# 発話型属性認証に向けた提示文章ごとの発話特徴分析

井戸 智斗志[1,a)]　石田 繁巳[2]　稲村 浩[2]

**概要**：インターネットサービスでは，Bot による自動操作を防止するため，人間と Bot を識別する CAPTCHA 認証が用いられている．画像認識技術の進歩により Bot が高精度に CAPTCHA 認証を突破するようになり，Bot に突破されない新しい CAPTCHA 認証技術の研究開発・利用が進んでいる．本研究では誤記を活用した発話型属性認証システムを提案し，その実現に向けて必要となる，提示文章ごとの発話特徴の分析を示す．誤記を含めた文章を提示すると Bot の発話速度は一定であるのに対し，人間は詰まりや言い淀みにより発話速度が一定ではないと考えられる．このような人間的発話特徴が発生する文章を特定するため，本稿では複数種類の誤記のそれぞれを含む文章を提示し，人間と Bot の発話データを収集して分析を行った．その結果，認証文の中央で，文節の 5 文字目に「結合」の誤記を加えることで 74%識別できることを確認した．

## 1　はじめに

インターネットの利用が拡大する中で，悪意のある Bot による Web サイトへの攻撃が発生している．例えば，悪意のある自動化された Bot が大量のアカウントを作成するアカウントの不正作成が挙げられる．作成したアカウントは，詐欺やスパムの発信元として利用される．悪意のある Bot は経済的損失を引き起こすため，Bot による攻撃を防ぐための対策が重要になっている．

Bot による攻撃を防ぐために，操作を行っている対象が人間か Bot かを識別する属性認証が用いられる．属性認証の概念は，人工知能研究におけるチューリングテストに関連している．チューリングテストは，1950 年にイギリスの数学者アラン・チューリングによって提唱されたテスト [1] である．機械が人間と区別がつかないほどの知的な振る舞いをするかどうかを評価する．

一般に，人間と Bot を識別する属性認証では CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) 認証が用いられる．CAPTCHA 認証は，人間には理解できるが機械では解析しにくい情報を問題として提示し，問題に正解することで対象者が人間であることを証明する手法である．多くの Web サイトでは画像認識型の CAPTCHA 認証が用いられている．

画像認識技術の進歩に伴って Bot が CAPTCHA 認証を突破するようになり [2]，Bot に突破されない新しい属性認証技術の研究開発が進んでいる．例えば，機械には判別困難な文字列を用いた語彙性判断型 CAPTCHA 認証 [3] やタイポグリセミアを用いた Multi-model CAPTCHA [4] が提案されている．

しかしながら，機械学習技術の著しい発展に伴い，単に人間と機械の判断能力の差異に基づくのではなく，より本質的な人間の行動特性に着目した，新たな視点による属性認証の枠組みが求められる．深層学習モデルは，人間の認知や判断を模倣する能力を急速に高めており，従来は人間にしかできないとされていた視覚的・言語的タスクにおいても，高精度な処理が可能となっている．従来の CAPTCHA 認証のような人間判別技術が機械によって容易に突破されるだけでなく，機械が人間らしい振る舞いや応答を実現することも可能となっている．

本研究では，新たな属性認証技術として発話型属性認証システムを提案する．認証時に読み上げる認証文章に誤記を含ませ，発話リズムの変化により属性認証を行う．誤記を含む認証文章を提示することで，人間が読み上げた場合には詰まりや言い淀みなどが確認されると考えられる．本稿では，認証文章として望ましい誤記の位置や種類を特定するために，文字認識特徴と発話特徴を分析した．発話型属性認証システムに人間と Bot の発話音声を入力して評価した結果，最大 74%で識別できることを確認した．

本稿の構成は以下の通りである．第 2 節では音声を用い

[1] 公立はこだて未来大学大学院 システム情報科学研究科
[2] 公立はこだて未来大学 システム情報科学部
[a)] g2125004@fun.ac.jp

た CAPTCHA 認証と合成音声の検知に関する関連研究を示す．第 3 節では提案する発話型属性認証システムの設計を示し，第 4 節においては認証文章の特徴を特定して，提案システムの属性認証性能を評価する．最後に第 5 節でまとめとする．

## 2　関連研究

### 2.1　音声ベースの CAPTCHA 認証

#### 2.1.1　発話型 CAPTCHA

Shah らは，発話による CPATCHA 認証を提案した [5]．認証者は画面に提示される認証文を読み上げて認証を行う．システムは，自然な人間の発話か，認証文を正しく発音しているかを評価する．自然な人間の発話は GMMs，認証文の確認は Pocketsphinx と正規化 Levenshtein 距離を用いる．2 つの評価基準を共に突破した場合に，人間と識別される．識別精度は 73.7%であった．

