Personalized Learning Through Gamification: Multi-Agent and Large Language Model Approaches

Kiem Hoang, Tin Huynh, Duong Ham Tran, Tri Pham, Binh Tran
Center of Data and Artificial Intelligence
The Saigon International University
Email: {hoangkiem, huynhngoctin, tranhamduong, phamxuantri, tranlehaibinhk12}@siu.edu.vn

***Abstract*—Personalized learning aims to adjust learning objectives, content, methods, and pace to suit the needs, preferences, and abilities of each individual. To effectively meet the goals of personalization, educators must prepare content tailored to each learner. This can lead to overload and potentially impact quality when there are too many aspects to consider for each learner. Hence, the demand for automatically generating content tailored to each learner is a critical necessity. In this study, we propose a framework that utilizes a Multi-Agent System combined with Large Language Models (LLMs) to automatically generate personalized content for higher education. To help each student acquire new knowledge in an enjoyable manner, this research focuses on generating games based on course materials. The proposed solution promises to deliver engaging experiences and enhance learning motivation by integrating playful activities into the learning process tailored to each individual.**

***Index Terms*—LLM, Multi-Agemt, Personalized Learning, Gamified Learning**

I. Introduction

Personalized learning is considered a significant advancement in education, enabling better support for all learners while respecting the diversity and individuality of each person [1]. For example, in a personalized learning system that incorporates coding games, students learn and experience programming tasks through fun activities suited to their skill levels, from writing basic code for beginners to debugging or optimizing algorithms for advanced learners.

The National Academy of Engineering in the United States has emphasized the development of personalized learning systems as one of the "Grand Challenges" of the 21st century, alongside other initiatives with the potential to drive global change [2]. Over the past decade, research in personalized learning has continued to grow, highlighting the importance and challenges of this field and underscoring the need for further efforts to find effective solutions [3].

To effectively meet the objectives of personalization, each teacher must prepare different content for each student, or multiple teachers and teaching assistants may be required for each student. However, according to statistics from the Vietnamese Ministry of Education, there is a significant disparity in the teacher-to-student ratio [1]. Specifically, in 2021, the ratio was 76,576 permanent faculty to 1,905,956 students; while in 2022, a similar ratio was maintained with 78,190 faculty for 2,145,426 students. On average, each teacher was responsible for teaching about 25 students in 2021 and about 27 in 2022. This disparity not only leads to faculty overload but also makes it difficult to ensure quality when implementing personalized teaching for each student. Therefore, meeting the personalization needs of the majority of learners today is not feasible without the serve of technology.

In this study, we propose a personalized learning system for higher education that integrates Multi-Agent Systems and Large Language Models (LLMs) to implement game-based learning models. This system utilizes LLMs to automatically generate educational content, including questions, game scenarios and materials, aiming to optimize the enjoyable learning experience for each student individually. Meanwhile, Multi-Agent Systems are incorporated to analyze learning styles, goals, and personal preferences, coordinating complex tasks such as creating interactive games and tracking learning progress. This solution aims to meet the demands for personalization at scale and the practical deployment capability in higher education while also reducing the workload for lecturers as well as educators. The system not only supports students in learning effectively but also enhances motivation by integrating engaging gamification elements such as rewards, leaderboards, and personalized tasks.

The main contributions of this study can be summarized as follows:

- Defined the problem of personalized learning as a formulaic expression focusing on content generation and gamification.
- Proposed a framework utilizing a multi-agent architecture and Large Language Models (LLMs) to create games that are tailored to the needs, preferences, goals, content, and learning abilities of each student.
- Provide an illustrative use case conducted at Saigon International University as a case study to demonstrate the effectiveness of the proposed framework.

II. Related works

Since 1958, psychologist B.F. Skinner demonstrated that "teaching machines" could enhance students' ability to learn independently, allowing them to complete tasks on their own

