AI-based Dynamic Quiz System: Disaster Knowledge King

Chen-Hao Tseng[1] and James Yichu Chen[1]

[1] National Taiwan University of Science and Technology, Taiwan
chtseng@9999gmail.com, yie@mail.ntust.edu.tw

Abstract

Amidst escalating global climatic threats, effectively disseminating disaster preparedness knowledge to the public remains a critical challenge. This study introduces Disaster Knowledge King (DKK), an automated, dynamic quiz system designed to modernize disaster risk reduction (DRR) education. While traditional educational games suffer from static, "dead" databases that require manual updates, DKK leverages Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) to facilitate real-time content creation.

The system features a fully automated pipeline that scrapes global news from sources like AP and Reuters, filters for recency, and utilizes the Gemini API to generate multiple-choice questions. By grounding responses in professional Ready.gov guidelines via RAG, DKK provides users with both situational awareness and actionable response strategies, and system is implemented as a LINE Bot to ensure high accessibility. While the DKK reduces the production cost to approximately $0.002 per question set, its G-Eval and UAT scores significantly exceed the median threshold. This study indicates that integrating AI-driven automation with gamified social media interfaces offers a scalable and efficient pathway for enhancing public disaster literacy.

Keywords

LLMs, RAG, automation, DRR, disaster literacy, education, LINE Bot, multiple choice questions.

1 Introduction

1.1 Background

For decades, humanity has confronted the escalating threat of anthropogenic global warming, characterized by an increase in extreme climatic events and natural disasters. While disaster management experts over the world have developed sophisticated mitigation strategies and response frameworks, a persistent challenge remains: the effective spread of “disaster preparedness knowledge” and “environmental literacy” to the general public.

The 2022 debut of OpenAI’s ChatGPT precipitated a revolutionary transformation in both professional productivity and daily life, signalling a new era of AI-augmented workflows. Consequently, the application of Large Language Models (LLMs) has expanded across diverse sectors, fostering unprecedented levels of innovation and generation. Within this context, experts have begun leveraging these models to enhance disaster education through the pedagogical adaptation of complex materials, the design of immersive practice scenarios, and the high-level analysis of learner performance.

1.2 Motivation

The classic mobile game “Knowledge King” is a landmark educational title that has achieved widespread popularity in Taiwan. The game utilizes a vast database of multiple-choice questions across diverse fields, challenging users’ knowledge in a competitive "contest mode." By presenting information in a thrilling and engaging format, Knowledge King has successfully demonstrated how gamification can enhance general knowledge. This study argues that “an educational tool modeled after Knowledge King is an ideal method for disseminating “disaster preparedness knowledge” and “environmental literacy” to the general public.”

However, this study identifies a critical limitation within the current Knowledge King model: the static nature of its database. Because the question pool is "dead," it requires labor-intensive and time-consuming manual updates to renew. This prevents the content from real-time information, so it’s a drawback that must be addressed.

1.3 Objective

This study aims to develop a dynamic quiz system titled "Disaster Knowledge King" (DKK). Designed as an advanced iteration of the Knowledge King framework, DKK focuses specifically on disaster prevention education. The system provides multiple-choice questions based on real-time disaster information, accompanied by comprehensive explanations and recommended response strategies. The most significant innovation of DKK is its ability to maintain a contemporary database; unlike traditional static systems, DKK utilizes Large Language Models to automate the generation of questions. By leveraging the successful engagement strategies of its predecessor, DKK serves as a highly efficient educational tool for spreading “disaster preparedness knowledge” and “environmental literacy”.

2 Literature Review

As extreme weather and associated disasters are the biggest contemporary global concern, “disaster risk reduction (DRR)” is imperative that many papers have valued its integration into education and explored the better methods to practice it. Research argues that DRR in education is a psycho-social concept that the successful integration of DRR requires a transition in instructional design toward localized and participatory approaches, emphasizing individual specialized action and active engagement within local communities and environments[1]. Nahid Aghaei [2] et al. also focus on the DRR education strategy for raising knowledge, and the critical point is to transfer accessible information with high quality, especially through the community involvement to share knowledge.