#### 2.1.2　Deepfake CAPTCHA

Lior らは，本物そっくりな声の「ディープフェイク」と呼ばれる Bot を見分けるために D-CAPTCHA を提案した [6]．電話相手に対してちょっと変わった要求をして，答え方を評価する．例えば，歌ってみてや咳払いをしてなどを要求する．評価項目は 4 つあり，本物の声と同じか，要求を指示通りできたか，声に変なところはないか，要求をすぐにできたかである．全ての項目を突破すると人間と識別される．識別精度は 91%以上であることが確認された．

### 2.2　合成音声検知手法

#### 2.2.1　ポップノイズ検知手法

合成音声を検知する手法として，望月らは人間が発声する際に生じるポップノイズの特性を利用した検出方法を提案した [10]．ポップノイズ区間に含まれる音素情報を用いることで，声の生体検知の頑健性が向上することが分かっている．そのため，生体検知を行いやすい認証文をシステム設計に反映した．従来手法と比較して，誤受理率を約 60%低下させることができた．

#### 2.2.2　機械学習手法

Bot による音声型 CAPTCHA の突破を阻止する方法として，機械学習で合成音声を検出するアプローチが考えられる．Li らは自己教師あり事前学習モデル HuBERT に基づく合成音声検出手法 HuRawNet_modified を提案した [7]．自己教師あり学習は，未学習の特徴を抽出することが可能なことが確認されている．音声処理のための事前学習モデル HuBERT に，音声認識や音声合成などの目的タスクに合わせた追加学習をすることにより高い性能を達成することが報告されている [8]．本手法により，学習データに含まれない音声も検出できることが示されている．しかしながら，Kassis らは複数の合成音声検出手法に対して検出されない合成音声生成技術を確立することで，既存の認証システムが突破できることを示した [9]．認証文を読み上げる音声ベースの属性認証では，Bot に突破されるため，別のアプローチを取り入れる必要がある．

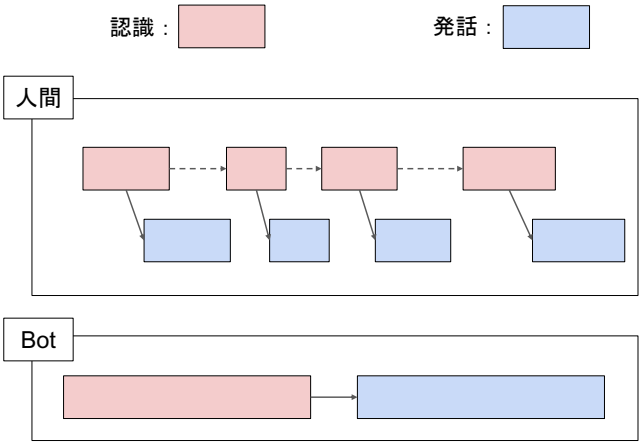


**図 1**　人間と Bot の認識から発話までの違い

## 3　発話型属性認証システム

### 3.1　キーアイデア

本システムのキーアイデアは，人間と Bot が誤記を含む文章を読み上げるときの文字認識特徴と発話特徴に基づいて属性認証することである．**図 1** に示すように人間は文章を読み上げる際に，文章を分解して認識と発話を逐次的に処理していくが，Bot は与えられた文章全体を認識して，一括で音声に変換する．提示される文章に誤記が含まれる場合，人間は文章を認識するまでに時間がかかり，詰まりや言い淀みが発生しやすく，発話リズムが一定ではなくなると考えられる．これに対し，Bot は文章内の誤記の有無で認識速度に多少の変化は生じるものの発話後への影響はなく，発話リズムは一定であると考えられる．

具体的には，文字認識特徴では，人間と Bot が文字列を認識してから理解するまでの文字認識コストに着目する．人間は誤記の有無によって文字認識コストが異なる．例えば，下記の文章を比べた場合，後者の方が認識コストが高いことが考えられる．

さくらが いろどり はるのおとずれを つげる
さらくが いどろり はるのおとれずを つげる

前者は一般的に使われる単語のみで構成されているため，人間にとって認識コストは低いと考えられる．後者は誤記を含んでおり，修正に時間がかかるため認識コストが高くなる．

