5ZE-01

音声認識と生成AIを統合した Web会議向け議事録支援システムの開発

原 由紀亜† 児玉 春樹‡ 林 佑弥§ 小島 正樹†

東京薬科大学生命科学部† 名古屋工業高等学校‡

会津大学コンピュータ理工学部§

1. はじめに

近年、テレワークの普及によりWeb会議が日常化している。Web会議は柔軟な働き方を可能にする一方、議事録作成には多大な労力を要し、情報共有の遅延を招く要因となっている。既存のツールは、リアルタイム性や要約精度、コスト面に課題が残る場合が多い。そこで本研究では、議事録作成の効率化を目的として、音声認識モデルWhisperおよび生成AI Geminiを統合した議事録支援システムを開発した。本システムは、基盤にFastAPIとWebSocketを採用し、拡張性の高い非同期処理環境を構築している。本稿では、本システムの構成と、ストリーミング処理における安定性を向上させる実装手法について述べる。

2. 開発背景

既存の自動化ツール[1]の多くは、録音後のバッチ処理や単なる発言の羅列に留まり、会議中の迅速な情報共有や、膨大な記録からの情報の構造化というニーズを十分に満たせていない。特に長時間の会議では、テキスト量が増加し、後から文脈を把握することが困難となる課題がある。そこで本研究では、通信プロトコルにWebSocketを採用し、HTTP通信のオーバーヘッドを排除することで高度なリアルタイム性を確保した。さらに、バックエンドのFastAPI上で音声認識と生成AIをパイプライン処理させることで、即時的なテキスト化と文脈理解に基づく議事録生成を両立させる構成とした。

3. システム構成

システムは、クライアントサイド、バックエンドサーバー、および外部AIモデルの3層構造で構成される。

3.1 全体アーキテクチャ

システムの全体構成を図1に示す。クライアントから送信された音声データは、サーバーでのバッファリングを経て音声認識モデルへ投入される。認識結果は生成AIにより構造化され、即座にクライアントへフィードバックされる非同期処理構造となっている。

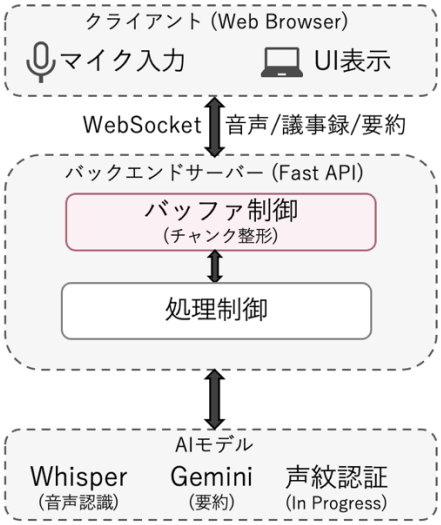

図1 システム構成

3.2 クライアントサイドの実装

クライアントサイドは、Webブラウザ上で動作するSingle Page Application (SPA) [2]として実装した。音声取得にはWeb Audio APIおよびMediaStream Recording APIを使用している。取得した音声ストリームは、ネットワーク負荷と処理効率のバランスを考慮し、小さなチャンク（バイナリデータ）に分割して送信される。また、サーバーからの切断時に自動再接続を行うロジックを実装し、長時間の会議における接続の安定性を担保している。

3.3 バックエンドとAI処理

バックエンドには、Pythonによる高速なWebフレームワークであるFastAPI[3]を採用した。WebSocketのエンドポイントを構築し、複数のクライアントからの同時接続を非同期で処理する。

Development of Minutes Support System for Web Meetings by Integrating Speech Recognition with Generative AI

† Yukia HARA, School of Life Sciences, Tokyo University of Pharmacy and Life Sciences

‡ Haruki KODAMA, Nagoya Technical High School

§ Yuya HAYASHI, School of Computer Science and Engineering, University of Aizu

† Masaki KOJIMA, School of Life Sciences, Tokyo University of Pharmacy and Life Sciences

