

# AN ADAPTIVE AND INTERACTIVE EDUCATIONAL GAME PLATFORM FOR ENGLISH LEARNING ENHANCEMENT USING AI AND CHATBOT TECHNIQUES

Yichen Liu<sup>1</sup>, Jonathan Sahagun<sup>2</sup> and Yu Sun<sup>3</sup>

<sup>1</sup>Shen Wai International School, 29,  
Baishi 3rd Road Nanshan Shenzhen China 518053

<sup>2</sup>California State Polytechnic University, Los Angeles, CA, 91748

<sup>3</sup>California State Polytechnic University, Pomona, CA, 91768

## **ABSTRACT**

*As our world becomes more globalized, learning new languages will be an essential skill to communicate across countries and cultures and as a means to create better opportunities for oneself [4]. This holds especially true for the English language [5]. Since the rise of smartphones, there have been many apps created to teach new languages such as Babbel and Duolingo that have made learning new languages cheap and approachable by allowing users to practice briefly whenever they have a free moment for. This is where we believe those apps fail. These apps do not capture the interest or attention of the user's for long enough for them to meaningfully learn. Our approach is to make a video game that immerses our player in a world where they get to practice English verbally with NPCs and engage with them in scenarios they may encounter in the real world [6]. Our approach will include using chatbot AI to engage our users in realistic natural conversation while using speech to text technology such that our user will practice speaking English [7].*

## **KEYWORDS**

*Machine Learning, NLP, Data Mining, Game Development.*

## **1. INTRODUCTION**

In the modern age, though many claim there are more opportunities than ever before, there is still a massive disparity between the rich and the poor. This is especially due to the fact that such opportunities cannot be accessed by just anyone. Language is one of the major barriers, especially for English, the most used language both by country and by population, and it is seen by many as a door to unlocking knowledge and catching up to the rest of the world, given that most resources are either written in, or translated to, English. Students in poorer regions do not have an environment conducive to learning all aspects of English, often learning with teachers who can barely speak it themselves, and textbooks decades old. Many rural students end up being Receptively Bilingual (knowing the words and being able to read/write but can't speak), which is obviously unideal, as speaking is what gives first impressions [8]. Thus, it is absolutely critical to create tools pushing for the learning of English in a comprehensive manner, providing an environment that especially encourages speaking [9]. In this way, we can start to bridge the gaps between people of different social economic status with a simple language difference.

The education market is expansive, and given the proliferation of English, there have obviously been hundreds of thousands of companies around the world offering their own take [10]. There are general language apps, such as Duolingo, which focus on all four aspects (reading, writing, listening, speaking) of dozens of languages. However, they're usually aimed at adults (kids versions are all very lacking), and spread themselves too wide to be effective beyond basic conversational level. They also generally operate with a very automated learning style, such as mundane drag and drops or listening exercises. I have also looked at the larger market of English-only apps, especially for children. Yet, what we have found is that while most of them operate on the effective principle of "learn by play," they skew a little bit too much towards the play [11]. There are entire games focused around a single word, often a trivial one, such as "Carrot." Even when more learning-heavy exercises are present, it still focuses on vocabulary and listening, as it becomes a more mundane and responsive task, rather than an active one. It is important to take advantage of the young age of children to create a conducive learning environment, rather than setting them up for more work later on.

In our own method, we try to create an environment that incorporates learn by play but also focuses heavily on speaking. The effectiveness of gamified approaches to learning cannot be denied, but a balance must be met between fun and learning, as always. A good way to do this is by setting learning objectives to being the keys of certain checkpoints or to unlocking certain rewards, instead of, like in so many other apps, having them simply as a "added feature," for example, collecting words while riding a horse. There are many existing tools which can teach vocabulary and basic grammar, or even listening, and there indeed are apps which focus just on that. Thus, our goal is simply to establish a unique environment and differentiate from existing tools—though later on such features can also be added on for a comprehensive software. A focus on speaking aims to directly fight against Receptive Bilingualism, and with an AI able to respond in real time, children are able to develop confidence through real interaction, instead of simply responding [12]. When text to speech is incorporated, students are then given the opportunity to mimic the pronunciation and try it out, with the reward given when the computer deems it accurate, providing real time comparisons.