発話特徴では，人間と Bot の誤記周辺における発話方法に着目する．人間は馴染みでない文章を即座に読み上げる際に，発話が流暢ではなくなると考えられる．例えば，「やしささと おもやりの こごろは せかいを あかるく てら すともふしびです」という文章を制限時間を設けて読み上げ

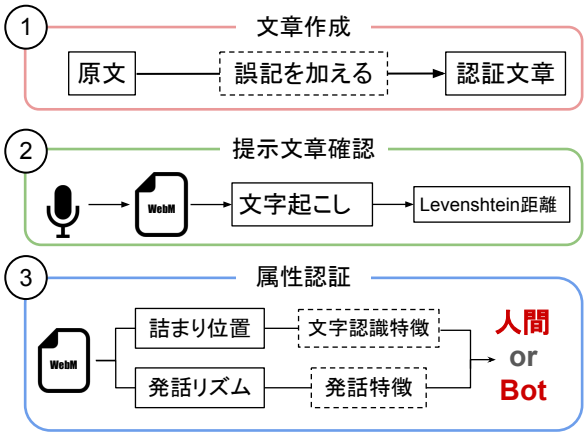


図 2　発話型属性認証システムの概要

た場合，流暢ではなくなると考えられる．Bot が読み上げた場合，機械的に流暢な発話をするため，流暢性は一定であると考えられる．

### 3.2　システム概要

図 2 に提案する発話型属性認証システムの概要を示す．提案システムは文章作成ブロック，提示文章確認ブロック，流暢性評価ブロックの 3 つのブロックで構成される．文章作成ブロックでは，あらかじめ認証文章を作成し，認証対象者に提示する．認証対象者が提示された文章を読み上げると提示文章確認ブロックはその音声をマイクを用いて取得し，文字起こしした上で提示文章と比較してその差分を算出する．流暢性評価ブロックでは，入力された音声の詰まり位置と音素時間に基づいて人間であるか Bot であるかを識別する．

以降では各ブロックについて詳述する．

### 3.3　文章作成ブロック

文章作成ブロックでは，発話によって人間か Bot かを識別可能な認証文章を作成して認証対象者に提示する．認証文章の元となる原文を用意して，適切な誤記を加える操作を行い，人間は流暢に発話できないが，Bot は流暢に発話できる文章を作成する．具体的な作成方法は，第 4 節で実験的に検討する．

### 3.4　提示文章確認ブロック

提示文章確認ブロックでは，はじめに人間または Bot による読み上げ音声をマイクを用いて取得し，音声ファイルに変換する．さらに，ソフトウェアを用いて文字起こしを行い，読み上げた文章と提示文章の類似度を計算して比較する．

類似度は，文字起こしした文章の長さ $L_d$ と元の文章の長さ $L_o$ を用いて以下のように定義する．

$$\text{類似度} = 1 - \frac{D_L}{\max(L_d, L_o)} \tag{1}$$

ここで，$D_L$ は Levenshtein 距離であり，2 つの文字列がどの程度異なっているかを示している [11]．Levenshtein 距離が小さいものほど似た文字列であることを示している．

Levenshtein 距離は，文字の挿入や削除，置換えによって 1 つの文字列を別の文字列に変形するのに必要な手順の最小回数として与えられる．Levenshtein 距離の求め方の例として，「みいえだいかく」という文字列から「みらいだいがく」という文字列に変換する方法を挙げる．以下に示すように最低でも 3 回の手順が必要になるため，Levenshtein 距離は 3 となる．

(0) みいえだいかく
(1) みいだいかく (「え」を削除)
(2) みらいだいかく (「ら」を追加)
(3) みらいだいがく (「か」を「が」に置換え)

式 (1) で定義される類似度が閾値を上回った場合は認証を継続する．本稿では，提示された文章を適切に読んでいるかを判定するため，文章全体の 50%以上が一致していれば適切に読んでいると判断し，閾値を 0.5 とした．閾値を下回った場合は，認証を棄却し，文章作成ブロックにおける文章作成から再度やり直す．

### 3.5　流暢性評価ブロック

流暢性評価ブロックでは，入力された音声が誤記を含む文節で非流暢な発話となっているかを，詰まり位置と音素時間の 2 つで評価する．誤記を加えた位置は，文章作成ブロックから受け取る．

詰まり位置の評価では，入力された音声の中で詰まりのある位置が誤記を含む箇所であるかを評価する．詰まりを無音区間と定義して検出する．スペクトルサブトラクション法により入力された音声から雑音を除去して，短時間エネルギーを計算する．そして，平均短時間エネルギーがあらかじめ設定された閾値を下回った区間を無音区間とする．無音区間の終了時刻から 2 秒間の音声を文字起こしし，誤記を含む文節の文字列が確認された場合に誤記による詰まりと判定する．本稿では無音区間判定の閾値は 10%とした．

