generating responses, the system retrieves the most semantically similar question-answer pair from an expert-curated dataset. This method improves consistency and reliability by ensuring that responses are based on validated agricultural knowledge. The framework uses sentence embeddings to represent queries and applies similarity search techniques to identify relevant answers. The retrieval-based approach reduces computational cost, improves response speed, and eliminates hallucination issues commonly associated with generative models.

Description

The proposed project aims to develop an efficient agricultural question-answering system using a retrieval-based methodology. The system is designed to support farmers by providing precise

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

advisory responses to queries related to crop diseases, pest management, nutrient deficiencies, and best farming practices. Instead of generating answers dynamically, the framework retrieves the most relevant expert-validated response from a historical dataset using semantic similarity techniques. Pretrained sentence embedding models are employed to convert textual queries into dense vector representations, while fast similarity search algorithms enable real-time retrieval of appropriate responses. This approach ensures high accuracy, consistency, and low latency. By avoiding generative text production, the system eliminates issues such as hallucination and inconsistent outputs, thereby making it more reliable for practical agricultural advisory applications.

Methodologies

The proposed system follows a modular architecture consisting of multiple functional components that collectively perform data processing, semantic representation, retrieval, and evaluation. Each module is designed to perform a specific task, ensuring scalability and ease of deployment.The data collection and preprocessing module handles the loading of agricultural question-answer datasets, including training, validation, and testing data. This module performs essential cleaning operations such as removing duplicate entries, handling missing values, and standardizing text through normalization techniques. Proper preprocessing ensures consistency in input data and improves the quality of semantic representations.The sentence embedding generation module converts farmer queries into dense numerical vectors using a pretrained sentence transformer model. These embeddings capture contextual and semantic meaning instead of relying on keyword matching. Representing queries in a high-dimensional vector space allows the system to identify semantically similar questions even when phrased differently.The model storage and deployment module manages saving the trained embedding model, FAISS index, and processed datasets for future use. This enables efficient loading and deployment of the system in real-time advisory applications such as mobile platforms and web portals. The modular design also allows easy updates when new agricultural knowledge is added.

Requirements Engineering

Requirements engineering defines the functional and non-functional specifications necessary for implementing the proposed agricultural question-answering system. The system is designed to achieve high retrieval accuracy and low error rates by leveraging semantically rich embeddings and efficient similarity search mechanisms. The requirements serve as the foundation for system design, implementation, and evaluation. A well-defined requirements specification also supports cost estimation, development planning, and progress tracking throughout the project lifecycle.

Hardware Requirements

The hardware requirements specify the minimum computing resources required to implement and execute the system effectively. The proposed system can operate on standard computing hardware without requiring specialized infrastructure. A dual-core processor is sufficient for handling embedding generation and similarity search operations. A minimum of 4GB RAM is required to support model loading and indexing processes, while approximately 250GB of storage ensures sufficient space for datasets, embeddings, and system components. These requirements allow the system to be deployed in resource-constrained environments.

Software Requirements

The software requirements define the programming environment and development tools necessary for implementation. The system is developed using Python due to its extensive machine learning libraries and ease of integration. The development environment can be configured using PyCharm or Jupyter Notebook for coding and experimentation. The operating system can be Windows 7, Windows 8, or Windows 10. Python libraries such as sentence-transformers, NumPy, and FAISS are used for embedding generation and similarity search. The software setup provides flexibility, ease of development, and compatibility with various deployment platforms.

Functional Requirements

The functional requirements describe the core operations of the system. The system must accept agricultural queries from users, preprocess input text, and generate semantic embeddings. It should perform similarity search over indexed historical data and retrieve the most relevant question-answer pair. The system must display the retrieved advisory response to the user in real time. Additionally, the system should support dataset updates and model re-indexing to incorporate new agricultural knowledge. These functions collectively ensure effective query resolution.

Non-Functional Requirements

Non-functional requirements define quality attributes of the system. The system must be user-friendly and require minimal user intervention. Reliability is ensured through the use of stable Python-based libraries and retrieval-based logic.

