

複数音声同時字幕提示ユーザインタフェースの開発

鈴木拓弥¹

概要：本研究は、会議やワークショップなど、複数の話者が同時に話す状況で活用することを想定した字幕提示のユーザインタフェースに関する研究である。聴覚障害者の支援には、手話やノートテイク、要約筆記などの手法が一般的に用いられている。しかしながら、聴覚障害者はグループワークや会議などの複数の話者が同時に話す状況においては、手話や字幕からだけでは十分に情報を取得できない場合がある。この課題を改善するために、事前に資料を送る、複数の話者が同時に話さない、ファシリテーターが話者を調整するなどの配慮が行われている。しかし、これらの配慮が常に行われるとは限らない。また、補助者が必要となる場合も多く、個人が簡便な手続きで問題を解決することができるような仕組みを準備する必要があると考え、これまでの支援や配慮とは異なる支援手法を検討した。本研究では、会議やグループワークなど、健聴者を中心とした中に少数の聴覚障害者が参加するような場面で、聴覚障害者を支援するアプリケーションを開発した。本アプリケーションは、複数の話者によって音声と同時に多発的に発せられる場面において活用することを前提としており、複数の音声を同時にリアルタイムで字幕化し、聴覚障害者に話者を区別しやすい状態で提示するための音声入力インターフェースを有する Web アプリケーションである。本発表では本アプリケーションの開発プロセスと検証について発表する。

キーワード：聴覚障害、情報保障、音声認識、複数話者、複数音声

1. はじめに

聴覚障害者への支援では、ノートテイク、手話通訳、要約筆記などの手法が中心的に用いられている。近年では、音声認識技術などの情報通信技術を活用した方法が成果を上げている。しかしその一方で、解決が困難な問題も存在する。例えば、学校や職場において聴覚障害者が直面する困難な場面として、グループワークや会議などが挙げられる。複数の話者が同時に話している状況では、聴覚障害者は手話や字幕だけでは十分な情報を得ることができない。そのため、事前に資料を用意する、複数の話者が同時に話さないようにする、司会者が話者を管理したりすることなどで解決している。しかし、これらの解決手法は常に利用できるわけではない。基本的には他者の介入を必要とすることが多い。また、これらの解決手法を用いることで、情報交換の質が低下すると解釈されることもある。結果として聴覚障害者の参加を得にくくなることもある。この問題の解決には、これまでの支援や配慮とは基軸の異なる対策が必要であるのではないかと考えた。

本研究では、複数の話者によって音声と同時に多発的に発せられる場面において使用することを前提とした支援アプリケーションを開発した。はじめに既存の手法を調査し、使いやすさと導入しやすさを両立したアプリケーションの実装方法を検討した。実装方法を決定した後、聴覚障害者にとって使いやすく、複数の音声を提示するためのユーザインタフェース（以下 UI）を開発した。開発後、手法が有効かどうか、効果を検証し、改善点を得た。

2. 関連研究の調査と提案手法の検討

筆者は聴覚障害者のための国立大学である筑波技術大学産業技術学部において教職に就いている。筆者のこれまでの教職経験から、会議やグループワークなど、健聴者が多数を占める中に少数の聴覚障害者が加わる場面においては、聴覚障害者はコミュニケーション上、多くの課題を抱えていると考えている。仮に健聴者の多くが手話を使うことができる場合や、手話通訳士に通訳を依頼できる場合であっても、情報発信量を調整することもある。挙手の上、交代で発言するなどの工夫や、司会者が話者を調整するような工夫、文書化された資料の配布などである。しかしながら、会議やワークショップに参加する健聴者が手話を使えるような状況は希であり、常に手話通訳士に依頼できるとは限らない。また、これらの条件が整っていたとしても、当事者である聴覚障害者の心理的負担など、別の課題もある。このような課題に対し、本研究では、集団において発せられる音声を、他者の介入を受けることなく、可能な限り簡便な手段でリアルタイムに字幕提示する手法を検討した。

発話から字幕提示に至るまでの手法について、はじめに要約筆記について検討した。他社の介入を必要とするが、要約筆記による字幕提示は、有効な情報保障手段の一つとして一般的に広く活用されている。字幕提示タイミングの遅れよりも、正確に情報を伝えることが重視される場面や、講義や講演など、基本的に単一方向で情報が伝達される場合における利用が一般的である。一方で、双方向なやり取りが多くなる会話などの場合、正確性よりも字幕提示タイミングの遅れの方が問題となる可能性が考えられた。