音素時間の評価では，入力された音声の誤記を含む文節の音素時間が平均音素時間より長いかによって流暢性を評価する．音素とは音声学における最小単位である具体的単音である [18]．音素時間は音素解析を行い，各音素の時間を計測する．そして，音声全体及び誤記を含む文節に含まれる音素のそれぞれについて平均値を計算し，誤記を含む文節の平均音素時間の方が長い場合は誤記による非流暢であると判定する．

最後に，詰まりがある，あるいは誤記による非流暢である場合は人間と識別し，それ以外の場合は Bot と識別する．

図 3　録音環境

## 4　実験的模索

提案システムを実現するためには，人間と Bot とで発話特徴が異なる認証文章を作成することが重要である．認証文章は，Bot が簡単に認識して流暢に発話できる一方で，人間には認識しづらく，流暢に発話することが難しい文章にする必要がある．しかし，誤記を加える位置や種類によって，どのような認識や発話の違いが発生するのかに関する論文はない．

そのため，複数の誤記タイプを用意して実験的に模索した．具体的には，文章中の誤記の位置や種類によって人間と Bot のそれぞれでどんな発話特徴が現れるのかを実験的に評価する．さらに，模索した結果から分かった提示するに望ましい文章を用いて，再度人間と Bot の音声データを収集する．提案システムで示した流暢性評価を用いて識別可能であるかを評価する．

### 4.1　実験準備

#### 4.1.1　音声データの収集

**図 3** に，人間の音声データの収集環境を示す．人間の音声データは，Apple MacBook Air (2020) に FIFINE AmpliGame-A6VW マイクを接続し，データ収集用に実装した Web アプリケーションを用いて収集した．Web アプリケーションは，録音ボタンが押されたら読み上げ文章を表示するとともに，10 秒間の録音を行うように実装した．録音の残り時間が被験者に見えるように，録音の残り時間をカウントダウン表示した．被験者には録音ボタンを押した後に表示される読み上げ文章を 10 秒以内に読むように指示を与えて音声データを収集した．

Bot の音声データは，文章のテキストデータから自動音声として得た．LLM (Large Language Model) に誤記を含む文章を与えて誤記を修正させた上で，自動音声を用いて音声データに変換した．

#### 4.1.2　認証文章のデータセット

実験的に文字認識特徴と発話特徴を見つけ出すために，誤記の位置や種類が異なる文章を作成した．文章の中でどこにどんな誤記があるのかを明確にすることが重要であることから，文章を分かち書きにして，文節数をもとに文章を 3 等分し，3 分割した文章のそれぞれに 5 種類の誤記をそれぞれ加えた文章を作成した．そのため，1 つの文章から誤記を含む文章 15 個を作成した．

誤記を含む文章の具体的な作成工程は以下の 5 つである．

(1) 17 文字程度の文章を生成
(2) 文章を分かち書きに変換
(3) 文章を全てひらがなに変換
(4) 文節数を基に 3 等分し，文章を前方，中央，後方に分割
(5) 指定した位置の，文字数が多い文節に誤記を加える

工程 (1) では，誤記を含む文章のもととなる文章を用意した．「17 文字程度」は，文章を 3 等分にした際に，それぞれに文節が 2 つ含まれることを想定している．17 文字は，俳句の文字数と同じであり，3 つに分割するアイデアのもととなっている．

工程 (2) では，工程 (1) で用意した文章を分かち書きに変換した．分かち書きには，オープンソフトウェアの MeCab [*1] (辞書：mecab-ipadic-NEologd) を用いた．MeCab は，日本語のテキストを単語や文節などの形態素に分割し，それらに品詞や活用形などの情報を付加するための形態素解析エンジンである．文節の切れ目は，品詞が名詞，動詞，副詞，感動詞，形容詞，形容動詞，連体詞，接頭詞のとき，その単語の始まりを文節の切れ目と定義した．文節の切れ目には全角スペースを挿入し，1 つの文章にした．

工程 (3) では，分かち書きした文章をすべてひらがなに変換した．漢字をひらがなに変換するために MeCab を用いた．

工程 (4) では，文章の文節数を基準に 3 等分し，それぞれを前方，中央，後方の 3 つの部分に分割した．この分割は，分析目的の「どこにどんな誤記があるか」の「どこ」に該当する．