However, the main challenge remains the deficiency in conceptual understanding regarding DRR, directly contributing to resource scarcity and the stagnation of social development in this area. As it is difficult to enhance public disaster literacy through traditional methods, utilizing social networks or media offers a viable and potential channel[1, 2].

Because of resource scarcity, utilizing Artificial Intelligence to assist material production is a promising method. Biancini et al.[3] have introduced a methodology for generating multiple choices question by LLMs. By providing the context and specific prompts, GPT-3.5 could be proficient in generating MCQs, which is evaluated through the non-parametric Friedman test. This study illustrates the viability of LLMs to produce MCQs and provides valuable insights for the future.

Additional research either demonstrates the capacity of LLMs to generate complex questions—specifically those involving analysis and synthesis—which highlights this model’s suitability for educational curricula. GPT-4 has exhibited the greatest versatility in constructing tasks of different formats and at multiple cognitive levels in alignment with Bloom's Taxonomy.[4] Concurrently, Shahab Sheikhalishahi et al. [5] compared the performance of ChatGPT-4o, ChatGPT-5, and Gemini 2.5 Flash on Persian internal medicine subspecialty board questions; the results reveal the superior performance of Gemini 2.5 in text-based clinical reasoning. Among existing LLMs, Gemini appears to possess the most robust textual understanding and Inference abilities.

Based on the concept above, content of DDK education should customize for diverse communities and environments, so the response strategies would be highly varied that LLMs can leverage Retrieval-Augmented Generation (RAG) to produce more ideal and accurate responses. Yunfan Gao et al.[6] thoroughly introduce RAG which effectively integrates parametric knowledge of LLMs with non-parametric external data sources, thereby mitigating hallucinations and inaccuracies; this establishes it as one of the essential frameworks for deploying large language models. The fundamental components of RAG are ”Indexing”, “Retrieve” and “Generation”, while indexing further consisted of data indexing, chunking and embedding. As a method combining retrieval and generation, RAG can offers multiple avenues for future research.

While early retrieval models rely on sparse representation (e.g., BM25 or TF-IDF), they frequently struggle with the vocabulary mismatch problem and complex semantic queries. Alternatively, by using pre-trained transformers to generate dense vector embeddings, these models can identify deeper conceptual connections between queries and documents. Consequently, dense retrieval offers a more robust framework for capturing context and improving overall retrieval precision[7].

The application of AI has emerged as a significant trend in disaster prevention, particularly within DRR education. There have been several examples, for instance, Ömer Cem Karacaoğlu et al.[8] explore AI-driven strategies in disaster education to determine their feasibility in fostering innovative and proactive preparedness. The research reveals that “Text classification guides students by identifying the topics covered in the educational material and helps focused learning. Question-answer applications enhance the learning experience by automatically extracting answers from texts.”

In addition to AI, DRR education can be integrated with others technology, such as chatbot, for the purpose of innovative pedagogies to enhance learning efficacy. Okan Yetişensoy [9] explores the effectiveness of AI-assisted chatbots in disaster education by comparing chatbot-integrated instruction against conventional pedagogical methods. The result reveals a significant positive impact on cognitive learning outcomes, with the experimental group demonstrating superior knowledge retention compared to the control group. Furthermore, semi-structured interviews suggested that learners favored the chatbot approach, citing its student-centered design and engaging pedagogical features as key drivers for more effective education. The research also suggests that “technology combining XR and chatbots would be helpful in enabling students to experience disaster situations firsthand and enhance the skills necessary for dealing with disasters.”

Squire’s research suggests that the immersive nature of computer and video games provides a comprehensive framework for learning that surpasses traditional pedagogies in global reach and effectiveness[10]. Fletcher's research highlights the superiority of interactive tools, noting that students retain 75% of information through simultaneous seeing, hearing, and interacting, compared to just 40% of visual data and 20% via auditory instruction[11]. Therefore, gamification might be a suitable pedagogy for DRR education.