そこで、音声と字幕化に関する既往研究を調査した。丸

¹ 筑波技術大学産業技術学部
Division of Industrial Technology, Tsukuba University of
Technology

山ら[1]によれば、発声と字幕とのタイミングのズレの許容限界は1秒程度とされている。下郡ら[2]によれば字幕の内容の精度よりも字幕提示のタイミングの方が内容理解に対する影響が大きいことが示されている。これら既往研究調査の結果から、発話から字幕提示までの所用時間を1秒以下とすることが望ましいことが分った。

次に発話から字幕提示までの時間を短縮するための手法について検討した。要約筆記時には、発話から字幕提示まで一定の時間を要することが知られている。有海の研究[3]により、発話から字幕の提示まで、入力の違い要約筆記で4秒台、入力の遅い要約筆記で5秒~10秒程度の遅延が生じることが示されている。複数話者による発話の場合、これに加えて話者識別のための時間が加わるため、より多くの時間を要することが想定される。そのため、複数話者の音声を時間差1秒以内で提示するには、要約筆記では不可能であると判断した。

次に音声認識による字幕化について調査した。本研究は学術的意義とは別の目的として、開発した手法やアプリケーションを無償で公開することを目的としている。そのため、調査にあたっては、無償で音声認識エンジンを利用可能であることを重視した。まず無償で使用することのできるGoogleドキュメントによる音声認識を試したところ、発話から1秒以下で字幕が提示されることが分った。しかしながら、Googleドキュメントを用いた手法は、複数話者の発話を区別して提示できない。また、Googleドキュメントの本来の目的が文書作成を目的としているため、テキスト提示時に連続した文章となり、複数話者の発話を提示するUIとしては適さない。

そのため、Googleドキュメントと同等の音声認識速度を有する別の手法を検討した。音声認識による日本語字幕提示ソフトウェアとして、UDトーク[4]がある。UDトークは日本では広く活用されている聴覚障害者支援ソフトウェアである。個人利用については無償で利用することができる。発話から1秒~2秒程度で字幕提示が可能であるが、発話から字幕が提示されるまでの速度はGoogleドキュメントより遅い。また、UDトークはiOS、Androidアプリケーションであり、マルチプラットフォームに対応したフロントエンドを持たない。PC環境においては他のソフトウェアとの連携を前提としており、必然的にプラットフォーム毎に異なるUIで活用することとなる。加え、iOS、Androidアプリケーションにおいても、複数話者に対応したUIは不十分であり、複数の発言者を区別して把握しにくい。そのため話者が交代する時にはUDトークのインストールされた機器を回し使う等の利用が行われ、音声認識に要する時間とは異なる理由で情報伝達に時間を要する場合がある。以上より、話者を識別しやすいUIを伴い、複数話者による日本語音声遅延なく字幕として提供できる、無償のマルチプラットフォームに対応したアプリケーションを開発す

ることとした。

次に、聴覚障害者が複数の健聴者に向けて発信するための手法について検討した。複数の健聴者との会話において、自然な流れで聴覚障害者が会話に入るための手法の検討である。口話のできる聴覚障害者は発話することで会話に参加しやすい。一方、口話を用いない場合には挙手等によって会話を遮らないと会話に参加することができない。また、聴覚障害者が健聴者から情報を受信する場面における音声と字幕のタイミングのズレは、聴覚障害者が健聴者の集団に対して発信する場面においても問題となる可能性がある。

精神的ストレスという側面では、利用時とは異なる問題も存在する。導入に関する負荷である。これまで述べたような聴覚障害者の受発信時のストレスを解消できたとしても、利用開始までのコストが大きいと、健聴者に活用を促しにくく、活用されにくい。本研究は冒頭で述べたように、健聴者の集団における聴覚障害者のコミュニケーション上の困難な課題を解決することを目的としているため、導入が容易である点や、特別なデバイスを求めない点も重要であると考えた。