工程 (5) では，指定した位置（前方，中央，後方）の中で文字数が最も多い文節を選び，最初と最後の文字以外に誤記を 1 つ加えた．誤記の種類は，**表 1** に示す 5 種類であることがわかっている [12]．しかし，英語における誤記であるため，日本語に適用する必要がある．**表 2** に日本語に適用した誤記を示す．

誤記を含む文章のデータセット作成において，17 文字程度のランダムな文章は LLM を用いて生成した．使用した LLM は，Google DeepMind が開発した Gemini 1.0 Pro(以下 Gemini) である．データセット作成に向けて，Gemini を使用して 100 個の文章を生成した．1 つの原文に対して

[*1] MeCab: https://taku910.github.io/mecab/

表 1　誤記の種類 [12]

| Class | Target Word:*AIRPLANE* |
|---|---|
| Transposition | AIR**LP**ANE |
| Erasure | AIR_LANE |
| Substitution | AIR**K**LANE |
| Deletion | AIRPLNE |
| Insertion | AIR**F**PLANE |

表 2　誤記の種類の日本語適用

| 誤記の種類 | 日本語例: はなびらが |
|---|---|
| 入替 | はな**らび**が |
| 削除 | は　びらが |
| 代用 | はな**ひ**らが |
| 結合 | はびらが |
| 挿入 | はなびら**は**が |

誤記の位置と種類が異なる 15 タイプの文章を作成した．15 タイプとは，誤記の種類が 5 つ (入替，削除，代用，結合，挿入) ×誤記の位置が 3 つ (前方，中央，後方) であることを示している．そのため，100 × 15=1500 個の文章を作成した．

### 4.2　実験対象

人間の被験者は，20 代の大学生 15 人である．各被験者には，100 個の文章のそれぞれに対して 15 種類の誤記タイプが均等に含まれるように構成された文章群を提示した．100 個の文章から生成された 15 タイプの誤記パターンを 15 通りの順序で並び替え，それぞれを 1 人の被験者に割り当てた．各被験者に異なる順序で誤記タイプが分散するよう設計されている．被験者 15 人に，それぞれ 100 個の文章を読み上げてもらい，1500 個の音声データを収集した．

Bot は，LLM を用いて文章を理解することを想定し，音声データを収集した．Bot が認証を突破することを想定した場合，誤記を含んだ文章を修正して読み上げることが想定される．そのため，プロンプトには誤記を含む文章の前に，「以下の文章を修正してください。」を加えて入力した．使用した LLM は，Gemini 1.0 Pro，GPT-4，GPT-4 omni(以下 GPT-4o) の 3 種類である．LLM から受け取った文章は，VOICEVOX [*2] を用いて音声データに変換した．VOICEVOX は，日本語音声合成エンジンおよびソフトウェアで，テキストを高品質な音声に変換するためのツールである．キャラクター番号は，2 を使用した．

### 4.3　文字認識特徴分析

人間と Bot の文字認識特徴を比較するために，誤記の修正率を評価した．誤記の修正率は，音声データを文字起こしした上で，誤記を含まない元の文章と比較することで評価した．人間と Bot から収集した発話音声データを

[*2] VOICEVOX: `https://github.com/VOICEVOX/voicevox`

表 3　各対象ごとの修正率 [%]

| 誤記タイプ | | 人間 | Gemini | GPT-4 | GPT-4o |
|---|---|---|---|---|---|
| 種類 | 位置 | | | | |
| 入替 | 前方 | 27 | 40 | 30 | 27 |
| | 中央 | 36 | 39 | 35 | 26 |
| | 後方 | 43 | 29 | 37 | 26 |
| 削除 | 前方 | 18 | 31 | 24 | 21 |
| | 中央 | 17 | 31 | 30 | 23 |
| | 後方 | 24 | 43 | 38 | 33 |
| 代用 | 前方 | 53 | 53 | 50 | 39 |
| | 中央 | 51 | 48 | 41 | 44 |
| | 後方 | 48 | 49 | 41 | 45 |
| 結合 | 前方 | 18 | 38 | 29 | 25 |
| | 中央 | 15 | 38 | 35 | 26 |
| | 後方 | 25 | 42 | 37 | 32 |
| 挿入 | 前方 | 17 | 37 | 34 | 27 |
| | 中央 | 39 | 44 | 36 | 30 |
| | 後方 | 37 | 40 | 41 | 37 |

