音声認識を用いた
青空文庫振り仮名注釈付き音声コーパスの構築の試み

佐藤文一[1] 吉永直樹[2] 豊田正史[2] 喜連川優[3,2]
[1]国立国会図書館 [2]東京大学生産技術研究所 [3]大学共同利用機関法人情報・システム研究機構
f-sato@ndl.go.jp, {ynaga, toyoda, kitsure}@tkl.iis.u-tokyo.ac.jp

概要

読書バリアフリー法が制定され，読みの困難な人のためのアクセシビリティへの要求が高まっている．本研究では，OpenAI が公開している Whisper による音声認識を用いて多種多様な読みを推定するためのコーパスの大規模化が可能かを検討する．コーパスの作成にあたっては正解テキストデータとして青空文庫のテキストと，音声データとして「サピエ」の音声デイジーをそれぞれ使用し，両者を文レベルで対応させ，音声認識で複数の認識候補を取得し，読みの推定をおこなっている．合計 5545 万文字，3520 時間の振り仮名注釈付き音声コーパスを構築した．この過程で明らかとなった課題についても説明する．

本研究によって得られたコーパスについては公開を予定している．

1 はじめに

2019 年に「視覚障害者等の読書環境の整備の推進に関する法律」（読書バリアフリー法）[1]が制定され，2023 年に「電子図書館のアクセシビリティ対応ガイドライン 1.0」[2]が公開されるなど，音声読み上げ機能（TTS）等に対応したアクセシビリティに係る要件が整理されてきている．そのため，今まで活字中心であった図書についても，audio book，テキストデイジー，マルチメディアデイジーなど，音声での読み上げを可能とする形式での提供が求められるようになった．このような背景から，漢字の振り仮名の自動付与(読みの推定)の精度向上が強く望まれている．

筆者らは，振り仮名が付与された国立国会図書館の書誌データ[3]のタイトルや青空文庫[4]の公開作品のテキストデータと視覚障害者情報総合ネットワーク「サピエ」[5]が視覚障害者に提供している点字データを用いて，大規模日本語振り仮名注釈付きコーパスを構築してきた．このコーパスをもとに 203 語の同形異音語の出現頻度を分析し，その中から 93 語，読みの総数 223 個に対して BERT[6]の転移学習による読み推定を行った[7, 8]．しかし，読みの推定は，同形異音語以外にも，「音読み訓読み」「人名と普通名詞」等と多数存在し，また，出現頻度の少な過ぎる読みも含まれるため，機械学習による読みの推定精度向上のためには，更なる振り仮名注釈付きコーパスの大規模化が必要であった．

本研究では，OpenAI が 2022 年に公開した高性能な音声認識モデル Whisper[9]を使って，振り仮名注釈付きコーパスの拡張が行えないかを検討した．具体的には，音声データとして，「サピエ」が視覚障害者に提供している音声デイジーを使用した．音声デイジーの xml データにより文単位に音声データを分割し，音声認識を行った結果と青空文庫のテキストの対応を取る．次に両者の文と，音声認識結果の複数の候補と，単語の読み辞書を用いて，読みを推定する．課題も見つかり今後の改善方針を明らかにした．