After we have completed the program design and released the prototype, we need to design some reasonable experiments to verify our hypothesis. In order to make the experiment more realistic and reliable, we sampled from different populations to ensure the stability of the experimental results [13]. Our two experimental designs are as follows :

- 1) In order to verify the stability of the process, we selected 30 different teachers and teacher assistants from 10 different regions. We ask teachers to record their standard pronunciation and ask teachers to test our English learning games with standard grammar and pronunciation. In the end, most teachers achieved high scores in the game. The accuracy of game prediction is as high as 93%.

- 2) In order to test the effectiveness of our games on English learning, we find 150 students from 20 different regions in China. They have different backgrounds and experiences in learning English. The final result shows that our software is the most effective for students who have been studying English for 1-3 years.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

## **2. CHALLENGES**

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

### **2.1. Language or cultural barrier**

When creating our first chat bot we quickly noticed how many words could be used interchangeably to carry the same meaning or a valid response to a question especially for new learners of a language who may not be familiar with colloquialisms and idioms of the language [14]. For example, colloquially in American English the biggest pizza is referred to as a “large pizza” but ordering a “huge pizza” isn’t grammatically incorrect. Another issue with particular word choices is catching terms and trying to interpret what they meant which can be caused due to a language or cultural barrier. Back to a pizza example, a pizza topping that may be common outside of the US could be considered very unusual in the US. In this case we would want the chatbot understand that the uncommon topping that is being order is a topping that isn’t offered and respond according, for example “Sorry we don’t off that as a topping” vs a more generic unhelpful response of “Sorry we don’t quite get that, may you repeat that.” The former will help the conversation move forward while the latter may stall the flow of the conversation and hinder the user’s learning.

### **2.2. Answering open ended questions**

Our goal is to teach conversational English by simulating real world scenarios. To accomplish thisgoal, we want to simulate a natural conversation. Doing so involves open ended questions, allowing users to answer how they feel is natural which can be difficult to account for. Unfortunately, we cannot fully predict how many new English learners will answer. The best remedy to solve this issue is to grow our user base and collect users’ responses and adjust our chatbot to accommodate how people respond by having more guided responses.

### **2.3. Delay of response**

As we are trying to gamify learning English, one of the main goals is to make the learning process fun and approachable. A big obstacle to fun is waiting. One issue we are facing is the delay between the user’s response and the computer’s response and can cause the user to wait [15]. We are relying on cloud services to handle our chatbot and speech to text. A few solutions we are looking forward to try out include better compression of our audio recording of the user, streaming the audio directly to our speech to text services vs wait sending a audio file after the user finishes speaking, and implementing a webhook that sends information directly to the other cloud services. We had to scrap our implementation of text to speech because the user’s input lag would have been so great it would have made our game practically unplayable.

## **3. SOLUTION**

We want to teach people English by making the learning fun by gamifying the process.

Introductory language textbooks use scenarios like ordering food and asking where the library is to immerse learners into scenarios, they may encounter in countries that speak that language. We are building upon that by using speak to text technology and chatbot artificial intelligence in a video game to further immerse learners and encourage them to practice English by speaking it. By gamifying English learning, we are hoping to build the habits that will motivate and engage with our users to keep practicing English every time the game is launched.

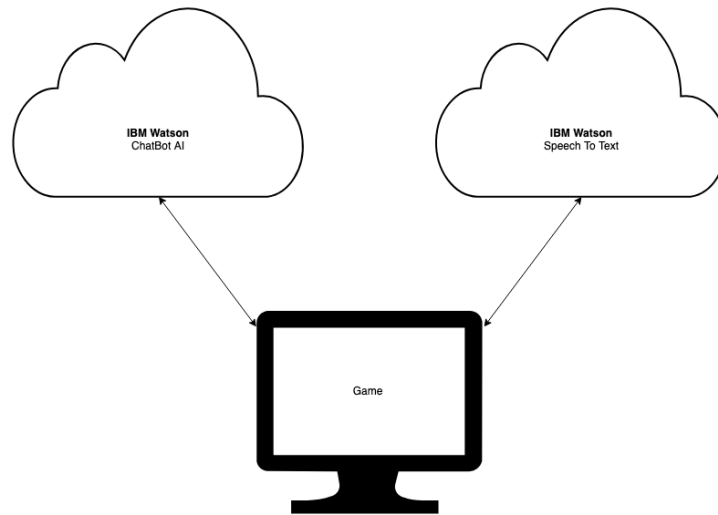


Figure 1. The overview of the project

Our game connects to two cloud services provided by IBM's Cloud services: IBM's Watson Speech to Text and Watson Assistant. We use Watson Speech to Text service to allow our users to practice speaking English and showing them how they pronounce their words encouraging our users to enunciate. The second service is Watson Assistant, which is IBM's chatbot solution. We use the chatbot to simulate real life situations to immerse our users in natural conversation.

