Vol.2021-AAC-15 No.13
2021/3/6

書誌データ・青空文庫・点字データを用いた振り仮名注釈付き日本語コーパスの構築

佐藤文一[†1]　吉永直樹[†2]　喜連川優[†3]

概要：重度視覚障害者は，パソコンで画面読み上げソフトを使用して，漢字交じりの文書を音声で聞いているが，しばしば読み誤りが発生する．例えば，「表に出る」を「ひょうにでる」と読み上げられると理解が困難になる．この問題に対しては，機械学習に基づく統計的手法を用いて，前述の表（ひょう，おもて）のような同形異音語の読みを推定するアプローチが有望であるが，モデルの学習には正しい振り仮名が付いた大量の文が必要になる．そこで我々は，振り仮名が付与された国立国会図書館の書誌データの雑誌タイトルや，校正済みの点字データなどを活用して，機械学習に基づく読み推定モデルの学習に必要となる振り仮名付きの日本語コーパスを構築した．具体的には，まず，書誌データのタイトルとその振り仮名のペア，青空文庫のテキストと官公庁が公開する障害者向けの広報テキストなどの PDF テキストと，それらに該当する点字のデータから，対応する文のペアをパターンマッチングで選び出す．次に，この文対に対して，既存の振り仮名注釈付きコーパスや形態素解析辞書などから事前に収集した漢字に対する振り仮名候補に基づく文字レベルのマッチングを行い，振り仮名注釈付き日本語コーパスを構築する．約 4.1 億文字の漢字仮名交じり文の中から，約 3.5 億文字の文に含まれるすべての漢字に対して振り仮名の自動注釈を行った．これにより同形異音語を含む文を選び出せることを確認した．

キーワード：コーパス，振り仮名，同形異音後，情報障害

Construction of a Japanese corpus with furigana annotations using bibliographic data, Aozora Bunko, and Braille data

FUMIKAZU SATO[†1]　NAOKI YOSHINAGA[†2]　MASARU KITSUREGAWA[†3]

１．はじめに

「視覚障害者等の読書環境の整備の推進に関する法律」(通称：読書バリアフリー法）が 2019 年 6 月に公布・施行されている．この法律の理念は，「障害の有無に関わらず，すべての国民が等しく読書を通じて文字・活字文化の恵沢を享受すること」である[1].

障害者への情報アクセシビリティを向上するための施策が施行されてきているが，視覚障害者の情報障害に対しては，まだ克服すべき課題が多い．重度視覚障害者は，パソコンで画面読み上げソフトを使用して，漢字交じりの文書を音声で聞いているが，しばしば読み誤りが発生する．例えば，「表に出る」を「ひょうにでる」と読み上げられると理解が困難になる．また，文部科学省が推進している GIGA スクール構想[2]により，初等中東教育の生徒が，端末を使い，教科書以外を音声で聞く機会が増えていくと思われる．このとき，誤った漢字の読みを繰り返し聞くことは望ましいことでは無い．このような背景から，視覚障害者が被る情報障害の解消のためにも，音声読み上げの精度の更なる向上が強く望まれている．

この問題と関連して，近年，自然言語処理では汎用言語モデルの BERT[3]などを用いて，文脈による同義語の分類の研究が行われている．この同じ意味を持つ同義語を文脈で見つけ出すことができるのであれば，逆に，文脈によって，その読みを変える同形異音後（読み方が複数ある単語)に応用でき，結果的に読みの推定を行うことができる．しかし，このモデルの学習には，正しい振り仮名が付いた大量の文が必要になる．正しい振り仮名を付ける作業を人手で行うと，膨大な時間と労力が必要になってしまう．

本研究では，この機械学習を用いた音声読み上げ精度の改善に必要となる，学習データの構築コストの問題を，既に校正済みの振り仮名データを利用することにより，人手ではなく自動で行うことで解決する．具体的に，校正済み振り仮名データと元の漢字仮名交じり文のデータから，パターンマッチングにより対応関係にある文のペアを選び出し，漢字の振り仮名候補を考慮して、その文対から文字レベルで漢字とその振り仮名の対応をとるという手順で振り仮名付きコーパスを自動構築した．

しかし，この手順を自動で行い振り仮名注釈コーパスを構築するためには，主に次の課題がある．

(1) 網羅的な漢字の振り仮名候補の収集方法
(2) テキストの前処理時で必要となる，様々な記法の注記・注解・目次等の自動削除
(3) 文書対から読みが異なる漢字仮名交じり文と対応する振り仮名の文のペアを抽出する手法

