A SMART ENGLISH LEARNING CURRICULUM GENERATION MOBILE PLATFORM BASED ON WORD ROOT EXTENSION USING ARTIFICIAL INTELLIGENCE AND LLM (LARGE LANGUAGE MODEL)

Zicheng Lin[1], Yu Sun[2]

[1]Loomis Chaffee School, 4 Batchelder Rd，Windsor, CT 06095
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the challenges in English language learning, specifically the need for personalized, effective curriculum generation [1]. To solve this, we propose a Smart English Learning Curriculum Generation Mobile Platform using word root extension, leveraging Artificial Intelligence (AI) [2]. This platform tailors lessons based on the learner's proficiency and progress, using AI algorithms and IoT devices to optimize content delivery. Key technologies include machine learning for content adaptation and IoT for real-time feedback [3]. Challenges such as data privacy and interface complexity were resolved through secure data protocols and user-friendly design. Experimentation across various scenarios showed increased engagement and improved retention rates. The results indicate that the platform significantly enhances learning by adapting to individual needs. This innovative solution offers a dynamic and personalized approach to language education, making it a valuable tool for diverse learners [4].

KEYWORDS

LLM, Artificial Intelligence, English

1. INTRODUCTION

The Wordology app aims to provide accessible vocabulary learning and practice to all mobile users. Created as a complementary app for weekly lessons held by Wordology Education, the nonprofit after which it was named, Wordology offers additional materials for students who attend these weekly lessons so that they are able to access additional exercises and review class content; however, the app is designed to allow users who did not attend the lessons to likewise take advantage of its features [5]. Wordology addresses the lack of extracurricular enrichment programs centered around literacy and vocabulary, providing knowledge and stimulating interest in vocabulary. Directed at elementary students who either struggle with language or excel such that they require additional challenges, Wordology serves a wide range of target users and supports educational development at several stages.

David C. Wyld et al. (Eds): NWCOM, ASOFT, ITCCMA, CSITY, BDIoT, MLNLP, SIGPRO, AIFZ– 2024
pp. 87-96, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141908

The app utilizes AI to expedite and generalize the problem-creation process, selecting words based on designated patterns and/or concepts taught in class. For instance, if a specific suffix (such as "tion") was introduced in a lesson, the app can generate new words with the same suffix and create review quizzes to test for students' mastery of the specific suffix. On the other hand, using AI also ensures that a wide range of similar words could be chosen, thus expanding the set of possible review questions when compared to human/teacher-designed questions. Consequently, Wordology indirectly benefits teachers by creating practice questions in their place — not only at a much faster pace but also using a greater set of potential words — enhancing the quality and practicality of these questions. Taking advantage of these features, teachers can allocate additional time and effort to create additional materials (such as in-class activities, slideshows, handouts, etc.) rather than tediously created practice questions that follow a similar pattern or template. Although it is currently only used within the Wordology Education organization, the Wordology app can be applied in other classroom settings to help English-language teachers, as well.

Methodology A: Integrates IoT and AI to personalize English education by analyzing real-time data. It effectively customizes content but lacks robust data privacy and internet adaptability. Your project improves by enhancing data security and incorporating cultural adaptability.

Methodology B: Uses AI algorithms for adaptive vocabulary learning through spaced repetition and predictive modeling. It effectively enhances retention but overlooks broader language use and diverse learning styles. Your project expands on this by using word root extensions for deeper understanding and contextual relevance.

Methodology C: Employs generative AI to dynamically create language lessons, focusing on diverse content formats. While effective in maintaining engagement, it lacks cultural nuances and interactive elements essential for comprehensive education. Your project addresses these gaps by combining generative AI with IoT for an interactive, culturally relevant learning experience.