The IBM Watson Speech to Text service transcribes audio to text to enable speech transcription capabilities for applications.

```

public void StartRecord()
{
    if (!MicrophoneDetected)
    {
        return;
    }
    audioSource.clip = Microphone.Start(null, true, 20, maxFreq);
    recording = true;
    RecordingReady = false;
}

public void StopRecording()
{
    if (recording)
    {
        Microphone.End(null);
        FilePath = SavWav.Save("recording.wav", audioSource.clip, Application.persistentDataPath);
        recording = false;
        RecordingReady = true;
    }
}
  
```

Figure 2. Code of recording

To use IBM's Speech to Text service we first need to have an audio file. Above is the code we use to record our user's voice. We are using Unity's Microphone to record the audio source. Unfortunately, it is in an uncommon audio file used by Unity so we encode it Waveform Audio File (WAV) format before saving to the user's computer. Once it's saved, we are ready to use the Speech to Text service.

```

private IEnumerator SpeechToText(){
    status = Status.Pending;
    WWWForm form = new WWWForm();
    string filePath = FilePath;
    print(filePath);
    byte[] bytes = File.ReadAllBytes(filePath);
    using UnityWebRequest www = UnityWebRequest.Post(url, form);
    www.SetRequestHeader("AUTHORIZATION", Authenticate("apikey", apikey));
    www.uploadHandler = (UploadHandler)new UploadHandlerRaw(bytes);
    www.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    www.SetRequestHeader("Content-Type", "audio/wav");
    yield return www.SendWebRequest();
    if (www.result != UnityWebRequest.Result.Success)
    {
        Debug.Log(www.error);
        Debug.Log(www.result.ToString());
        status = Status.Error;
    }
    else
    {
        string jsonResponse = www.downloadHandler.text;
        if (jsonResponse.Contains("\"results\": []"))
        {
            Trascript = "";
            Confidence = 0;
        }
        else
        {
            int indexOfTranscript = jsonResponse.IndexOf("transcript");
            indexOfTranscript += "transcript\": \".Length;
            print(indexOfTranscript);
            int endOfTranscript = jsonResponse.IndexOf("\",", indexOfTranscript);
            print(endOfTranscript);
            int lengthOfTranscript = endOfTranscript - indexOfTranscript;
            Trascript = jsonResponse.Substring(indexOfTranscript, lengthOfTranscript);

            int indexOfConfidence = jsonResponse.IndexOf("confidence", endOfTranscript);
            indexOfConfidence += "confidence\": \".Length;
            print(indexOfConfidence);
            int endOfConfidence = jsonResponse.IndexOf(",", indexOfConfidence);
            print(endOfConfidence);
            int lengthOfConfidence = endOfConfidence - indexOfConfidence;
            Confidence = double.Parse(jsonResponse.Substring(indexOfConfidence, lengthOfConfidence));
        }
        print(Confidence);
        status = Status.Success;
    }
}
}

```

Figure 3. Code of speech to text

Above is our code used to send the WAV file we saved of the user's recording to IBM. It is sent over using a HTTPS request. The request we sent contains the audio recording as a byte array and our API key for authentication. In return we expect a JSON response. There are two keys we are looking for in this JSON, the transcript of the audio recording the IBM's confidence level.

