テクニカルノート

日本語点字翻訳における例外事例適用閾値の自動調整

山崎 節[†] 小野 智司[†] 中山 茂[†]

本稿では，エラーから事例知識を自動獲得する日本語点字翻訳方式を提案する．従来方式では，例外事例を適用する閾値を，経験的に設定する必要があった．本稿では，事例適用の閾値を個々の事例ごとに設定し，自動的に調整することで，従来は再利用される頻度が低かった事例を効果的に再利用できる方式を提案する．実験により，閾値を自動調整することで，従来よりも 28%の分かち書きのエラーを削減でき，市販システムよりも少ないエラーで点字翻訳を行えることを確認した．

Threshold Adjustment for Exceptional Case Application in Japanese-to-Braille Translation

Takashi Yamasaki,[†] Satoshi Ono[†] and Shigeru Nakayama[†]

We propose a Japanese-to-Braille translation method which can acquire case knowledge from errors. Although existing method involves a common threshold for applying exceptional cases that must be manually adjusted, the proposed method has a threshold for each case, and automatically adjusts the thresholds in order to reuse cases effectively. Experimental result shows that the proposed method can get rid of 28% segment errors of the existing method.

1. はじめに

日本語点字翻訳（点訳）は，漢字かな混じりの文書を，発音に近い点字表記へと変換する処理である[1)~3)]．点訳は独自の曖昧な規則に従って行う必要があるが，「意味の理解を助ける場合には区切って書く」など，計算機上で明確化が困難な規則が存在する．

著者らは，ユーザによるエラーの修正や，既存の点訳文書から自動的に例外的知識を獲得し，事例として再利用する点訳方式 J2B☆を提案している[4),5)]．J2B はルールベース推論（Rule-Based Reasoning: RBR）と事例ベース推論（Case-Based Reasoning: CBR）[6)]を併用し，精度と速度を両立した点訳を行うことができる．しかし J2B では，事例適用の閾値を手動で調整する必要があり，有効に再利用されない事例が存在していた．類似度を自動的に調整する方式も提案されているものの[7)]，類似度のみの調整ではすべての事例を有効に活用することは難しい．

本稿では，J2B において，事例適用の閾値を個々の事例ごとに自動調整するよう改良した Adaptive J2B（aJ2B）を提案する．aJ2B は，従来は利用頻度が低かった事例を効果的に再利用でき，よりエラーの少ない点訳を行うことができる．

2. 点字翻訳の概要と問題点

2.1 日本語点字翻訳

単語間に空白が挿入されていない日本語を点字に翻訳するには，入力された漢字かな混じりの文（原文）に対して，分かち書き，漢字かな変換，および発音に近い表記への変換（点字表記変換）の 3 段階の処理を行う必要がある（**図 1**）．点字表記変換された文字列は点字と 1 対 1 で対応しているため，点字と同一のものとして扱う．また，本稿では，空白を挿入して区切る箇所を ‘■’ を用いて表す．

点訳における分かち書きは，点訳独自の規則に従い，一般の文節区切りよりも短く，形態素解析よりも長く区切る必要がある．たとえば，複合語は，後に続く語の品詞，長さや発音の自然さに応じてより細かい分かち書きを行う（**図 2** (a)）．点字表記変換に関しても同様に曖昧な規則が存在する．たとえば，数量や順序を

† 鹿児島大学工学部情報工学科
Department of Information and Computer Science, Faculty of Engineering, Kagoshima University

☆ 文献 4) では対話型点訳支援システムを J2B と表記しているが，本稿では自動点訳に主眼を置くため，文献 4) のシステムにおける自動点訳処理部を J2B と表す．

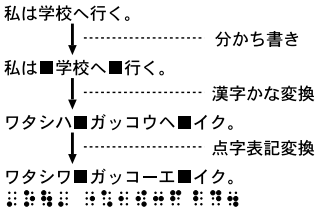
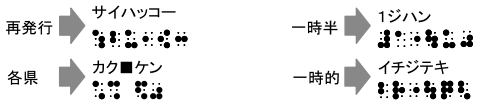


図 1 点字翻訳の手順
Fig. 1 Japanese-to-Braille translation process.



