

アクセシブルな発話発表を支援する ウェブアプリケーションの開発

馬場 哲晃^{1,a)} 塩野目 剛亮^{b)}

概要：情報保障の中でもノートテイク（または要約筆記）は、講義や発表の様子を文字起こしすることで聴覚障害当事者に同時通訳を行う取り組みとして、多くの大学や学会等で頻繁に利用されている。例えば東京都立大学では、ダイバーシティ推進室が中心となりノートテイクを行う学生アルバイトの育成支援を行っており、学内の聴覚障害学生に対して日頃よりノートテイクを実施している。このように当事者を支援する仕組みが整う一方で、ノートテイク対象者である教員や発表者も情報保障を意識した発話を実践する必要がある。しかし、普段はノートテイクのない環境に慣れているユーザにとって、単発的なノートテイク実施下での発表や授業において、必ずしも常に「聞きやすい」「文字起こししやすい」発話をするには困難な側面がある。またノートテイクと呼ばれる文字起こしユーザ側は、発表者に対してアクセシビリティ確保を目的とした発話指摘がしづらい環境にある。そこで我々は学会発表を対象として、発表者に対して即時に発話スピードや発話内容のフィードバックを自動で提供することで、アクセシブルな発話を支援するウェブアプリケーションを開発した。これにより、発話支援はもとより、ノートテイクが指摘せずとも発表者の自発的な発話改善を促進すると期待できる。

1. はじめに

情報保障の中でも要約筆記は聴覚障害者当事者への情報保障手段の一つであり、支援者を要約筆記者等と呼ぶ。要約筆記の手法に応じて、ノートテイク、OHC テイク、パソコン要約筆記等と呼び方が細分化できるが、本稿ではパソコンを利用した要約筆記を対象として話を進める。

有海ら [1] は、講義場面における要約筆記結果を詳細に分析し、発話量の 4~6 割程度が文字に変換されていたことや、音声-文字変換時の平均タイムラグが 3~8 秒程度であったこと等、要約筆記における特徴を幅広くまとめている。要約筆記者に必要な文字入力速度は一般的に 100~120[CPM] (Character Per Minute, 本稿では日本語における 1 文字を Character として考える) とされており、先の発話量に対する変換率が 4 割の場合、入力速度 120[CPM] の要約筆記者が入力可能な発話速度は 300[CPM] 程度と概算できる。300[CPM] は NHK がニュース原稿作成に利用する発話速度として広く知られており [2], 300 文字はかな

漢字混じり文を一文字ずつカウントしたものである。文字換算基準において、このような文字数やモーラ数、漢字をよみがな置き換えてのカウント等が考えられるが、本稿では 300[CPM] という指標を利用者側が目指しやすいよう、かな漢字混じり文を 1 文字ずつカウントすることとした。

以上より、標準的な要約筆記支援が行われている授業や発表において、発表者は 300[CPM] 程度での発話が望まれることがわかるが、よほどの早口にならない限りは発話速度を他者に指摘されることは少ない。しかし、坂本らはニュース番組におけるアナウンサーの発話を 1.5 倍速で提示した場合、人工内耳及び補聴器装用者は健聴者群と比較し、聴取能力が有意に低下することを示しており [3], 要約筆記者のみならず、難聴者に対しても適切な速度で発話することは重要である。

発話速度以外にも、フィラー（有声休止）や長文は文章要約の妨げになるため、要約筆記において発表者は意識する必要がある他、視覚障害の観点からは「これ」、「ここ」等の指示代名詞も極力利用しないことが望まれる。以上の背景を踏まえた上で先行事例との差異に触れつつ、本システムの開発を本稿では述べる。

