ENHANCING PUBLIC SPEAKING CONFIDENCE: AN AI-POWERED DEBATE PRACTICE APP WITH REAL-TIME FEEDBACK

Yutong Huo [1], Moddwyn Andaya [2]

[1] Sierra Canyon School, 11052 independence Ave Chatsworth, CA 91311
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This project introduces an AI-powered debate practice app designed to help users improve their public speaking and debate skills [9]. The app simulates real public forum debates by letting users input topics, choose roles, and engage with an AI opponent in various debate phases. It allows for flexible, on-demand practice and gives users instant feedback based on their choices [1]. To test the app's effectiveness, five users participated in a survey after using it. The results showed an average score of 8.0 in both preparedness and confidence, proving the app helped users feel more ready and self-assured. The app also stands out when compared to other public speaking methods, such as therapy, solo prep, or structured classes [2]. Unlike those, this app offers an interactive, real-time experience that helps users practice impromptu responses under pressure. Overall, this tool provides a practical and accessible way to build communication confidence and improve debate performance.

KEYWORDS

AI debate app, Speaking skills, Instant feedback, Confidence building

1. INTRODUCTION

I do public forum debate personally, and this debate is mainly about preparing, practicing and eventually performing. In public forum debate, having abundant practice and the quick improvisation are both very crucial to be a good debater [6]. However, the premise of playing these out is being calm and confident, and this is what most debaters struggle with. In a similar vein, Wan Mustapha, Ismail, Deepak & Elias (2010) investigated the level of anxiety in oral communication among the final year Bachelor of Business Administration learners in a Malaysian University. They discovered that 45% of the learners had a high level of anxiety towards oral communication. Moreover, it's found that about 77% of the general population fears public speaking [10]. Therefore, this project helps the debaters to be better prepared by practising with AI and gaining more sense of how to deal with impromptu issues in real debate.

In Section 5, I compared three different methods that aim to improve public speaking and reduce anxiety. The first study used structured classroom preparation to reduce test anxiety, but it relied heavily on human planning and didn't simulate real-time situations. The second article emphasized the importance of preparation and confidence but focused more on mindset and solo practice rather than interactive training. The third paper analyzed the physical and mental patterns of anxiety and suggested therapy and awareness but lacked an active way to practice speaking. All three had good intentions—improving confidence, reducing stress, and encouraging preparation—but none of them offered on-demand, realistic practice. My debate app improves on

David C. Wyld et al. (Eds): MLNLP, ASOFT, CSITY, NWCOM, SIGPRO, AIFZ, ITCCMA – 2025
pp. 113-121, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.152008

these by giving users a space to actually experience debates with AI, practice any section they want, and get used to impromptu speaking under pressure. It's flexible, interactive, and available anytime, making it more useful for real growth in communication skills.

I simulate a real debate using AI to help the debaters better practice before real competition. I decided to do this because I personally experienced the nervousness before debating in competitions and I want to make a project that helps the debaters to gain more practice and confidence. Users can choose the section that they are practising which is more flexible, and the users can target the section they want to practice directly. Compared to debate training classes, this app saves a lot more time and expenses, since the users can use it anytime, anywhere, and have more training without a particular limit of training time. Users will also have a better user experience by using the app instead of other methods. Unlike human being, the AI has endless ideas as the user's opponent. This enables the user to exploit more opinions that may exist in real debate competitions [7].

In the experiment section, I tested my AI-powered debate app with five users to see if it improved their confidence and preparedness in public speaking or debate. After using the app, each person answered two questions—one on how prepared they felt and the other on their confidence—using a 1-to-10 scale. The results showed an average score of 8.0 for both questions, meaning the app was effective in helping users feel more ready and confident. The highest score of 10 appeared more often in the confidence question, suggesting the app had a stronger emotional impact there. Even the lowest score was a 6, which still reflects a positive experience. Overall, the test showed that my app is helpful for people looking to practice and improve their speaking skills. It also confirmed that giving users a chance to simulate real debates through AI can boost both their confidence and performance.

2. CHALLENGES

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

2.1. AI Response Trigger Explanation

Skeptical Question: Why isn't the AI responding after I speak?