また、日本語特有の問題についても検討した。日本語の音声認識時には、音声を正しく認識するだけではなく、日本語特有のかな漢字変換を経て最終的に文字化される。西欧語の場合とは異なり、仮に音声を音として正しく認識していたとしても、正確にかな漢字変換されない場合があり、西欧語よりも多くの修正が必要となる。このため、日本語の音声認識時には、誤りを容易に修正できるUIを実装することで、変換精度の問題を緩和できるのではないかと考えた。

また、聴覚障害者に対する字幕提示時には、第一筆者のこれまでの研究から、一定量の字幕をログとして提示することで、時間を遡って確認でき、内容理解が深まることが分っている[5][6]。そこでユーザの求めに応じて余白量を調整し、文字をできるだけ多く画面内に提示できるようにUIを検討することとした。

以上の要件を全て満たすソフトウェアを設計、開発した。開発時に重視した点を要約する。以下の6点である。

- 1) 発話から字幕提示までの所要時間が1秒以下であること。
- 2) 複数話者の音声を識別しやすいUIを有すること
- 3) テキスト入力を可能とするなど、口話を行わない聴覚障害者が発言しやすいUIを有すること
- 4) アプリケーションは特別なデバイスを要求せず、スマートフォンやPCなど、環境を問わず様々なプラットフォームにおいて同様のUIで動作すること。また、容易に導入、活用できること
- 5) 確定後の再編集がしやすいUIを有すること
- 6) 余白量を調整できるなどし、一定量のテキストを提示できるUIを有すること

3. アプリケーションの開発

アプリケーション開発にあたり、はじめに音声認識エンジンについて検討した。日本語の音声認識し、文字情報に変換できる仕組みは複数存在する。開発に着手した2019年2月時点では、IBM Watson Speech to text, Microsoft Azure Speech to Text, AmiVoice, Google Cloud Speech to Text などがあった。上記の内、Google Cloud Speech to Text は月間60分という制限内であれば無償で利用できるが、これらの手法は基本的に全て有料であり、容易に活用できない。

次に検討した手法はブラウザであるGoogle Chromeに内蔵されている音声認識エンジンの活用である。Google Chromeの音声認識であれば、利用時間に関わらず無償であり、活用は容易である。Google Chromeを利用する必要があるものの、もしGoogle Chromeがインストールされていない場合に、同ソフトウェアをインストールする以外のコストを必要としない。また、Webアプリケーションとすることで、前章で述べたマルチプラットフォーム間での動作を統一しやすく、開発や導入に関わる負担を大幅に軽減させることができる。このことから、本ソフトウェアはWebアプリケーションとして開発した。音声を認識して字幕表示するだけの基本機能を持つプロトタイプを作成してテストしたところ、音声認識から字幕表示されるまでのタイムラグは1秒以下であり、ほぼラグのなく音声を文字に変換可能であることが分った。そこで話者それぞれのPCやスマートフォン上の本アプリケーションをWebソケットで繋ぐ仕様と、複数人が同時に使える状態とした。また、本アプリケーションを利用するためのユーザ管理は独自の仕組みを導入するのではなく、各種のソーシャルアカウントと連携させることで、ソーシャルログイン可能とした。本仕様により、本アプリケーション側においては、ユーザ管理の仕組みを開発する必要がなくなる。利用者側にとっては、新たにユーザ登録する必要がなく、いずれかのソーシャルアカウントを所持している場合には、ユーザ登録の必要がなく直ぐに利用が可能となる。もしユーザがアカウントを所持していない、あるいはアカウント連携に抵抗感を持つ場合には、いずれかのソーシャルサービスで新たにアカウントを作成してもらうこととした。以上の仕様により、2章で述べた6つの条件の内、項目番号1番と4番を同時に満たすことができると判断した。

次に本アプリケーションのUIについて検討した。本アプリケーションは複数話者の音声を識別しやすいUIを有する必要がある。テキスト情報を主体としたコミュニケーションツールとして、Twitter, Skype, Lineなどが挙げられる。これらは時系列に従い、発言毎に枠囲いされたテキストを上下に積み重ねるUIを共通して有している。自身の発言と他者の発言については、左右に分離することで、自身の発言と他者の発言とを区別して視認しやすいUIとし