3 Methodology

This study constructs the Disaster Knowledge King (DKK), which is organized into three subordinate sections: the Generation System, the Database, and the Answer System, with several detailed modules under each section. The specific system architecture is illustrated in Figure 1. Specifically, this study develops a variety of functions into corresponding modules and then integrates associated modules into a primary section. Consequently, each section executes its respective tasks independently, allowing the DKK to stream results from all sections to form a complete, dynamic quiz system.

In the Generation System, the server executes a procedure at each interval to generate a series of intact question sets. First, the News Module collects real-time news sources concerning disaster events occurring globally. Second, the Filter Module ensures the news is fresh and removes expired question sets from the database. Third, the AI Manufacture Module utilizes a Large Language Model (LLM) to generate the question set, subsequently employing the RAG method with specific disaster response documentation to recommend response strategies. Lastly, the Packaging Module organizes the LLM output into a specific format and uploads it to the Database..

In the Database, this section consists of the "DB Calling Module" and the "Cloud DB". Its primary function is the storage of all question sets; it also serves as the connection between the Generative and Answer systems while transferring the DKK's essential properties: disaster preparedness knowledge and environmental literacy.

In the Answer System, this study adopts the LINE Bot as the operational medium for the DKK, substituting traditional application software with a near-ubiquitous social media interface. Upon user initiation, the Question Module retrieves and renders a question from the database. Subsequently, the Options Module presents four processed alternatives to the user. Following the user's input, the Answer Module displays the correct answer, an explanation, and the response method, using specific colour to indicate whether the user's response was correct.

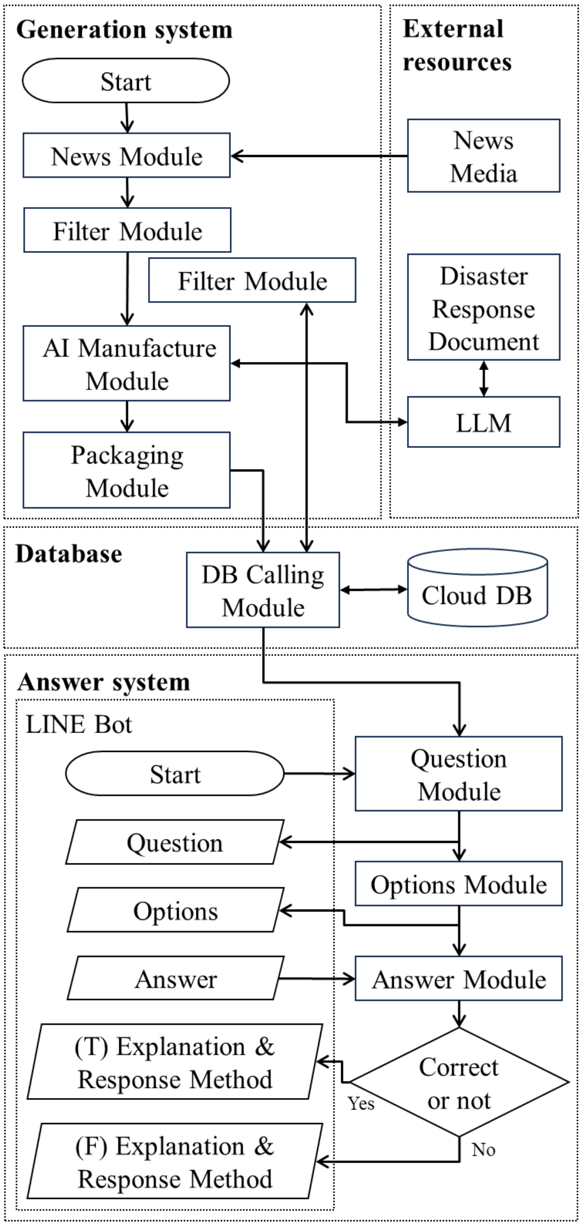

Figure 1 Disaster Knowledge King Disaster Knowledge system architecture diagram

4 Implementation

Diverse APIs and LLMs are integrated into DKK's framework, with key research outcomes focusing on prompt engineering, production pipeline design, and dynamic disaster prevention methodologies, and the whole system development is based on Python.