Whisper large-v2 を用いて文字起こしし，MeCab(辞書：mecab-ipadic-NEologd) を用いてひらがなに変換した．誤記を加える前の原文も同様にひらがなに変換し，発話音声データから得られたひらがなの文章と完全一致するかを確認した．

集計した結果を**表 3** に示す．表の赤色・青色セルは各対象の修正率の最大値・最小値をそれぞれ示している．人間は 15 人の合計である．

人間は，前方に代用の誤記を加えた場合，修正率が最大で 53 であった．反対に，中央に結合の誤記を加えた場合，修正率が最小で 15 であった．Gemini と GPT-4 は前方に代用の誤記を加えた場合，修正率が最大で 53 と 50 であった．GPT-4o は，後方に代用の誤記を加えた場合，修正率が最大で 45 であった．反対に前方に削除の誤記を加えた場合，3 つの LLM で修正率が最小となり，31, 24, 21 であった．

人間と Bot の修正率の差を確認するため，各 LLM の修正率から人間の修正率を引いて，絶対値を計算した．人間と Bot の修正率差の絶対値を**表 4** に示す．表の赤色・青色セルはそれぞれ，比の最大値・最小値を示している．LLM Mean はの人間と Bot の修正率差の絶対値の平均値である．

文章の中央に結合の誤記を加えた文章において，人間と Bot の修正率の差が平均で大きい傾向が認められる．全ての LLM において，人間よりも修正できることがわかる．反対に，人間と Bot の修正率差の絶対値では，各 LLM の一番低い誤記タイプはバラバラである．これは各 LLM の特徴が要因であると考えられる．この結果から，発話型属性認証に向けては，中央に結合の誤記を加えた文章を提示することが望ましいと言える．

表 4　人間と Bot の修正率差の絶対値

| 誤記タイプ | | Gemini | GPT-4 | GPT-4o | LLM |
|---|---|---|---|---|---|
| 種類 | 位置 | | | | Mean |
| 入替 | 前方 | 13 | 3 | 0 | 5.33 |
| | 中央 | 3 | 1 | 10 | 4.67 |
| | 後方 | 14 | 6 | 17 | 12.33 |
| 削除 | 前方 | 13 | 6 | 3 | 7.33 |
| | 中央 | 14 | 13 | 6 | 11.00 |
| | 後方 | 19 | 14 | 9 | 14.00 |
| 代用 | 前方 | 0 | 3 | 14 | 5.67 |
| | 中央 | 3 | 10 | 7 | 6.67 |
| | 後方 | 1 | 7 | 3 | 3.67 |
| 結合 | 前方 | 20 | 11 | 7 | 12.67 |
| | 中央 | 23 | 20 | 11 | 18.00 |
| | 後方 | 17 | 12 | 7 | 12.00 |
| 挿入 | 前方 | 20 | 17 | 10 | 15.67 |
| | 中央 | 5 | 3 | 9 | 5.67 |
| | 後方 | 3 | 4 | 0 | 2.33 |

表 5　単語の内の誤記の位置と詰まり [個,%]

| 結合位置 | 音声数 | 詰まった音声数 | 詰まり発生率 |
|---|---|---|---|
| - | 5 | 0 | - |
| 2 | 56 | 12 | 21.43 |
| 3 | 25 | 3 | 12.00 |
| 4 | 8 | 1 | 12.50 |
| 5 | 5 | 2 | 40.00 |
| 6 | 1 | 0 | 0.00 |
| 合計 | 100 | 18 | - |

### 4.4　発話特徴分析

中央に結合の誤記を加えた際の，発話特徴について分析する．人間の音声データを 100 個聞き，削除された文字の位置と詰まりの関係を表 5 に示す．

削除された文字の位置が，単語内で誤記が発生した文字の位置を表す．加えられた誤記は結合であるが，削除した文字に着目する．音声数は，削除された文字の位置に誤記が加えられた文章を読み上げる音声の総数を示す．詰まった音声数は，誤記によって音声が詰まった音声の総数を示す．詰まり発生率は，各結合位置の音声数に対する詰まり音声数の割合である．