†1 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, the University of Tokyo
†2　東京大学生産技術研究所
Institute of Industrial Science, the University of Tokyo
†3 国立情報学研究所，東京大学生産技術研究所
National Institute of Informatics/Institute of Industrial Science, the University of Tokyo

(4) ペアの文に振り仮名を付ける際の，形態素解析器の形態素の分割誤りへの対処

本研究では，(1)は現代日本語書き言葉均衡コーパス[4]以外からも候補を集めた．(2)は，その都度プログラムの拡張で対応した．(3)は，漢字仮名交じり文を一度形態素解析器で振り仮名に変換し，パターンマッチングの終了後に元に戻す事で解決した．(4)は，形態素解析器の分割では無く，文字種による分割と分かち書きの空白を併用し，漢字の振り仮名は，送り仮名を取った漢字の部分だけを用いることにより解決した．

本研究で使用したデータは，

- 国立国会図書館の書誌データの・雑誌のタイトル（振り仮名のタイトルが提供されている）
- 社会福祉法人日本点字図書館が運営している視覚障害情報総合ネットワーク「サピエ」が視覚障害者に提供している校正済み点字データと青空文庫のデータ
- 厚生労働省の社会保障審議会 (障害者部会)[5]で公開している点字データと pdf
- 内閣府が公開している点字・大活字広報誌「ふれあいらしんばん」[6]の点字版と pdf

である．

本研究では，これらの読み情報が付与されたデータに注目し，文字レベルで振り仮名を付与したコーパスを構築した．結果として，約 4.1 億文字の漢字仮名交じり文の中から，約 3.5 億文字の文に含まれるすべての漢字に対する振り仮名を付与した日本語コーパスを構築した．また，これにより同形異音語を含む文を選び出せることを確認した．すべての漢字に振り仮名を付けることにより，普段日常的に使っている漢字を正確に読み上げるモデルの学習を行うためのコーパスとしての利用が期待できる．

本稿の構成は，次の通りである．第 2 節で，関連研究を概説する．第 3 節では，振り仮名注釈付き日本語コーパスを構築するためのそのコンポーネントを概説する．第 4 節では，第 3 節で構築したコンポーネントを用い，4 種類のデータに対して生成した振り仮名付き注釈コーパスの結果を報告する．第 5 節で，前節得られた振り仮名注釈付き日本語コーパスから，同形異音語の出現数を取得したので，その結果を報告する．第 6 節でまとめと今後の方向性について述べる．

２．関連研究

日本語の漢字の読みの推定の研究は，形態素解析，仮名漢字変換，音声合成と伴って，研究が行われてきている[7][8] .『現代日本語書き言葉均衡コーパス』(BCCWJ) は，広範な分野の 1 億 430 万語のデータを格納し，長短ふたつの言語単位を用いて形態素解析された振り仮名が付与されている．教科書に特化した，コーパスを用いた研究 [9]，多重の読みのコーパス化の研究[10]が行われている．

単語の読みを推定する問題は，単語の意味を正しく推定することと関係している．この単語の意味のモデル化について，自然言語処理では，単語をベクトルで表現する手法が広く研究されている．代表的な手法としては，2013 年に Mikolov らが考案した Word2Vec[11]により，語彙数より少ない次元のベクトル（単語埋め込み）を大規模テキストを用いた自己教師あり学習によって得る方法が提案されている．単語の埋め込みを用いることで，似た文脈で出てくる単語同士の類似性をとらえることができるようになった．

さらに 2018 年には，Devlin らにより，文脈を考慮して単語の意味を計算する事前学習モデルとして，Bidirectional Encoder Representations from Transformers (BERT)が提案された. 双方向 Transformer に基づき文脈を考慮して単語の意味を捉えるこのモデルでは, 例えば銀行の意味の bank と土手の意味の bank を文脈により区別することが可能である. この点に注目して，筆者らも，BERT を用いて，同形異音語の読みの推定[12]の研究を行っている．この研究での実験では，現代日本語書き言葉均衡コーパスから同形異音語として「国立(こくりつ/くにたち)」「表(ひょう/おもて)」「大分(おおいた/だいぶ)」を選び, 振り仮名(ラベル 0/ラベル 1）のラベル 1 の recall は，それぞれ，0.645，0.191，0.347 であった．このため，振り仮名を人手で修正し，それぞれの読みに対応した単語埋め込みベクトルを作成することで，クラス分類問題として読みの推定を行った．