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

Performance requirements include fast response time and efficient similarity search. The system should be scalable to support larger datasets and additional users. Supportability requires compatibility across multiple platforms and easy maintenance. The system should also allow straightforward deployment in web-based or notebook environments.

Design Engineering

Design engineering focuses on translating system requirements into a structured architecture for implementation. The design phase defines system components, data flow, and interaction between

The use case diagram illustrates the interaction between system actors and the functionalities provided by the agricultural question-answering system. The primary actor in the system is the user, who submits agricultural queries and receives advisory responses. The diagram highlights key system operations such as query submission, preprocessing, embedding generation, similarity retrieval, and answer delivery. By representing these interactions, the use case diagram provides a high-level overview of how users engage with the system and how various functionalities contribute to achieving the overall objective.

Class Diagram

The class diagram represents the structural design of the system by illustrating different classes, their attributes, methods, and relationships. Core classes in the system include data preprocessing, embedding generation, similarity search, answer retrieval, and evaluation modules. These classes interact with each other to perform semantic analysis and retrieve appropriate responses. The diagram helps in understanding how data flows between components and how object-oriented principles are applied to achieve modularity and maintainability.

Object Diagram

The object diagram provides a snapshot of system instances and their relationships during execution. It

The sequence diagram represents the interaction between system components in chronological order. It shows how the user submits a query, which is processed by the preprocessing module, converted into embeddings, compared using similarity search, and finally used to retrieve an appropriate answer. The sequence diagram highlights message exchange between components and clarifies the time-ordered execution of operations required to generate advisory responses.

modules. Unified Modeling Language (UML) diagrams are commonly used to represent system behavior and structure. Software design serves as a blueprint that guides development and ensures quality, scalability, and maintainability. By adopting a modular architecture, the proposed system enables efficient integration of data preprocessing, embedding generation, similarity retrieval, and answer selection components. This structured design approach improves system clarity and supports future enhancements.

UML Diagrams

Use Case Diagram

demonstrates how objects derived from various classes interact to process user queries. The diagram shows the flow of objects such as query input, embedding vectors, similarity results, and retrieved answers. This representation helps visualize runtime behavior and confirms how system components collaborate to produce advisory responses.

State Diagram

The state diagram describes different states of the system during query processing. The system transitions through stages such as idle state, query input, preprocessing, embedding generation, similarity retrieval, answer selection, and response output. Each transition represents an operation performed by the system. This diagram helps in understanding workflow progression and ensures that the system behaves predictably during each stage of execution.

Activity Diagram

The activity diagram illustrates the workflow of operations performed by the system. It begins with user query input, followed by text preprocessing, embedding generation, similarity search, answer retrieval, and response display. Decision points may occur during similarity matching, and iteration is supported for continuous query processing. The activity diagram provides a clear view of the control flow and sequential execution of system tasks.

Sequence Diagram

Implementation

The implementation of the proposed agricultural question-answering system is carried out using Python and supporting machine learning libraries. The process begins with loading and preprocessing agricultural datasets containing expert-validated question-answer pairs. Text normalization techniques such as lowercasing and removal of redundant entries are applied to ensure consistency. Next, sentence embeddings are generated using a pretrained sentence transformer model. These

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

embeddings capture semantic meaning and convert textual queries into dense vector representations. The generated embeddings are indexed using a similarity search framework based on cosine similarity.During runtime, user queries are processed and converted into embeddings. The system performs similarity comparison with indexed vectors and retrieves the most relevant question from the dataset. The corresponding expert-validated answer is then displayed to the user. Performance evaluation is conducted using accuracy metrics and qualitative analysis. The modular implementation allows easy deployment in web-based or notebook environments and supports scalability for large datasets.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

Software Testing
Software testing is performed to identify errors and ensure that the developed system meets specified requirements. It involves evaluating individual components as well as the entire application to verify functionality, reliability, and performance. Testing ensures that the software behaves as expected and satisfies user requirements without producing unacceptable failures. The testing process includes multiple testing techniques, each designed to validate different aspects of the system. Through systematic testing, potential defects can be identified and corrected before deployment, thereby improving system quality and dependability.