日本語文章を読み上げる際の停留時間と眼球移動距離の長さは，平均約 0.25 秒と平均約 5 文字と分かっている [13] [14]．音声化により眼球移動距離は短くなる [15] ため，3 文字目は文節を読み始めから認識される可能性が高い．見慣れない文字列でも，認識していればそのまま読み上げることができるため，詰まる確率が低いと言える．文節内の 5 文字目が削除された場合，詰まりが確認されたのは 2 個であった．確率で表すと，40.00%である．詰まる確率が一番高いのは，停留から動いた後に読み始める文字であるからと考えられる．黙読の場合は，文字列を認識できれば読み飛ばすことができるため，タイポグリセミア現象が発生しやすい．しかし，音読の場合は，文字列を正確に認識した上で発音しなければいけない．音声化により，眼球移動距離は短くなり，5 文字より短くなる．停留から動き出した瞬間に読み慣れない文字を認識することなるため，詰まりが発生していると考えられる．ただし，文節内の 5 文字目が削除された音声が少ないことには留意する必要がある．

### 4.5　評価方法

文字認識特徴分析と発話特徴分析から特定された，認証文章の文章特徴により人間と Bot を識別できることを評価する．識別は，流暢性評価により行う．流暢性評価は，詰まり位置と音素時間の 2 つの観点から評価する．人間が誤記を含んだ文章を読み上げた場合，発話リズムが崩れることが考えられる．そのため，入力された音声の発話リズムが乱れていれば人間と識別する．

詰まり位置は，誤記の文字の直前または，誤記を含む文節の直前で無音区間があることを評価する．無音区間の検出は，次の手続きでサンプルごとに量的基準を設定し，この基準に従って検出されたものを「無音」とみなす [16]．

(1) 音声データをサンプリングレート 44.1kHz で離散化
(2) スペクトルサブトラクション法により雑音を除去
(3) 短時間エネルギーを計算
(4) 平均短時間エネルギーの 10%を閾値に設定

無音区間を検出した際に，開始時刻と終了時刻を記録する．無音区間終了時刻の 1 秒前から 2 秒間の音声を切り出し，Whisper large-v2 を用いて文字起こしを行う．文字起こしした中に，誤記の箇所の文字が含まれていれば，詰まっていると判断する．

音素時間は，誤記の文節とその他の文節における，各音素の平均時間のズレを評価する．音素ごとにかかった時間を測るために日本語に対応した大語彙連続音声認識エンジンの Julius [17] を用いる．Julius は，音声と読み上げた文章 (ひらがな) を用意し，音素セグメンテーションキットで音声ファイルを音素単位の forced alignment をすることにより，音素ごとの時間の計測する．

流暢性自動評価には，各音素の平均時間との差が有効であることがわかっている [18]．そのため，音声全体の平均音素時間と誤記を含む文節の平均音素時間の差を評価することで，誤記の影響を評価することが可能と考えられる．音素時間は，平均音素時間よりも誤記を含む文節の平均音素時間が長い場合，流暢ではなくなったと判断する．

評価実験では，詰まったと判断される，または流暢ではないと判断された場合に，人間であると識別する．

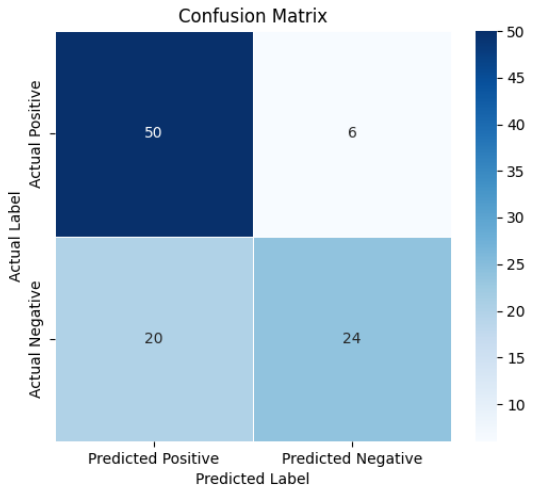


図 4　提案システムの正解率 (VOICEVOX)

### 4.6　評価結果

提案システムに，人間と Bot の音声をランダムに 100 回入力した結果を，図 4 に示す．真陽性が 50，偽陽性が 6，偽陰性が 20，真陰性が 24 であった．そのため，流暢性評価ブロックの正解率は 74%である．

### 4.7　考察

流暢性評価ブロックの識別率は 74%であり，提案システムの実現可能性は高いと考えられる．しかし，Bot を人間と識別する確率が，Bot を Bot と識別する確率とほぼ同じのため，改善が求められる．CAPTCHA 認証において重要なことは，Bot を人間と判断しないことである．提案システムでは，誤って Bot を人間と判断する可能性が高い．そのため，Bot を人間と判断しない識別要素が必要であると考える．具体的には，誤記の修正度合いを識別要素に加えることである．人間と LLM で誤記の修正の仕方は異なると考えられる．人間的修正の仕方と LLM 的修正の仕方を識別可能にすること，提案システムの実現可能性を高めることが期待される．