2. 先行事例

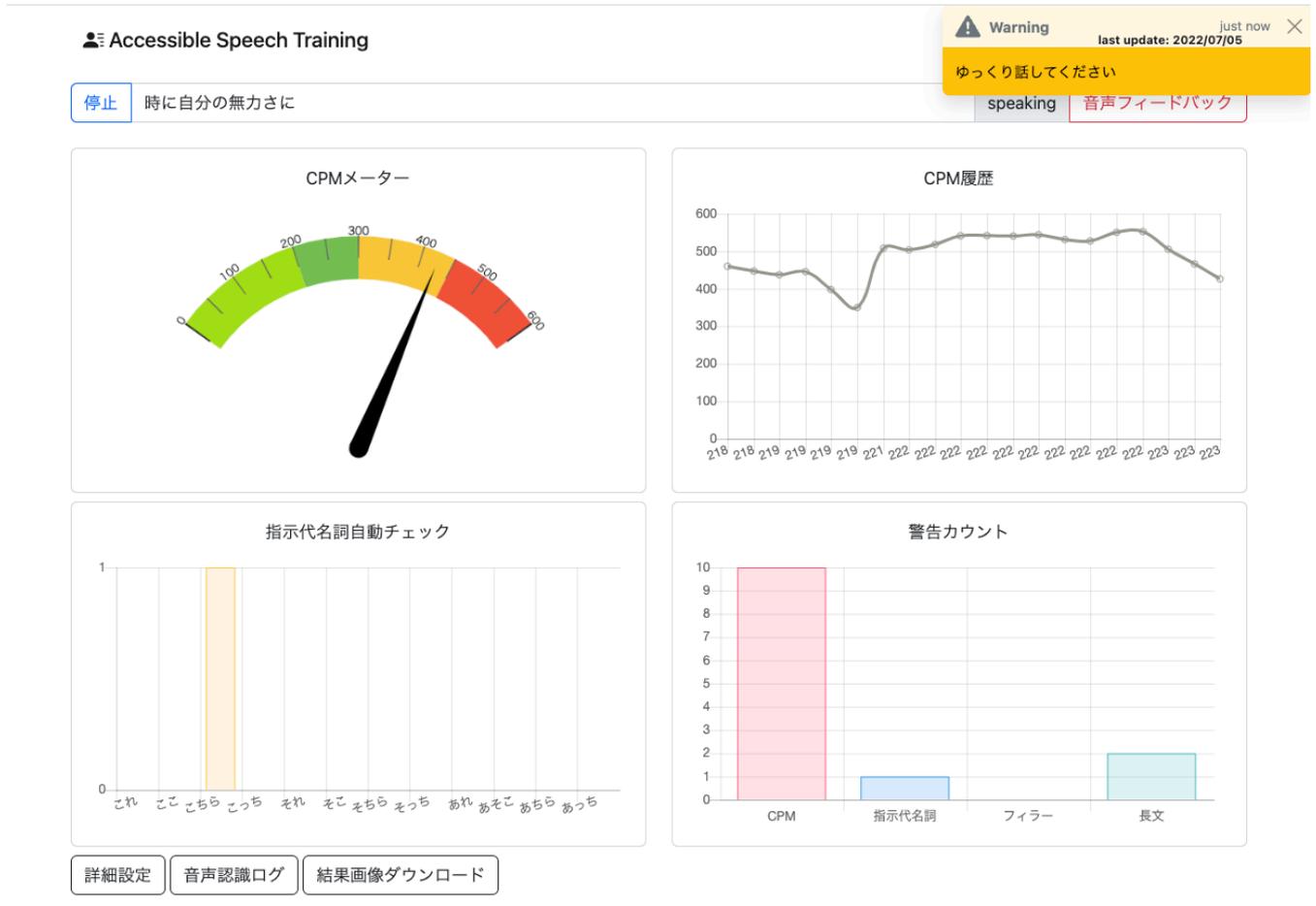
アクセシブルな発話支援の観点から、それらを機械シス

¹ 馬場哲晃
Tokyo Metropolitan University, 6-6, Asahigaoka, Hino, Tokyo, 191-0065, Japan

² 塩野目 剛亮
Teikyo University, 1-1, Toyosatodai, Utsunomiya, Tochigi, 320-8551, Japan

a) baba@tmu.ac.jp

b) shionome@ics.teikyo-u.ac.jp



© 2022 Tetsuaki Baba All rights reserved.

図 1 開発した Accessible Speech Training の画面. 画面上部に認識中文字列が表示されている. 左上:一分間の発話文字数を示す CPM(Character per Minute). 右上: CPM の時系列データ. 左下:発話された具体的な指示代名詞. 右下: CPM (設定値よりも早い話速で話した場合にカウント). 図右上に黄色背景で CPM 警告が出現している, 指示代名詞 (左下でカウントされるたびに増加), フィラー (フィラーが発話されるたびにカウント), 長文 (指定時間文の区切れがなく話が継続している場合のカウント).

テムによって支援する先行事例は少ないが, プレゼン支援の観点からはいくつかの先行事例が存在する.

