ChatGPT

エスペラント語リアルタイム文字起こしの高精度実現方法

はじめに

ZoomやGoogle Meetなどのオンライン会議で**エスペラント語のみ**を話す場合に、会話をリアルタイムに文字起こしする方法を検討します。システム環境（Windowsやクラウドなど）は問わず、参加者にとって使いやすく、**文字起こし精度が最優先**です。表示形式は会議ツール内の字幕でも別ウィンドウでも構いませんが、**リアルタイムで読みやすく表示**されることが重要です。本レポートでは以下の観点から最新情報に基づいて調査します。

- エスペラント音声に対応した音声認識エンジン（Google Cloud Speech-to-Text、OpenAI Whisper、Vosk など）とその精度・特徴
- 利用可能な会議プラットフォームの字幕機能や外部字幕連携の可否（Zoom、Google Meet、Microsoft Teams、Jitsi など）
- 自作・拡張が可能なツール（例：OBS Studio＋字幕プラグイン、Whisperベースのローカルアプリ等）
- 既存ソリューション（Web Captioner、Otter.ai、Subtitle Edit等）のエスペラント対応状況
- 音声入力のためのマイク設定や仮想オーディオデバイスを用いた音声取り込み方法
- 推奨される構成例（例：Windows＋OBS＋Whisper＋外部字幕表示アプリ）

各項目について、具体的な設定例やリンク、オープンソース/商用の別、料金体系にも触れながら解説します。

エスペラント対応の音声認識エンジンと精度評価

エスペラント語の音声を高精度にテキスト化できる音声認識エンジンとして、主に以下が挙げられます。

- **Google Cloud Speech-to-Text**（商用クラウドサービス） – Googleのクラウド音声認識。長らく主要言語中心でしたが、現在はエスペラント語も**プレビュー対応**されています [1]。正式対応は限定的ですが、Dialogflow（Googleの対話AI）の言語リファレンスではエスペラント語がSTT（Speech-to-Text）可能言語として示されています [1]。Googleの認識精度は一般に高く、エスペラントでも高精度が期待できます。料金は標準モデルで約0.016ドル/分程度（約1ドル/時） [2] [3] で、1か月60分までは無料枠もあります（V1　APIの場合）[^1]。クラウドゆえの**利点**は高精度・スケーラビリティですが、**懸念**としてインターネット接続必須・利用コスト発生があります。

- **OpenAI Whisper**（オープンソースモデル） – OpenAIが公開した多言語音声認識モデルです。700,000時間以上の音声で訓練され、約100言語に対応しています [4]。エスペラントもサポート対象に含まれており、モデルに言語を指定して高精度な文字起こしが可能です。Whisperの**大モデル**は非常に高精度で、エスペラント音声に対しても最先端に近い性能が報告されています。例えば、Mozilla　Common　Voiceのエスペラントデータセットで訓練された他エンジン（Voskモデル）で単語エラー率8.3%程度 [5] という結果がありますが、Whisper大規模モデルでも同等かそれ以上の精度が期待できます。Whisperの**利点**はオープンソースかつローカル実行可能な点で、ネット接続なし・使用料無料で利用できます。実装によってはリアルタイムに近い出力も可能で、OBSプラグイン「LocalVocal」のようにWhisper（whisper.cpp版）を

利用して**ローカルで遅延数百ms程度のリアルタイム文字起こし**を実現している例があります [6] 。一方で**懸念**は、大モデルほど計算資源を要し（CPUでは重くGPU推奨）、また公式にはストリーミング対応がないため工夫が必要な点です。ただコミュニティによるストリーミング拡張（例：Collabora WhisperLiveやWhisper-Streaming実装）も登場しつつあります。

- **Vosk**（オープンソースオフラインエンジン） – Kaldi系のエンジンを簡単に利用できるオフライン音声認識ライブラリです。Alpha Cephei社によって提供され、多数言語の既成モデルがあります。エスペラント語についてもCommon Voiceコーパスに基づくモデルが公開されており、**単語誤り率8.28%**という高精度が報告されています [5] （Common Voiceのテストデータにて）。モデルサイズは約50MBと比較的小さく、PCやモバイル、Raspberry Pi等でも動作可能です [7] 。Voskの**利点**は軽量で組み込みやすく、ネット不要・無料で使える点、**懸念**はWhisperなどに比べ音響モデルの表現力が若干劣る可能性や、導入に技術的手間がかかる点です（Python/Java/C#等での実装が必要）。