合計，本 3252 冊，5545 万文字，録音時間 3520 時間を青空文庫振り仮名注釈付き音声コーパスとして NDL Lab から公開する予定である．

2 関連研究

漢字の読み推定を機械学習で扱うためには，読みに漢字ごとに正しい振り仮名が付いた学習事例が必

要となる．そのため，機械学習の適用に耐えうる規模の言語資源を整備することが求められている．

漢字の読み推定は，形態素解析，仮名漢字変換，音声合成などのタスクと関連して，学習データの構築に焦点を当てて研究が行われてきた[10, 11, 12]．読み推定と同様に，単語単位の分類問題として定式化される語義曖昧性解消タスクでは，事前学習済みモデルである BERT の利用による性能改善が報告されている[13, 14, 15]．近年では，音声コーパスや音声データを使用して，読みに関する研究がおこなわれてきている[16, 17]．小林らは，「日本語話し言葉コーパス」の訓練データの不足を，疑似訓練データを追加することにより，同形異音語の読み推定での有効性を調査している[16]．増山らは，「国会審議映像検索システム」により，同期された音声とテキストから，同形異音語の出現数を分析している[17]．一方，藤本らは，テレビの字幕と音声認識ソフトウェア ESPNET2 をベースにして，約 2 万時間の大規模な音声コーパスを構築しており，今後読みの推定への応用が期待される[18]．筆者らは，振り仮名注釈コーパスを構築し，複数の同形異音語の読みの推定を行っている[7, 8]．

3 振り仮名注釈付き音声コーパス

本節では，最初に，本研究で構築した振り仮名注釈付き音声コーパスの書く作品毎に出力されるテキストファイルの具体的な構成例を示し，次に，音声認識を用いてそのデータを得る手法を説明する．

3.1 出力のデータ形式

後述する青空文庫の音声デイジー(読み上げ音声)から構築した振り仮名注釈付き音声コーパスの具体例を表 1 に示す．例は夏目漱石の「こころ」の一文である．

表 1　コーパスの出力の例

行番号	23	0000045.mp3		
進まない結婚を強いられていた			[音声認識結果]	
勧まない結婚を強いられていた。			[青空文庫テキスト]	
「解析結果:」				
勧	進	すす	同音_rubi	
「読み推定結果:」				
勧	すす	すす	進	
結婚	けっこん	けっこん	結婚	
強	し	し	強	

1-3 行目はそれぞれ，分割された音声データのファイル名，音声認識の結果のテキスト，対応する青空文庫のテキストである．4 行目から始まる「解析結果」は，青空文庫の文字列，音声認識の文字列，読み，備考である．6 行目から始まる「読み推定結果」は，青空文庫の文字列，推定の読み，形態素エンジンの読み，音声認識の文字列，備考である．

解析結果には，Whisper による音声認識結果と元テキストに不一致の個所があるときに，不一致となる文字列と両者に共通する読みを抽出した結果が含まれている．青空文庫のテキストに漢字が含まれているときは，「読み推定結果」以降で，漢字とその推定した読みを示している．

なお，形態素の解析は MeCab-ipadic-neologd を使用した[20, 21]．

3.2 音声デイジーの xml の解析

音声デイジーは，視覚障害者が利用しやすいように，音声と目次等が xml で構造化・階層化され，文単位，章や節の間での移動が可能である [20]．この xml を解析し，目次と見出しの情報と文単位での音声データの開始時間と終了時刻を収集する．

3.3 音声データの分割

前項で得た xml の文単位での開始・終了時刻に基づいて，mp3 の音声データを分割する．このとき，Whisper への入力のため，bitrate="16k"で変換した wav 形式の音声データも保存する．

3.4 音声認識候補の収集

前項で得た文単位の wav の音声ファイルに対して，Whisper による音声認識を行い，テキストを得る．Whisper はバージョン 20230124 を使用し，model='medium,' beam_size = 5 で音声認識を行った．Whisper は，現在までの認識したトークン列から，次のトークンを予測する方法で音声認識を行っている．

Whisper は次のトークンを予想する時，ビームサーチにより候補の数を絞っている．このため，Whisper の decording.py とその関連のソースコードを部分的に修正することにより，文頭から逐次的に処理しているビームサーチでの他の認識候補を取得し，後述する「読みの推定」における候補として使用することができる．

3.5 音声認識結果のテキストの前処理

Whisper の音声認識結果には，モデル由来の「ご視聴ありがとうございました」のような入力音声に含まれないテキストが含まれることがあるので，これらを削除する．また，一部の音声デイジーには，青空文庫には無い，「注記」や「視覚障害者の理解を助けるための説明」が含まれている．長い注記の場合は，音声認識のテキストと青空文庫のテキストから対応する文を見つけるときの阻害要因であるので，削除することが望ましい．