Response: The AI only responds once you hit the "set" button. When you speak, the app doesn't know yet if it's your turn to go again or if it's the AI's turn. Once you press "set," it tells the app that your side and speaker role are locked in, and that's when the AI knows to jump in and respond. Without pressing it, the system doesn't have all the context it needs, so it just waits for you to confirm your side and turn before reacting.

2.2. App Uses Beyond Debates

Skeptical Question: What else can this app be used for, besides debates?

Response: You could totally use it for public speaking and presentations too! It's a great tool for practicing how you communicate and getting instant feedback. Instead of just focusing on debates, you could set it up to help you gather opinions or practice your speech in front of an audience. It's a versatile way to refine how you present ideas, whether you're speaking by yourself or with others, and you can even use it to improve your argumentation and response skills in real-time.

2.3. Selecting Debate Sections to Practice

Skeptical Question: How do we pick different sections to practice?

Response: Just click on the phase button and choose the section you want to practice. It's super easy—each phase is listed there, so you can focus on whichever part of the debate you need to work on, whether it's your opening statement, rebuttal, or closing. It gives you the flexibility to practice what's most important to you at any time.

3. SOLUTION

When first loading up the app, you will first encounter the setup page to get the public forum debate started. The setup menu first contains a username input field so the other members of the debate can refer to you as such. Right under it are two input fields where you will type the pro and con topics that will be debated over the session. And finally a dropdown field to choose if you want to be a pro or con side and another field to choose whether you'd like to be a first or second speaker during the debate.Both dropdown fields also allows you to choose a random choice. Once you are done with the setup, you are now then displayed on the main debate chat or the main gameplay screen. In this screen you have full control of the debate session and practice as many phases of the debate process. First you have the main chat right in the middle of the screen where you can type your response or inputs relative to the current phase. The top bar of the screen where you can control each part of the debate process such as the phase itself, the side (pro or con) that is currently speaking, and the speaker (first or second). Depending on which side and speaker, the chat will activate or deactivate whether it is your turn or not. Overall for visual appeal, each speaker is assigned a cartoon character at the bottom of the chat to indicate individually who's talking.

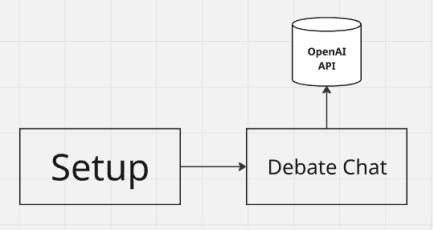

Figure 1. Overview of the solution

The first and most important component of this app is the AI Manager. This manager script handles the connection between the app and the OpenAI API [8]. The API itself talks to the OpenAI's server itself and handles everything AI related that we are doing. For this project we are only concerned about the 'chatting' part of AI where we write a prompt and have an AI output based on our input.

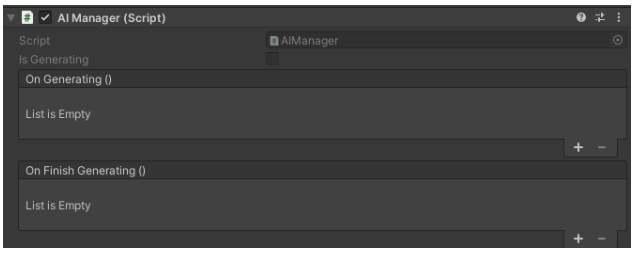

Figure 2. Screenshot of AI manager

```
public async Task SendChat(string input, List<Message> messages)
{
    if (isGenerating)
    {
        Debug.LogWarning("Cannot send chat while generating...");
        return;
    }
    if (string.IsNullOrEmpty(input) || messages == null || messages.Count == 0)
    {
        Debug.LogError("Invalid input or messages provided.");
        return;
    }

    isGenerating = true;
    OnGenerating?.Invoke();

    // Add user message
    messages.Add(new Message(Role.User, input));

    // Create request
    ChatRequest request = new ChatRequest(messages, model: Model.GPT4o);

    // Send request and get AI response
    ChatResponse response;
    try
    {
        response = await client.ChatEndpoint.GetCompletionAsync(request);
    }
    catch (System.Exception ex)
    {
        Debug.LogError($"Error in AI generation: {ex.Message}");
        isGenerating = false;
        OnFinishGenerating?.Invoke();
        return;
    }

    // Extract AI response text
    string fullResponse = response.FirstChoice;

    // Simulate typing effect and wait for it to complete
    await RunTypingCoroutine(fullResponse, messages);
}
```