(a) Segmentation examples. (b) Expression examples.

図 2 曖昧な点訳の例
Fig. 2 Examples of ambiguous translation.

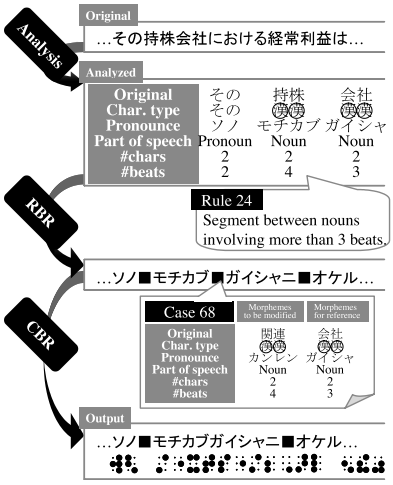


図 3 自動点訳処理部
Fig. 3 Process flow of automatic translation.

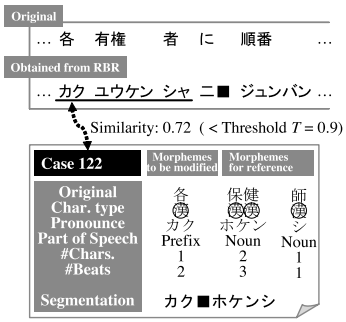


図 4 再利用されない事例の例
Fig. 4 Example of the case not reused.

表す語は，原則として数字を用いて書くが，数量や順序の意味が薄れた語は，意味の理解を妨げない限り，かなを用いて書く（図 2 (b)）．上記のような点訳独特の規則は，計算機上でルール化することが困難であり，エラーのない自動点訳を行うことは難しい．

2.2 従来の点訳方式 J2B

J2B は，RBR と CBR を順次適用する[8] ことで，精度と速度を両立させた点訳方式である[4]（**図 3**）．J2B は，点訳の手引き[2] などに記載されている規則のうち，基本的な点訳の規則をルールとして用い，ルール化が困難な例外的知識を事例として用いる．事例は 1 つのルールに属することになるため，ルールと同じ結論部を持つ事例（正事例）の保持が不要であり，適用されたルールと異なる結論部を持つ事例（例外事例）のみを保持すればよい．このため，CBR 単体で問題解決を行う場合に比べ，事例ベースのサイズを抑えることができる．また，適用されたルールの例外事例のみを検索するため，すなわち，ルールが事例のインデックスの役割を果たすため，事例検索時間を抑えることができる．

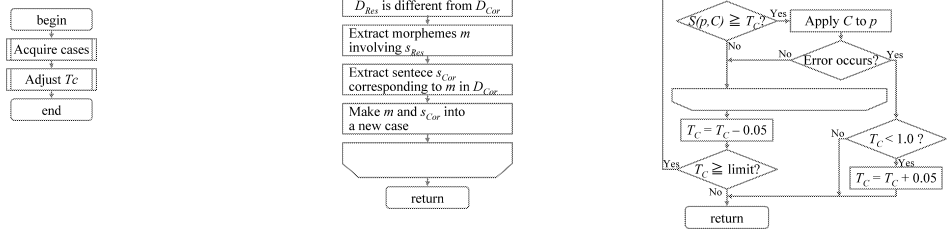
J2B では，ルールに対する正事例を保持しないため，例外事例を適用すべきかどうかを，閾値 T に基づいて判断する．すなわち，点訳対象文書における注目箇所と，例外事例との類似度が T を超えた場合に例外事例を適用する．