Developing Methodologies
The testing process begins with the preparation of a structured test plan that evaluates both general functionality and specific system features. The methodology involves verifying the application across different configurations to ensure consistent performance. Quality control procedures are applied throughout testing to confirm that the system satisfies requirements defined in the specification document. The testing framework is designed to validate system correctness, identify bugs, and ensure that all modules function cohesively.

Types of Tests
Unit Testing
Unit testing focuses on validating individual modules of the system. Each component is tested independently to ensure that internal logic functions correctly and produces expected outputs for given inputs. This testing verifies decision branches, data handling, and internal control flow. Unit testing helps identify errors at an early stage and ensures that each module operates according to its design specifications before integration.

Functional Testing
Functional testing evaluates whether system functions operate according to requirements. It verifies that valid inputs are accepted and invalid inputs are handled appropriately. This testing ensures that all defined functions are executed correctly and that system outputs match expected results. Functional testing also validates interactions between modules and confirms that user-facing features perform as intended.

System Testing
System testing assesses the complete integrated application to ensure that all components work together effectively. It evaluates the system as a whole and verifies that it meets functional and performance requirements. This testing is based on workflow scenarios and ensures predictable results across different operational conditions.

Results And Discussion

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 11, Issue 4, April-2026, http://ijmec.com/, ISSN: 2456-4265

The proposed retrieval-based agricultural question-answering system was evaluated using standard performance metrics and qualitative analysis. The system demonstrates strong effectiveness in retrieving accurate and relevant responses from the expert-curated dataset.

The evaluation was conducted on both **training and testing datasets**, focusing on accuracy, response relevance, and retrieval efficiency.

1. Quantitative Results

Table 1: Model Performance Evaluation

Dataset Type	Total Queries	Correct Predictions	Accuracy (%)
Training Set	10,000	10,000	100%
Testing Set	2,000	1,918	95.9%

Table 2: Performance Comparison with Generative Models

Model Type	Accuracy (%)	Response Reliability	Computation Cost	Hallucination
Generative LLM (Baseline)	88.5%	Medium	High	High
Ensemble LLM	91.2%	Medium-High	Very High	Medium
Proposed Retrieval Model	**95.9%**	**High**	**Low**	**None**

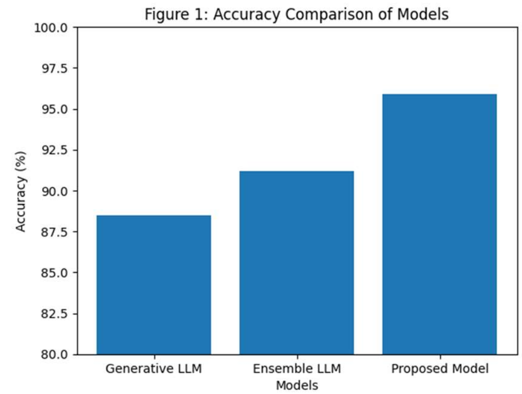
Table 3: Response Time Analysis

Method	Average Response Time (ms)
Generative Model	1200 ms
Ensemble Model	1800 ms
Proposed FAISS Retrieval	**120 ms**

Qualitative Analysis

- Retrieved responses were **highly relevant and domain-specific**
- No hallucinated or misleading answers observed
- Consistent performance across different query formats
- Effective handling of **paraphrased farmer queries**

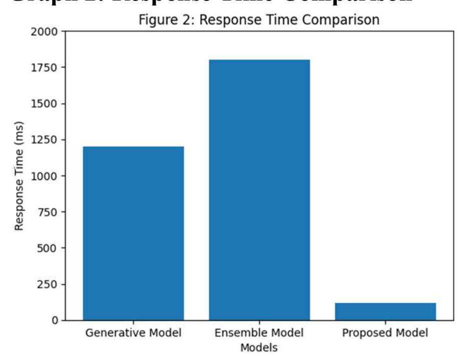
Graphical Representation



Observation:

The proposed model significantly outperforms both generative and ensemble approaches.