栗原らによる「プレゼン先生 [4]」は, 魅力あるプレゼンテーションを支援するシステムである. 一般書におけるプレゼン評価指標を参考にアイコンタクト, 発話速度, 言いよどみ (フィラー), 抑揚をリアルタイムに検出し, 発表者にフィードバックする. 本研究では特に警告表示手法を参考にした. 実際に言いよどみが生じた際, プレゼン先生では画面上に即座に警告が表示される. 魅力的なプレゼンテーションを目的としている観点で本研究とは異なるが, 本研究をインクルーシブデザインの観点から実践していく場合に有益な知見を多く含んでいる.

杉山ら [5] は, 就職活動を控えた学生を対象に, 話速を定量的に評価・改善するセルフチェックサービス「Wasokun」を開発した. チェック項目を話速のみに限定し, 面接終了後に解析結果をフィードバックするシステムであり, 話者

への即時的なフィードバックはしない.

Hikaru Tsuyumine による「PresentationSpeed」は iOS アプリとしてリリースされており*1, 録音した音声データから文字数及び読み上げ速度をバッチ処理によって計算する. 先の Wasokun 同様に発話の最中においてユーザーフィードバックがないため, リアルタイムアプリケーションとしての対話性が乏しい. 一方でスマートフォン等のモバイル端末で動作完結する点を本研究では参考とした.

以上の先行事例を基に, 本稿では以下の通り開発するシステムの仕様を定めた.

- (1) 手軽さを考慮し, 様々な端末で動作すること
- (2) ユーザの発話に対して即時的フィードバック (発話速度, 指示代名詞, フィラーに関する警告を含む) を提供すること

*1 <https://www.presentationsspeed.site/>

3. 実装

先に掲げた仕様 (1) を実現するにあたり、多くの端末で動作する観点から JavaScript を利用し、ウェブブラウザで動作することを前提とした。本研究では自動音声認識が必須となるが、自動音声認識はすでに Web Speech API として W3C によって策定されている。2022 年 7 月 6 日現在ではいまだドラフトではあるが、Chrome 及び Safari においてすでに機能が実装されているため、音声認識エンジンには Web Speech API を利用する。ただし、Chrome, Safari 間において動作仕様に細かな違いがあるため、本稿では Chrome を動作保障対象として設計している。実装には Web Speech API を使いやすくした p5.speech*2 に一部修正を加えたライブラリを利用した。音声認識については誤りが含まれる可能性があるが、本研究は意味解釈をシステム仕様上必要としないため、認識誤りテキストはそのままで解析処理を行う。結果として、ユーザが発話した内容が完璧に解析されているわけではない。音声認識結果は、途中経過をふくめ適時認識結果をプログラム内で参照できるため、音声認識開始時刻を t_0 (単位は [ms])、時刻 t_1 における認識文字数を n 、CPM のセンシング感度を S とした場合、1 分間あたりの文字数 CPM を以下の数式で定義する。ただし CPM' は前回計算時の値、 n は小さいほど CPM 誤差が大きくなるため、実装時は $n \geq 20$ 、 S は 10 としている。

$$CPM = CPM' - \frac{CPM' - n \times (\frac{1000 \times 60}{t_1 - t_0})}{S}$$

次に仕様 (2) を実現するにあたり、発話内容から指示代名詞やフィルター等を検出するため、JavaScript 上で形態素解析が利用できる kuromoji.js*3 を用いた。kuromoji の形態素解析精度に関しては公式ウェブサイト*4 を参照されたい。

実装したウェブアプリケーション [6] を図 1 に示す。利用手順は以下の通りとなる。

- (1) 指定リンク*5 をブラウザで開く
- (2) 開始ボタンを押し音声認識をスタート
- (3) 適時認識結果が各グラフや警告にて表示される
- (4) 停止ボタンを押してシステムを止める
- (5) 必要に応じて結果画像や音声認識ログをダウンロードする

3.1 その他機能

図 1 左下にそれぞれ「詳細設定」「音声認識ログ」「結果画像ダウンロード」ボタンを用意した。「詳細設定」ボタンを



図 2 詳細設定項目

表 1 テスト結果一覧

	1	2	3	4
-r[say command wpm option]	133	157	186	210
target CPM	250	300	350	400
actual CPM	254	302	348	401
last measured CPM	256	303	345	402
average measured CPM	236	278	327	372
accuracy (last/average) [%]	99/94	99/92	98/93	99/92

