

# 仮想会議における参加者発話ログと生成 AI を活用した議論の質向上に関する研究

阿比留幹大<sup>1</sup> 鶴田直之<sup>1</sup>

**概要:** 本研究では、三次元仮想空間を利用した仮想会議において、参加者の発話ログと生成 AI を利用して議論の活性化と参加者の理解度の向上を図る機能を提案する。具体的には、参加者各自がマイク付きのパソコンを利用していることを利用して、参加者の発言のログを取得し、これを生成 AI を使用して分析することにより、会議中の疑問の即時解消機能による議論の停滞防止と、同時発話により聞き取れなかった場合の容易な確認機能および議論の即時要約機能による参加者の理解度向上を図った。実験では、会議の沈黙の時間の減少に成功し議論の促進が行われ、より明確に会議の理解度が向上したことが確認できた。

**キーワード:** 仮想会議, 発話ログ, 要約, 生成 AI

## Research on Enhancing the Quality of Discussions in Virtual Meetings Using Participant Speech Logs and Generative AI

KANTA ABIRU<sup>†1</sup> NAOYUKI TSURUTA<sup>†1</sup>

**Abstract:** This study proposes a system to enhance discussions and improve participant comprehension in virtual meetings conducted within a three-dimensional virtual space by utilizing participant speech logs and generative AI. Specifically, by leveraging the fact that each participant uses a microphone-equipped computer, speech logs are collected and analyzed using generative AI. This enables the prevention of discussion stagnation through an immediate clarification function for resolving questions during meetings, an easy confirmation function for cases where simultaneous speech makes understanding difficult, and a real-time summarization function to improve participant comprehension. Experiments demonstrated a reduction in silent periods during meetings, facilitating discussions and leading to a clearer improvement in meeting comprehension.

**Keywords:** virtual meeting, speech log, summarisation, generative AI

### 1. 研究背景と目的

昨今新型コロナウイルス蔓延による在宅勤務の増加やリモートワーク推進といった働き方改革が今なお進み浸透しつつある。この働き方改革によりオンライン会議の機会が増加し、Zoom や Microsoft Teams といったアプリケーションが急速に普及している。ただし、これらのツールは時間や場所に縛られない利便性を提供する一方で、従来の対面会議に比べて以下のような課題が顕在化している。

- 臨場感・没入感の欠如
- 対面にはない心理的違和感
- 議論の質や理解度の低下

これらの課題に対する新たな解決策として、筆者らの先行研究では現実に近い環境を再現した三次元仮想空間を活用した会議システムを開発した[1][2][3]。この会議システムでは従来のオンライン会議が抱える課題を解決するために設計されており、以下のような機能を有している。

1. 参加者のアバターの生成機能
2. アバターの移動及び位置同期

3. スライドや動画の共有機能 (図 1 参照)
4. フォトリアルな会議室のモデリング (図 2 参照)
5. VR 機器での会議参加及びVR 用アバターの生成



図 1 先行研究のスライド機能



図 2 先行研究のフォトリアルな会議室

<sup>1</sup> 福岡大学  
<sup>†1</sup> Fukuoka University

しかし、先行研究ではプレゼンテーション形式に重きを置いてシステムを開発したため、これは発表者が情報を伝達し、聴衆が理解することに関しては効果的であったが、双方向的かつ積極的な意見交換が行われるディスカッションの形式ではまだまだ課題が残っているのが現状である。具体的には、多数の参加者が双方向的かつ積極的に議論し、意見を出し合いながら結論を導き出す場面では、発言の偏りや議論の停滞といった課題があり、対面を含む会議にもともとあった課題も顕在化した。特に、参加者全員が均等に意見を共有し、それらを整理しながら意思決定を進める環境において、議論の透明性や活性化が不十分であるという限界が見られた。

そこで、本研究は、双方向的かつ積極的な意見交換が行われる会議で参加者の議論をより活性化し、会議の質を向上させ対面会議に勝るとも劣らないシステムを開発することを目的とする。具体的には、発話ログの自動取得・可視化機能や生成 AI による議論の整理・支援機能を導入することで、参加者全員が主体的に会議へ関与できるシステムを実現し、これまでの限界を克服することを目指す。

なお、本研究では、双方向的かつ積極的な意見交換が行われる会議を質の高い会議と定義する。

## 2. 基本技術

本研究で用いた開発環境とツールについて述べる。

### ● Unity[8]

Unity は、リアルタイム 3D 開発プラットフォームであり、ゲーム開発だけでなく、シミュレーション、VR/AR、インタラクティブなアプリケーションの開発にも広く利用されている。本研究では、三次元仮想空間を構築するための基盤として Unity を採用した。また、音声認識には Unity に付随する DictationRecognizer class を使用した。