Figure 3. Screenshot of code 1

The main function of the AIManager is the SendChat function that has a parameter string input that is the user's input message to the AI and a list of Messages which is what you would call the 'history' of the chat that the AI use for context when messaging so it can keep track of who said what. The Message class simply stores the Content which is basically what the chat says (like a simple string message) and the Role which is who sent this content (e.g User, Assistant). This function is an public async function that allows other scripts to await for the AI's response before continuing code. When the function runs, we first add the user's input straight into the list of messages so the AI has context of what the user said. Then we make a 'ChatRequest' which acts like the 'settings' of the chat session containing the data about what AI model we should use, how many 'tokens' should a response be, and what the chat's messages should be (and this is where we pass in the list of messages) [11]. We then call upon the api client itself, access the chat (ChatEndpoint), and await for a response back from the AI (GetCompletionAsync) passing in the ChatRequest we made. After getting the AI's output, we store it into a variable using response.FirstChoice.

During the gameplay itself, we have mentioned that there is a top bar on the debate chat screen that allows you to take control of the entire debate itself [12]. These controls are simple and change the basic variables within our code. Those variables are then passed into the AI where it will do the rest for us.

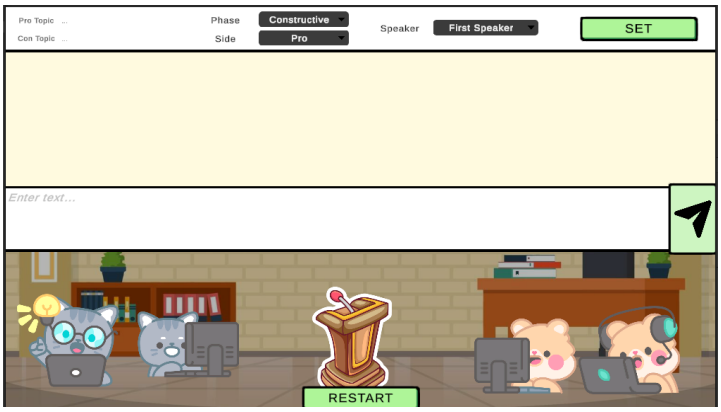

Figure 4. Screenshot of the game 1

```
void Start()
{
    ai = GetComponent<AIManager>();
    setupPanel.SetActive(true);

    ai.OnGenerating.AddListener(() => SetInteractable(false));
    ai.OnFinishGenerating.AddListener(() => SetInteractable(true));
    ai.NewMessageChunkReceived += (chunk) =>
    {
        chatBoxField.text += chunk;
    };

    sideDropdown.onValueChanged.AddListener((value) =>
    {
        switch (value)
        {
            case 0: currentSide = Side.Pro; break;
            case 1: currentSide = Side.Con; break;
        }
    });

    speakerDropdown.onValueChanged.AddListener((value) =>
    {
        switch (value)
        {
            case 0: currentSpeaker = Speaker.First; break;
            case 1: currentSpeaker = Speaker.Second; break;
        }
    });

    doneSetupButton.onClick.AddListener(OnDoneSetup);
    setButton.onClick.AddListener(() => OnSet());
    sendButton.onClick.AddListener(OnSend);

    startingProFirst = proFirstCharacter.position;
    startingProSecond = proSecondCharacter.position;
    startingConFirst = conFirstCharacter.position;
    startingConSecond = conSecondCharacter.position;
}
```

Figure 5. Screenshot of code 2

In this code, we will look at the main brain script of the entire app. At the start function of the script, we will handle what happens when the controls on the top bar of the debate chat is changed. We first handle anything on certain AI events. The AIManager script has OnGenerating and OnFinishGenerating Action events that we will take advantage of. We simply will use this to turn the chat box interactable or not so the user doesn't accidentally send any chats to the AI while it's already generating something. There is another Action event that outputs the new messages from the AI which we simply add to the current chatbox to see [13].