音声認識では，「注」は，「中」，「ちゅう」，「じゅう」，「重」，「10」など色々な文字に認識されうる．これらの間違えやすい文字をプログラムで検出し，半自動で削除を行った．

3.6 テキストの前処理

既存研究[8]を参考に，青空文庫のテキストの前処理として下記を行った．

- ルビ，入力注の削除
- JIS X 0213 の面区点番号の漢字への変換
- 見出し，ルビとその漢字のデータの収集

3.7 音声認識結果と元テキストの対応付け

音声認識結果のテキストと青空文庫のテキストは，文単位で対応がとれていないため，以下の手順により対応する文ペアを抽出した．

まず，前処理で得られた目次の情報から，青空文庫の本と音声デイジーの本のペアを，さらに見出しの情報から，章単位のペアを抽出した．青空文庫と音声デイジーの本は，現代仮名遣いにした訳者や版が異なっているが，編集距離の情報と句読点の位置を手掛かりに，最終的に，この章単位のペアから，文単位のペアを，抽出した．

3.8 音声認識結果と元テキストの不一致個所の解析

前述で得られた音声認識の結果と青空文庫の文のペアに対して，編集距離のアルゴリズムを使って，不一致する個所を見つける．不一致する個所の両者に共通の読みがあるかを調べ，一致した読みを，この個所の読みと推定する．例えば，「年」を「歳」と認識した時は，元テキストと音声認識結果のそれぞれの読みのリストを作成し，この両者の読みが一致する読みを探索する．ただし，ルビがある時は，ルビの読みを最優先で行い，次に形態素解析器の読み，次にその他の読みと一致するかを調べる．この例では，形態素エンジンの読みの「とし」で，両者の一致した読みが見つかっている．

読みの一致を調べるために，二通りの方法で行っている．一つは，仮名のあいまい一致である．「ず」と「づ」，「じ」と「ぢ」，「は」と「わ」，「へ」と「え」，「ひ」と「ぴ」のような濁音・半濁音の比較，「う」と「ー」などは，一致する読みとみなす．上記で一致が見つからないときは，音素に変換して読みの一致を調べる．jaconv.hiragana2julius[23]で音素に変換し，4 音素以上の時，編集距離が 1 の時は一致とみなし，8 音素以上の時は距離が 2 までを一致とみなした．例えば，「電報」と「電報」の場合は，それぞれの読みの音素が「deNpo」と「reNpo」の時は，「d」と「r」の 1 音素が異なっているだけなので、両者を一致としている．

青空文庫の送り仮名は，現在の仮名遣いと異なっている[23] ．Whisper は現代文で学習しているため，両者の送り仮名は異なっている．例えば，「起こす」と「起す」は，どちらの読みも「おこす」である．この不一致の解析中に送り仮名の不一致の対応も行っている．

以上により，音声認識結果と青空文庫の両者の不一致個所の対応を取ることにより，音声認識の誤り個所を明示することができる．

3.9 青空文庫の漢字の読みの推定

青空文庫のテキストの漢字の読みは，形態素解析器の形態素ごとにまとめている．従って，漢字の形態素は音声認識結果と「不一致」，「一致」，「不一致と一致の混在」3 種類に分類できる．「不一致」の場合は，前述の不一致の解析結果の読みを採用する．前述の例では，「年」の読みは「とし」である．次に「一致」の場合は，ビームサーチの結果の複数の読みの候補を使用して読みの推定を行う．例えば，「年」の複数の読みの候補が['年', '歳', 'と']の場合は，自分自身と同じ文字を除くと，候補は['歳', 'と']となる．「年」と「歳」の読みと，「年」と「と」の読みの一致を調べる．一致した読みがある場合は，そ