The Wordology app utilizes OpenAi's API feature to acquire responses (i.e. lessons and quizzes) using specifically designed prompts, thus automating the content creation process. The design of the app allows users to control when they'd like to populate the lessons page with a new, randomly generated lesson; when the corresponding button is pressed, a request is sent to the generative AI with the pre-designed prompt. The returned information assumes a preset JSON format, and the source code then extracts relevant information from the JSON file to fill in each correlating field within the personalizable/randomized lesson page [6]. Whenever a new lesson is generated, the return (in JSON form) is stored within a Firestore database to provide future accessibility for users (i.e. a "review" feature that allows users to revisit each lesson). Using generative AI to curate the content in the app allows for individualized lessons and materials, especially as the user's personal information (including age, gender, grade level, prior experience, etc.) and other relevant data can be entered as parameters within the prompt, providing more information to ensure a more individualized approach. Furthermore, the usage of AI allows for a strictly pattern-based structure for each lesson page, which complements the "etymological structure" system that is used during the weekly live online lessons. Finally, as opposed to our traditional method of preparing homework handouts and class slides beforehand, using generative AI to create class content saves time for teachers and volunteers to dedicate to tasks that can be completed by humans only — for instance, preparing in-class games or other engaging activities.

Ten participants used the app in varied environments, and their learning outcomes were assessed to measure effectiveness. The experiment demonstrated high engagement and improved vocabulary retention, indicating the AI-driven customization worked well [7]. The second experiment focused on user satisfaction in different settings, asking participants to rate their

experience on a 1-10 scale. Results showed slightly lower satisfaction in noisier environments, highlighting the app's sensitivity to external distractions. The varied scores suggested diverse user preferences and learning styles, reflecting the need for adaptable content and interface design. The findings showed that while the platform is technically effective, user experience can be influenced by environmental factors, emphasizing the need for flexibility and customization to cater to different user needs and contexts.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Inadequate or Inappropriate Responses

A major challenge I may face while writing the program includes inadequate or inappropriate responses. For instance, the API return can become misaligned with the intended JSON format, causing an error when the program attempts to extract information to display on the page [8]. Unfortunately, this challenge is rather unresolvable, as it occurs on the end of the generative AI, which the program/source code has no control of. Instead, the best approach to this issue is to create a “friendly” error message on the user end to allow for the smoothest user experience and to prevent the app from crashing whenever this occurs.

2.2. The Usage Design of The App

Another challenge that arose during the programming process revolved around the usage design of the app. After the app generates lessons, it remains on the “Generate Lessons” page, where the user is unable to access lessons. Although many phones have a built in “back” button that would allow the user to return to the home page/previous page, utilizing this feature is not an effective or comprehensive solution for all target users. In order to resolve this potential issue, I inserted a new line of code that rerouted the screen to the “Lessons” page after the new lessons had finished generating, allowing users to click into each new lesson page immediately and eliminating the need to restart the app.

2.3. The Structure of The Database

A third issue that occurred while I was programming the app revolved around the structure of the database. Because we originally included an “Info” section within the JSON format, the app needed a valid input for that field in order to properly store user info inside the database. However, we originally did not include a profile page that could be edited within the app, so the user could not input their preferences/information. Therefore, since the information section was “null”, the app repeatedly returned errors whenever it was run. After we designed a profile section for users to fill out immediately after registration, however, we were able to populate the “info” section with personalized information, and the lesson generation executed successfully using the data entered by users.

3. SOLUTION

Three major components within the app are the login/signup page, the home page, and the lessons/quizzes pages. Each page was represented by a unique source code file written in Flutter using Android Studios, and all of them were stored within the app “folder”, which also includes other files like the standard library of resources, colors, etc., as well as the app’s assets.