- **その他エンジン**: Microsoft Azure Speech-to-TextやAmazon Transcribeなど主要クラウドサービスはいずれもエスペラント語は**未対応**です（主要な国語のみ対応）。IBM Watsonも同様にエスペラント非対応です。したがって現状では上記のGoogle Cloud、Whisper、Voskあたりが現実的な選択肢です。加えて、Mozilla DeepSpeechの後継であるCoqui STTには有志がエスペラントモデルを訓練した例がありますが、報告されている単語誤り率は20–30%程度と高めで [8] 、WhisperやVoskには及びません。商用サービスでは、AI字幕サービスの**DeepL TranslateやMetaの子会社**によるサービスなど一部でエスペラント「文字起こし結果の翻訳先」としての対応は見られるものの、**音声入力として直接対応**した事例はほぼありません。近年ではWebexが英語音声をエスペラント含む100言語以上に字幕翻訳する機能を提供していますが [9] 、**今回の要件（エスペラント「音声」そのものの文字起こし）とは異なる**ため注意が必要です。

以下に主要エンジンの特徴をまとめます：

エンジン / サービス	エスペラント対応状況	種類・提供形態	精度の目安	料金・ライセンス
Google Cloud Speech-to-Text	公式プレビュー対応 [1] （正式版では限定）	クラウドAPIサービス	高精度（エラー率数%台）	従量課金（標準モデル約$0.016/分 [2] ）
OpenAI Whisper	対応（100言語対応モデルに含む）	オープンソースモデル（ローカル実行）	非常に高精度（大モデルで数%台）	無料（MITライセンス、要CPU/GPU）
Vosk (Kaldi系)	対応（既成モデル有り）	オープンソースエンジン（ローカル実行）	高精度（WER約8% [5] ）	無料（Apache 2.0ライセンス）
Microsoft Azure Speech	非対応	クラウドAPIサービス	–	–（主要言語のみ対応）
Amazon Transcribe	非対応	クラウドAPIサービス	–	–（主要言語のみ対応）
Otter.ai (参考)	非対応（英・西・仏のみ） [10]	SaaS字幕サービス	–	月額制（対応言語外のため適用外）

※上記の精度目安は公開情報や有志検証によります。エスペラントは話者層が限られるため、エンジンによっては訓練データが少なく精度が下がる可能性もあります。とはいえCommon Voiceプロジェクトなどで蓄積された音声があるため、対応エンジンでは実用十分な精度が得られると考えられます。

会議プラットフォームの字幕機能と外部連携

続いて、オンライン会議プラットフォームごとの字幕対応状況を見ていきます。エスペラントのリアルタイム文字起こしという観点では、**プラットフォーム標準の自動字幕機能はどれも対象外**のため、外部サービスやカスタム連携が鍵となります。

- **Zoom** – Zoomは英語や日本語など主要言語向けに自動ライブ字幕機能を提供していますが、エスペラント語は対応していません。しかし、Zoomには**サードパーティ字幕API連携**の仕組みがあります [11]。ホストが「手動字幕の設定」でAPIトークン（字幕用URL）を取得し、それに対して字幕テキストをHTTP POSTで送信することで、Zoom会議内の字幕欄にリアルタイム表示できます [11] [12]。この方法を使えば、**外部の音声認識エンジンで取得したエスペラントのテキストをZoomに表示**可能です。実際、プロの速記者サービスやOtter.ai（英語限定ですが）のZoom連携もこのAPI方式で実現されています。したがって、**Zoomは外部文字起こし結果の字幕表示に対応可能**なプラットフォームと言えます。ホストもしくは指定の字幕提供者がスクリプトやツールを用いて、音声認識エンジンからのテキストを逐次Zoomに送信する運用になります。