The other controls on the top bar are the side (pro or con) and speaker (first or second) dropdown fields that simply set an enum on the script that will be used later on another component. After the field's are set, we simply add a functionality to each button like the done setup button (for the setup screen), the set button (that finalizes your choices on the the top bar) which we will look at

in the next component, and the send button which is simply the button to send your input to the chat box.

One of the most important functions of the app is actually now telling the AI information about our debate so it can make such actions. The AI itself has been given specific instructions to respond to certain commands and say certain things based on its parameters. This is where parts from the second component goes.

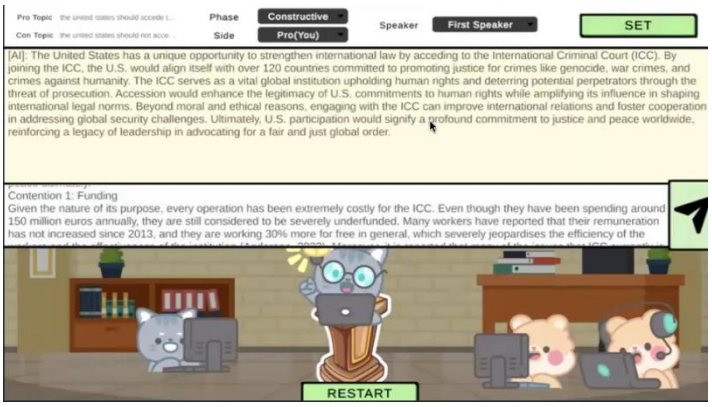

Figure 6. Screenshot of the game 2

```
1 reference
async Task OnSet()
{
    string sender = GetSender();
    string command = $"[command] phase {GetDropdownText(phaseDropdown)} {sender} {currentSide} {currentSpeaker}";
    messages.Add(new Message(Role.User, command));
    await ai.SendChat(command, messages);

    chatBoxField.text += "\n\n";
}

1 reference
void OnSend()
{
    string finalText = $"[You]: {userInputField.text}";
    chatBoxField.text += finalText + "\n\n";

    messages.Add(new Message(Role.User, finalText));

    userInputField.text = "";
}
```

Figure 7. Screenshot of code 3

When we have pressed the set button on the top bar, this basically tells the AI information for the debate. First we get the sender from the GetSender() function which simply compares the current speaker and side to our chosen speaker and side from the setup, if it equals ours, then we are the speaker, otherwise it's the AI currently sending. The AI can either be our teammate or any speaker of the opposing team and that's where the choices from the top bar dropdowns come in. When the set button is pressed, the current side and speaker chosen from those dropdowns is passed into this 'command' as a parameter. The AI instruction has a command that basically feeds information into the AI such as the current phase, the sender, side, and speaker. This tells the AI three things, should it talk or not (if the sender is user, then don't speak and just retain the information right now), which side and turn is speaking, and what should it say based on the current phase of the debate. This command is then passed in as an input to the AI where we then get the AI's decision of letting the user speak or typing something.

4. EXPERIMENT

4.1. Experiment 1

I want to test how well my debate practice app fits with a public speaker, someone trying or doing debate, or the general audience trying to get into both by seeing how confident and prepared users feel after using it. This experiment is important because it helps show whether the app is actually helpful for people who want to improve their public speaking or debating skills.

I planned to deploy a short questionnaire after each user had used the app, a few questions specifically structured to gauge whether the users felt more prepared and confident about public speaking or debating after using the app. Then, each question was rated on a 1-to-10 scale. I had five people test the app and then took the survey right after based on their experience. The aim was to assess the effectiveness of the app in enhancing users' sense of readiness and confidence. The results were tabulated and summarized for further analysis and evaluation of the responses.