## 5　おわりに

本研究では，誤記を活用した新たな発話型属性認証システムを提案した．人間と Bot が誤記を含む文章を読み上げた音声データを収集し，発話特徴の分析を行った．分析の結果，誤記による非流暢性を利用した属性認証の有効性を示した．さらに，文節の中央 5 文字目に結合の誤記を加えた文章が，人間と Bot を識別する提示文として適していることを明らかにした．提案システムを用いた識別実験では，74%の精度で両者を判別可能であることを確認した．

## 参考文献

[1] 松原仁．チューリングテストとは何か．人工知能学会誌, Vol. 26, No. 1, pp. 42–44 (2011).

[2] Searles, A. and Nakatsuka, Y. and Ozturk, E. and Paverd, A. and Tsudik, G. and Enkoji, A. An Empirical Study & Evaluation of Modern CAPTCHAs. *Proceedings of the 32nd USENIX Security Symposium*, pp. 3081–3097 (2023).

[3] 下倉良太，佐藤遼平，飯國洋二．人に容易で機械に判別困難な非語を用いた語彙性判断型 CAPTCHA 認証．日本音響学会誌， Vol. 79, No. 9, pp. 438–446 (2023).

[4] 久保田萌々，藤川真樹，鈴木真樹史．タイポグリセミアを用いた Multi-model CAPTCHA の提案と評価. 産業応用工学会論文誌, Vol. 11, No. 1, pp. 54–64 (2023).

[5] Shah, M. and Harras, K. Hitting Three Birds with One System: A Voice-Based CAPTCHA for the Modern User. *2018 IEEE International Conference on Web Services (ICWS)*, pp. 257-264 (2018).

[6] Lior, Y. and Guy, F. and Fred M, G. and Yisroel, M. Deepfake CAPTCHA: A Method for Preventing Fake Calls. *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pp. 608–622 (2023).

[7] Li, L., Lu, T., Ma, X. and Yuan, M. Voice Deepfake Detection Using the Self-Supervised Pre-Training Model HuBERT. *Applied Sciences*, Vol. 13,No. 14, p. 8488 (2023).

[8] rinna 株式会社．”日本語の音声に特化した事前学習モデル HuBERT を公開”．NEWS. 2023-04-28. https://rinna.co.jp/news/2023/04/20230428.html, (参照 2024-10-28).

[9] Kassis, A. and Hengartner, U. Breaking Security-Critical Voice Authentication. *2023 IEEE Symposium on Security and Privacy*, pp. 951–968 (2023).

[10] 望月紫穂野, 塩田さやか, 貴家仁志．話者照合のためのポップノイズの発生頻度を考慮したプロンプト文を用いた声の生体検知，情報処理学会研究報告，Vol. 2017-MUS-115, No. 57, pp. 1-6 (2017).

[11] 難波知康，清水聡一，天野英晴，味岡 義明，新井正敏，今井大輔．レーベンシュタイン距離と最小二乗法を用いた標識認識アルゴリズム. 自動車技術会論文誌， Vol. 41, No. 4, pp. 883–888 (2010).

[12] Marques, M., Jiang, X., Dufor, O., Berrou, C. and Kim-Dufor, D.-H.: A Connectionist Model of Reading with Error Correction Properties, *7th Language and Technology Conference, LTC 2015*, pp. 304–317 (2015).

[13] 斎田真也．読みと眼球運動，眼球運動の実験心理学，名古屋大学出版会，pp. 167–197, (1993).

[14] 神部尚武．日本語の読みと眼球運動，読み-脳と心の情報処理，朝倉書店，(1998).

[15] 森田愛子，髙橋麻衣子．音声化と内声化が文章の理解や眼球運動に及ぼす影響，教育心理学研究，Vol. 67, No. 1, pp. 12–25 (2019).

[16] 横井聖宏，馬場康輔，須藤秀紹，山路奈保子．発話中の「間」がプレゼンテーションに対する聴衆の支持に与える影響. 日本感性工学会論文誌, Vol. 15, No. 3, pp. 363–368 (2016).

[17] 河原達也，李晃伸．連続音声認識ソフトウェア Julius. 人工知能学会誌，Vol. 20, No. 1, pp. 41–49 (2005).

[18] 瀧田寿明，中臺久和巨，星野准一．児童による音読の流暢性自動評価手法. 情報処理学会論文誌， Vol. 57, No. 3, pp. 922–930 (2016).