### ● Photon[4]

Photon は、マルチプレイヤー対応のネットワークソリューションであり、リアルタイム通信を容易に実現するための SDK を提供している。本研究では、仮想会議システムにおける参加者間の同期処理や、リアルタイムなデータ共有を実現するために Photon を採用した。Photon Unity Networking 2 (PUN2) を用いることで、アバターの移動や会話内容の同期など、仮想空間内での多人数同時接続を効率的に実装できた。

### ● RPC 通信[5]

Remote Procedure Call (RPC) は、分散システムにおける通信技術の一つで、異なる端末間での同一関数の呼び出しを可能にする。本研究では、Photon の RPC 通信機能を活用し、クライアント間でのデータ共有を実現した。

### ● Photon Voice 2[6] [7]

Photon に付随した、ボイスチャットを簡単に追加することができる SDK である。遅延の小さな音声対話を実現できるとされている。

### ● ChatGPT (gpt-3.5-turbo)

ChatGPT は、OpenAI が開発した生成 AI であり、高度な自然言語処理能力を持っている。本研究では、ChatGPT を仮想会議システムに統合し、以下の機能を実現した

- **リアルタイム要約:** 取得した発話ログファイルから会議内容を自動で要約し、参加者が進行状況を把握しやすくする。
- **議論支援アドバイス:** 取得した発話ログファイルから会議が停滞した際に、次の議題や方向性を提案する。

## 3. 提案手法

### 3.1 システム概要

本システムに追加する機能として下記の 3 つを追加した。

1. リアルタイムで発言を記録する発話ログ記録機能
2. 1 で記録した発話ログファイルをもとに要約を作成する機能
3. 1 で記録した発話ログファイルをもとに会議の進行や内容に関するアドバイスを行う機能

本機能は、オンライン会議や仮想空間会議における発言の被りがあった際に、後から見返すことや、のちの要約や会議に関するアドバイスを作成するために実装した。

発言を記録する際に、発言者の情報も一緒に記録することで、発言者の区別が容易に実現できる。これは従来の対面会議ではボイスレコーダーなどを使ったとしても発言者の特定などが大変だといった混信の問題も解決でき、大きなメリットになると考えられる。また、これによってアドバイス機能では会議全体だけでなく、個人に対してのアドバイスも行うことができ、これもメリットになると考えられる。

また、書記の機能も担っていることから、記録ミスの低減や記録者の負担を軽減し、会議への集中度を高め、会議の質の向上も期待できる。

参加者が発言した内容をリアルタイムで他のユーザーと共有することで、会議中の透明性を高め、参加者間の理解を促進し発言内容が各クライアントに記録されるため、議論内容を後から確認することが可能になる。

本機能は、対面会議やオンライン会議でも考えられる課題である、参加者による会議の理解度のばらつきを、重要な結論や論点を即座に確認可能にすることで減らし、会議の質を向上させるために実装した。

### 3.2 システムの利用の流れ

これらの機能を用いた会議の進行について述べる (図

3). 会議中に発生する3つの場面: 生じる発言の聞き逃し, 会議中の参加者ごとの理解度のばらつき, 会議中の沈黙・停滞, に対処するためにシステムを次のように利用する.

1. まず複数人の参加者が3次元仮想会議空間に入室し, 議論や意見交換を行う. その際, 参加者全員の発言した内容は会議参加者全員が持っている発話ログファイルに保存されていく.
2. 同時発話などによる発言の聞き逃しが生じた際に, 参加者自身のローカルにある発話ログファイルを見返して聞き逃しを防ぐ.
3. 会議中にアイデアが出ない場合や, 会議の進行が止まってしまう事態になった場合, 操作している画面内のUIボタンを押すことで, 要約やアドバイスの生成がリアルタイムで行うことができ, 会議中の停滞の解決や会議の要点を即座に確認可能にし, 参加者ごとの理解度のばらつきを減少させる.



図3 システムの利用の概念図

### 3.3 システムの動作の流れ

次に, 本システムの内部の動作の流れを説明する (図4).