Name	Preparedness (Q1)	Coinfidence (Q2)
Emily	8	9
Marcus	6	7
Sarah	9	8
Jason	7	6
Natalie	10	10

Figure 8. Table of experiment 1

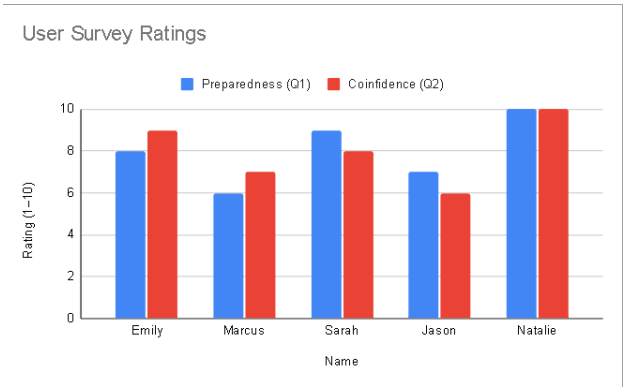

Figure 9. Figure of experiment 1

For Question 1 (preparedness), the average score was 8.0, while for Question 2 (confidence), the average was 8.0 as well. To calculate the overall average across all responses, we added all scores together (80 total points from 10 answers) and divided by 10, giving an overall average score of 8.0. Both questions had the same overall response average, which shows that users found the app equally helpful for improving both preparedness and confidence. That said, the highest individual response (a perfect 10) appeared more in Question 2, indicating a slightly stronger emotional impact on confidence for some users. On the flip side, the lowest score in either question was a 6, which still reflects a positive experience overall. There were no significantly weak responses, but confidence had slightly more variation, possibly because public speaking confidence can be harder to build than knowledge or structure, even with practice.

4.2. Experiment 2

I do public forum debate personally, and this debate is mainly about preparing, practicing and eventually performing. In public forum debate, having abundant practice and the quick improvisation are both very crucial to be a good debater. However, the premise of playing these out is being calm and confident, and this is what most debaters struggle with. In a similar vein, Wan Mustapha, Ismail, Deepak & Elias (2010) investigated the level of anxiety in oral communication among the final year Bachelor of Business Administration learners in a Malaysian University. They discovered that 45% of the learners had a high level of anxiety towards oral communication. Moreover, it's found that about 77% of the general population fears public speaking. Therefore, this project helps the debaters to be better prepared by practising with AI and gaining more sense of how to deal with impromptu issues in real debate.

I simulate a real debate using AI to help the debaters better practice before real competition. I decided to do this because I personally experienced the nervousness before debating in competitions and I want to make a project that helps the debaters to gain more practice and confidence. Users can choose the section that they are practising which is more flexible, and the users can target the section they want to practice directly. Compared to debate training classes, this app saves a lot more time and expenses, since the users can use it anytime, anywhere, and have more training without a particular limit of training time. Users will also have a better user experience by using the app instead of other methods. Unlike human being, the AI has endless ideas as the user's opponent. This enables the user to exploit more opinions that may exist in real debate competitions.

5. RELATED WORK

The paper "The Effect of Study Preparation on Test Anxiety and Performance" by Yusefzadeh et al. explores how scheduled study routines reduce anxiety and improve academic results [5]. Their system worked by organizing class activities like presentations, summaries, and Q&A to create regular engagement. Compared to my AI-powered debate app, their method relies on human structure and scheduled events, while mine gives on-demand practice with AI anytime. Unlike their method, my system lets users face spontaneous, debate-style situations with instant feedback, making it more flexible and realistic for practicing confidence. I think my app is better for real-time decision-making training.

The article "How Can I Get Better at Presenting?" by David Novak shares five tips for public speaking, focusing on preparation and confidence [4]. One key point is that 90% of stage anxiety comes from not preparing enough. Novak emphasizes practicing and knowing your content as the main way to reduce nerves. Compared to my AI debate app, his approach is more about solo prep and mindset, while mine creates a live, back-and-forth debate experience to simulate pressure. My app gives users real-time feedback and unpredictable prompts—something static prep can't offer. That makes my tool better for building quick-thinking skills.