３．振り仮名注釈付き日本語コーパス作成のためのコンポーネント

大量の漢字仮名交じりテキストデータと振り仮名データまたは点字データから，対応する文を見つけ出し，振り仮名注釈コーパスを構築するために，下記の 5 個のコンポーネント(プログラム)を作成した．

- 漢字・英文字・記号に対する振り仮名候補の収集
- 点字のバイナリーファイルの仮名テキストへの変換
- 入力の漢字仮名交じりのテキストと点字の仮名テキストの前処理
- 前処理で作成された二つのテキストから対応するペアの文をパターンマッチングで選び出す
- ペアの文に対して文字種ごとに読み候補の正しさを検査する verifier プログラムを用意して振り仮名注釈を行う

まず振り仮名と点字の仮名について概説し，次に，それぞれのコンポーネントを説明する．

3.1 振り仮名と点字の仮名について

振り仮名は，「書き言葉の振り仮名」と「話し言葉の振り仮名」の二つがある．「書き言葉の振り仮名」は，通常，一般的に使われている書き方である．

例. 僕は東京からパリへ行く。

ぼく　は　とうきょう　から　パリ　へ　いく。

「話し言葉の振り仮名」は，耳に聞こえるとおりに書くのが原則であり，「は」は「わ」，「へ」は「え」のように長音符を使用するのが特徴である．

例. ぼく　わ　とーきょー　から　パリ　え　いく。

形態素解析器 MeCab [13]を使うと，書き言葉と話し言葉の 2 種類の振り仮名が取得できる．

これに対して，点字での振り仮名は，話し言葉の振り仮名がベースであるが，仮名は 1 種類だけである．

例　ぼくわ　とーきょーから　ぱりえ　いく。

また，数えるための漢数字の振り仮名は，数字を使用する．

例. 三回表に

3 かい　おもてに

したがって，MeCab で得られる話し言葉の振り仮名と点字の仮名は，完全には一致していない．

3.2 漢字・英文字・記号に対する振り仮名候補の収集

漢字の振り仮名の候補は

- 現代日本語書き言葉均衡コーパス(BCCWJ)から取得した語彙
- MeCab[13]の IPA 辞書(csv ファイル)
- MeCab-ipadic-neologd[14]の辞書(tsv ファイル)
- 青空文庫のルビから取得した語彙
- 書誌データのタイトルに含まれる括弧無いにあるルビから取得した語彙(ただし人出でチェック)

から作成する．

漢字の振り仮名の候補部分だけを使用するため，送り仮名の仮名を取り除いている．

例 行く　-> [行　い]

2 文字の漢字の片方の振り仮名が未定のときは，他方の振り仮名から推定し，漢字の振り仮名の候補に追加する．

書誌データの振り仮名は書き言葉の振り仮名であり，他の 3 種類のデータは点字の仮名のため，書き言葉と話し言葉の振り仮名の両方を振り仮名候補に加えている．

例　[東京 とうきょう　とーきょー]

英文字と記号の振り仮名候補は，

- 現代日本語書き言葉均衡コーパス(BCCWJ)から取得した語彙
- MeCab の IPA 辞書(csv ファイル)
- MeCab-ipadic-neologd の辞書(tsv ファイル)

から収集する．

例 [% ぱーせんと]

Z [Z ぜーた]

[NONSENSE なんせんす]

記号の振り仮名候補には更に点字特有の振り仮名も追加している．

例　[① (1)]

3.3 点字バイナリーファイルの仮名テキストへの変換

点字は基本的に 2 列 x3 行の 6 個の点で表されるが，データ形式が複数存在している．そこで，BES，BSE，BET の３種類の形式の点字データをひらがなのテキストに変換するプログラムを作成した．

3.4 入力テキストの前処理

漢字仮名まじり文の入力テキストと，点字の仮名のテキストは後述の処理を行うため，前処理を行う．

青空文庫の漢字仮名交じりのテキストに対しては，

- 英数字を半角に正規化
- ルビを取り除く．なお，ルビおよび省略されたルビに対応した漢字とルビとその位置も後述の処理のため取得する．
- 入力注を取り除く

厚生労働省と内閣府の公開の pdf から，漢字仮名交じり文は，OCR ソフトウェア(ABBYY FineReader PDF 15[15])でテキストに変換し，英数字は半角に正規化する．サピエの点字データから仮名に変換されたテキストの前処理として，表紙，目次，注記，注解，奥付を取り除く．