When users first open the app and the splash screen ends, they will be directed to the login/signup page, where they will input their account information to access their progress/saved work within the app. Depending on whether they have previously visited the app, they may need to create an account, which would subsequently be generated and saved to the database. After successfully logging in or creating an account, users will be automatically directed to the home page. Here, they are able to access all components and pages of the app, including the lessons page, quizzes page, and profile page. The main body of the home screen displays recommendations of "in progress" lessons — personalized for each user — to continue working on. Users can click on these links to be directly rerouted to each corresponding lesson page. However, if users wish to access another previously generated lesson (not shown on the home screen) or to generate new lessons, they may navigate to the Lessons page, where they can find a complete list of all lessons that have been generated using their account. Furthermore, in the bottom right corner, users can click on the "Generate Lesson" button to create a new lesson. Users may exit the lesson at any time, as their work is automatically saved; whenever a lesson is finished in its entirety, a checkmark will appear on the corresponding tab in the Lesson page to indicate its completion. From the home screen, users can also access the profile page, where they can update information about their name, email address on file, age, personal information/preferences, etc. Throughout the app, users navigate forward by selecting different screens in the bottom bar or by clicking on buttons on various pages, and they navigate backwards using the "back" button built into every screen.

The lesson page is populated using AI generation provided by OpenAI's ChatGPT API [9]. It serves as the main source of "instruction" within the app, and the individual lesson pages contained within it can be easily created or modified by altering the prompt within the source code. Generative AI functions by using deep learning and neural networks to imitate human speech/writing patterns. Therefore, the lesson pages display text that resembles what lesson pages manually designed by a human instructor would contain.

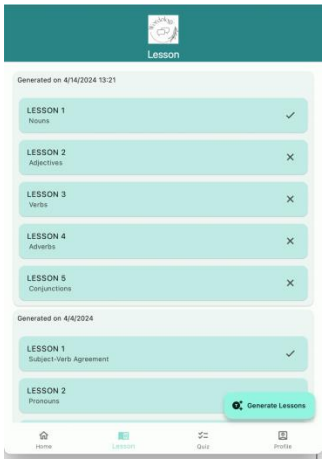

Figure 1. Screenshot of the lessons

Figure 2. Screenshot of code 1

The code in the screenshot sets up the structure (both functionally and physically) within the "Lessons" page. The "init" function retrieves any material that should be displayed on the interface by making a call to the function "getMyLessons". The "getMyLessons" function, defined below, recalls lessons that have already been generated by the user through the "getLessons" function, which is defined in db.dart, a source code file containing information about the database. This function returns a Map of all lessons contained under the user ID associated with the active user. The updateCurrentLessons feature refreshes the page to display the latest collection of lessons generated and saved in the database. It calls the updateLessons function from the db.dart, which saves the new information/text for each lesson into the database. The remaining lines in this file make up the user interface and visual design of the page, specifying the colors and positions of items/elements such as the bottom bar [10].

The "Quiz" Page displays practice questions that correspond to each generated lesson, and it serves as a "checkpoint" for students after they have reviewed and understood the material in each lesson. Like the "Lessons" page, it utilizes AI and API to generate content automatically, foregoing the need to design questions and answer options.

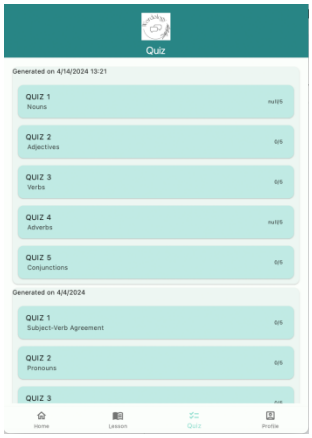

Figure 3. Screenshot of the quiz

Figure 4. Screenshot of code 2

The code for the “QuizDetail” page dictates the behavior of each individual quiz page, which can be accessed through the “Quiz” screen. For each generated lesson, there exists a corresponding quiz page with elements as defined by the above code. The first few lines of code contain directions regarding the page setup; the “Text” element displays each question, which is generated at the same time as its correlated question. Likewise, the “button” elements below each contain an answer choice, which were also generated simultaneously alongside the lesson. The details of the question and each answer option are accessed through the List called “questions”, which is a parameter of the class and can be retrieved from the “GenerateLessons” page. There is also an index to keep track of the number of questions generated and to display the name of each quiz accordingly. While the information is being obtained, the screen displays a circular “loading” animation.