押すと、警告基準となる CPM と長文の基準となる秒数を個別設定できる (図 2 参照)。初期設定では CPM は 400、長文警告は 10 秒以上としている。「音声認識ログ」ボタンは音声認識結果がテキスト形式で表示され、ダウンロードできる。結果画像ダウンロードボタンでは 4 つのグラフを画像としてダウンロードできる。

4. ベンチマークテスト

実装したシステムに関して、音声認識精度や形態素解析そのものに関しては利用 API やライブラリに依存するところであるが、CPM の精度に関しては一定の信頼性評価が必要と考え、ベンチマークテストを行った。ベンチマークテストの目的は、本システムにて算出される CPM がどの程度妥当な数値を算出しているかを検証する。

4.1 方法

macOS にて利用可能な say コマンドを利用し、任意の文章に対して任意の CPM にて音声読み上げを行い、本システムにより CPM を計算し、設定値との誤差を検証する。ただし say コマンドでは word per minute の設定はできるが日本語に対応した速度設定ができない。say コマンドに関して 2 バイト文字の場合にどのような word per minute を算出するのかに関する資料を見つけないことができなかったが、オプション値を変更することで発話スピードは変更できるため、実際にいくつかのパラメータを試しつつ実測値が設定 CPM となった速度で計測を行った。

4.2 実験

測定する発話速度は 250,300,350,400[CPM] とし、それぞれの文章は夏目漱石「ぼっちゃん」[7] の冒頭 1 段落目、

*2 <https://github.com/IDMNYU/p5.js-speech>
 *3 <https://github.com/takuyaa/kuromoji.js/>
 *4 <https://www.atilika.com/ja/kuromoji/>
 *5 <https://tetsuakibaba.jp/project/accessbileSpeech/>