4.1 Generation system

The Generation system serves as the fundamental driver of overall operations. The system is comprised of four distinct modules to establish a production pipeline: the News Integration Module, the Filter Module, the AI Manufacture Module and the Packaging Module.

4.1.1 News Module

To acquire global disaster information, the system integrates three primary news sources—the Associated Press, Reuters, and Channel News Asia—selected for their credibility and extensive regional coverage. The news integration workflow proceeds in three stages: first, defining extraction parameters (keywords, media outlets, and volume); second, utilizing the NewsAPI.org service to retrieve headlines, publication times, and URLs; and third, employing the BeautifulSoup library to scrape full web content from those URLs.

Once ingested, the News Module categorizes the data into specific disaster types, including “avalanche, earthquake, fire, explosion, extreme heat, drought, flood, hazardous materials, hurricane, typhoon, debris flow, pandemic, severe weather, thunderstorms, lightning strike, tornado, tsunami, volcano, wildfire, and snowstorm.”, and standardizes publication dates into a unified format (YYYY-MM-DD HH:MM:SS) for database consistency.

Finally, the module encapsulates this data into a structured schema—comprising news content, date, disaster type, and URL—before transmitting it to the Filter Module for processing.

4.1.2 Filter Module

The core characteristic of the DKK’s questions is their "dynamism," signifying that all information must maintain a high degree of real-time relevance and freshness. Consequently, the system must strictly manage the temporal validity of both newly imported news data and existing items within the database.

In the first step, the Filter Module receives data packets from the News Module. It compares the "publication date" of each entry against a pre-defined "update threshold" set in the system parameters, thereby filtering out data that exceeds the permissible latency.

In the second step, the system executes a stale data purging operation. Utilizing a data deletion function, the Filter Module applies a specific retention rule: "any data record with a publication date older than N days (N being the retention period defined by system parameters) shall be removed." If the deletion of records results in a specific disaster category containing zero items, the module invokes the table dropping function to remove the empty category table. This process ensures the structural integrity and operational stability of the Generation system.

In the final step, the Filter Module encapsulates all qualified data into a standardized format—"news content, publication date, disaster type, and news URL—and transmits it to the AI Manufacture Generation Module to serve as the foundational material for subsequent question generation.

4.1.3 AI Manufacture Module

The core of the DKK generation system is the AI Manufacture Module, which integrates the Gemini API, Retrieval-Augmented Generation (RAG), and structured data schemas to produce "news-dissemination-oriented" educational content. Gemini was selected as the primary LLM for its superior semantic parsing and reasoning capabilities. To ensure structural reliability, the module utilizes pydantic and google-genai libraries to enforce a predefined schema, guaranteeing that every output includes a high-quality question, four options, and a detailed explanation.

Prompt engineering serves as the study’s central research focus, guided by two principles: "multi-layered iterative development" and "contextual supplementation." The former breaks down complex tasks into high-success sub-tasks through iteration, while the latter applies directional constraints to ensure strict adherence to the desired content. Together, these principles enable the system to achieve superior and stable generation quality.

A distinctive feature of the DKK is the "response method" included in every explanation to meet Disaster Risk Reduction (DRR) goals. To ensure these methods are accurate, the research employs RAG by grounding the Gemini API in professional Ready.gov guidelines. This workflow involves converting guidelines to Markdown, indexing them via the google-genai SDK, and generating grounded responses. Ultimately, prompt engineering remains most vital, as it dictates the overall educational effectiveness of the module.

The following are the prompts used for each of the five production steps in AI Manufacture Module:

prompt1: f'You are now a global disaster intelligence expert. Please organize the article from link {news[“news_URL”]} into four key points, focusing on the damages caused.'

prompt2: 'Based on these four key points, create a multiple-choice question with four options(A, B, C, D), designed so that people can use reasoning to arrive at the answer and understand what happened. Please also provide the correct answer.'

prompt3: 'Then provide an explanation for the answer in under 70 words, and ensure that the text does not contain the words \'option\' or \'A, B, C, D\'." '

prompt4 : Next, based on the knowledge and content in "fileSearchStores", please tell everyone how to respond if they are personally involved in the case of this article, within 70 words.

prompt5: ‘Based on the content above, return the “Question”, “Option_A”, “Option_B”, “Option_C”, “Option_D”, “Correct_Answer”, “Explanation”, and “Response_Method”. Do not change anything, especially the question and the options.’