3.5 前処理で得られたテキストからパターンマッチングによるペアの文の取得

漢字仮名交じりのテキストと仮名のテキストの前文から，該当する文のペアを作成する．

次の例を用いて概説する．

A: … 吾輩はコーヒーを飲む。…

AA: ... わがはいわこーひーをのむ。…

B: ...わがはいわ　こーひーを　のむ。…

BB: … わがはいわこーひーをのむ。…

パターンマッチングを行うための前処理として，形態素解析器(MeCab+ NEOLOGD)で漢字仮名交じり文(A)を形態素解析により話し言葉の振り仮名(AA)を得る．ここで，点字の仮名と同じにするため，カタカナはひらがなに変換しておく．点字の仮名(B)に対しては，分かち書きの空白を削除したテキスト(BB)を作成する．この振り仮名と点字の仮名のテキストは通常は完全には一致しておらず、文同士の対応も取れていない。

一致しない理由として、形態素解析器の漢字の振り仮名の誤り、数字の表記の違い、その他の理由で、文が追加されていたり、語句が異なっていたりする。そこで，レーベンシュタイン距離をベースにしたアルゴリズムで，パターンマッチングを行い，ペアの文を見つける．レーベンシュタイン距離は，二つの文字列が最小編集距離で表される距離である．この時，区切り文字でテキストが分割出来るときは，分割する．区切り文字は，句点だけだと非常に長い文になる場合があるため，句読点と括弧と一一で分割を行う．以上により，話し言葉の文(AA)と仮名の文(BB)のペアの文を得る．

次に話し言葉の文(AA)に対応する，元の漢字仮名交じり

表 1　最終出力の例

漢字かなまじり	ふりがな	Verifierの理由
吾輩	わがはい	漢字-ルビ
は	わ	ひらがな
	'	分かち書き
東京	とーきょー	漢字-語彙
	'	分かち書き
空港	くーこー	漢字-語彙
から	から	ひらがな
	'	分かち書き
パリ	ぱり	カタカナ
へ	え	ひらがな
	'	分かち書き
行	い	漢字-語彙
った	った	ひらがな
。	。	記号

文(A)を前文の中から特定する. 同様に分かち書きの空白を削除した文(BB)から, 元の分かち書きの仮名の文(B)を前文から特定する. 以上の処理で, 漢字仮名交じり文(A)と分かち書きの文(B)のペアが得られた.

3.6 振り仮名注釈の作成

ここでは，前述で得られたペアの文から，漢字の振り仮名を見つける．以下例文にそって，漢字の部分を中心にアルゴリズムの概略を説明する．

例：A: 吾輩は東京空港からパリへ行った。
B: わがはいわ　とーきょー　くーこーから　ぱりえ　いった。

まず，入力の漢字仮名交じり文を文字種で分割する．文字種は，数字，英文字，ひらがな，カタカナ，漢字，記号の 6 種類で，それぞれ振り仮名をチェックするための Verifier を作成した．例えば，漢字の verifier では，ルビ，収集した振り仮名候補、省略されたルビの順に一致しているかを調べる．収集した振り仮名の一致は，グラフ探索で行う．

例：　横軸に東京空港
縦軸に点字の未処理の文字列
とうきょー　くーこーから　ぱりえ　いった.

座標[0, 0]から，文字列の長い方から語彙にあるかを調べる．最初は，

東京空港　東京空　東京　東

[2, 5, 東京 とーきょー] [1,2, 東, とー]

[2, 5, 京 きょー]

[4, 9, 空港 くーこー] [3,7, 空港, くー]

[4, 9, 港 こー]

上記で得られた 2 次元ルートを，深さ優先探索で，ルートを選ぶ．このとき，振り仮名の各文字に対して，同一カテゴリの文字かを確認する．たとえば, [へ」と「え」,「は」と「わ」, 長音は同一文字とみなす．また，匹(ぴき)，かわ(がわ)のように，濁点や半濁点に変化している文字も同一とみなす．最後に，次の文字種と組み合わせて送り仮名をチェックする．分かち書きの空白があるときは，分かち書きを優先し, そこまでの漢字と振り仮名のペアを verifier は出力する．なお、漢数字で、例えば「一匹」が「1 ぴき」の場合の時は、この漢数字を分割して処理する.

以上の処理を書く文字種ごとに verifier を実行することにより，表 1 に示す出力を最終的な振り仮名注釈コーパスとして得た.