2.3 J2B の問題点

閾値 T は全事例で共通であり，T および類似度計算における属性ごとの重み ω_i を経験的に設定することで，事例の効果的な再利用を試みていた．T が大きいほどよく似た箇所にのみ事例を適用することになり，事例の誤用を防ぐことができるものの，事例が適用できる箇所は減少する．反対に T が小さいほど事例を積極的に適用するが，事例の誤用が増加する．J2B では $T = 0.9$ と設定していたが，事例によっては T が高いため，有効に再利用されない事例が存在した．たとえば，接頭辞の後ろは基本的に区切らずに書くが，意味の理解を助ける場合は，発音も考慮して区切る必要があり，**図 4** の「各」の後ろは区切る必要がある．RBR の出力は基本的なルールに従い，「各」の後ろを誤って区切らなかった．事例ベース内には，「各」の後ろで区切る事例 122 が存在したが，接頭辞に続く名詞が異なっていたため，類似度が閾値を超えず，事例 122 が適用されずにエラーとなった．事例 122 を適切に再利用するためには T を下げる必要があるが，T を下げることにより他の事例の誤用が増加してしまう．

3. 提案する AJ2B

本稿で提案する AJ2B は，事例適用の閾値を事例ご



(a) Training procedure. (b) Process flow of case acquisition. (c) Adjustment algorithm of T_C.

図 5 事例ベース生成手順
Fig. 5 Process flow of case base generation.

とに設定し，自動調整するよう J2B に改良を加えたものである．

3.1 知 識 表 現

J2B および AJ2B では，形態素解析によって得られた読みを出力の基本とし，事例およびルールによって出力を修正することで点訳を行う[4),5)]．事例およびルールは，注目している形態素（修正対象形態素）およびその前後の形態素（参照形態素）における，字面，字種，読み，品詞，文字数，および拍数の 6 種類の属性を参照する．各ルールは優先度を持ち，複数のルールが適用できる場合は，優先度が最も高いルールを適用する．

事例は，属するルール番号，および閾値 T_C を持つ．入力文書の注目する箇所 p と事例 C との類似度 $S(p,C)$ は式 (1) で計算する．

$$S(p,C) = \frac{\sum_{n=1}^{m} \sum_i (\omega_i \times \delta_{i,n}(p,C))}{m \sum_i \omega_i} \qquad (1)$$

ここで，m は C に含まれる形態素数，ω_i は属性 i の重みであり，$\delta_{i,n}(p,C)$ は，p と C において，形態素 n における属性 i の値が一致する場合は 1，しない場合は 0 とする．

3.2 自 動 点 訳

AJ2B における自動点訳の処理手順は，事例適用の際に，各事例の閾値 T_C を参照する点を除けば，J2B と同様である（図 3）．入力文書の形態素情報をもとに，まずルールを適用し，次に適用されたルールの例外事例を検索する．類似度が閾値 T_C を超える事例が存在する場合は，事例を適用する．閾値 T_C を超える事例が複数個存在する場合は，最も類似度の高い事例を適用し，類似度が同一の場合は新しい事例を優先する．

3.3 事例ベースの生成

AJ2B における事例ベース生成手順を**図 5** に示す．まず，既存の点字文書から事例を自動獲得し，次に，事例ごとの閾値 T_C を調整する．

3.3.1 事例の自動獲得

事例を自動獲得する手順を図 5 (b) に示す．AJ2B は，既存の点字文書 D_{Cor} および D_{Cor} の原文 D_{Ori} を用いて，事例の自動獲得を行う．まず，すでに獲得済みの事例とルールを用いて原文を自動点訳する．このとき，自動点訳で用いた形態素情報やルール，事例の適用履歴を保持しておく．次に，自動点訳結果 D_{Res} において，正解と一致しない文字列 s_{Res} を抽出し，s_{Res} を含む最短の形態素列 m と，m に対応する正解文 S_{Cor} とを組にして，新たな事例 C とする．s_{Res} に対してルール R が適用されている場合には R を C のインデックスとする．