ている。しかしながら、自身と他者を区別しやすい一方で、他者を区別するのは主にアイコンのみであり、複数の他者を区別しにくい。また、これらのソフトウェアは余白を広く取る傾向があり、設定されている余白量をユーザが自由に調整できない。そこで本アプリケーションのUIは、アイコンと色によって話者を区別し、余白量をユーザが調整できる仕様とした。本仕様により、一章末尾で述べた5つの条件の内、項目番号2番と6番を満たすことができる。

次に発話のできない聴覚障害者が発言しやすい仕組みを検討した。口話ができる聴覚障害者は多いが、音声認識可能な発話ができる聴覚障害者は比して少なく、また発話に抵抗感を持つ聴覚障害者もいる。そのため、音声認識に加え、テキスト入力をUIに加えることとした。また、このテキスト入力UIに音声認識された結果を戻し、テキストを再編集可能な仕様とした。本仕様により、一章末尾で述べた6つの条件の内、項目番号3番と5番を同時に満たすことができる。

以上の仕様により、当初要件を全て満たしたプロトタイプを開発した。開発したプロトタイプのログイン直後の画面を図1で示す。また、ルーム入室後の画面を図2で示す。



図1 ログイン直後の画面
Fig. 1 Screen image of after log in



図2 ルーム入室後の画面
Fig. 2 Screen image of after entering the room

本アプリケーションの利用方法について説明する。ユーザはソーシャルアカウントの一つである Google アカウントを利用してログインし、フロントページでルームを設定する。図 1 はルームがひとつだけ設定された状態を示している。多数のルームを設置することもできる。ルーム設定後「招待コード (URL)」を発行し、他のユーザを設定したルームをクリックのみで参加させることができる。ユーザを識別しやすくすることは本アプリケーションの開発において重要であるため、ゲスト利用であっても、自身の発言を表示する時の背景色や文字色などは自由に設定できる仕様とした。図 2 はルームの中に 3 人が参加し、会話している様子を示している。ゲストユーザではルームに参加することはできるが、ルームは設置できない。「テキスト入力エリア」は文字入力のために利用する。一般的なテキストエディタと同様である。プロトタイプ時点ではエンターによる確定ができず、設定した時間の経過を待つ、あるいは「決定する」ボタンを押して入力を確定する仕様となっていた。「録音開始」ボタンはトグル方式になっており、押すと音声認識が始まり、ボタンは認識中を意味する反転表示となる。音声認識された結果は「テキスト入力ウインドウ」に戻り、ユーザが設定した時間だけウインドウ内に留まる。その間に結果を修正できるようにした。「決定する」ボタンを押すか、設定した時間経過によって入力が確定されると、変換後の文字が下部のメインウインドウに表示される。同時に反転されていた「録音開始」ボタンが元の状態に戻る仕組みである。つまり、プロトタイプ開発時点では、発話毎に「録音開始」ボタンを発話する度に押す必要があった。加え、プロトタイプ時点では音声認識された結果が「テキスト入力ウインドウ」に表示されている間は他者が入力内容を確認できず、決定後に共有される仕組みとなっていた。

開発後、プロトタイプを用いて検証を行った。聴覚障害者が就業する企業 3 社に依頼し、以下の状況で試用した。試用後、ヒアリングを実施した。本アプリケーションは複数の健聴者に少数の聴覚障害者が交じり会話する場面で活用することを想定しているが、プロトタイプの検証では、反対に少数の健聴者が複数の聴覚障害者に対して説明する場面でも検証した。

- 1) 複数の健聴者 (2 名) と聴覚障害者 (1 名) が会話する状況
- 2) 複数の健聴者 (5 名) と複数の聴覚障害者 (2 名) が参加する会議
- 3) 1 人の健聴者が複数の聴覚障害者 (9 名) に対してプレゼンテーションする状況

検証の結果、以下の改善点を得た。

- ・ 録音開始を毎回押さずに連続して変換される仕組みが欲しい。
- ・ 最上行だけではなく、テキスト入力ウインドウにも、

変換中の状態を表示して欲しい。

- ・ 「決定する」ボタンに加え、エンターでも入力確定できるようにして欲しい。
- ・ 「録音開始」という表現が適切ではないので、別の表現に変えて欲しい。
- ・ かな漢字変換前のテキストを確認できれば、多少の誤認識や誤字を補完できる。
- ・ 音声変換処理が上手くいかず、処理待ちになることがあるので、処理を強制的に中断できるようにして欲しい。
- ・ Web アプリのため容易に使えるのだが、招待コードが長いので、入力するのは現実的ではなく、他の通信手段を用いてコードを受け取る必要がある。簡便な方法で入室できるようにして欲しい。