表 2　書誌データの収集結果

フォルダー名	入力のタイトルの全文字数	入力のタイトル数	収集タイトルの文字数	収集率（文字数に対して）	収集のタイトル数	収集率（行数に対して）
out_tsv_file0098	3071077	100000	2971162	0.967	97714	0.977
out_tsv_file0100	3032425	100000	2924748	0.964	97210	0.972
out_tsv_file0099	2974427	100000	2889438	0.971	97588	0.976
out_tsv_file0094	2965127	100000	2814521	0.949	95919	0.959
out_tsv_file0096	2952156	100000	2869692	0.972	97725	0.977
以下省略						
合計	388554355	18162545	328620835	0.846	16116338	0.887

4．振り仮名注釈コーパスの収集

この節では，収集した振り仮名の候補の結果と 4 種類の振り仮名注釈コーパスの収集結果を報告する．また，収集の際に，気づいた事柄を概説する．

4.1 収集率について

本節で用いる収集率は下記のように定義している.

- 収集率(文字に対して) = 収集した文の文字数　/ 全体の文字数
- 収集率(行数に対して) = 収集した文の行数　/ 全体の行数

以下の例で説明する.

a: もう少しです。

b: もう　すぐです。

a は，文字種で分解すると，ひらがな・漢字・ひらがな・記号に分解される．それぞれの文字種毎の verifier のプログラムが走り，漢字の verifier で，「少」の振り仮名が無いため，verifier は[", "]を出力する.

一方,

b: もー　すこし　です。

の場合は，漢字 verifier は，['少','すこ']を出力する．これらの文字種毎に得られた verifier の出力を連結して，元の a,b と比較して，両方が一致した時，A の文字数を収集率の文字に加える．したがって，漢字を含まない文も収集した文の中に含まれている．また，漢字 verifier が正しい出力の時でも, 例えばひらがな verifier の所で不一致と判定した場合は，その文は収集の対象にはならない.

4.2 漢字・英文字・記号の振り仮名候補の収集結果

現代日本語書き言葉均衡コーパスは，誤りの振り仮名も含まれているので，出現数が 5 以下の振り仮名は，取り除くようにした．MeCab の IPA 辞書は，(株)，(有)の略語を含んでいるので，対応して振り仮名候補の収集を行った.

以上により, 漢字 410,885 個, 英文字 15,210 個, 記号 1,326 個を収集した.

4.3 書誌データの収集結果

国立国会図書館の書誌データのタイトルは，下記のようにタイトルとその形態素分割の結果と振り仮名で構成されている．

例:「野山の花」 「野山の　花」 「ノヤマノ　ハナ」

したがって既に漢字仮名交じり文と振り仮名文のペアが得られているので，他の 3 種類のデータとは異なり，ペアの文を見つけるためのパターンマッチングは不要である．なお，振り仮名は，書き言葉の振り仮名が基本で，形態素エンジンでの結果を人手で修正しているとのことである．

タイトル総数: 19,633,431 に対して，漢字を含まないタイトルを除外して，10 万タイトルで分割して，振り仮名注釈コーパスの作成を行った結果が表 2 である．タイトル数に対する収集率は，88.7%であった

4.4 青空文庫の収集結果

青空文庫のテキストファイルと，サピエの点字ファイルを人出で対応付けを行った後，振り仮名注釈コーパスの作

表 3　青空文庫の収集結果

著者名	作品名	入力の全文字数	入力のペアの全行数	収集文字数	収集率（文字数に対して）	収集行数	収集率（行数に対して）
夏目漱石	吾輩は猫である 倫敦塔カーライル博物館幻影の盾琴のそら音一夜薤露行趣味の遺伝坊っちゃん 草枕 虞美人草坑夫三四郎それから 門 こころ道草 明暗	2383251	148552	2333171	0.979	146310	0.985
谷崎潤一郎	細雪(上巻 中巻) 痴人の愛	524838	34266	512831	0.977	33574	0.98
宮沢賢治	グスコーブドリの伝記雁の童子銀河鉄道の夜	71475	3977	63513	0.889	3729	0.938
島崎藤村	夜明け前(第一部 第二部)	802220	18220	671402	0.837	16393	0.9
堀辰雄	美しい村風立ちぬ	97130	5196	93444	0.962	5050	0.972
山本周五郎	樅ノ木は残った(第一部 第2部 第三部)	568207	47396	552553	0.972	46377	0.979
紫式部	源氏物語	1130549	49268	1081297	0.956	47654	0.967
中里介山	大菩薩峠	4501913	324864	4368882	0.97	317900	0.979
吉川英治	宮本武蔵 新・水滸伝 源頼朝 神州天馬侠 私本太平記 鳴門秘帖 新書太閤記 源頼朝 黒田如水 江戸三国志	7914791	631620	7522975	0.95	612612	0.97
有島武郎	或る女 カインの末裔 小さき者へ 生まれいずる悩み 親子	466717	20942	433129	0.928	19742	0.943
横光利一	旅愁 御身	720369	40069	689640	0.957	38909	0.971
伊藤左千夫	野菊の墓	33404	1867	32166	0.963	1815	0.972
田山花袋	蒲団	194890	11905	180623	0.927	11275	0.947
野村胡堂	銭形平次捕物控	717492	54103	667812	0.931	50967	0.942
森鴎外	山椒大夫 二人の友 最後の一句 高瀬舟 高瀬舟縁起	52753	3647	51049	0.968	3550	0.973
芥川竜之介	トロッコ 蜜柑 お時儀 鼻 藪の中 杜子春 魔術 ひょっとこ 玄鶴山房 河童	97297	5814	88029	0.905	5669	0.975
合計		20277296	1401706	19342516	0.954	1361526	0.971