[1]https://moet.gov.vn/thong-ke/Pages/thong-ko-giao-duc-dai-hoc.aápx

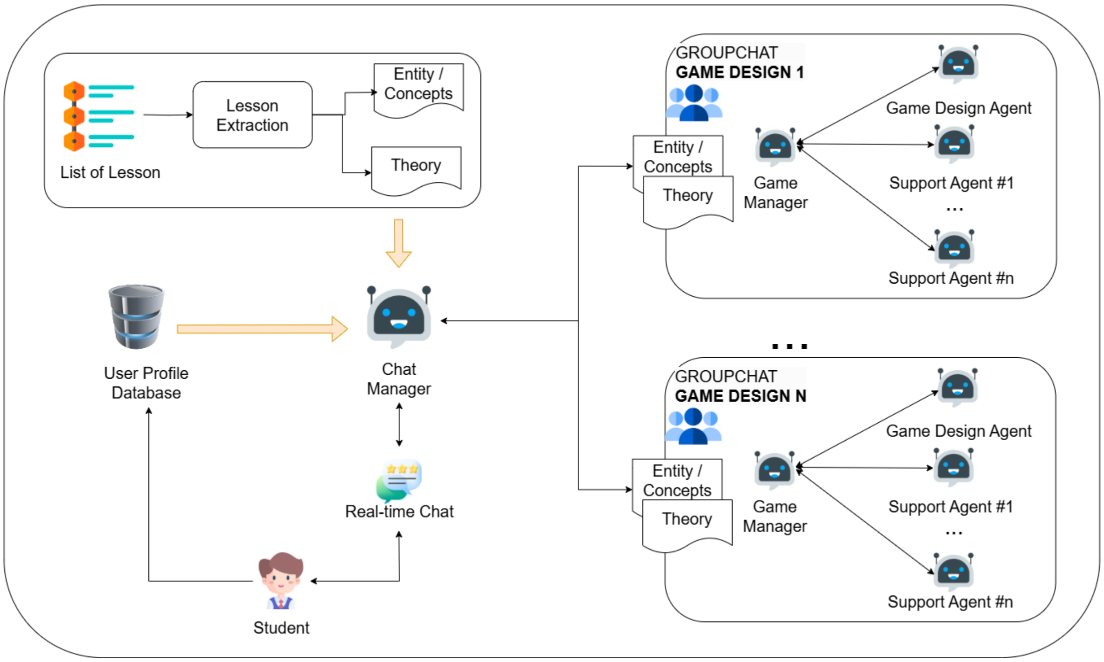

Fig. 1. Overview of Multi-Agent Systems for Automatically Generating Gamified Learning.

and adjust their learning pace [4]. This innovation opened a new approach for the application of technology in education. In the following decades, researchers continued to explore digital-based learning methods, which proved to be more effective than traditional classroom settings [5]. From around 2008, personalized learning began to attract significant attention from many researchers [6], with various approaches such as adaptive learning systems [7] and traditional online learning tools [8]. These systems primarily used rule-based algorithms or simple statistical models to tailor educational content to each learner's ability. However, these tools have limitations such as a lack of flexibility, difficulty in understanding and meeting the complex needs of personalized learning, especially when the educational content includes interdisciplinary fields or requires the processing of advanced linguistic analytics.

Recently, the advent and emergence of Large Language Models (LLMs) [9] has introduced an entirely new approach, overcoming the limitations of previous systems. With their ability to process and understand natural language as humans do, LLMs can generate customized learning content, from quizzes [10] and questions [11] to lesson plans [12], not only tailored to individual learning needs but also based on complicated data across various contexts [13]. Moreover, these models can provide real-time feedback [14], offer detailed explanations, and foster motivation, significantly enhancing the interactivity and effectiveness of the learning process. Beyond merely providing content, LLMs have also been deployed in complex systems such as Multi-Agent Systems [15] and Mixture-of-Experts (MoE) [16]. In these systems, each agent or specialized model is responsible for a specific domain or task, such as solving mathematical problems, analyzing text, or providing scientific feedback [17]. Compared to traditional systems that rely on a single tool or algorithm, this approach is more flexible and efficient, effectively meeting the diverse personalized learning needs of learners.

Gamified Learning [18] is a teaching method that transforms the learning process into an engaging and enjoyable experience by incorporating game-specific elements, rather than relying solely on traditional lectures. Over the past decade, gamified learning has attracted significant attention from the research community due to its superior ability to motivate and drive the learning process while enhancing student engagement [17]. This method has proven effective in boosting learning motivation and enhancing personalized Learning. Firstly, Gamified Learning enables learners to customize content according to their preferences, learning styles, and personal goals [19]. For example, those who enjoy challenges can choose puzzle-based minigames, while others might prefer simulation games or storytelling. This personalization fosters a comfortable and enjoyable learning experience. Secondly, integrating game elements such as points, levels, rewards, and immediate feedback encourages active participation, enhances memory retention, and improves the practical application of knowledge [18], [20]–[22]. For instance, in language learning, games like word matching or fill-in-the-blank exercises have significantly enhanced memory retention. Despite the benefits that Gamified Learning offers, implementing it comes with its own set of challenges. The process of creating a minigame requires significant investment in content, visuals, programming, and testing. To address these difficulties, we have developed a personalized learning system that automatically transforms lesson content into minigames. This system not only saves time but also expands the scalability of Gamified Learning, making the learning journey more engaging and effective.