れを推定の読みとして採用する．

一致した読みが無い場合は，仮名にした読みの先頭または末尾で一致するかを調べる．

以上でも見つからなかった場合は，形態素の読みを，そのまま採用する．

「一致と不一致が混在の場合」は，先頭の漢字の読みが一致する読みがある場合は，その読みを採用する．この漢字の読みの推定の個所では，形態素にまとめて分割する操作も行っている．たとえば「揚子江支流」は，「揚子江」と「支流」の形態素に分割されるが，音声認識の結果が，「陽子公子流」であるため，不一致の個所の「江支」を，形態素と一致するように分割している．

3.10 構築したコーパスの統計情報

個々の作品に対し、収集した文の文字数の全体の文字数に対する割合を収集率として定義したとき，収集率が 50%以上の作品は，作家数 118 人，作品数 (重複タイトル有り) 3252 冊，文字数 5545 万文字，録音時間 3520 時間であった．作家別の録音時間を，表 2 に示す．

録音時間は，作家によって，大きく異なっており，江戸川乱歩の録音時間が圧倒的に多い結果となった．

この振り仮名付き音声コーパスについては国立国会図書館の NDL Lab より公開を予定している．

表 2　作家別の文字数と録音時間

著者名	江戸川乱歩	吉川英治	山本周五郎	夏目漱石	中里介山	中略	左川ちか	村岡博
作品数(重複あり)	251	78	216	73	24		1	1
作品数(重複無し)	94	43	131	50	19		1	1
total青空文字数	13121992	10195861	5773133	5075712	2859434		234	166
文字収集率	0.701	0.593	0.715	0.712	0.648		0.731	0.771
total青空行数	650502	538933	367455	310638	145491		30	24
行収集率	0.748	0.624	0.74	0.721	0.671		0.7	0.208
total青空録音時間(時:分:秒)	815:02:00	536:20:08	330:38:53	336:59:59	284:32:09		0:01:19	0:02:02
録音収集率	0.746	0.625	0.738	0.72	0.67		0.675	0.227
accept文字数	9195783	6047102	4127797	3616410	1852802		171	128
accept行数	486550	336359	271775	223838	97606		21	5
accept録音時間(時:分:秒)	607:39:06	335:03:41	243:59:15	242:47:44	190:46:08		0:00:53	0:00:28

4 考察

この節では，振り仮名付き注釈付き音声コーパスを構築した時に判明した課題を 3 点報告する．

1 点目は漢字の読みの推定に関する課題である．今回 beam サーチの結果を利用して，複数の逐次認識候補を収集したが，候補の数が不十分な場合がみられた．Whisper の出力する日本語文字列は，漢字等の一文字を複数トークンに分割することがあるので，ビームサーチの評価結果に基づいて 1 トークンずつ遷移先を選択していくと，本来の文字の情報が壊れた無効なトークンとして候補に含まれる場合がある．このため，場合によっては十分な数の有効な読みの候補の収集ができない．一定割合無効なトークンが含まれることを見越して収集する候補の数を増やす事が必要である．

2 点目として，読みが確定できない場合における対処である．例えば，「私」が正しく認識された場合は，「わたくし」と「わたし」のどちらの読みを採用すべきか決定できない．この課題は音素に分解することによって一定の解決が見込まれるものであるが，音素を収集すべきかについては，今後の検討課題である．

3 点目は，青空文庫と音声認識の結果の対応づけである．例えば「来い」を「恋」と音声認識した場合は，漢字の「来」の対応する文字列が無い．対応づけは読みの推定には必須では無いが，今後検討したい．

以上のように本論文における実装には課題が存在するが，今後改善方法を検討したい．

上記に加えて，Whisper の model を「medium」から「large-v3」に変更する予定である．これにより，音声認識の精度向上が見込めるため，コーパスの収集できる文字数と録音時間が増えることが期待できる．

5 おわりに

本論文では，音声デイジーの xml の情報をもとに，OpenAI の Whisper の音声認識結果と，青空文庫のテキストから，5545 万文字， 3520 時間の振り仮名注釈付き音声コーパスを構築し，漢字の読み推定における課題を明らかにした．