(1) 音声認識処理：受信した音声データは、OpenAI の Whisper モデルを用いてテキスト化される。なお、認識精度向上のためバッファリング機構を実装している。
(2) 要約と構造化：文字起こしされたテキストは、Google の Gemini API[4]へと渡される。システムプロンプトとして「議題の抽出」「決定事項の特定」「アクションアイテムのリスト化」を指示しており、Gemini は入力された散文的な会話テキストを、即座にこれら 3 点の項目に構造化して出力する。

4. 実装と評価

4.1 音声処理の安定化

WebSocket 通信におけるネットワークの揺らぎや無音区間の発生に対処するため、本システムでは以下の 2 点を実装した。第 1 にサーバー側のバッファ制御である。受信パケットを一時蓄積し、Whisper の推論に最適なチャンク長へ整形してから処理を行うことで、発話が不自然に分断されることを防ぎ精度を維持した。第 2 にタイムアウト管理である。データ途絶や応答遅延時に接続を自動リセットするロジックを導入し、長時間会議においてもフリーズしない安定した継続記録を可能にした。

4.2 動作検証

開発したシステムの有用性を確認するため、模擬的な Web 会議環境において動作検証を行った。検証では、約 10 分間の会議音声を送信し、文字起こしおよび要約生成の挙動を確認した。図 2 に実行画面を示す。画面上には Whisper によるリアルタイムな文字起こしが表示され、終了時には Gemini により構造化された「議題」「決定事項」「アクションアイテム」が出力されている。検証の結果、発話からテキスト化までの遅延は数秒程度に抑えられ、要約も文脈を正しく捉えていたことから、本システムが議事録作成の効率化に寄与することを確認した。

図 2 システムの実行・出力イメージ

5. まとめ・今後の課題

本稿では、Web 会議における議事録作成の効率化と情報の構造化を目的として、FastAPI と WebSocket を用いたリアルタイム会議支援システムを開発した。OpenAI Whisper による音声認識と Google Gemini による文脈理解、さらに声紋認証技術を統合して発言者の特定を可能にする構成とし、サーバーサイドでの適切なバッファ制御を実装することで、ネットワークの揺らぎや長時間の会議においても安定して動作するシステムを構築した。これにより、会議中の議論を即座に可視化し、発言者と紐づいた決定事項の抽出をリアルタイムに行うことが可能となる。

今後の課題としては、以下の 3 点が挙げられる。第 1 に、認識精度および要約品質の定量的な評価である。本稿では動作検証による定性的な確認を行ったが、今後は実際の会議データセットを用いた単語誤り率（WER）の測定や、要約の正確性について被験者を用いた評価実験を行い、実用性の検証を進める。第 2 に、専門用語への適応能力の向上である。Whisper は汎用性が高い反面、特定の業界用語や略語の認識には課題が残るため、ユーザー辞書やプロンプトによる補正機能を強化する。第 3 に、声紋認証機能の実装の深化と検証である。本システムは話者特定を可能とするアーキテクチャを採用しているが、リアルタイム通信との完全な統合やチューニングは現在実装の途上にある。今後は、複数話者の識別精度の検証を進めるとともに、誤判定を即座に修正できる UI の実装を行い、システムの実用性を高めていく予定である。

6. 参考文献

[1] toruno 編集部. “議事録作成におすすめな文字起こしアプリ｜スマホ対応も紹介”. リコージャパン ウェブマガジン. 2025-06-03. https://www.ricoh.co.jp/magazines/column/trn-meeting-minutes-apps/, (参照 2025-07-26).
[2] Meta Platforms. React. https://react.dev/
[3] Tiangolo. FastAPI. https://fastapi.tiangolo.com/
[4] Google. “Gemini API Overview”. Google AI for Developers. https://ai.google.dev/docs