To implement a personalized learning support system through gamification, this study proposes a multi-agent architecture to efficiently manage and automate the creation of minigames. Within this system, each minigame is developed by a group chat comprising LLM-based Agents, each assigned specific roles depending on the type of minigame required. For example, an image-generation Agent is responsible for creating visual illustrations, while a content-generation Agent focuses on developing scenarios, questions, and other relevant materials. This coordinated collaboration among Agents enables the production of engaging minigames with diverse and rich content, tailored to meet the personalized learning objectives of individual students.

III. Problem Statement

Input: $U = \{u_1, u_2, \dots, u_n\}$, where each u_i is a representative vector encapsulating the attributes of user i. Each vector $u_i = (D, C, T, G)$ is defined as follows:

- $D = \{sex, age, major, level, \dots\}$ denotes a vector comprising the personal attributes of the user, such as Gender (sex), Age (age), Major ($major$), and other relevant details.
- $C = \{CGPA, GPA, GAS, \dots\}$ denotes a vector representing the user's academic performance, including metrics such as Cumulative Grade Point Average (*CGPA*), Semester Grade Point Average (*GPA*), and Subject Grade Average (*GAS*), among others.
- $T = \{t_1, t_2, \dots, t_q\}$ represents a vector of the user's topics of interest (*Topics of Interest*), where each t_i corresponds to a specific subject or thematic area.
- $G = \{g_1, g_2, \dots, g_p\}$ signifies a vector of the user's gamification preferences (*Gamification Preferences*), where each g_i denotes a type of game the user prefers for learning. In this study, two game types are utilized: *4 Pics 1 Word* and *Coding Game*.

Output: A personalized set of questions or exercises q, generated to align with the individual user's needs, preferences, and competencies. Formally, for every $u \in U$, $t \in T$, and $g \in G$, the output is defined as:

$$Q = \{q_j\}_{t,g}^{u} \tag{1}$$

Objective Function

$$f(U, T, G) = \arg\max \frac{1}{n} \sum_{i=1}^{n} \frac{1}{d} \sum_{j=1}^{d} \text{satisfied}(q_j^i) \tag{2}$$

where

- $satified(q_{j,i})$ quantifies the satisfaction level of user u_i with the generated question or exercise q_j, measured on a scale ranging from 1 (least satisfied) to 5 (most satisfied).
- n denotes the total number of users in U, and d represents the number of questions generated for each user.

IV. Our Approaches

A. Method overview

The system is designed to leverage the power of LLM Agents to automate the process of creating minigames. Figure 1 illustrates the overall structure and operation of the system, which includes the following key components below.

*1) **Lesson Extraction**:* The system begins by analyzing learning materials through the Lesson Extraction module, which extracts the set $L\{k, t\}$, containing the key elements from each lesson, as follows:

$$L = \{k, t\}$$

where

- k: A set of keywords related to concepts, definitions, theorems, and other important elements in the lesson.
- t: A set of detailed content related to the concepts, definitions, theorems, etc., in the lesson.

*2) **Chat Manager**:* The Chat Manager serves as the ***central component*** responsible for interacting with users and selecting appropriate content to send to the Game Design Group Chats.

*3) **Game Design Group Chat**:* Each group chat operates as an independent unit comprising multiple agents, including the Game Design Agent (GDA), Support Agents (SA), and the Game Manager (GM), each with specific roles as follows:

- **Game Design Agent (GDA)**: Responsible for designing minigames (MG), including creating questions, challenges, and scenarios based on the extracted keywords and contents. The MG includes questions, challenges, and a scenario.
- **Support Agents (SA)**: A collection of supporting agents, $S_A = \{sa_1, sa_2, \dots, sa_n\}$, where each sa_i has specialized capabilities, such as accessing the internet, querying internal databases, retrieving information, generating multimedia content, or optimizing game interfaces.
- **Game Manager (GM)**: Control the overall coordination of the game design process, ensuring synchronization, receiving MG content (questions, challenges, and scenarios) from the GDA, sending specific requests to each sa_i in S_A to perform supporting tasks, and collecting user feedback to enhance game personalization.