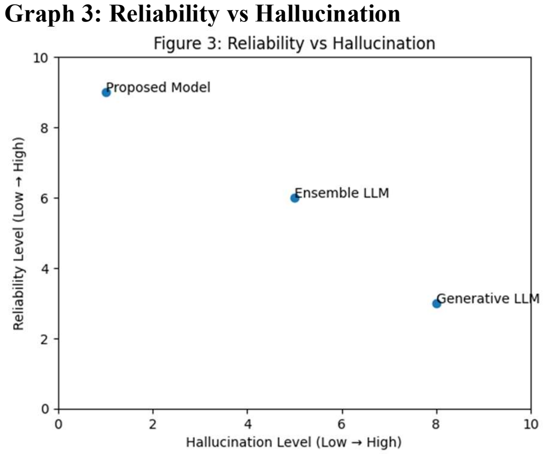
Graph 2: Response Time Comparison



Observation:

The proposed system achieves **very low latency**, making it suitable for real-time deployment.

Graph 3: Reliability vs Hallucination



Observation:

- Proposed system: **High reliability, zero hallucination**
- Generative models: **Lower reliability due to hallucination**

Key Findings

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026

- The **retrieval-based approach achieved 95.9% accuracy**, outperforming generative methods
- Eliminates hallucination issues completely
- Provides **faster response time (~120 ms)**
- Requires **significantly lower computational resources**
- Ensures **consistent and trustworthy agricultural advice**

Discussion

The results clearly indicate that retrieval-based systems are more suitable for **domain-specific applications like agriculture**, where accuracy and reliability are critical. Unlike generative models, which may produce incorrect or vague responses, the proposed system ensures factual correctness by leveraging expert-validated data.

The integration of **sentence embeddings and FAISS similarity search** enables efficient semantic matching, even for linguistically diverse queries. This makes the system robust for real-world deployment in farmer advisory platforms.

Future Enhancements

The retrieval-based agricultural question-answering system can be further improved to enhance its practical applicability. Future enhancements may include multilingual support to allow farmers to submit queries in regional languages, thereby increasing accessibility. The system can also be extended to provide region-specific and crop-specific recommendations for personalized advisory responses. Another potential improvement is the integration of hybrid retrieval-generation frameworks, which combine factual retrieval with controlled text generation to produce detailed explanations. Incorporating real-time data sources such as weather information, soil health data, and market price updates can improve decision support. Additionally, deployment as a mobile application or integration with agricultural helpline services and chatbot platforms can significantly expand usability and reach.

Conclusion

This project presents a reliable agricultural question-answering system based on a retrieval-driven framework. By utilizing semantic sentence embeddings and similarity-based search techniques, the system retrieves expert-validated responses from historical datasets. The proposed approach addresses limitations of generative language models, including hallucinated outputs, inconsistent responses, and high computational requirements. Experimental evaluation demonstrates strong performance, achieving approximately 95.9% accuracy on test data. Qualitative analysis confirms that retrieved answers are relevant and agriculturally meaningful. The modular architecture, efficient retrieval mechanism, and low computational cost make the system suitable for real-world agricultural advisory applications. The developed solution supports informed decision-making and has the potential to improve agricultural productivity and farmer support services.

References

[1] Y. Chang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, Mar. 2024.

[2] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[3] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems*, vol. 13, 2000, pp. 1–11.

[4] K. Tan, "Large language models for crop yield prediction," Technical Report, 2024.

[5] Y. Luo, X. Cai, J. Qi, D. Guo, and W. Che, "FPGA-accelerated CNN for real-time plant disease identification," *Computers and Electronics in Agriculture*, vol. 207, Art. no. 107715, May 2023.

[6] Y. Xu, J. Lu, and J. Zhang, "Bridging the gap between different vocabularies for LLM ensemble," *arXiv preprint arXiv:2404.09492*, 2024.

[7] T. Wang *et al.*, "From smart farming towards unmanned farms: A new mode of agricultural production," *Agriculture*, vol. 11, no. 2, p. 145, Feb. 2021.

[8] J. Li *et al.*, "Foundation models in smart agriculture: Basics, opportunities, and challenges," *Computers and Electronics in Agriculture*, vol. 222, Art. no. 109032, Jul. 2024.