3.3.2 閾値の自動調整

事例 C の閾値 T_C を調整する手順を図 5 (c) に示す．まず，T_C を初期値 1.0 に設定する．次に，D_{Ori} において，C が属するルール R_C を適用可能な箇所（p とする）すべてに対し，C の適用を試みる．すなわち，$S(p,C) \geq T_C$ であれば C を適用し，適用した結果が正しければ処理を続け，エラーである場合には閾値の調整を終了する．すべての p に対してエラーが発生しなかった場合には，T_C を下げて上記の処理を繰り返す．

J2B では，区切りと読みのエラーを修正する事例を用いるが，区切りに関する事例はより柔軟な適用が必要であるため T_C の下限を 0.7，読みに関する事例では 0.9 とした．

4. 評 価 実 験

4.1 準 備

計算機科学の入門書の 1 章分（17,181 文字）および 1 台の計算機（CPU: Pentium4 3.2 GHz，RAM:

2 GByte）を用いて，aJ2B の有効性を検証する実験を行った．比較対象として，J2B および市販システム Extra Ver.4.0 を用いた．点訳の知識を有するボランティアが作成し，表記法 2001 年版に準拠するよう著者らが修正した点字文書を正解とした．前半部（8,659 文字）を用いて事例ベースの作成を行い，後半部（8,522 文字）を用いて評価を行った．

Extra は，前半部でエラーとなった箇所のうち，ユーザ辞書に登録が可能な 42 個の単語を人手で登録した．J2B および aJ2B は，点訳の手引き[2] および点字表記事典[3] を参照して作成した 94 個のルールを利用するものとし，130 個の事例を前半部から獲得した．J2B の閾値 T は 0.9 とし，J2B および aJ2B は ChaSen を用いて形態素解析を行い，ω_i の調整は行わず，すべての属性で 1.0 とした．評価項目として，後半部におけるエラー数，事例獲得時間（aJ2B は T_C の調整時間を含む），および，後半部の翻訳時間に着目した．

4.2 実験結果

実験結果を**図 6** に示す．それぞれの方式におけるエラーを，分かち書きのエラー（区切りエラー）と，漢字かな変換やかな表記など，分かち書き以外のエラー（読みエラー）とに分類した．図 6 より，aJ2B は J2B よりも区切りエラー数を 28%削減できることが分かる．aJ2B では，図 4 のような接辞に関するエラーや，数詞に関するエラーなどを改善できていた．読みに関する事例では，T_C が 0.9 に設定され，J2B と同様となった．

aJ2B のエラー数は Extra よりも 17%少ない．また，Extra への単語登録は人手で約 40 分の時間が必要であったのに対し，aJ2B による事例獲得および閾値調整は約 36 秒と，精度改善の手間および時間を大幅に短縮できている．なお，aJ2B の翻訳時間は Extra よりも若干速い程度であった．

4.3 考察

一般に，快適に読める点字文書の品質の基準として 1 ページあたり 1 カ所のエラーが望ましいとされている．点字文書は 32 文字 × 20 行で 1 ページとなり，本実験で使用した文書を点字に翻訳すると 28 ページとなる．Extra は 1 ページあたり平均 4.7 カ所，aJ2B は平均 3.9 カ所のエラーとなり，aJ2B でも快適に読める点字文書の品質には至っていない．しかしながら，Extra は，即時性が要求される場合や，人手で点訳を行う場合の前処理としては，十分に有用であるとされている．よって，上記の用途であれば，aJ2B は知識の自動獲得によって実用的な精度で点訳を行えると考える．

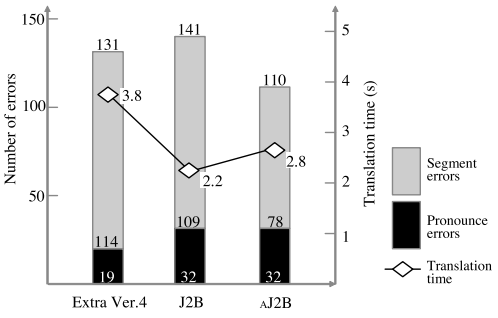


図 6 実験結果
Fig. 6 Experimental results.