*4) **User Profile Database**:* The User Profile Database used to store personal information, learning behaviors, and feedback, providing critical data for personalizing learning content. The system is designed to be highly scalable, enabling the deployment of various games to meet diverse learning objectives and the personalized needs of each student

B. Gamified Learning Generation

1) 4 Pics 1 Word game: "4 Pics 1 Word" is a popular game where players are provided with four pictures (images) and must guess the keyword based on the hints from the pictures. Figure 2 illustrates the process in which various Agents collaborate to create a game level. For example, generating

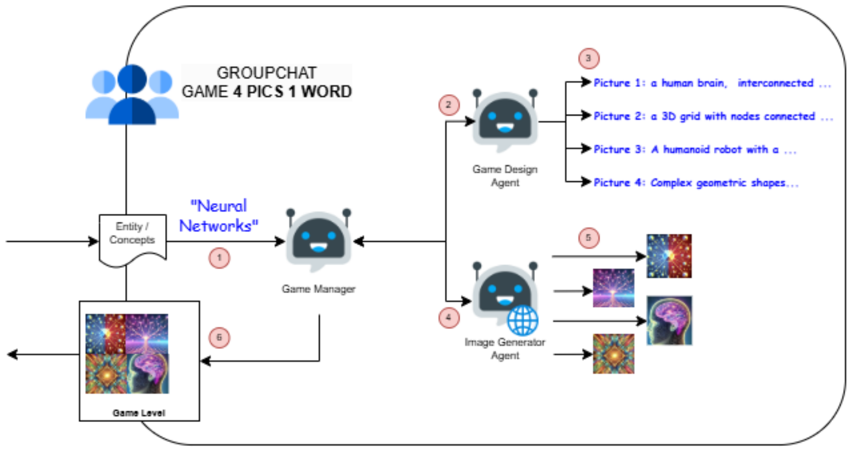

Fig. 2. The Agents collaborate to design the game 4 Pics 1 Word game.

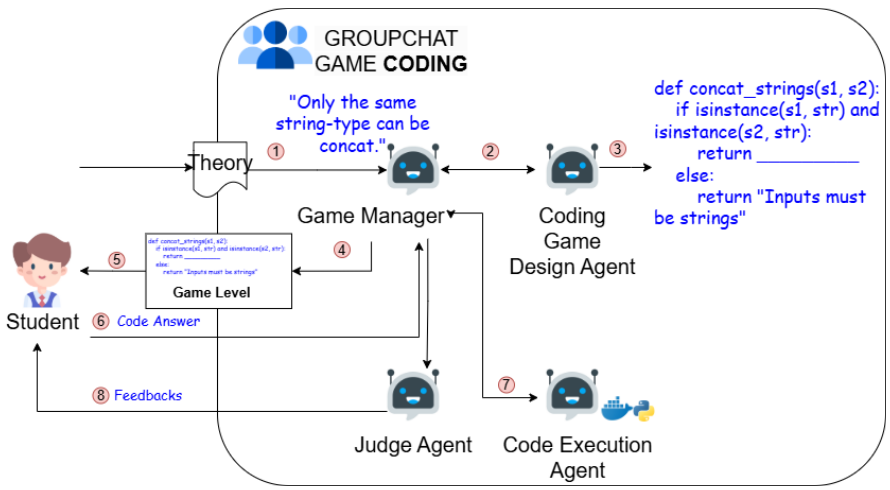

Fig. 3. The Agents collaborate to design the coding game.

a "4 Pics 1 Word" game level with the keyword "Neural Networks" involves the following steps:

Game Manager: The process begins when the Agent receives information from the Chat Manager specifying that the game topic is "Neural Networks." Based on this information, the Agent sends a request to the Game Design Agent to generate descriptions for four images related to the topic. Once the descriptions are provided by the Game Design Agent, the Agent forwards them to the Image Generator Agent to produce the corresponding images. Finally, the Agent collects the generated images from the Image Generator Agent and compiles the descriptions and images to construct a complete game level.

Game Design Agent: The process begins when the Agent receives a request from the Game Manager specifying the topic "Neural Networks." Based on this topic, the Agent proceeds to generate descriptions for the corresponding images.

- Image 1: A human brain, symbolizing biological neural networks.
- Image 2: A 3D grid connecting nodes, representing network structures.
- Image 3: A humanoid robot, illustrating AI applications.
- Image 4: Complex geometric shapes, representing neural algorithms.