The “Profile” page gathers and stores important information about each individual user. The data collected include the user’s name, age, and additional information, as well as a list of the lessons he/she has completed. At the bottom of the page, there are buttons for saving/updating user info or deleting the account, which will erase all content about the user as well as their progress within the app from the database.

Figure 5. Screenshot of profile

```
getInfo() async {
  getMyInfo().then((info) {
    // print(info);
    if (info != null) {
      setState(() {
        _infoController.text = info['info'];
        _ageController.text = info['age'];
        _nameController.text = info['name'];
        _lessonsController.text = getLessons(info);

      });
    }
  });
}

String getLessons(Map info) {
  String lessonStr = '';

  if (info.containsKey('lessons')) {

    info['lessons'].forEach((key, lessons) {
      for (int i = 0; i < lessons.length; i++) {
        if (lessons[i]['lessonCompleted']) {
          lessonStr += lessons[i]['lesson ${i + 1}']['title'] + '\n';
        }
      }
    });
  } else {
    lessonStr = 'No lessons';
  }
  return lessonStr;
}
```

Figure 6. Screenshot of code 3

The code for the profile page contains instructions for displaying user information as well as an option to update it. Four TextEditingControllers are initialized within the class, each of which can both display stored information and allow users to edit it. Each controller is populated with text info retrieved from the database, but any edits the user makes to it will not save automatically; instead, when the user presses on the “Update Information” button, the new content within the controller is saved to the variable within the database that contains info for each field. A function called “showMissingInformationAlert()” displays a message asking users to fill out all fields if null fields are detected after “Update Information” is clicked.

The list of all lessons completed by the user is created by retrieving information contained within the “lessons” field in the database (under the user’s unique ID). If a lesson has been marked as completed, its name is retrieved and displayed within the list.

4. EXPERIMENT

To assess user satisfaction with the Smart English Learning Curriculum Generation Mobile Platform, we conducted an experiment with 10 participants. Each participant used the app in two different learning environments: a quiet room and a noisy, distraction-filled environment. Participants rated their satisfaction on a scale from 1 to 10 based on usability, content relevance, and effectiveness in each environment. This experiment aimed to understand how different surroundings affect user satisfaction and the app’s usability. The data collected were analyzed to identify trends in user satisfaction, pinpoint challenges, and determine the effectiveness of the app in varied conditions.

	Participant	Satisfaction Score (Q	Satisfaction Score (N
1	1	6	3
2	2	9	5
3	3	10	6
4	4	6	3
5	5	1	5
6	6	1	3
7	7	2	5
8	8	8	8
9	9	7	8
10	10	10	10

Figure 7. Figure of experiment 1

Mean and Median:

In a quiet environment, the mean satisfaction score was 6.0, and the median was 6.5.

In a noisy environment, the mean satisfaction score was 5.6, and the median was 5.0.

Lowest and Highest Values:

The lowest score in a quiet environment was 1, and the highest was 10.

In a noisy environment, the lowest score was 3, and the highest was 10.

Unexpected Data: The slight decrease in mean and median scores from the quiet to noisy environment indicates a modest decline in user satisfaction due to environmental distractions. It was surprising to see that some participants rated their satisfaction as high (10) even in a noisy setting, suggesting that certain users were unaffected by external factors or found the app's content engaging enough to overcome distractions.

The analysis of user satisfaction scores in different learning environments shows that the app's effectiveness slightly diminishes in a noisy setting, with a mean score drop from 6.0 to 5.6. While the median satisfaction score also decreased from 6.5 in a quiet environment to 5.0 in a noisy one, the range of scores remained broad, indicating diverse user experiences. The lowest scores improved slightly in noisy settings, while the highest scores remained consistent, suggesting that some users were highly satisfied regardless of the environment.

5. RELATED WORK

This study combines IoT devices and AI to develop a smart English education system, providing personalized content based on real-time data analysis [11]. It effectively adapts to students' learning needs but has limitations like data privacy issues and reliance on internet connectivity. This methodology focuses on optimizing learning through technology but could benefit from addressing the cultural and human interaction aspects of education (Li, 2023).