4.1.4 Packaging Module

The Packaging Module processes source response from Gemini, which is received as a Python dictionary, by assigning a unique item ID, labeling options, and merging the "Explanation" and "Response Method" into a single string separated by a "&&&" delimiter. This concatenation strategy ensures the dataset can fit into the pre-defined database schema while allowing both data points to be stored in one block for future scalability.

To generate the unique ID, the module queries the database to identify existing entries and randomly selects a new value from a range of 1 to k, where k is an adjustable parameter tailored to table capacity. The finalized data is encapsulated into a standardized "questions dataset" having a schema: (‘ID’, ‘Question’, ‘Option A’, ‘Option B’, ‘Option C’, ‘Option D’, ‘Correct Answer’, ‘Explanation & Response Method’, ‘Publication Date’, ‘News Title’, ‘News_URL’).

Finally, the DB Calling Module uploads this dataset to the target database via a data-appending function to complete the run.

4.2 Answer system

This study adopts a chatbot as the primary vehicle for the answering system, with LINE selected as the implementation platform. Given its high penetration rate and convenience, the LINE platform allows users to access and engage with educational resources through a low technical barrier and at zero cost.

4.2.1 LINE Bot

The LINE Bot facilitates user interaction through message templates and Rich Menus. The LINE Bot’s internal logic serves as the central controller for the entire answering system, as Figure 2 shows. Its development involved chatbot account creation, Vercel project establishment, API configuration, and UI design.

At the core of the system is the "Central Processor," a module built on the official LINE SDK. This processor governs the DKK-robot’s operational modes and integrates database workflows to provide real-time question generation and feedback. The Central Processor operates via a five-step workflow: “initializing the Flask framework”, “connecting to the LINE Messaging API”, “monitoring webhook data”, “parsing incoming information” and “executing module results”.

Depending on the parsed input, the Central Processor triggers one of three functional modules—Question, Options, or Answer—each performing data operations via a DB Calling Module connected to a cloud database. To ensure a professional user experience, the interface was crafted using the LINE Bot Designer and Flex Message Simulator. The resulting JSON layouts are embedded directly into the bot's logic, providing users with an intuitive and clear educational interface.

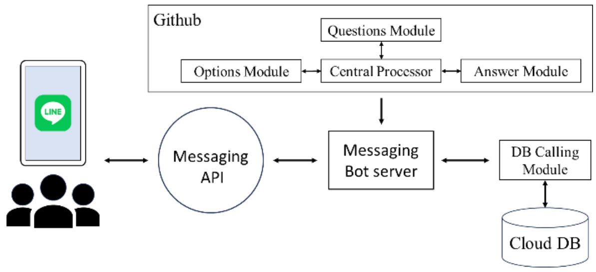

Figure 2 architecture framework of LINE Bot

4.2.2 Question Module

When receiving a response ‘Disaster’, the Central Processor invokes the Question Module to initiate the question generation process. The Question Module randomly fetches a record from the cloud database and deconstructs it into an ‘information packet,’ which is then returned to the Central Processor for subsequent process. Throughout its operation, the Question Module interacts with the database via the DB Calling module, utilizing several functions for connection configuration, table deletion, data deletion and data fetching.

The operational workflow of the Question Module designed in this study is as follows:

1. Data availability verification: The module first confirms the existence of question datasets within the database. If nothing is found, an error notification is transmitted to the Central Processor.

2. Random selection: If question dataset is available, the module retrieves all item IDs in database and utilizes the Python random library to select a single ID randomly.

3. Data extraction: Based on the selected ID, the module fetches the complete question dataset, including ID, Question, Option A, Option B, Option C, Option D, Correct Answer, Explanation & Response Method, Publication Date, News Title, News_URL.

4 .Message formatting: The module generates a text message formatted as ‘disaster-[ID]: Question’.

5. Information packet encapsulation: The text message, question data are integrated into an information packet, which is then returned to the Central Processor.