Image Generator Agent: Generating four images based on the descriptions provided by the Game Design Agent.

2) Coding game: Coding Game is designed to encourage students to develop programming skills through practical and application-driven challenges. The system leverages the power of a multi-agent architecture to provide an interactive and effective learning environment. In the game, students complete short programming exercises with varying levels of difficulty, such as completing source code or debugging errors. The operation process is based on the coordination of multiple agents, each responsible for specific tasks to support learners. For example, generating an coding games around the statement *"Only the same string type can be concatenated"*, involves the following steps:

Game Manager Agent: The Game Manager Agent begins by receiving the user's request and confirming the input information. It then forwards the learning concept to the Coding Game Design Agent to generate a suitable exercise. Based on the user's abilities and needs, the Game Manager Agent selects an appropriate type of exercise, such as completing source code or debugging errors. After determining the exercise's type and difficulty level, it sends these details along with the learning concept to the Coding Game Design Agent to create the programming task. Once the exercise is designed, the Game Manager Agent delivers it to the user and receives their response, which is subsequently forwarded to the Code Execution Agent for code execution.

Coding Game Design Agent: Based on the game type, difficulty level, and provided concept, the Coding Game Design Agent generates a programming exercise, such as the following *"complete the source code"* task:

```
def concat_strings(s1, s2):
  # Check if both inputs are strings
  if isinstance(s1, str) and isinstance(s2
     , str):
    return ________  # Fill in the missing
         part to concatenate the strings
  else:
    return "Inputs must be strings"
```

Code Execution Agent: Receives the user's response from the Game Manager Agent and executes the code.

Judge Agent: Evaluates the execution results, analyzes the source code, and provides detailed feedback to the student, including suggestions or improvements if necessary.

V. Implementation & Demonstration

A. Multi-Agents Frameworks

We utilized microsoft/autogen to develop an advanced LLM workflow. A hierarchical structure in autogen allowed a managing agent to coordinate subordinate agents, decomposing complex tasks into modular, scalable sub-tasks. Meanwhile, autogen studio refined interaction protocols and visualized conversational flows in real-time, ensuring seamless agent collaboration. This setup effectively demonstrated the practical capabilities of LLMs in addressing multi-step, sophisticated problems.

B. 4 Pics 1 Word Game Demonstration

For the keyword "Neural Networks", the system demonstrated the ability to generate a cohesive game level based on agent collaboration. The detailed responses from the agents are presented as follows:

- **Game Design Agent:** Generating images descriptions for creating four images.
  - A close-up of interconnected neurons, with electrical impulses traveling along the axons, symbolizing biological neural networks.
  - A scientist sketching a complex neural network diagram on a glass board, showcasing layers of nodes and interconnections.
  - A computer screen visualizing a 3D model of a neural network, with multiple layers and nodes lighting up in sequence.
  - An artificial intelligence robot with a transparent head, revealing a glowing neural network inside, representing machine learning.
- **Image Generation Agent:** Generating four images shown in Fig 4.
- **Game Manager:** Responding to the complete question for the user.
  - **Question:** Guess the keyword.
  - **Images:** Four images generated by the Image Generation Agent shown in Fig. 4.

C. Coding Game Demonstration

The coding game results demonstrate the system's ability to automatically generate, evaluate, and provide feedback for programming exercises. Below is an example of how the system processes the statement about "String Concatenation" using its multi-agent framework:

- **Game Design Agents:** generates a Python programming task based on the statement.

```
# Here is a Python code snippet that
   can be used:
# Fill in the blanks to correctly
   concatenate the strings.

name = "John"
greeting = "Hello "

# Fill in the blanks with the correct
   operator
message = _______ + _______

# This should print "Hello John"
print(message)
```

- **User:** submits their response

```
message = greeting + name
```

- **Code Executor Agent:** tests the user's solution using a provided test case.

```
name = "John"
greeting = "Hello "
message = greeting + name
assert message == "Hello John"
```

- **Judge Agent:** provides detailed feedback to the learner.

```
"Great job! You've correctly filled in
the blanks with the correct variables to
complete the string concatenation. Your
understanding of Python variables and
string operations is commendable! Keep
up the good work."
```

VI. Conclusion

This study proposed a multi-agent framework powered by Large Language Models (LLMs) to support the automation