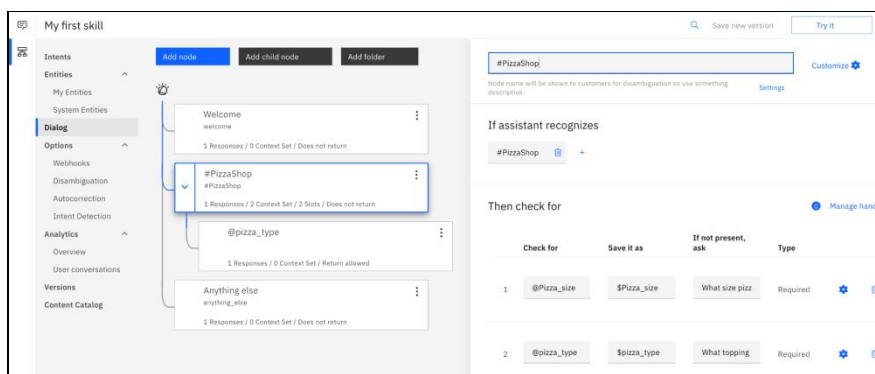


Figure 4. Screenshot of sample conversation

Above is a sample conversation we developed for a pizza shop. To key terms or entities, we check for are pizza size and pizza toppings. The user can ask for both for example “I would like to order a large cheese pizza” and that will satisfy our chatbot. In the case where the user only orders the size, our chat bot is able to adapt the conversion and ask the user what type of pizza they would like.

```
private IEnumerator Session()
{
    watsonStatusText = "waiting for session";
    service.CreateSession(OnCreateSession, assistantId);

    while (!createSessionTested) { yield return null; }

    watsonStatusText = "session created";
    connectionAlive = true;
    watsonStatusText = "Waiting for Watson";
    service.Message(WatsonsResponse, assistantId, sessionId);

    while (!waitingForUser) { yield return null; }

    while (connectionAlive)
    {
        watsonStatusText = "Waiting for User";

        while (waitingForUser) { yield return null; }

        var input = new MessageInput()
        {
            Text = UserMessage,
            Options = new MessageInputOptions()
            {
                ReturnContext = true
            }
        };

        if (service == null) { break; }
        if (connectionAlive == false) { break; }

        service.Message(WatsonsResponse, assistantId, sessionId, input: input);
        watsonStatusText = "Waiting for Watson";

        while (!waitingForUser) { yield return null; }
    }
}
```

Figure 5. Main code used to communicate with Watson

Here we have the main code used to communicate with Watson. The Session function creates a session to Watson Assistant where the assistantID variable is the specific scenario we want to connect to. Once the session has been established, Watson sends a message greeting the user starting the conversion for the user to follow. This function will loop indefinitely either waiting for the user’s response or Watson’s response to the user.

```
public bool SendMessageToWatson(string message)
{
    if (waitingForWatson)
    {
        return false;
    }
    else
    {
        UserMessage = message;
        waitingForUser = false;
        waitingForWatson = true;
        return true;
    }
}
```

Figure 6. Function of sending message

The function we created to send messages requires a string that is saved as an instance variable that will be read once the Session function is ready to use it. The message parameter of the SendMessageToWatson function in practice comes from the transcription of the user’s recording

from the Speech to Text service. Once the UserMessage variable is set, the waitingForUser and waitingForWatson boolean variables are set so the Session function can continue.

```
private void WatsonsResponse(DetailedResponse<MessageResponse> response, IBMError error)
{
    Log.Debug("Watson's Response: {0}", response.Result.Output.Generic[0].Text);
    WatsonResponse = response.Result.Output.Generic[0].Text;
    waitingForUser = true;
    waitingForWatson = false;
}
```

Figure 7. Code of Watson's response

When Watson is sent the user's response we wait for Watson's response. When Watson sends its response to the user that triggers a callback function in which we save Watson's response, used to display the response in game, and we again toggle the waitingForUser and waitingForWatson variables, setting the game state so that the user may respond.

Our biggest limitation we are facing is gathering enough users of varied diverse backgrounds to further program our scenarios to capture common errors and edge cases in how people respond. We plan to keep developing our chat bot to catch more edge cases and develop more scenarios to focus on specific topics such as creating a character who you can ask for directions and creating bus stops that players can use to navigate the world using the characters directions. The biggest technical limitation is the processing time from the speak to text process and then taking that text and having the chatbot process it. Since both processes are done in the cloud the user may experience a long delay from the time the user finishes his speech to the game's on screen response. We originally intended to have the text to speech response to answer the player but that would have delayed the computer's response even further which would make the game feel unplayable.

## 4. EXPERIMENT

### Experiment 1

We considered a lot of issues when designing experiment 1. In order to eliminate all possible influencing factors and to ensure the diversity of samples, we selected 30 different teachers and teacher assistants from 10 different regions in the US. 10 of them are teaching elementary school, 10 of them are teaching middle school, 10 of them are teaching high school. We ask teachers to record their standard pronunciation and ask teachers to test our English learning games with standard grammar and pronunciation. The result Table shows below:

|                           | Pass With High Score | Pass With Low Score | No Pass |
|---------------------------|----------------------|---------------------|---------|
| high school teacher       | 9                    | 1                   | 0       |
| middle school teacher     | 10                   | 0                   | 0       |
| elementary school teacher | 9                    | 1                   | 0       |

Figure 8. Table of data

From the data in the table, we can tell most teachers achieved high scores in the game. The accuracy of game prediction is as high as 93%. Base on the experiment 1 we can prove that users can pass the game if they have good English skill.