- **Google Meet** – Meetには自動字幕（英語や日本語、スペイン語など）機能がありますが、エスペラント語はサポート外です。また、2025年現在、MeetはZoomのような外部字幕サービス連携APIを公開していません。したがって**Meetでエスペラント字幕を表示する公式な方法はない**のが現状です。対策としては、会議とは別に**字幕用のウィンドウを共有**する方法があります。例えば、主催者が自分のPC上でエスペラント音声を認識するアプリを動かし、その結果テキストを常に画面一部に表示（またはウィンドウ表示）させ、それをMeetの画面共有機能で他参加者に見せる、といった手法です。この場合、字幕はMeetのインターフェースではなく共有コンテンツとして表示されます。または各参加者が各自で音声認識ツールを動かし自分の画面上に字幕をオーバーレイする（例：Chromeブラウザ拡張で音声入力をキャプション化する）手も考えられます。現状Meetに直接字幕を流し込むことはできないため、**別ウィンドウ方式での代替表示**が必要です。

- **Microsoft Teams** – Teamsも英語等に対応したライブ字幕機能がありますが、エスペラント語は対象外です。また会議内への外部字幕入力APIは提供されていません（Teamsライブイベントではサードパーティ字幕者の手動入力は可能ですが開発者向けAPIは限定的です）。したがって、基本的には**Google Meetと同様に外部ウィンドウ共有等で代替**するしかありません。Teamsの機能として、PowerPoint Liveや字幕翻訳の機能がありますが、いずれも入力元が英語など限定のため、エスペラント話者同士の会話には適用困難です。よって**Teams単体では直接解決できず、外部ツールとの併用が必要**です。

- **Jitsi Meet** – オープンソースのJitsiは、自前でサーバーを立てることでカスタマイズ可能なプラットフォームです。Jitsiには「Jigasi」というモジュールを用いた字幕機能の拡張があり、設定によってGoogle CloudやIBM、**Vosk、さらにはWhisper**等のエンジンと連携して自動字幕を提供できます [13]。実際、JigasiはGoogle Cloud Speech-to-Text APIやVoskサーバーを呼び出して複数参加者の音声を文字起こしし、Jitsi Meetの画面上に字幕として流すことが可能です [14] [15]。VoskについてはDockerイメージを用意することで無料で動かせるようになっており [15]、エスペラントのVoskモデルをロードすれば**完全オープンソース環境でエスペラント字幕付き会議**を実現できます。またコミュニティ実装でWhisperをバックエンドに

するオプションも存在します [13]。もっとも、Jitsiを利用するにはホスティングやサーバー構築の知識が必要で、エンドツールとして手軽に使うにはハードルがあります。しかし技術的には**エスペラント音声のリアルタイム字幕を公式UI内に表示できる唯一のプラットフォーム**といえます。既存のJitsi公共サーバー（meet.jit.si等）ではデフォルトでこの機能は無効な場合が多く、自身で環境構築・設定するか、対応したサービスプロバイダを利用する必要があります。

- **Cisco Webex** – Webexはリアルタイム翻訳字幕機能を有し、英語の発話をエスペラントを含む100+言語に機械翻訳して表示するサービスを提供しています [16] [17]（有償のリアルタイム翻訳オプション）。しかしこれは**発話は英語等**が前提であり、エスペラント話者同士の会話そのものを直接文字起こしする機能ではありません。エスペラント音声をそのまま認識する機能はWebexにもありません。ただしWebexでもZoom同様、手動字幕入力者を指定できるので、人手または外部プログラムによるキャプション入力は不可能ではありません [18]。総じて、**エスペラント音声を自動で認識するプラットフォーム標準機能はない**ため、ZoomやJitsiのような**外部字幕取り込み可能なもの**か、Meet/Teamsのように**画面共有等で疑似的に表示**する方法を採る必要があります。

以下に主要プラットフォームの対応可否を表にまとめます：

プラットフォーム	自動字幕のエスペラント対応	外部からの字幕入力可否	備考・代替策
Zoom	×（自動字幕は主要言語のみ）	○（字幕用APIで外部テキスト入力可 [11]）	API経由で外部STT結果を表示可能 [11]
Google Meet	×（主要言語のみ）	×（公式な外部字幕APIなし）	別ウィンドウ共有で字幕を見せる対処
Microsoft Teams	×（主要言語のみ）	×（外部字幕APIなし）	外部ツールでオーバーレイ表示など
Jitsi Meet (自前運用)	–（標準機能なし）	○（JigasiでSTT連携可 [13]）	Vosk/Whisper等統合で字幕表示可能