Finally, once receiving this packet, the Central Processor triggers the Options Module, passing the information packet as an execution parameter. Notably, test message will be display later along with options.

4.2.3 Options Module

Immediately following the Item Module, the Options Module is activated to process the ‘option content’ within the item dataset. This module features two critical design priorities: First, to mitigate potential spatial bias in LLM-generated outputs, an Option Shuffling Mechanism re-randomizes the choice sequence. Second, since the LINE Bot interface is inherently stateless and the complete interaction workflow (Category Selection → Item Generation → Answering → Feedback) requires three distinct exchanges, a cloud-based session memory mechanism was developed. This ensures that the context of the user's answering process is accurately tracked and retrieved. The operational workflow is as follows:

1. Extraction and shuffling: The module retrieves the four original options and applies a randomization algorithm to re-map the correct answer to a new position.

2. Session Management: To support concurrent multi-user access, a unique 8-digit Temporary ID is dynamically generated (e.g., 10125002), encoding the session status, category, and item number.

3. Data Persistence: A temporary dataset, including the shuffled options, metadata, and a start-time timestamp, is uploaded to the database's 'Memory' table.

4. UI Generation: The module constructs a Carousel Flex Message where each card serves as an interactive option containing the description and a postback trigger.

The Temporary ID serves as a primary key within the database's Memory table to uniquely identify each transient answering record. The 8-digit ID follows a specific encoding rule: 10,000,000 + (Category Code * 100,000) + (Item Number * 100) + Session ID.

In the end, Central Processor will display the question and options after receiving the Carousel Flex Message.

4.2.4 Answer Module

When receiving the user's response (containing a newline delimiter \n), the Central Processor invokes the Answer Module. The module retrieves the corresponding session record from the database's Memory table to compare the user's input with the correct answer. Given the high information density of the feedback, Flex Messages are adopted to provide a structured, customizable, and visually clear template for the user.

The operational workflow of the Answer Module is as follows:

1. Message parsing: The incoming string is deconstructed by using the \n and - delimiters to extract the item category, the Temporary ID, and the user's selected option.

2. Temporal validation: The module utilizes the datetime library to compare the current system time against the session timestamp. If the response exceeds the predefined time limit, an error message is returned.

3. Memory purging: To maintain database efficiency and minimize storage overhead, the module invokes a deletion function to remove the temporary record from the 'Memory' table once it is processed.

4. Conditional formatting: The system performs a logic check between the user's answer and the correct entry. The Flex Message adopts a green color scheme for correct answers and a red scheme for incorrect ones.

5. Flex message construction: The feedback's Flex Message is assembled into three distinct layers: upper block displaying the correct answer; middle block containing the explanation explanation; and lower block showing the publication date and the source URL.

6. Response method: After separating the response method from Explanation_Plue_Response_Method, the module displays it in another Flex Message template, which synchronize the pattern with feedback's template.

7. Final Dispatch: The finalized Flex Message of feedback and response method are returned to the Central Processor and transmitted to the user.

This structured interface allows users to quickly grasp the correct knowledge and background information, with the option to access the original news manuscript via the embedded hyperlink for deeper learning.

4.3 Database

To support the high-frequency and real-time data access requirements of the DKK, this study establishes a database. The database also serves as the connection for information transition between the independently operated Generation System and the Answering System. To ensure constant availability, the database is deployed on the Google Cloud.

4.3.1 Cloud DB (google sheet)

To mitigate the high operational costs of maintaining a MySQL instance on GCP, this study constructs an alternative storing framework by leveraging Google sheet, which is free under 15 GB. The database consists of only one single table 'disaster,' which contains 200 data slots (a capacity set by the DKK's Parameter Control Panel). Each data slot is in a uniform schema: 'ID', 'Question', 'Option A', 'Option B', 'Option C', 'Option D', 'Correct Answer', 'Explanation & Response Method', 'Publication 'Date', 'News Title' and 'Source URL.'