## Experiment 2

Before designing the second experiment, we considered how to design the experiment to show that our software is effective for English learning. First of all, the selected student sample should not come from a country where English is a native language. Secondly, in order to ensure the diversity of the samples, the distance between these students should be far enough. Finally, we chose China as the sample area for the experiment. We find 150 students from 20 different regions in China and divide them into 3 groups. Group1 is the students who have learned English 1 -3 years before the test. Group2 the students who have learned English 3 - 5years before the test. the students who have learned English more than 5 years before the test. They have different backgrounds and experiences in learning English. We ask them to download the game and try to learn English with it. After few weeks we ask them to finish the survey and collect the result, the graphic shows below:

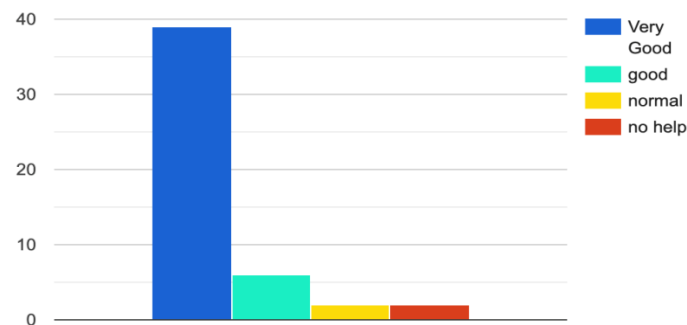


Figure 9. Result of 50 students from 1 - 3 years experience group

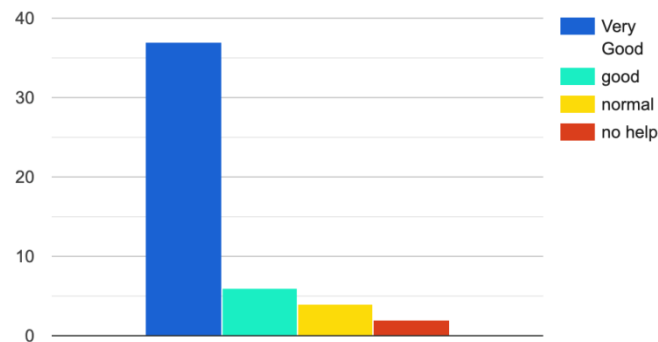


Figure 10. Result of 50 students from 3 - 5 years experience group

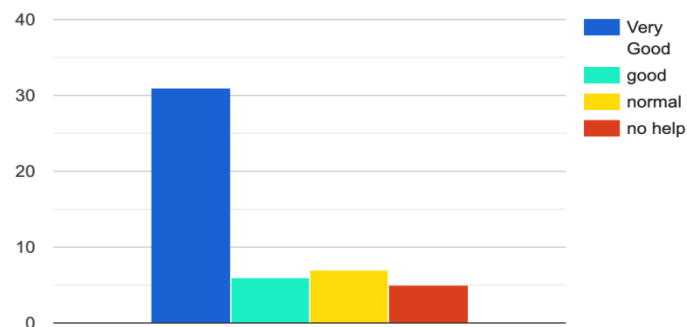


Figure 11. Result of 50 students from more than 5 years experience group



The final result shows that our software is the most effective for students who have been studying English for 1-3 years.

In order to eliminate all possible influencing factors and to ensure the diversity of samples, we selected 30 different teachers from 10 different regions in the US. The accuracy of game prediction is as high as 93%. We can prove that users can pass the game if they have good English skills.

in order to ensure the diversity of the samples, we find 150 students from 20 different regions in China divide them into 3 groups base on their English experiment, we prove that our software is the most effective for students who have been studying English for 1-3 years.

## **5. RELATED WORK**

Chen and Tsai proposed a game-based English learning system with a context-aware interactive learning mechanism, which can appropriately provide learners with corresponding game-based English learning scenarios based on the learner's geo location [1]. The proposed system aims to construct an augmented reality game-based learning environment that combines virtual objects with real scenes. This game does not involve voice-based interaction as what we have accomplished. In addition, this game is based on a hard-coded environment without AI-based conversations.