※上記の“外部からの字幕入力可否”はプラットフォームの機能としてテキストデータを字幕に流し込めるかどうかを示しています。ZoomやJitsiは専用インターフェースがありますが、MeetやTeamsにはなく、画面共有や各端末ごとの工夫が必要になります。

カスタム字幕ツール・自作ソリューション

プラットフォームに直接依存しない形で、**自前で字幕表示環境を構築**することも可能です。ユーザー自身が音声認識エンジンを動かし、その結果を好きな形式で表示するアプローチです。以下、考えられるツールや拡張例を紹介します。

- **OBS Studio＋字幕プラグイン** – OBSはオープンソースのブロードキャストソフトですが、仮想カメラ機能やプラグイン拡張を活用することで字幕合成ができます。有力なのが「*LocalVocal*」というOBSプラグインで、PC内でWhisper（whisper.cpp版）を動かして**ローカルでリアルタイム音声認識**を行い、OBS上にテキストを表示するものです [6] [4]。LocalVocalはWindows/Mac/Linux対応で、100言語以上に対応したWhisperモデルを使用し、クラウド不要・遅延も最小限（GPUなしでも動作可）とされています [6]

[4]。具体的には、OBSでマイクやPC音声にこのプラグインのフィルターを掛けると、音声→テキスト変換され、指定したテキストソースに字幕として表示されます[19]。また字幕を逐次*.txt*や.srtファイルに書き出す機能もあり[20]、外部アプリからそのファイルを読んで表示することもできます。OBSを使う利点は、**自分の映像に字幕を焼き付けて仮想カメラ出力**できる点です。例えばZoomやMeetで自分のカメラ映像としてOBSの仮想カメラを選べば、自身の映像下にエスペラント字幕を合成して配信できます。これなら相手は特別な設定なしに字幕付き映像を視認できます。ただし自分の発話しか字幕にならない点や、映像として合成されるためテキストのコピーや検索ができない点には注意です。またOBS経由にせずとも、LocalVocalはファイル出力や他アプリ連携もできるため、例えばOBSで音声→テキストした結果をZoomの字幕APIに送信するスクリプトを書くといった拡張も考えられます。

- **Whisperベースのローカルアプリ** – Whisperを利用したGUIアプリも登場しています。その一つが「**Buzz**」と呼ばれるWindows/Mac向けのオープンソース音声認識アプリです。Buzzは内部でWhisperを使用し、マイク入力やシステム音をリアルタイムに文字起こしして画面上に表示できます。Buzzでは入力言語にエスペラントを指定可能で（Whisper対応言語のため）、ライブ記録モードで字幕的にテキストを流し続けることができます[21][22]。Windowsの場合、システム音声を取り込むには仮想オーディオデバイス（後述）を使い、Buzzのマイクとして指定する必要があります[23]。Buzzは比較的簡単に使えますが、表示テキストのレイアウト調整や他アプリへの埋め込みは手作業になります。その他、CUIツールですがLinux向けの*nerd-dictation*（Voskを使ったリアルタイム字幕ツール）[24]などもあり、用途に応じて選択できます。

- **その他のツール・プラグイン**: ブラウザを使う方法として、ChromeのWeb Speech APIを用いたWebアプリ「**Web Captioner**」があります。これは無料のサイトで、マイクの音声をブラウザ内で文字起こしし字幕表示するものです。しかしChromeの音声認識エンジン自体がエスペラント非対応のため、Web Captionerでエスペラントを選択することは**できません**[25]（言語リストに存在しない）。他方、商用の自動議事録サービス**Otter.ai**は優れた話者分離やUIを備えていますが、対応言語は英語（および最近追加されたフランス語・スペイン語）のみで[10]、エスペラントには対応していません。同様に**字幕編集ソフトのSubtitle Edit**にも音声→テキスト変換機能があります。Subtitle EditはWhisperやVoskを内部利用して動画音声から字幕を書き起こす機能を持ち[26]、エスペラントでもエンジンを用意すれば変換できます。ただしこれはリアルタイム会議向けではなく**オフライン音源の文字起こし・字幕化**用途です。会議中の逐次処理には向きませんが、録画したエスペラント会議の文字起こしには活用できます。