表 4　厚生労働省の公開データの収集結果

フォルダー名1	フォルダー名2	入力の全文字数	入力のペアの全行数	収集文字数	収集率（文字数に対して）	収集行数	収集率（行数に対して）
6_社会保障審議会障害者部会の資料	（243）第61回資料4	44734	2351	28656	0.641	1795	0.764
6_社会保障審議会障害者部会の資料	（431） 第85回参考資料1－2	42371	1910	37571	0.887	1815	0.95
6_社会保障審議会障害者部会の資料	（368） 第80回参考資料2	30620	1536	23975	0.783	1318	0.858
6_社会保障審議会障害者部会の資料	（359）第79回資料	30502	1410	21996	0.721	1132	0.803
6_社会保障審議会障害者部会の資料	（357）第78回資料(修正版)	29656	1335	21084	0.711	1053	0.789
6_社会保障審議会障害者部会の資料	（392） 第82回参考資料1	29184	1331	26283	0.901	1259	0.946
以下省略							
合計		1012193	46775	780694	0.771	40547	0.867

表 5　ふれあいらしんばんの収集結果

フォルダー名	入力の全文字数	入力のペアの全行数	収集文字数	収集率（文字数に対して）	収集行数	収集率（行数に対して）
ふれあいらしんばん(73)Ｖｏｌ．７３B3528R04191937	7387	375	6648	0.9	351	0.936
ふれあいらしんばん(67)Ｖｏｌ．６７B3528R04030091	7140	399	6144	0.861	375	0.94
ふれあいらしんばん(63)Ｖｏｌ．６３B3528R03929475	7126	279	3836	0.538	241	0.864
ふれあいらしんばん(66)Ｖｏｌ．６６B3528R03997661	6914	394	5679	0.821	359	0.911
以下省略						
合計	97559	5288	78628	0.806	4825	0.912

成を行った．収集した著者と作品での結果が表 3 である．全文字数に対する収集率は，95.4%であった．

4.5 厚生労働省の公開データの収集結果

厚生労働省社会保障審議会障害者部会では，資料として，pdf と点字データが別々ではあるが公開されている．点字データは全部で約 550 のタイトル数である．この pdf から OCR ソフトウェアで得られた漢字仮名交じり文と該当する点字データに対して，135 タイトルに対して振り仮名注釈コーパスを作成した結果が表 4 である．全文字数に対する収集率は，77.1%であった．

4.6 内閣府の公開データからの収集結果

内閣府政府広報室より，「ふれあいらしんばん」の pdf と点字データが定期的に刊行されている．バックナンバーを含めて 15 タイトル入手した．振り仮名注釈コーパスを作成した結果が表 5 である．全文字数に対する収集率は，80.6%であった．

Vol.2021-AAC-15 No.13
2021/3/6

4.7 4種類の振り仮名注釈コーパスの収集結果から

4 種類の振り仮名注釈コーパスの収集結果をまとめたのが，表 6 である．書誌データの 3.3 億文字が大半を占めるが，全部で 3.5 億の文字数の大規模振り仮名注釈コーパスを構築することができた. まだ細かくは分析していないが，収集時に気づいたことを以下に記する．

① 収集した漢字の振り仮名には，当て字の振り仮名が多数含まれていた．例えば，「東京」の振り仮名に「あっち」 [こっち」が含まれている．基本的には，既に漢字仮名交じり文に対応した仮名の文のペアが得られているので，おそらく問題にはならないと予想される．