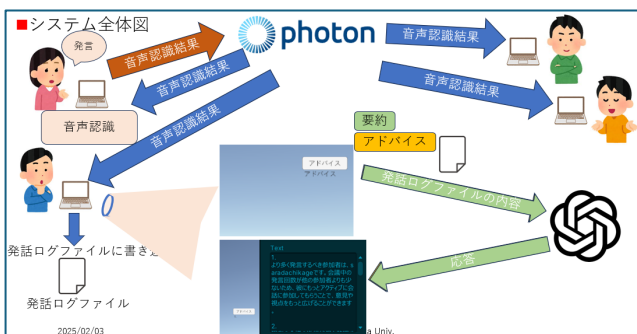


図4 システムの動作の流れ

1. 仮想会議空間で参加者が発言を行うと, 発言した参加者自身のローカルで音声認識の処理を行い, RPC通信機能を使用して全参加者に発言者と発言内容の情報をブロードキャストする.

2. 音声認識結果 (発話内容) が送られてきたら, 参加者各々のローカルにある発話ログファイル (以下 TextLog) に認識結果を記録する.
3. 要約やアドバイスを生成する UI ボタンが押されると, TextLog の中身を読み込み, プロンプトとして ChatGPT に送信する.
4. ChatGPT からの応答は操作画面の UI に格納されて表示される.

### 3.4 機能実装の詳細

音声認識機能は, Unity の DictationRecognizer クラスを用いて, Windows プラットフォーム上で音声認識を実現した.

音声認識結果をローカルファイルに記録するログ管理機能では, スレッドセーフなデータ構造である ConcurrentQueue を利用し, ログエントリを蓄積した. Unity のコルーチン機能を用いて, キューに保存されたデータを非同期でファイルに書き込む仕組みを実現している. なお, 書き込みエラー時に最大3回まで再試行することで, 信頼性の高い記録を保証している.

RPC 通信を用いた認識結果の共有では, 音声認識の初期化, 音声認識の開始と停止, 再起動, 音声認識結果の処理を行っている.

リアルタイム要約機能では, ChatGPT API を利用して会話履歴を管理し, プロンプトを API に送信し, 返答を保持する. 文脈に基づいた応答を生成するため, 保持する会話履歴は role (役割) と content (内容) の2つの属性を持った json 形式で累積している. role は "system" (指示文), "user" (ユーザーからの入力), "assistant" (ChatGPT の応答) のいずれかを指定し, content は実際のテキストデータを格納する. json の解析には JsonUtility.FromJson を使用した.

リアルタイム・アドバイス機能は, TextLog をもとに, 発言者ごとの発言内容を解析, ChatGPT へのプロンプト作成, 生成されたアドバイスの表示に至る一連のプロセスから成る. 発言者と発言内容のペア (タプル) をリスト形式で作成し, 発言者毎の発言回数も記録する.

## 4. 実験評価

### 4.1 実験・評価方法

以下の手順で実験を行う.

1. 被験者2~4名の1グループで仮想会議空間に入室し議論を行う.
2. 会議室に入室後, 議題を提示し10分間実装した機能は使用せずに会議を行う.
3. 議題による実験結果への影響を少なくするために議題は統一

4. 10 分後、発話ログファイルで過去の会話の確認や要約とアドバイスを生成して読む時間を取り、議論を再開する。
5. 議論再開後、発話ログファイルを見ながら議論を行い、参加者の任意のタイミングで要約やアドバイスを生成してもらう。なお、要約やアドバイスの生成は何回行ってもいいこととする
6. 結論が出た時点で終了とする
7. 会議終了後、事後アンケートで評価を行う。

評価のポイントは以下の通りである。

- リアルタイムでの発話ログファイルの確認による発言の聞き逃しの減少
- 要約による参加者ごとの理解度のばらつきの減少
- アドバイスによる会議の停滞が減少

本実験では、被験者のこれまでの対面会議やオンライン会議の経験に基づく体感で評価を行ってもらう。

#### 4.2 実験結果

被験者は 10 人の協力で合計 4 回の会議を実施した。

- 会議全体の時間は平均 18 分 56 秒(機能を使用しなかった 10 分間も含む)
  - 同時発話などによる発言の聞き逃しの回数は平均 1.0 回
  - 最初の生成を除いて実験中に要約やアドバイスを生成した回数は平均 3.5 回
  - 要約・アドバイスの生成前の議論の停滞の回数は平均 5.3 回
  - 要約・アドバイスの生成後の議論の停滞の回数は平均 0.0 回
- ※議論の停滞は 10 秒以上の沈黙があったときに 1 回とカウントする。

実験後のアンケートの結果を以下に示す。(1: 全くそう思わない ~ 10: 非常にそう思う)