4.3.2 DB Calling Module

This study utilizes the distinct data source by leveraging specialized SDK "gspread". This library provides Python-based programmatic access, enabling this study to develop the DB Calling Module. This module is integrated into both the Generation system and Answering systems to serve as the primary interface for database interaction.

The DB Calling Module comprises seven functional functions: Connection Setting, Table Creating, Table Dropping, Table showing, Data Appending, Data Dropping and Data fetching. By adopting this modular architecture, functional components across the dual systems can execute database operations simply by invoking the relevant function, regardless of the underlying data source.

5 Result

After registering a LINE account, users can scan the QR-Code (Figure 3) , click the “Add friend” block, then initial interaction with the DKK-robot which displays the comprehensive result.

Figure 3 QR-Code of DKK-robot

5.1 DKK Operation

The Generation System features fully automated, production capabilities. By utilizing a scheduled server task, the system initiates its operational pipeline daily at 13:00 (GMT+8), consistently performing news scraping, data filtering, item generation, and packaging.

Regarding economic efficiency, the DKK system operates with zero human labor costs. Infrastructure expenses include $1.67 per day for GCP MySQL and $0 for Google Sheets. With the Gemini-2.5-flash-lite model priced at $0.40 per million tokens and an average consumption of 5,000 tokens per item, the production cost per question is approximately $0.002.

5.2 DKK’s question set & AI generation

All question sets in DKK’s are made by AI, and generation at every steps are evaluated by G-Eval, a widely recognized benchmark in the NLP community. G-Eval is an open-source framework using LLM-as-a-judge with chain-of-thoughts (CoT) to evaluate LLM outputs,

"This study recorded 26 data points collected between March 20 and March 28, 2026. Based on the official default evaluation criteria shown in Figure 4, DKK’s average G-Eval scores for the first four production stages were 0.949, 0.950, 0.904, and 0.865, respectively, significantly exceeding the quality threshold (0.5). These results confirm the LLM integrated within DKK exhibits excellent robustness.

```
clarity = GEval(
    name="Clarity",
    evaluation_steps=[
        "Evaluate whether the response uses clear and direct language.",
        "Check if the explanation avoids jargon or explains it when used.",
        "Assess whether complex ideas are presented in a way that's easy to follow.",
        "Identify any vague or confusing parts that reduce understanding."
    ],
    evaluation_params=[LLMTestCaseParams.ACTUAL_OUTPUT],
)
```

Figure 4 Official default evaluation criteria

"Table 1 presents an example of the DKK’s AI Manufacture Module generation, and it is based on the Source new: https://reurl.cc/ZldgVV

Table 1 Question set of DKK

Dataset Items	Content
Question	Considering the damages caused by the Wang Fuk Court fire, which of the following statements best summarizes the multifaceted nature of the disaster's impact?
Option A	The fire primarily caused minor property damage and a few injuries, with the main impact being a temporary disruption to local services.
Option B	The primary damages were limited to the immediate structure of the building, with minimal loss of life and no significant long-term economic consequences.
Option C	The fire resulted in a catastrophic loss of life, extensive structural destruction of multiple buildings, substantial financial strain on victims and the community, and profound psychological trauma.
Option D	The damages were solely economic, with significant insurance payouts compensating for the loss of personal belongings and the structural rebuilding efforts.
Correct Answer	C
Explanation	The correct choice encompasses the widespread fatalities, the severe destruction across multiple residential blocks, the considerable financial aid required for victims and rebuilding, and the lasting psychological impact on survivors. The other choices significantly underestimate the scale and breadth of the devastation
Response Method	In the event of a home fire, victims should immediately prioritize escape. If there's smoke, stay low and crawl to an exit. Before opening a door, feel it for heat; if hot, use an alternate escape route. Close doors behind you to slow the fire's spread. Once safely outside, call emergency services. If unable to escape, close the door, seal gaps, and signal for help from a window.

5.3 Users interface

Figure 5 illustrates the interface of the DKK-robot. By simply clicking the button “START” and “Select”, user is able to engage in the DRR educational gameplay.