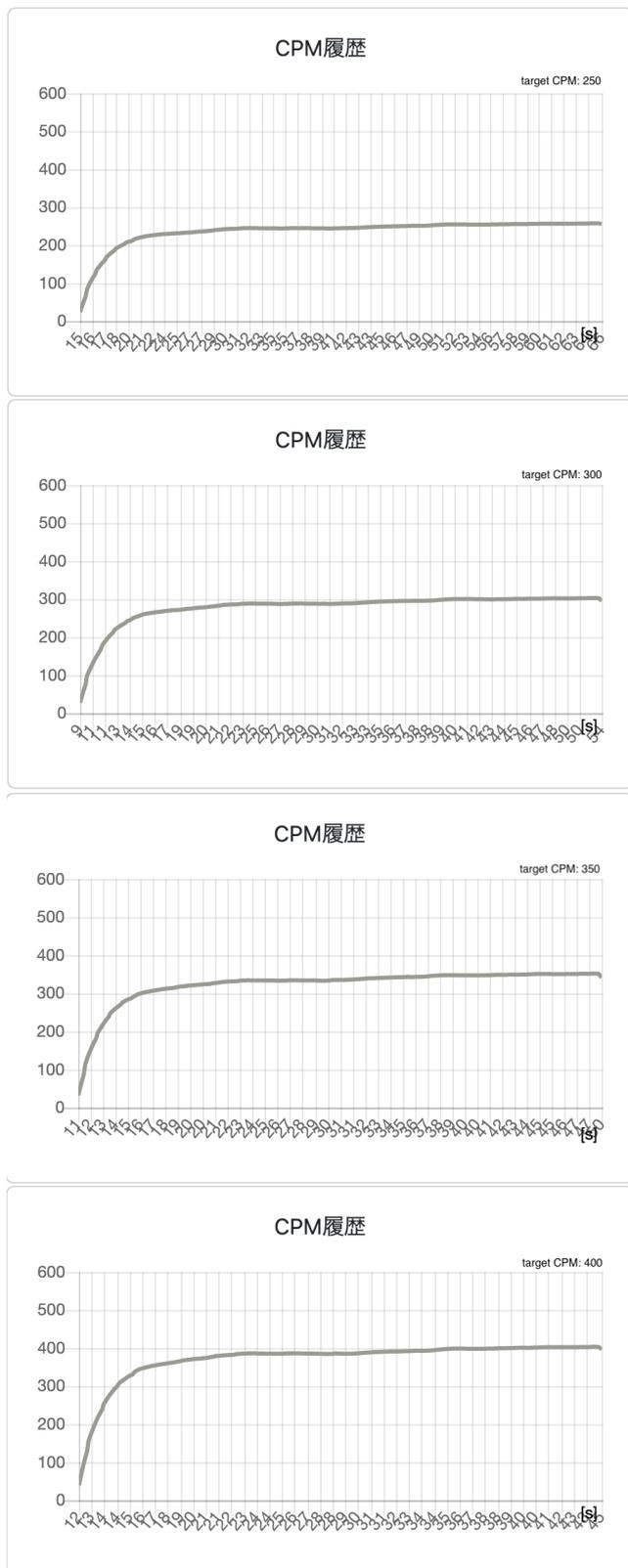


図 3 上から順に 250,300,350,400[CPM] をターゲット CPM に設定した際の音声自動読み上げによるシステム計算結果

全 237 文字を利用した。なお音声読み上げで利用した say コマンドにおいて、読み上げ音声は kyoko に設定している。実際に利用した音声テキスト、データは本研究 github ページ [6] を参照されたい。それぞれのテスト条件を表

1 に示す。-r は say コマンドにおけるオプションであり、wpm[word per minute] を設定できるが先に述べたとおり、2 バイト言語に関してはどの単位が word となるのか基準が示されていないため、実測 (60 * (全文字数/再生時間)) にて actual CPM を算出している。実際に読み上げされた文章を本システムにて計測し、文章終わり時の CPM を last measured CPM, 読み始めから読み終わりまでの計測された CPM の平均値を average measured CPM とした。average measured CPM は前回計算値との差分による急激な変化に対する感度調整により、237 文字の短い計測において音声認識開始時の CPM 上昇が抑えられてしまうため (図 3), last measured CPM に対して accuracy が劣るが、last measured CPM は 98 – 99[%] と非常に精度が高いことがわかる。

認識時においても CPM が対話的に大きく揺らぐことがなく、本実験環境下においては安定して CPM を算出できている (図 3 参照)。図 3 は上から順に 250,300,350,400[CPM] をターゲット CPM とした時系列計測結果である。

5. おわりに

今後は各種発表会や学会での運用において、本ウェブアプリケーションを利用することを目指す。継続した実験やベンチマークテストを通じてアプリケーションの改善に努め、アクセシブルな発表支援に寄与したい。

謝辞 本アプリケーションの開発にあたり、ご意見をいただいた情報処理学会アクセシビリティ研究会運営委員の皆様にご感謝いたします。

参考文献

- [1] 有海順子, 四日市章: 聴覚障害学生へのパソコン要約筆記における音声-文字変換の特徴, 障害科学研究=Japanese journal of disability sciences / 障害科学学会編集委員会 編, Vol. 33, pp. 55-66 (オンライン), 入手先 (<https://cir.nii.ac.jp/crid/1520853833250730496>) (2009).
- [2] 矢野 香: 【NHK 式+心理学】 一分で一生の信頼を勝ち取る法, ダイアモンド社 (2014).
- [3] 坂本 圭, 小淵千絵, 城間将江, 松田 帆, 関恵美子, 荒木隆一郎, 池園哲郎: 聴覚障害者における早口の音声聴取に文の有意性が及ぼす影響, *AUDIOLOGY JAPAN*, Vol. 61, No. 3, pp. 209-215 (オンライン), DOI: 10.4295/audiology.61.209 (2018).
- [4] 栗原一貴: プレゼン先生: 音声情報処理と画像情報処理を用いたプレゼンテーションのトレーニングシステム, 第 14 回インタラクティブシステムとソフトウェアに関するワークショップ, *WISS*, 2006, pp. 59-64 (オンライン), 入手先 (<https://cir.nii.ac.jp/crid/1574231876075581440>) (2006).
- [5] 杉山 渉, 中村亮太, 上林憲行: クラウド型音声認識 API を用いて適切な話速を定量的に評価・改善するセルフチェックサービス, 情報処理学会全国大会講演論文集, pp. 4.797-4.798 (オンライン), 入手先 (https://jglobal.jst.go.jp/detail?JGLOBAL_ID=201402209439600770) (2014).
- [6] BABA, T.: *Accessible Speech Trainer*, github.
- [7] 夏目漱石: 坊っちゃん, 筑摩書房 (青空文庫).