1-1.議事録ファイルをリアルタイムで閲覧できたことが、会議の進行に役立ったと感じましたか？

平均 9.1 点

1-2.他の参加者の発言が正確に記録されていると感じましたか？

平均 7.7 点

1-3.議事録ファイルをリアルタイムで閲覧できたことで会議の理解度は向上しましたか？

平均 8.7 点

1-4.議事録ファイルをリアルタイムで閲覧し会議の流れを確認することで、会議への積極的な参加ができましたか？

平均 8.6 点

1-5.発言の聞き逃しがあった際に議事録ファイルをリアルタイムで確認できたことは役に立ちましたか？

平均 8.9 点

2-1.リアルタイム要約機能による要約の内容は正確だと思いましたか？

平均 9.3 点

2-2.リアルタイム要約機能が、会議の内容理解に役立ったと感じましたか？

平均 9.0 点

2-3.要約機能により、自分の会議への理解度が向上したと感じましたか？

平均 9.0 点

2-4.要約機能により、参加者間の理解度のばらつきが減少したと感じましたか？

平均 8.6 点

2-5.要約によって議論の方向性が明確になったと感じましたか？

平均 8.1 点

2-6.要約機能を使うことで、会議の進行や意思決定が迅速化したと感じましたか？

平均 8.3 点

3-1.アドバイスの結果、全員が参加しやすい雰囲気になりましたか？

平均 9.3 点

3-2.アドバイス機能により、新しい視点や意見が取り入れられましたか？

平均 8.8 点

3-3.アドバイス機能により、新しい視点や意見が取り入れられることで、意見の偏りが解消されたと感じましたか？

平均 8.5 点

3-4.アドバイス機能により、発言者の偏り(発言が少ない人)が解消されたと感じましたか？

平均 8.7 点

3-5.アドバイス機能が、会議中の沈黙の時間や停滞の時間を減少させるのに役立ったと感じましたか？

平均 8.8 点

3-6.アドバイス機能により、議論がより深まったと感じましたか？

平均 7.5 点

3-7 アドバイスは、会議の進行に効果的だったと思いますか？

平均 8.4 点

4-1.本システムを総合的に評価してください  
(1: 非常に悪い ~ 10:非常に良い)

平均 9.0 点

実験結果より、発話ログ記録機能では、発言の聞き逃し自体の回数は少なかったが、アンケートより発話ログをリアルタイムで閲覧できることは有効的だということが示された。また、要約機能ではアンケート結果より参加者ごとの理解度のばらつきが減少し、参加者自身の会議の理解度が向上したことが確認できた。アドバイス機能では生成の前後で会議の停滞の回数の減少が確認できると共に、アンケートからアドバイス機能が議論の活性化と積極性が向上したことが確認できた。



図 5 実験中の会議の様子

```
neo: 朝忙しかったら  
momoka: うん  
momoka: 確かに  
neo: 準備するの難しいね今  
momoka: 確かに  
neo: パパって準備できる  
neo: 栄養価の栄養価と満足度の高い  
neo: やってないもんね  
momoka: 安くない温野菜温野菜  
momoka: 温野菜はい  
neo: 温野菜だな最近百均でもあるしね温野菜チンでできる  
neo: 蒸すやつね  
momoka: うん  
neo: でも魚欲しくね  
momoka: 魚欲しい  
neo: ね  
momoka: 魚欲しいなあ  
neo: しゃけかサバって言ったね  
momoka: しゃけですね  
neo: しゃけほしいね  
momoka: もし  
momoka: だったら  
momoka: えでも焼き魚とかだったらもう焼くだけじゃないですか  
momoka: あそうそうそう
```

図 6 発話ログファイルの中身(一部抜粋)

```
### 会議ログ要約:  
参加者はモノの操作方法や視点変更についての話題から、朝ごはんに関する様々な意見や好みについてディスカッションを展開しました。  
-  
パンとご飯、おかずや味噌汁の具材、朝食運びにおける好みや準備の難しさ、温野菜や焼き魚の手軽さなどが話題に上りました。  
-  
参加者は朝ごはんにおける栄養価や満足度、料理の手軽さなどを考慮しながら議論を進め、最適な朝食の構成について検討しました。
```