② 4 種類のデータすべてで，個々のファイルで見ると収集率にかなりのばらつきがあった. 厚生労働省と内閣府の PDF データについては，図表などを含む会議資料を含んでおり，必ずしも PDF に含まれるテキストを忠実に点字に変換していないため，OCR ソフトウェアで得られたテキストと点字のテキストの齟齬が大きいため，収集率が低い．このため文のペアを見つけるアルゴリズムの改良が必要と思われる．

③ 青空文庫の本に該当するサピエの点字データは，現在全国共通基盤で運営しているサピになる前の，「ないーぶ　ねっと」の頃のデータのため，目次・注記・注

表 6　4 種類の結果のサマリー

データの種類	入力の全文字数	入力のペアの全行数	採用文字数	採用率（文字数に対して）	採用行数	採用率（行数に対して）
書誌データ	388554355	18162545	328620835	0.846	16116338	0.887
青空文庫	20277296	1401706	19342516	0.954	1361526	0.971
厚生労働省	1012193	46775	780694	0.771	40547	0.867
ふれあいらしんばん	97559	5288	78628	0.806	4825	0.912
合計	409941403	19616314	348822673	0.851	17523236	0.893

表 7　同異音語の収集結果

	分類	書誌データ		青空文庫		厚生労働省		ふれあいらしんばん		合計	
	一致の種類	単語を含む	単語が一致	単語を含む	単語が一致	単語を含む	単語が一致	単語を含む	単語が一致	単語を含む	単語が一致
1	大分	10255	3710	233	222	2	2	1	1	10491	3935
2	だいぶ	114	8	230	221	0	0	0	0	344	229
3	おおいた	10141	3702	3	1	2	2	1	1	10147	3706
4	国立	20138	19079	0	0	66	66	4	4	20208	19149
5	こくりつ	19556	18835	0	0	66	66	4	4	19626	18905
6	くにたち	582	244	0	0	0	0	0	0	582	244
7	表	194870	32688	2952	1237	346	154	33	4	198201	34083
8	おもて	2621	1121	1426	1090	1	0	3	3	4051	2214
9	ひょう	192249	31567	1526	147	345	154	30	1	194150	31869
10	一言	1691	1685	427	422	1	1	0	0	2119	2108
11	ひとこと	1502	1502	422	418	1	1	0	0	1925	1921
12	いちげん	133	127	0	0	0	0	0	0	133	127
13	いちごん	56	56	5	4	0	0	0	0	61	60
14	最中	118	117	184	172	0	0	0	0	302	289
15	さいちゅう	108	107	172	160	0	0	0	0	280	267
16	さなか	5	5	9	9	0	0	0	0	14	14
17	もなか	5	5	3	3	0	0	0	0	8	8
18	角	38366	7723	3609	2196	4	1	2	1	41981	9921
19	かく	33309	5868	2943	1723	3	0	1	0	36256	7591
20	かど	3965	1503	499	368	1	1	1	1	4466	1873
21	すみ	449	109	64	25	0	0	0	0	513	134
22	つの	643	243	103	80	0	0	0	0	746	323
23	人気	7581	7312	316	272	0	0	0	0	7897	7584
24	にんき	7574	7305	231	187	0	0	0	0	7805	7492
25	ひとけ	1	1	46	46	0	0	0	0	47	47
26	じんき	6	6	39	39	0	0	0	0	45	45
27	一目	1429	1179	281	229	1	1	0	0	1711	1409
28	ひとめ	1045	1041	200	200	1	1	0	0	1246	1242
29	いちもく	384	138	81	29	0	0	0	0	465	167
30	上方	1795	1787	280	259	0	0	0	0	2075	2046
31	かみがた	1149	1141	269	248	0	0	0	0	1418	1389
32	じょうほう	646	646	11	11	0	0	0	0	657	657
33	上手	6977	6972	554	550	0	0	3	3	7534	7525
34	じょうず	6853	6848	516	512	0	0	3	3	7372	7363
35	かみて	109	109	16	16	0	0	0	0	125	125
36	うわて	15	15	22	22	0	0	0	0	37	37
37	人事	25695	23566	93	38	2	2	0	0	25790	23606
38	じんじ	25695	23566	32	31	2	2	0	0	25729	23599
39	ひとごと	0	0	61	7	0	0	0	0	61	7