"Scared Speechless" by Nancy Bartosek highlights Professor Ralph Behnke's research on public speaking anxiety [3]. He found anxiety peaks psychologically when people are assigned to speak, and physically when they actually begin. His work aims to reduce anxiety through therapy and understanding anxiety patterns. Compared to my AI debate app, which helps users actively practice debate situations, Behnke's research focuses more on awareness and passive strategies. My app provides repeated real-time experiences, letting users gradually build confidence by doing—not just learning about the fear. That's why I think my system gives a more hands-on, practical way to overcome speaking anxiety.

6. CONCLUSIONS

One thing I think could be better is the animation and scene transitions, especially when the speakers deliver their speeches [14]. Right now, the transitions between debate phases feel a bit stiff, and the animations for the speakers aren't as dynamic as I'd like. It doesn't really capture the energy of the debate, and I think that could be improved. To fix this, I could work on making the animations smoother and adding more character expression to match the tone of each speech. Another thing I'd like to improve is the app's responsiveness. Sometimes the AI takes a little longer to respond, especially when it's dealing with more complex topics. Speeding up the response times would definitely make the experience feel more seamless.

Finally, I think there's room for more customization in the app. Right now, everything looks pretty standard, so giving users the option to personalize their debate experience with different character designs or backgrounds could make it feel more unique to each person [15]. These improvements would help make the app more fun and engaging for everyone, plus they'd give users more control over how they interact with the debate.

REFERENCES

[1] Suliman, Wafika A., and Jehad Halabi. "Critical thinking, self-esteem, and state anxiety of nursing students." Nurse education today 27.2 (2007): 162-168.

[2] Kelly, Lynne, and James A. Keaten. "Treating communication anxiety: Implications of the communibiological paradigm." Communication education 49.1 (2000): 45-57.

[3] Lall, Hrithik, Bipranil Tapader, and Samapika Das Biswas. "Speech anxiety: Improper communication & fear of public speaking." International Journal of English Learning & Teaching Skills 1.4 (2019): 352-356.

[4] Yalçın, Alemdar, and Nursel Yalçın. "How to get best result from a presentation? How to increase effectiveness of a presentation?." Procedia-Social and Behavioral Sciences 9 (2010): 480-486.

[5] Salem, Ashraf Atta. "A Sage on a Stage, to Express and Impress: TED Talks for Improving Oral Presentation Skills, Vocabulary Retention and Its Impact on Reducing Speaking Anxiety in ESP Settings." English Language Teaching 12.6 (2019): 146-160.

[6] Boza, María. "The Great Debaters? Well, Close Enough." Fostering Habits of Mind in Today's Students. Routledge, 2023. 103-110.

[7] Schwardmann, Peter, Egon Tripodi, and Joël J. Van der Weele. "Self-persuasion: Evidence from field experiments at international debating competitions." American Economic Review 112.4 (2022): 1118-1146.

[8] Auger, Tom, and Emma Saroyan. "Overview of the OpenAI APIs." Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next. js. Berkeley, CA: Apress, 2024. 87-116.

[9] Parvis, Leo F. "The importance of communication and public-speaking skills." Journal of Environmental Health 63.9 (2001): 44-44.

[10] Menzel, Kent E., and Lori J. Carrell. "The relationship between preparation and performance in public speaking." Communication Education 43.1 (1994): 17-26.

[11] Gozalo-Brizuela, Roberto, and Eduardo C. Garrido-Merchan. "ChatGPT is not all you need. A State of the Art Review of large Generative AI models." arXiv preprint arXiv:2301.04655 (2023).

[12] Tumposky, Nancy Rennau. "The debate debate." The Clearing House: A Journal of Educational Strategies, Issues and Ideas 78.2 (2004): 52-56.

[13] Bhalerao, Harshali Rahul, and Ashutosh Mathur. "Awareness and Perception of Chatbots/Chatbox Amongst Customers." ECS Transactions 107.1 (2022): 1781.

[14] Dragicevic, Pierre, et al. "Temporal distortion for animated transitions." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2011.

[15] Jerome, Lee, and Bhavini Algarra. "Debating debating: a reflection on the place of debate within secondary schools." The curriculum journal 16.4 (2005): 493-508.