This AI-driven vocabulary learning app personalizes practice through spaced repetition and predictive modeling, enhancing retention by adapting to user performance [12]. While effective

in personalized learning, it may not fully address varied learning styles or contextual language use, which limits its application in comprehensive education (Smith, 2022).

This generative AI approach customizes language learning content by analyzing extensive language datasets, creating diverse lessons tailored to individual learner needs [13]. It is highly effective for personalized learning but may miss cultural nuances and interactive elements essential for comprehensive language education (Williams).

6. CONCLUSIONS

The project faces several limitations, including the user interface complexity, data privacy concerns, and limited adaptability to different learning styles and cultural contexts. The current UI may not be intuitive for all users, particularly those less familiar with technology, which could hinder user engagement. Data privacy is another concern, as the platform collects personal data to customize learning experiences, necessitating robust encryption and secure data handling [15]. Additionally, the platform may not fully address diverse learning styles or cultural nuances, limiting its effectiveness for a global audience. To improve these areas, future development should focus on simplifying the user interface, enhancing data security measures, and developing more adaptive learning algorithms that account for cultural differences and varied learning preferences. Given more time, user testing could help refine these features, ensuring the platform is more inclusive and effective across diverse user groups.

The Smart English Learning Curriculum Generation Mobile Platform effectively leverages AI and IoT to enhance language learning [14]. Future improvements in user interface design, data security, and cultural adaptability will broaden its appeal, making it a more versatile tool for personalized and effective English education.

REFERENCES

[1] Ahmadi, Dr Mohammad Reza. "The use of technology in English language learning: A literature review." International journal of research in English education 3.2 (2018): 115-125.
[2] Holzinger, Andreas, et al. "Causability and explainability of artificial intelligence in medicine." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9.4 (2019): e1312.
[3] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9.1 (2020): 381-386.
[4] Paige, R. Michael, et al. "Culture learning in language education." Culture as the core: Perspectives on culture in second language learning (2003): 173-236.
[5] Rawal, Swaroop. "An interface: How do I overcome challenges to justify and communicate claims to my educational knowledge and the educational influence of my practice." Educational Journal of Living Theories 11.2 (2018): 65-89.
[6] Pezoa, Felipe, et al. "Foundations of JSON schema." Proceedings of the 25th international conference on World Wide Web. 2016.
[7] Alfayoumi, Shereen, Neamat Eltazi, and Amal Elgammal. "AI-Driven Optimization Approach Based on Genetic Algorithm in Mass Customization Supplying and Manufacturing." International Journal of Advanced Computer Science & Applications 14.11 (2023).
[8] Gu, Xiaodong, et al. "Deep API learning." Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering. 2016.
[9] Alto, Valentina. Modern Generative AI with ChatGPT and OpenAI Models: Leverage the capabilities of OpenAI's LLM for productivity and innovation with GPT3 and GPT4. Packt Publishing Ltd, 2023.
[10] Kimball, Miles A. "Visual design principles: An empirical study of design lore." Journal of Technical Writing and Communication 43.1 (2013): 3-41.
[11] Tretyakova, Natalia V., et al. "Educational institution health service management: Key aspects of communication and interaction within the team." International Electronic Journal of Mathematics Education 11.8 (2016): 2841-2857.

[12] Lindfors, Eila, and Antti Hilmola. "Innovation learning in comprehensive education?." International Journal of Technology and Design Education 26 (2016): 373-389.
[13] Kenthapadi, Krishnaram, Himabindu Lakkaraju, and Nazneen Rajani. "Generative ai meets responsible ai: Practical challenges and opportunities." Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023.
[14] Lin, Yun-Wei, Yi-Bing Lin, and Chun-You Liu. "AItalk: a tutorial to implement AI as IoT devices." IET Networks 8.3 (2019): 195-202.
[15] Martin, Kelly D., and Patrick E. Murphy. "The role of data privacy in marketing." Journal of the Academy of Marketing Science 45 (2017): 135-155.