今後も多種多様な単語の読みの推定の精度の改善の手法の検討と，そのための，ベースとなる振り仮名注釈付きコーパスの拡充を行いたいと考えている．

参考文献

1. 視覚障害者等の読書環境の整備の推進に関する法律. e-Gov. https://elaws.e-gov.go.jp/document?lawid=501AC0100000049
2. 電子図書館のアクセシビリティ対応ガイドライン 1.0｜国立国会図書館―National Diet Library https://www.ndl.go.jp/jp/support/guideline.html
3. 国立国会図書館サーチが提供する OAI-PMH https://ndlsearch.ndl.go.jp/help/api/oai_pmh
4. 青空文庫　Aozora Bunko https://www.aozora.gr.jp
5. サピエとは https://www.sapie.or.jp/sapie.shtml
6. DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
7. 大規模振り仮名注釈付きコーパスを用いた同形異音語の読み分類. 佐藤文一 , 吉永直樹 , 喜連川優. 言語処理学会第 28 回年次大会講演論文集, 2022.
8.SATO, Fumikazu, et al. Building Large-Scale Japanese Pronunciation-Annotated Corpora for Reading Heteronymous Logograms. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. 2022. p.7113-7121.
9. RADFORD, Alec, et al. Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning. PMLR, 2023. p. 28492-28518.
10. 機械翻訳手法に基づいた日本語の読み推定. 羽鳥潤, 鈴木久美. 言語処理学会第 17 回年次大会, 2011. p.579-582.
11. 仮名漢字変換ログを用いた単語分割・読み推定の精度向上. 高橋文彦, 森信介. 情報処理学会研究報告, 2014. p.1-10.
12. 読み曖昧性解消のためのデータセット構築手法. 西山浩気, 山本和英, 中嶋秀治. 人工知能学会全国大会論文集 第 32 回全国大会 (2018). 一般社団法人 人工知能学会, 2018.
13. BERT を利用した教師あり学習による語義曖昧性解消. 曹鋭, 田中 裕隆, 白 静, 馬 ブン, 新納 浩幸. 言語資源活用ワークショップ発表論文集= Proceedings of Language Resources Workshop. No. 4. 国立国語研究所, 2019.
14. 事前学習済み BERT の単語埋め込みベクトルによる同形異音語の読み誤りの改善（福祉情報工学）. 佐藤文一, 喜連川優. 電子情報通信学会技術研究報告= IEICE technical report: 信学技報 119.478（2020）, 2020, p17-21.
15. BERT の Masked Language Model を用いた教師なし語義曖昧性解消. 新納浩幸, 馬ブン. 言語処理学会第 27 回年次大会発表論文集, 2021, p.1039-1042.
16. 疑似訓練データを用いた BERT による同形異音語の読み推定. 小林汰一郎, 古宮嘉那子, and 新納浩幸. 研究報告自然言語処理 (NL) 2022.3, 2022: p.1-5.
17. 国会審議における同形異音語の分析. 増山幹高, 松田謙次郎. 法學研究：法律・政治・社会. Vol.96 No.2. 2023, p.444-464
18. Yue Yin, Daijiro Mori, Seiji Fujimoto. ReazonSpeech: A Free and Massive Corpus for Japanese ASR. 言語処理学会第 29 回年次大会講演論文集, 2023. p.1134-1139
20. MeCab: Yet Another Part-of-Speech and Morphological Analyzer https://taku910.github.io/mecab/
21. GitHub - neologd/mecab-ipadic-neologd: Neologism dictionary based on the language resources on the Web for mecab-ipadic https://github.com/neologd/mecab-ipadic-neologd
22. DAISY2.02，DAISY3 等の仕様の日本語訳を公開します。 | 日本 DAISY コンソーシアム https://blog.normanet.ne.jp/jdc/?q=node/6
23.GitHub - ikegami-yukino/jaconv: Pure-Python Japanese character interconverter for Hiragana, Katakana, Hankaku, and Zenkaku https://github.com/ikegami-yukino/jaconv
24. 送り仮名の付け方 - 国語施策・日本語教育 文化庁 https://www.bunka.go.jp