- **自作スクリプト**: 技術的な余裕があれば、各エンジンのAPIやライブラリを直接使ってカスタムソリューションを作ることも可能です。例として、Pythonでマイク音声をWhisperやGoogle Cloud STTに送り、返ってきたテキストをウィンドウに表示する簡易アプリを作成できます。またそのテキストをZoom APIに送って字幕反映させるスクリプトも作れます。この場合、表示方法は自由度が高く、字幕のフォント・色・配置もカスタマイズ可能です。例えばTkinter等で大きな文字で常時最前面に出す字幕ウィンドウを作れば、どの会議アプリ上でも重ねて表示できます。自作の難点は開発コストですが、既存ツールを組み合わせても実現できない細かな要件（例えば専門用語辞書の組み込みや特定キーで字幕一時停止等）に対応できるメリットがあります。

音声入力の取り込み方法（マイク・仮想デバイス設定）

リアルタイム文字起こしを正確に行うには、**音声入力を適切にエンジンに渡す**ことが重要です。シナリオにより音声の取り込み方法が異なるため、考慮すべきポイントをまとめます。

1. **誰の発話を文字起こしするか** – 会議の全参加者の発言をまとめて字幕化したい場合、会議の混合音声（各参加者の声が混ざった出力）を認識にかける必要があります。一方、自分（主催者）の発言だけ字幕にして相手に見せたい場合は自分のマイク音声だけ処理すれば足ります。全員分を一つの字幕にする場合、同時発話があると正確さが落ちるため、可能なら発言者ごとに処理するのが理想ですが現実的には難しいでしょう。そこで通常は**会議のマスター出力音声をまとめてSTT処理**する形になります。

2. **音声の取り込み** – パソコン上で会議アプリと音声認識エンジンを両方動かす場合、**仮想オーディオデバイス**を用いて音声を中継するのが一般的です。例えばWindowsでは「VB-Audio Virtual Cable」[23]という無料ツールが使えます。Virtual Cableをインストールすると仮想的なスピーカー出力とマイク入力が一組作られます[23]。Zoom/Meetの**スピーカー出力**をこの仮想デバイスに指定し、音声認識アプリの**マイク入力**に同じデバイスを指定することで、会議の音声がそのまま認識エンジンに渡ります[23]。これにより自分のPC内で音声をループバックでき、マイクで物理的に拾わなくてもデジタル音声を直接処理できます。Macの場合は「BlackHole」や「Loopback」といった仮想オーディオデバイス、LinuxではPulseAudioのモニター機能等が類似の役割を果たします[27][28]。

3. **マイク設定と音質** – マイクから直接エンジンに送る場合は**クリアな音質**が重要です。エスペラントは発音が比較的明瞭と言われますが、雑音が多い環境だとどんなエンジンも誤認識が増えます。可能ならノイズキャンセリング機能付きマイクや、会議アプリ側のノイズ抑制を有効にすることが有用です。ただし会議アプリ側で過度に音声処理される（例：自動音量調整やエコーキャンセル）と認識に影響する場合もあります。自分のPC内で完結にするなら、Zoomの音響処理を切り、代わりにOBSでフィルタ処理してから認識するといった制御も可能です。いずれにせよ、**はっきり発音し、マイク入力レベルを適切に保つ**ことが基本となります。

4. **ディレイ（遅延）の許容** – リアルタイム性を重視すると、一度に処理する音声長を短く刻む必要があります。Whisperなどは高精度の代わりに2〜3秒ごとに結果を更新する実装が多いです（小刻みに音声を切り出して逐次認識）。ユーザー体感で**1〜2秒程度の遅れ**は許容範囲ですが、5秒以上遅れると違和感があります。エンジンとハードウェア次第ではありますが、設定でなるべく短いチャンクで処理する、音声認識結果の部分出力（中間結果）を表示する、といった工夫で**リアルタイム性と精度のバランス**を取ります。例えばGoogle CloudのストリーミングAPIは発話中から順次テキストを返すモードがあるので、それを使えば遅延を小さくできます。一方Whisper系は基本的に音声チャンク単位出力なので、1〜2秒区切りで切り上げて認識する改造実装もあります。このように**できるだけリアルタイムに近づける設定**も重要になります。