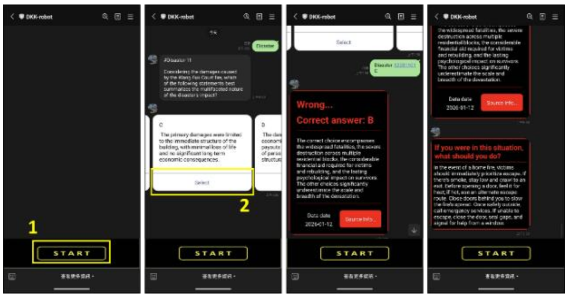

Figure 5 DKK-robot operation process

5.4 User acceptance testing (UAT)

This study utilized five customized questions to preliminarily evaluate user experience, each graded from 0 (disagree) to 5 (agree). 20 students and staff members from NTUST participated in the usability testing of DKK.

As illustrated in Table 2, the average UAT scores indicate that while system's usability is excellent, the tutorial content was rated as relatively moderate.

Table 2 UAT questions and scores

Questions	score
I find the operating process of this system very simple.	4.526
I believe most people can easily access and use this system.	4.579
I think the user interface of this system is clear and straightforward.	3.895 .
Compared traditional lectures, this system is more engaging for my learning.	3.684
The information presented by this system is fresh and interesting, and the knowledge I learned is very practical.	3.737

6 Future work

First, developing others more popular social media to be the interface of the DKK-robot, e.g. WhatsApp.

Second, while this paper prioritizes the execution framework, further research into prompt engineering is essential to refine the DKK questions' quality.

Third, various applications can be integrated into DKK, e.g. site safety induction for workers.

7 Conclusion

By integrating the LLMs with RAG, Chatbot, and automation, this study has "successfully developed a groundbreaking Disaster Risk Reduction education method". DKK is an advanced evolution of the classic mobile game Knowledge King, distinguished by its automatically and dynamically updating question database. Through a engaging gameplay process, users can effectively learn disaster-related news and practical knowledge from real cases.

While the average cost per question set in DKK is only $0.002, all G-Eval and UAT scores significantly exceed the median threshold. This study proposes a viable framework of DKK, so further research and development are needed to improve its practicality.

References

[1] W. Masocha, N. Takaidza, A. Manyani, and C. Mutseekwa, 'Disaster risk reduction integration into school curriculum: A global analysis,' 2025.

[2] N. Aghaei, H. Seyedin, and H. Sanaeinasab, 'Strategies for disaster risk reduction education: A systematic review,' *Journal of education and health promotion*, vol. 7, no. 1, p. 98, 2018.

[3] G. Biancini, A. Ferrato, and C. Limongelli, 'Multiple-choice question generation using large language models: Methodology and educator insights,' in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, 2024, pp. 584-590.

[4] H. K. Ilarionov, I. Domanetska, and O. Fedusenko, 'Features of the practical use of LLM for generating quiz,' 2024.

[5] S. Sheikhalishahi, A. Haddadi, S. Sadeghipour, F. Rafiei, and H. Soltani, 'Comparative performance of ChatGPT-4o, ChatGPT-5, and gemini 2.5 flash on Persian internal medicine subspecialty board exams,' *Scientific Reports*, 2025.

[6] Y. Gao *et al.*, 'Retrieval-augmented generation for large language models: A survey,' *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[7] J. Genesis, 'Retrieval-Augmented Generation: A Comprehensive Survey of Advances and Applications in Deep Learning Models,' *Available at SSRN 5203514*, 2024.

[8] Ö. C. Karacaoğlu, D. Karacaoğlu, and N. Özbaltan, 'Artificial Intelligence Based Solutions in Disaster Literacy Training,' in *10th International Conference on Lifelong Education and Leadership for ALL (ICLEL 2024)*, 2025: Atlantis Press, pp. 28-45.

[9] O. Yetişensoy, 'AI-enhanced disaster education in middle schools,' *Educational Technology & Society*, vol. 29, no. 1, pp. 205-222, 2026.

[10] K. Squire, 'Video games in education,' *Int. J. Intell. Games & Simulation*, vol. 2, no. 1, pp. 49-62, 2003.

[11] J. D. Fletcher, 'Effectiveness and cost of interactive videodisc instruction in defense training and education,' 1990.