なお，aJ2B におけるエラーの約 45%は，カタカナで表記された専門用語に関する形態素解析の失敗に起因していた．たとえば，「オンラインシステム」を 1 つの形態素と認識してしまったために，本来は「オンライン■システム」と区切るべき箇所を，区切ることができかった．ChaSen の辞書から複合名詞を取り除くことで，上記のエラーを解決でき，エラー数を半減できる．他の約 55%はルールや事例の不足に起因しており，より多くの事例を利用することでエラー数を減少させることができると考える．

5. おわりに

事例適用の閾値を自動調整する点訳方式 aJ2B を提案した．aJ2B は，個々の事例に応じた適切な閾値を用いることで，各事例を有効に再利用することができる．実験により，aJ2B は，事例の自動獲得および閾値調整により，J2B および，手動で辞書への単語登録を行った Extra よりも少ないエラーで点訳を行えることを確認した．今後，様々な文書に aJ2B を適用し，その有効性を評価するとともに，類似度を学習する方式[7] との併用を検討する．

参考文献

1) 福井：日本語自動点訳ソフト 4 種の精度の比較，第 2 回視覚障害リハビリテーション研究発表大会論文集，pp.114–117 (1993).
2) 日本盲人社会福祉施設協議会点字図書部会：点訳のてびき（第 3 版）(2002).
3) 日本盲人福祉研究会：最新点字表記辞典（第 5 版）(2002).
4) 小野，高木，浜田，水野，西原：事例知識を用いた日本語点字翻訳とエラー修正支援，ヒューマンインタフェース学会論文誌，Vol.5, No.4, pp.491–498 (2003).
5) 山崎，小野，中山：事例ベースとルールベースを用いた日本語点字翻訳，電気関係学会第 58 回九州支部連合大会，13-1A-05 (2005).

6) Bartsch-Spörl, B., Lenz, M. and Hubner, A: Case-Based Reasoning — Survey and Future Directions, *Proc. 5th Biannual German Conference on Knowledge-Based Systems*, pp.67–89 (1999).
7) Stahl, A. and Gabel, T.: Using Evolution Programs to Learn Local Similarity Measures, *Proc. 5th International Conference on Case-Based Reasoning*, pp.537–551 (2003).
8) Golding, A.R. and Rosenbloom, P.S.: Improving Accuracy by Combining Rule-Based and Case-Based Reasoning, *Artificial Intelligence*, Vol.87, pp.215–254 (1996).

(平成 17 年 11 月 30 日受付)
(平成 18 年 2 月 1 日採録)

山崎　節

昭和 58 年生．平成 17 年鹿児島大学工学部情報工学科卒業．現在，同大学院理工学研究科情報工学専攻に在学．主として知識ベースに基づく点字翻訳に関する研究に従事．

小野　智司（正会員）

平成 9 年筑波大学第三学群情報学類卒業．平成 11 年同大学院修士課程理工学研究科修了．平成 14 年同大学院博士課程工学研究科修了．平成 13 年日本学術振興会特別研究員．平成 15 年より鹿児島大学工学部情報工学科助手．博士（工学）．知識処理，機械学習，日本語点字翻訳等の研究に従事．電子情報通信学会，ヒューマンインタフェース学会，情報知識学会各会員．

中山　茂（正会員）

昭和 52 年京都大学大学院博士課程修了，同年上智大学助手，昭和 56 年京都工芸繊維大学助手，昭和 62 年兵庫教育大学助教授，平成 9 年より鹿児島大学工学部情報工学科教授．京都大学工学博士．平成 8 年情報文化学会学会賞受賞，平成 12 年九州工学教育協会賞受賞．主として，量子情報工学，群知能，分散オブジェクト，協調型仮想環境，並列 GA の研究に従事．電気学会，電子情報通信学会各会員．