解等がそれぞれの作品で微妙に異なっており，収集率を上げるためには分析が必要である．また，青空文庫と街頭の点字データの底本が異なっている等で，旧字旧仮名から新字新仮名への訳が大きく異なっている作品は，対応方法を検討する必要がある．

④ 書誌データは，非常に幅広い年代のタイトルが含まれており，必ずしも振り仮名がタイトルに対応していないものが散見されており，これが収集率を下げている要因の一つである．例えば，振り仮名に書き言葉と話し言葉の両方の振り仮名が兵器されているようなケースもある．ひらがなの振り仮名ではあるが，「なほ」は「なお」，「づ」は「ず」にするなどの対応は行ったが，まだ細かな対応が必要である．

５．同形異音語の収集結果

この節では，今回作成したコーパスで同形異音語を収集したので結果を報告する．同形異音語として，「大分」「国立」「表」一言」「一目」「最中」「角」「人気」「一目」「上方」「上手」「人事」の 12 個に対して行った結果が表 7 である．

表 7 の「単語を含む」は，「大分駅」「表面」のように該当の単語を含んだ場合である．「単語の一致」は「大分」「表」のように該当単語と一致した時である．

また表 7 では，例えば，「表」の振り仮名の「ひょう」と「ひょー」は「ひょう」でまとめている．個々の漢字の振り仮名だけでは無く，下記の例のように，該当の同形異音語の「一目(いちもく)」を含むペアの文を収集することができる．

例: 「よい外戚をお持ちになった親王方も帝の殊寵される源氏には一目置いておいでになるのであるが、」「よい がいせきを おもちに なった しんのーがたも みかどの しゅちょーされる げんじにわ いちもく おいて おいでになるので あるが、」

６．おわりに

本論文では，漢字仮名交じりのテキストと点字の仮名テキストをパターンマッチングにより文のペアを抽出し，その文対に対して文字レベルのマッチングを行うのとで，大規模な振り仮名注釈付き日本語コーパスが構築できることを示した．

このコーパスでは，テキスト中のすべての漢字に振り仮名が注釈づけられているので，容易に同形異音語等の，個々のニーズに応じた文が抽出できることを示した．

今回，青空文庫からコーパスの作成に用いた作品には，長編時代小説が多く含まれているが，今後は，作家数を増やしたり，随筆等の短い作品も採用して，より幅の広いジャンルのテキストを含むようコーパスの拡張を行いたい．同時に，機械学習に基づく統計的手法を用いて，同形異音語の分類も行いたい．

今後も視覚障害当事者の観点から，視覚障害者の情報障害の課題に取り組んでいこうと思っている．

参考文献

[1] 令和２年版障害者白書（全体版） - 内閣府 https://www8.cao.go.jp/shougai/whitepaper/r02hakusho/zenbun/index-w.html

[2] GIGA スクール構想の実現について：文部科学省 https://www.mext.go.jp/a_menu/other/index_00001.htm

[3]Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[4] 概要 現代日本語書き言葉均衡コーパス（BCCWJ） http://pj.ninjal.ac.jp/corpus_center/bccwj/

[5] 社会保障審議会 (障害者部会)｜厚生労働省 https://www.mhlw.go.jp/stf/shingi/shingi-hosho_126730.html

[6] 点字・大活字広報誌「ふれあいらしんばん」｜政府広報オンライン https://www.gov-online.go.jp/pr/media/katsuji/index.html

[7] 羽鳥潤，鈴木久美. "機械翻訳手法に基づいた日本語の読み推定." (2011).

[8] 高橋文彦，森信介. "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." 研究報告自然言語処理 (NL) 2014.15 (2014): 1-10.

[9] 田中牧郎. "言語政策に役立つ，コーパスを用いた語彙表・漢字表などの作成と活用 (< 特集> 日本語コーパス)." 人工知能学会誌 24.5 (2009): 665-672.

[10] 小木曽智信. "多重の読みを持つテキストのコーパス化." 言語資源活用ワークショップ発表論文集= Proceedings of Language Resources Workshop. No. 1. 国立国語研究所, 2017.

[11] Mikolov, Toma, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations." Proceedings of the 2013 conference

[12] 佐藤文一，喜連川優. "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)." 電子情報通信学会技術研究報告= IEICE technical report: 信学技報 119.478 (2020): 17-21.

[13] MeCab: Yet Another Part-of-Speech and Morphological Analyzer https://taku910.github.io/mecab/

[14] 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

[15] PDF Software Open, Read & Edit PDFs FineReader PDF https://pdf.abbyy.com/ja/