推奨構成例と運用方法

以上を踏まえ、実際にどのような構成でエスペラント字幕を実現できるか、一例を示します。ここでは**「Windows　PC上でZoom会議を開催し、参加者全員のエスペラント会話をリアルタイム字幕表示」**するケースを想定し、**高精度かつ比較的構築しやすい**手段を組み合わせます。

推奨構成例: Windows＋OBS＋Whisper＋Zoom字幕

構成概要: Windows上でOBS StudioとLocalVocalプラグイン（Whisper利用）を使い、Zoom会議の音声をリアルタイムにテキスト化。字幕表示はZoomのAPIを使い、他の参加者にも見える形で提供します（Zoomの字幕欄に表示）。

必要な要素:

- Windows 10/11 PC（できればGPU搭載で性能高めのものが望ましい）
- OBS Studio（無料）＋ LocalVocalプラグイン [6]
- VB-Audio Virtual Cable（無料仮想オーディオデバイス） [23]
- Zoom デスクトップクライアント（最新版、ホスト権限）
- OpenAI Whisperモデルデータ（プラグインに同梱の小〜中モデル、または自分でGGML版モデルを用意）

セットアップ手順の例:

1. **仮想オーディオ設定**: VB-Cableをインストール。Windowsのサウンド設定で、既定の出力デバイスを「CABLE Input」に変更します（またはZoomアプリ内のスピーカー出力をCABLE Inputに指定）。これにより、Zoom会議内の全音声が仮想デバイス経由で流れるようになります [23]。

2. **OBSの音声入力**: OBS Studioを起動し、オーディオソースとして「CABLE Output」（仮想オーディオの出力側）を追加します。これでOBS上で会議の混合音声をモニターできる状態になります。

3. **LocalVocalプラグイン導入**: OBSにLocalVocalプラグインをインストールします（OBSフォーラムまたはGitHubからダウンロード [6]）。インストール後、先ほどのオーディオソース（CABLE Output）にフィルターとして「LocalVocal」を追加し、言語をEsperanto（エスペラント）に設定します。内部でWhisperモデル（デフォルトは小モデル等）がロードされ、音声認識が走り始めます。

4. **字幕テキストの取得**: LocalVocalのフィルター設定で、出力方法を「OBSテキストソース」に送るよう指定します（プラグインのクイックスタート手順に沿って設定） [19]。これにより、OBS内のテキストオブジェクトに認識テキストが逐次表示されます。また併せて「.srtファイルに字幕を書き出す」オプションを有効にしておきます [20]。これで、指定フォルダに字幕が蓄積されるとともに、最新の字幕行が逐次ファイル更新されます。

5. **Zoomへの字幕連携**: Zoomミーティングを開始し、「字幕のオプション」から「手動による字幕付与を有効」「API経由の字幕配信を許可」をオンにします（管理者権限設定が必要な場合あり）。ホスト用の「字幕APIトークン（URL）」を取得します [29]。別途用意した簡易スクリプト（Python等）で、手順4の出力テキスト（例: 最新の字幕行を保持するTXTファイル）を監視し、新しい字幕が出るごとにその内容をZoomのAPI URLにHTTP POSTします [12]。POSTのたびに`seq`パラメータ（連番）をインクリメントし、字幕テキストを送信する実装です [30]。これによりZoom側にリアルタイム字幕が流れます。

6. **表示確認と調整**: Zoom参加者の画面下部にエスペラント語字幕がほぼリアルタイムで表示されることを確認します。認識精度や遅延を見ながら、Whisperモデルを必要に応じ変更します（小モデルで速度優先／大モデルで精度優先）。また語末の句読点や大文字など自動整形が必要ならWhisperの設定を調整します。字幕の出るタイミングが遅ければチャンク長を短くしたり、逆に途切れすぎる場合は長めにします。

必要に応じて専門用語や人名の綴りは事前にカスタム辞書（Whisperには直接適用できませんが、Googleなら**バイアス**機能でカスタマイズ可[31]）を使うことも検討します。