上記回答の内、「かな漢字変換前のテキストを確認できれば、多少の誤認識や誤字を補完できる」という意見について補足する。本意見は上記の検証の 1 において、2 名の健聴者と 1 名の聴覚障害者が会話する状況において得られた。検証時には、1 名の健聴者のすぐ横に聴覚障害者が並び座り、健聴者のノート PC 上で動作するプロトタイプを聴覚障害者が確認できる状況下で実施した。プロトタイプ時点では、発話者の環境にはかな漢字変換前のテキストがリアルタイムで表示されているが、他者へは変換確定後にテキスト情報が共有される仕様となっていた。そのため、他者はかな漢字変換前の状態を確認できず、かな漢字変換後の結果しか確認できない。そのため、かな漢字変換前のテキストも共有できるようにすることで、誤変換が生じても正しい内容を推測し、補完しやすくなるのではないかと考えた。これらの意見をフィードバックし、プロトタイプを改修した。

4. プロトタイプの改修

プロトタイプによる検証後、ヒアリング結果をとりまとめてフィードバックした。プロトタイプに以下の機能を追加して改修した。

- 1) かな漢字変換前のテキストを共有できる機能
- 2) 音声認識時の処理待ちを強制終了する機能
- 3) エンターで入力確定とする、ユーザが設定を変えた場合に過去の発言分についても遡ってその設定を適用するなど、UI の強化
- 4) 連続で音声認識し、設定した無音時間ごとに区切って変換する機能
- 5) Google アカウントに加え、Twitter アカウントによるソーシャルログイン機能
- 6) 短縮コード及び QR コードによる招待機能

以上の改修を行った後、アプリケーションをリリースした。

5. まとめ

本研究では、聴覚障害者向けに複数の音声を即時字幕提示する UI を持つアプリケーションを開発し、効果を検証した。その結果、聴覚障害当事者から多くの改善点を得てプロトタイプを改修し、実用可能なアプリケーションをリリースした。

また、開発後の当事者による試用とヒアリングの過程において、音声の誤認識や誤変換が生じていても、かな漢字変換前の状態をリアルタイムで共有することで、誤った内容から正しい内容を推測できる可能性があることも分かった。

本研究においては、定性的に開発した手法の有効性を調査したのみであり、検証や有効性の評価は限定的である。アプリケーションは実用を開始しており、既に多数の活用事例を得ている。今後も実用を続けながら、本ソフトウェアの効果について定量的な調査を実施し、手法の有効性を検証したい。

謝辞 本研究は JSPS 科研費 19K21745 の助成を受けたものです。

参考文献

- [1] 丸山一郎, 阿部芳春, 沢村英治, 三橋哲雄, 江原暉将, 白井克彦. ニュース字幕の提示タイミングずれに対する許容特性, 電子情報通信学会技術研究報告, ヒューマンコミュニケーション基礎 99(123), pp.21-28, 1999.
- [2] 下郡信宏, 池田朋男, 関矢陽子. 英語字幕による会議支援: 字幕の精度と表示タイミングが理解に及ぼす影響, 情報処理学会研究報告, Vol.2010-GN-75 No5, pp.1-6, 2010.
- [3] 有海順子. 聴覚障害学生に対するパソコン要約筆記の特徴に関する研究: 大学の授業場面・支援者・当事者の要因から, 筑波大学, 2013, 博士論文
- [4] UD トーク. <https://udtalk.jp/> (確認 2020 年 10 月)
- [5] 鈴木拓弥. 聴覚障害学生を対象としたデザイン実技演習支援に関する研究, 筑波技術大学テクノレポート, Vol.18 No.2, pp.68-72, 2011
- [6] 鈴木拓弥, 長嶋祐二. 聴覚障害学生向け実技演習における実演履歴提示ソフトウェア SZKISS の開発と有効性の検証, 電子情報通信学会論文誌 D (情報・システム), J101-D(2), pp.359-368, 2018.02