図 7 生成された要約(一部抜粋)

```
Text  
1. もっと発言すべき参加者:  
- neoはもっと積極的に発言する必要があります。momokaとの対話でバランスを取りながら、新たな視点や提案をもたらすことで議論を活性化させましょう。  
2. 会議の進行案:  
- 現在のポイント: 朝ごはんに関する食事の構成についての議論が行われています。  
- 次に議論すべきポイント:  
- 栄養価や満足感、料理の手軽さなど、朝食運びの重要性を加味した結論を導くべきです。  
- 提案: 次は朝食の健康効果やエネルギー提供といった視点を盛り下げ、議論を深化させることが重要です。
```

図 8 生成されたアドバイス(一部抜粋)

## 5. 結論

本研究では、会議の質を向上させることを目的として、対面会議を超える参加者の双方向かつ積極的な議論が促進できる環境の構築をリアルタイム発話ログ記録機能や要約、アドバイスの生成機能を用いて提案・開発した。このシステムは、オンライン会議や対面会議にもともとあった従来の課題、特に発言の聞き逃しや議論の停滞、参加者間の理解度のばらつきといった問題を解決することを目指して設計された。

本システムでは以下の3つの機能を実装した。

1. **リアルタイム発話ログ記録機能:** 発言者ごとに発言内容を記録し、会議の透明性を高めるとともに、後の振り返りや要約・アドバイスの作成を容易にした。また、発言者の区別が容易になりこれは対面会議を超える可能性を示す。
2. **発話ログファイルを基にした要約機能:** 会議の流れや主要な結論を迅速に把握できるようにすることで、参加者ごとの理解度のばらつきを減らし理解度を向上させた。
3. **発話ログファイルを基にしたアドバイス生成機能:** 会議が停滞した際に次の議論の方向性を提示することで、議論の促進と参加者全員の積極的な関与を促した。結果的に議論の停滞が減少したことが確認された。

実験の結果、リアルタイムで記録された発話ログファイルと自動要約による情報共有の仕組みが、会議参加者の理解度を大幅に向上させたことが確認された。特に、発話ログファイルを活用した要約機能は、参加者の理解を深めるだけでなく、重要な論点への集中を促し、議論の効率化に寄与した。また、アドバイス機能の導入により、会議中の沈黙の時間や議論の停滞が減少し、全体的な活発な意見交換が実現した。これらの成果は、対面会議に匹敵する、あるいはそれをを超える質の高い会議体験の可能性を示している。

一方で、本研究にはいくつかの限界も存在した。たとえば、音声認識の精度やリアルタイム処理速度は、システムのパフォーマンスに大きく影響を及ぼす要因であり、特定の環境や参加者の発音に依存する場面もあった。また、ChatGPT を利用した要約やアドバイスの生成において、

トークン制限やプロンプトの設計による制約が見られた。これらの課題は、システムのさらなる改良と性能向上の余地を示している。

今後の展望として、本システムをより幅広い会議形態や参加者層に適用し、その効果を検証することが挙げられる。特に、感情認識や視線追跡などの高度な参加者行動分析機能を統合することで、議論の質を一層向上させることが期待される。また、多言語対応や大規模会議へのスケーラビリティの確保も重要な課題である。これらを通じて、オンライン会議の未来における標準的なシステムとしての地位を確立することを目指す。

## 参考文献

- [1] 内田 龍成, "3次元仮想会議システムにおける没入感向上に関する研究", 福岡大学工学部電子情報工学科卒業論文(2024年)
- [2] 齊藤 篤稀, "3次元仮想会議システムにおけるプレゼンテーション機能に関する研究", 福岡大学工学部電子情報工学科卒業論文(2024年)
- [3] 山田 朋史, "心理分析用3次元仮想会議室におけるプレゼンテーション機能の高度化に関する研究", 福岡大学大学院工学研究科電子情報工学専攻修士論文(2024年)
- [4] PUN2(Photon Unity Networking 2)で始めるオンラインゲーム開発入門  
<https://zenn.dev/o8que/books/bdcb9af27bdd7d>
- [5] 同期・RPC通信  
<https://zenn.dev/o8que/books/bdcb9af27bdd7d/viewer/2e3520>
- [6] Photon Voice 2 で Unity に超単純な通話機能を実装するだけ  
<https://qiita.com/aKquad/items/6cbd3cf76e4a0f0cc84c>
- [7] [Unity(C#)×PUN2]Oculus Quest でのボイスチャット実装方法  
<https://qiita.com/OKsaiyowa/items/1fcfe340645ff5e9aabc>
- [8] DictationRecognizer class in UnityEngine.Windows.Speech  
<https://docs.unity3d.com/ja/2021.1/ScriptReference/Windows.Speech.DictationRecognizer.html>