この構成では、**ホストPC一台で自動字幕生成から表示まで完結**します。参加者は通常のZoomクライアントで字幕を見るだけです。クラウドサービスは使用しておらず、Whisper処理はローカルなので通信遅延もなくプライバシー面も安全です[6]。精度もWhisper大型モデルを使えば非常に高くなるでしょう。ただしホストPCの負荷は上がるため、高性能CPU/GPUがあると望ましいです。また、万一精度に不満がある場合はGoogle Cloud STTに切り替えることも可能です。その際は手順3で代わりにOBSプラグイン「Google Speech Recognition」（RatWithACompiler氏作）を使うか、手順5のスクリプトでGoogle Cloudに音声を送信して結果を受け取るようにします[32]。Google Cloud利用時はクラウド料金が発生する点に注意してください。

別構成例: Meet/Teams＋別ウィンドウ字幕

Zoomのように字幕APIが使えない環境向けには、**字幕専用ウィンドウを参加者全員に見せる**方法が現実的です。この一例として、**Google Meet＋OBS＋仮想カメラ**を組み合わせる方法を紹介します。

- 基本は先ほどのOBS＋LocalVocalでエスペラント音声をテキスト化する部分は同じです。違いは表示方法です。OBS上で、自分のWebカメラ映像に字幕テキスト（テキストソース）をオーバーレイ表示し、そのOBSを**仮想カメラ出力**します。Meetでは自分のカメラにOBS仮想カメラを選択すると、**自分の映像に常に字幕スーパーが載った状態**で相手に配信できます。これにより、各参加者は特別な設定なしにあなたの話す内容の字幕を見ることができます。複数人が字幕付き映像を出したければ、各自が同様のOBS設定をする必要があります（ハードルは高いですが、可能ではあります）。あるいは主催者だけが全員分の音声を拾って自分の映像にまとめて字幕表示することもできますが、自分が話していないときも自分のカメラ枠をPin留めしてもらうなど運用が必要になります。

- もう一つは、字幕テキストを大きく表示した**ウィンドウ（またはブラウザページ）を共有**する方法です。例えば専用のHTMLページを作り、そこにOBSからの字幕テキストファイルをAjax等で読み込んで表示更新するようにすれば、ブラウザで開いたそのページにリアルタイム字幕が映ります。それをMeetの画面共有機能で共有すれば、参加者全員が閲覧できます。この方法だと自分の映像とは独立した枠で表示でき、文字も画面全体を使って大きく出せる利点があります。操作役は発言者とは別の人にして、誰か一人が常に字幕スクリーンを共有し続ける形でも良いでしょう。TeamsでもPowerPointやウィンドウ共有で同様のことが可能です。

- 共有による字幕表示の注意点は、**会議映像と字幕の両方を見るには画面スペースが分かれる**ことです。特に画面共有の場合、参加者側ではスライドを見るような扱いになるため、人によっては見づらいかもしれません。そこでMeetなら**サイドバイサイド**で共有とカメラ映像を並べる、Teamsなら**コンテンツのみ表示**に切り替えて字幕画面を大きくするといった工夫が必要です。また、小規模なら参加者各自が字幕ページを自分で開いてもらい、各自の画面で好きにレイアウトしてもらう方法もあります（URLを共有して各自閲覧）。

以上のように、プラットフォーム標準に頼らずとも**ツールと工夫次第でリアルタイム字幕**は十分実現可能です。特にエスペラントのように既存サポートが少ない言語では、オープンソースの力を借りて自前で構築することが現実解となるでしょう。

参考情報・まとめ

- **音声認識エンジン**: 高精度を最優先するならOpenAI Whisper（大モデル）やGoogle Cloud STTが有力。Whisperはオープンソースで無料だが要スペック、Googleはクラウド課金だがお任せで使える。Voskも8%前後の誤り率で実用的[5]。AzureやAWSはエスペラント非対応なので選択肢外。
- **会議プラットフォーム**: Zoomは外部字幕API対応[11]で柔軟性高い。Meet/Teamsは直接は無理だが画面共有などで代替可能。Jitsiは自前セットアップでエスペラント字幕をネイティブ実装できる強み[13]。
- **ツール/ソリューション**: OBS＋LocalVocal（Whisper利用）は主要言語以外でも使える強力な組み合わせ[6]。Web CaptionerやOtter.aiなど既存サービスはエスペラント未対応で使えない[25][10]。Subtitle Editはオフライン文字起こしに有用[26]。
- **音声設定**: 仮想オーディオデバイスで会議音声を認識エンジンへ入力[23]。マイクは高品質なものを使い、ノイズやエコーを抑制。Overlapping speech（かぶり発話）は避け、はっきり話すことで精度向上。
- **実装のポイント**: Whisperなど英語以外では句読点や大文字小文字の扱いが荒いことがあるので後処理で整えると読みやすいです（必要なら独自に文頭を大文字化など）。また専門用語（例えば固有名詞）は認識ミスが起きやすいため、事前に用語リストを周知したり、都度手修正できる体制が理想です。最終的なログ（議事録）用途なら後でSubtitle Edit等で校正・整形すると良いでしょう。