Fig. 4. 4 Images related to "Neural Network" concept generated by Image Generation Agent

of gamified personalized learning experiences. By integrating agents with specialized roles—ranging from game design and content generation to code execution and evaluation—the system successfully created interactive learning tools such as "4 Pics 1 Word" and coding games. Initial results demonstrated the framework's capability to deliver tailored educational content effectively, offering a scalable solution to enhance student engagement and learning outcomes. However, several aspects warrant further investigation. Future work will focus on addressing the limitations of gamified learning, particularly ensuring that game elements do not overshadow educational goals. Additionally, the framework will be refined to improve its robustness, scalability, and adaptability to a broader range of learning objectives and user preferences. Another promising direction is enhancing the quality of generated content by integrating Retrieval-Augmented Generation (RAG) methods, enabling the system to access and incorporate more relevant, domain-specific information in real-time. A key step forward will involve deploying the system in practical educational settings to validate its effectiveness, gather user feedback, and further optimize its design based on real-world application. This study underscores the potential of LLM-powered multi-agent systems in advancing personalized education and sets the stage for future innovations in interactive learning technologies.

References

[1] David Dockterman. Insights from 200+ years of personalized learning. *npj Science of Learning*, 3(1):15, 2018.

[2] Gordon Ellis. Grand challenges for engineering. *IEEE Engineering Management Review*, 37(1):3–3, 2009.

[3] Atikah Shemshack and Jonathan Michael Spector. A systematic literature review of personalized learning terms. *Smart Learning Environments*, 7(1):33, 2020.

[4] Anna K Chmielewski. The global increase in the socioeconomic achievement gap, 1964 to 2015. *American sociological review*, 84(3):517–544, 2019.

[5] Burrhus Frederic Skinner. Teaching machines. In *Higher Education in the United States: The Economic Problems*, pages 189–191. Harvard University Press, 1960.

[6] James Basham, Tracey Hall, Richard Carter, Jr, and William Stahl. An operationalized understanding of personalized learning. *Journal of Special Education Technology*, 31, 08 2016.

[7] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *Adaptive hypertext and hypermedia*, pages 1–43, 1998.

[8] John Bourne and Janet C Moore. *Elements of quality online education: Practice and direction*, volume 4. Olin College-Sloan-C, 2003.

[9] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[10] Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

[11] Nerses Yuzbashyan, Nicolay Banar, Ilia Markov, and Walter Daelemans. An exploration of zero-shot natural language inference-based hate speech detection. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 1–9, 2023.

[12] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[13] Janis Goldzycher, Moritz Preisig, Chantal Amrhein, and Gerold Schneider. Evaluating the effectiveness of natural language inference for hate speech detection in languages with limited labeled data. *arXiv preprint arXiv:2306.03722*, 2023.

[14] Lasha Abzianidze, Joost Zwarts, and Yoad Winter. Spacenli: Evaluating the consistency of predicting inferences in space. *arXiv preprint arXiv:2307.02269*, 2023.

[15] Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

[16] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[17] Ilaria Caponetto, Jeffrey Earp, and Michela Ott. Gamification and education: A literature review. In *European conference on games based learning*, volume 1, page 50. Academic Conferences International Limited, 2014.

[18] Sujit Subhash and Elizabeth Cudney. Gamified learning in higher education: A systematic review of the literature. *Computers in Human Behavior*, 87, 05 2018.

[19] Yujia Hong, Nadira Saab, and Wilfried Admiraal. Approaches and game elements used to tailor digital gamification for learning: A systematic literature review. *Computers Education*, 212:105000, 01 2024.

[20] Crystal Yunus and Kim Hua Tan. Exploring a gamified learning tool in the esl classroom: The case of quizizz. *Journal of Education and e-Learning Research*, 8:103–108, 01 2021.

[21] Daniel Strmečki, Andrija Bernik, and Danijel Radosevic. Gamification in e-learning: Introducing gamified design elements into e-learning systems. *Journal of Computer Science*, 11:1108–1117, 12 2015.

[22] Ni Wulantari, Azhariah Rachman, Mike Sari, Lulu Uktolseja, Agus Ikip, Saraswati Tabanan, Jl Pahlawan, Delod Peken, Kec Tabanan, Kabupaten Tabanan, Universitas Oleo, Kampus Hijau, Bumi Tridharma, Kec Anduonohu, Kota Kambu, Sulawesi Kendari, Tenggara, Stkip Muhammadiyah, Sungai Penuh, and Jawa Majalengka. The role of gamification in english language teaching: A literature review. *Journal on Education*, 06:2847–2856, 01 2024.