Wu and Huang developed a game-based system to help learners to improve their vocabulary [2]. Users can choose different settings, portfolio, and levels, in order to adjust the study mode and pace. This project focused on improving the motivation of vocabulary learning, while ours focuses on AI-based interaction and conversation to improve the practical aspect of English.

Hung and Young proposed a complete hardware device that provides an interactive English vocabulary learning board game [3]. A complete study has been done by collecting both quantitative and qualitative data. The results showed that game-embedded handheld devices could increase the interdependence of the group and help the learners to improve the engagement and immersion. Our system is independent from the devices, and the same application can be running cross platform in mobile, web, PC, and embedded environments.

## **6. CONCLUSIONS**

We created this game as a fun way to teach English with the main goal being to teach conversational English by simulating real world scenarios. Our approach was to use chatbot AI and speech to text technology by IBM to allow users to interact with NPCs in natural conversations by speaking with the NPCs. For example, the first scenario we created is a pizza parlor where the user orders a Pizza. The scene starts by walking up to the NPCs and interacting to start the conversation. The NPC starts the conversation by greeting the user and providing the user with info about what the scenario is about and what some appropriate responses are. The conversation continues by giving control to the user and allowing them to record their voice. Once the user is finished responding, their recording is sent to IBMs' a speech to text service and the output of that is sent to our chatbot which responds to the user. This is looped until the conversation and scenario reaches its end.

We sample 3 groups of 50 students from 20 different regions in China. The first group have 1 - 3 years of English experience, the second group have 3 - 5 years and the final group have 5 or more years. We had them take a proficiency test before testing our app for a control. After a week of

using our game, we had them take another proficiency test. The final result shows that our software is the most effective for students who have been studying English for 1-3 years.

To further enhance our research, we need more user's data to understand our shortcomings in our scenarios to see where we can improve our wording and guide our user's to appropriate responses. Without this data we can't not make meaningful changes to our conversations to make them feel more natural. We will be looking for users to help us test our chatbot and ways to make our game more fun and engaging.

To better immerse our users in the game we are looking to implement text to speech technology for the NPCs to respond. We also are looking into shortening the delay between the user's response to the NPC's. We believe we can achieve that by streaming the user's microphone to our speech to text service, then having a web hook redirect that outcome to our chatbot.

## REFERENCES

- [1] Chen, Chih-Ming, and Yen-Nung Tsai. "Interactive location-based game for supporting effective English learning." In 2009 International Conference on Environmental Science and Information Application Technology, vol. 3, pp. 523-526. IEEE, 2009.
- [2] Wu, Ting-Ting, and Yueh-Min Huang. "A mobile game-based English vocabulary practice system based on portfolio analysis." *Journal of Educational Technology & Society* 20, no. 2 (2017): 265-277.
- [3] Hung, Hui-Chun, and Shelley Shwu-Ching Young. "An investigation of game-embedded handheld devices to enhance English learning." *Journal of Educational Computing Research* 52, no. 4 (2015): 548-567.
- [4] Katzner, Kenneth, and Kirk Miller. *The languages of the world*. Routledge, 2002.
- [5] Barber, Charles, Joan C. Beal, and Philip A. Shaw. *The English language: A historical introduction*. Cambridge University Press, 2009.
- [6] Schnaars, Steven P. "How to develop and use scenarios." *Long range planning* 20.1 (1987): 105-114.
- [7] Dahiya, Menal. "A tool of conversation: Chatbot." *International Journal of Computer Sciences and Engineering* 5.5 (2017): 158-161.
- [8] Woods, Michael. *Rural*. Routledge, 2010.
- [9] Belsey, Catherine. *Critical practice*. Routledge, 2003.
- [10] White, Ellen Gould Harmon. *Education*. AB Publishing, 1903.
- [11] Ryu, Dongwan. "Play to learn, learn to play: Language learning through gaming culture." *ReCALL* 25.2 (2013): 286-301.
- [12] Cox, David R. "Interaction." *International Statistical Review/Revue Internationale de Statistique* (1984): 1-24.
- [13] Hahn, Frank. "Stability." *Handbook of mathematical economics* 2 (1982): 745-793.
- [14] Nevalainen, Sampo. "Colloquialisms in translated text. Double illusion?." *Across Languages and Cultures* 5.1 (2004): 67-88.
- [15] March, Wayne F. "Dealing with the delay." *Diabetes Technology & Therapeutics* 4.1 (2002): 49-50.