以上、現在考えられるエスペラント会話のリアルタイム文字起こしソリューションを網羅的に解説しました。完全なワンストップ製品はありませんが、紹介した組み合わせを用いれば**高精度な自動字幕**を実現できます。オープンソース技術の活用により、ニッチな言語コミュニティでも利便性の高い字幕環境を整えられることを期待します。

[^1]: Google Cloud STTのV1 APIではデータログを許可すれば60分/月まで無料枠、その後$0.016/分[33][34]（ログ不許可だと$0.024/分）となっています。V2 APIでは一律$0.016/分（50万分/月まで）[2]に値下げされました。いずれにせよ約1ドルで1時間音声を文字起こしできる計算です。

参考文献・情報源: 使用した情報には、OBS用Whisperプラグイン開発者による解説[6][4]、Voskエンジンのエスペラントモデル公開情報[5]、Zoomサポート記事[11]、Jitsiの公式リポジトリ[13]、Google Cloudのドキュメント[1]、各種ツールの仕様ページ等を含みます。詳細は各出典箇所をご参照ください。

---

1 Language reference | Conversational Agents | Google Cloud
https://cloud.google.com/dialogflow/cx/docs/reference/language

2 3 33 34 Speech-to-Text API Pricing | Google Cloud
https://cloud.google.com/speech-to-text/pricing

4 6 19 20 LocalVocal: Local Live Captions & Translation On-the-Go | OBS Forums
https://obsproject.com/forum/resources/localvocal-local-live-captions-translation-on-the-go.1769/

5 7 24 Esperanto Speech-To-Text Model For Vosk ASR : r/Esperanto
https://www.reddit.com/r/Esperanto/comments/s027vy/esperanto_speechtotext_model_for_vosk_asr/

8 Links to pretrained models - #60 by stergro - Mozilla Discourse
https://discourse.mozilla.org/t/links-to-pretrained-models/62688/60

9 16 17 18 Show real-time translation and transcription in meetings and webinars
https://help.webex.com/article/nqzpeei/Afi%C8%99a%C8%9Bi-traducerea-%C8%99i-transcrierea-%C3%AEn-timp-real-%C3%AEn-%C3%AEnt%C3%A2lniri-%C8%99i-seminare-web

10 Bonjour! ¡Hola! Otter.ai Expands AI Meeting Assistant to Support ...
https://otter.ai/blog/bonjour-hola-otter-ai-expands-ai-meeting-assistant-to-support-french-and-spanish

11 12 29 30 Using a third-party closed captioning service
https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0060368

13 14 15 GitHub - jitsi/jigasi: Jigasi: a server-side application acting as a gateway to Jitsi Meet conferences. Currently allows regular SIP clients to join meetings and provides transcription capabilities.
https://github.com/jitsi/jigasi

21 22 23 27 28 Live Recording | Buzz
https://chidiwilliams.github.io/buzz/docs/usage/live_recording

25 Supported Languages · VRCWizard/TTS-Voice-Wizard Wiki · GitHub
https://github.com/VRCWizard/TTS-Voice-Wizard/wiki/Supported-Languages

26 Subtitle Edit - Nikse.dk
https://www.nikse.dk/subtitleedit

31 Speech-to-Text supported languages | Google Cloud Documentation
https://docs.cloud.google.com/speech-to-text/docs/speech-to-text-supported-languages

32 Closed Captioning via Google Speech Recognition | OBS Forums
https://obsproject.com/forum/resources/closed-captioning-via-google-speech-recognition.833/