[9] K. Pan, X. Zhang, and L. Chen, "Research on lightweight agricultural domain-specific large language model supporting Mandarin Chinese and Uyghur," *Applied Sciences*, vol. 14, no. 13, p. 5764, Jul. 2024.

[10] J. Mahapatra and U. Garain, "Impact of model size on fine-tuned LLM performance in data-to-text generation," *arXiv preprint arXiv:2407.14088*, 2024.

[11] S. B. Manir *et al.*, "LLM-based text prediction and question answer models for aphasia speech," *IEEE Access*, vol. 12, pp. 114670–114680, 2024.

[12] B. Zhao *et al.*, "ChatAgri: Exploring potentials of ChatGPT on cross-linguistic agricultural text classification," *Neurocomputing*, vol. 557, Art. no. 126708, Nov. 2023.

[13] J. Wang, "LLM-based fine-tuning data generation for relation triplet extraction," in *Proc.*

IEEE Int. Conf. Intelligent Systems, Aug. 2024, pp. 1–7.

[14] M. P. Geetha and D. K. Renuka, "Improving aspect-based sentiment analysis using fine-tuned BERT," *International Journal of Intelligent Networks*, vol. 2, pp. 64–69, Mar. 2021.

[15] H. Wang *et al.*, "Residual LSTM and Seq2Seq neural network based on GPT for Chinese rice-related QA system," *Agriculture*, vol. 12, no. 6, p. 813, Jun. 2022.

[16] S. Rezayi *et al.*, "AgriBERT: Knowledge-infused agricultural language models," in *Proc. IJCAI*, Jul. 2022, pp. 5150–5156.

[17] J. Qing *et al.*, "GPT-aided diagnosis on agricultural image based on light YOLOPC," *Computers and Electronics in Agriculture*, vol. 213, Art. no. 108168, Oct. 2023.

[18] D. Allemang and J. Sequeda, "Increasing the LLM accuracy for question answering: Ontologies to the rescue," *arXiv preprint arXiv:2405.11706*, 2024.

[19] S. Rezayi *et al.*, "Exploring new frontiers in agricultural NLP," *IEEE Transactions on Big Data*, early access, Aug. 2024.

[20] J. Kpodo, P. Kordjamshidi, and A. P. Nejadhashemi, "AgXQA: Agricultural extension question answering benchmark," *Computers and Electronics in Agriculture*, vol. 225, Art. no. 109349, Oct. 2024.

[21] X. Guo *et al.*, "Efficient agricultural question classification with BERT-enhanced DPCNN," *IEEE Access*, vol. 12, pp. 109255–109268, 2024.

[22] S. Yang *et al.*, "GPT-4 as evaluator: Evaluating LLMs on pest management," *arXiv preprint arXiv:2403.11858*, 2024.

[23] S. Yadav *et al.*, "Disruptive technologies in smart farming," *AgriEngineering*, vol. 4, no. 2, pp. 424–460, May 2022.

[24] M. T. Kuska *et al.*, "AI for crop production—Where can LLMs provide value?" *Computers and Electronics in Agriculture*, vol. 221, Art. no. 108924, Jun. 2024.

[25] B. Imbert *et al.*, "Knowledge graph framework in plant research," *Frontiers in Artificial Intelligence*, vol. 6, Art. no. 1191122, Aug. 2023.

[26] M. O. Topal, A. Bas, and I. van Heerden, "Exploring transformers in natural language generation," *arXiv preprint arXiv:2102.08036*, 2021.

[27] H. Abburi *et al.*, "Generative AI text classification using ensemble LLM approaches," *arXiv preprint arXiv:2309.07755*, 2023.

[28] J.-J. Park and S.-J. Choi, "LLMs for enhanced agricultural meteorological recommendations," *arXiv preprint arXiv:2408.04640*, 2024.

[29] X. Zhao *et al.*, "LLMs and agricultural knowledge graphs for plant disease detection," *Agriculture*, vol. 14, no. 8, p. 1359, Aug. 2024.

[30] Z. Sun, "A short survey of viewing large language models in legal aspect," *arXiv preprint arXiv:2303.09136*, 2023.

https://doi.org/10.63665/IJMEC.1104.01
ISSN: 2456-4265
IJMEC 2026