形態素解析のエラー分析

鍛治 伸裕[†] 森 信介[‡] 高橋 文彦[◦] 笹田 鉄朗[‡] 斉藤 いつみ[§] 服部 圭悟[*] 村脇 有吾[◦] 内海 慶[¶]

[†]東京大学 生産技術研究所　　[‡]京都大学 学術情報メディアセンター

[◦]京都大学大学院 情報学研究科　　[§]NTT メディアインテリジェンス研究所

[*]富士ゼロックス株式会社　　[◦]九州大学大学院 システム情報科学研究院

[¶]デンソーアイティーラボラトリ

kaji@tkl.iis.u-tokyo.ac.jp

{mori,takahashi,sasada}@ar.media.kyoto-u.ac.jp

saito.itsumi@lab.ntt.co.jp

keigo.hattori@fujixerox.co.jp

murawaki@ait.kyushu-u.ac.jp

kuchiumi@d-itlab.co.jp

1 はじめに

近年，ソーシャルメディアの拡大により，多種多様なテキストデータを処理する必要性が高まりつつある．しかしながら，そうしたテキストにおいては，新語や固有名詞といった未知語が頻出し，また話し言葉が多用されるため，一般的な形態素解析器の精度は決して十分なものではない．

形態素解析器の更なる精度向上のためには，現状の解析誤りがどのような要因によるものであるのかを分析し，明らかにすることが必要不可欠であると考えられる．このような問題意識にもとづき，形態素解析におけるエラーの分析を行った．

2 仮名漢字変換ログを利用した単語分割器のエラー分析

文章を作成する際に使用する仮名漢字変換システムのログデータは，ノイズの混じったアノテーション付きコーパスとみなすことができる．そのため，最近では，この仮名漢字変換ログを付加的な学習コーパスとして活用することによって，単語分割器の精度を向上させるという試みが行われている [5]．本節では，仮名漢字変換ログを用いた単語分割器を対象としたエラー分析について述べる．この手法の詳細については文献 [5] を参照されたい．

2.1 実験データ

まずはじめに，実験で用いた学習コーパスおよびテストコーパスの概要を説明する (表 1)．

2.1.1 現代日本語書き言葉均衡コーパス

現代日本語書き言葉均衡コーパス (BCCWJ)[2] の Core データから 56,753 文，6,025 文を抽出し，それぞれを BCCWJ-train，BCCWJ-test とした．

2.1.2 ツイートデータ

2014/05/19-2014/05/22，2014/06/02-2014/06/04 に収集した 2,659,168 件のツイートから無作為に 1,592 件のツイートを選択し，人手でアノテーションを行った．アノテーション基準は BCCWJ の短単位に準拠し，これに加えて活用語尾を分割する．これらのツイートから，メンション，ハッシュタグ，URL，ティッカーシンボルを除いた本文部分を抽出した．これらのツイッター特有のシンボルを除いた理由としては，正規表現で抽出が可能なので，単語分割の対象にならないと判断したためである．次に，本文に改行を含むツイートは改行文字前後で文を分割した．これらの処理によって，2,976 文を得た．これを 8 : 2 に分け，それぞれを TWI-train, TWI-test とした．

表 1: 実験で用いる言語資源

学習コーパス		
記号	文数	単語数
BCCWJ-train	56,753	1,324,951
TWI-train	2,354	29,460
記号	エントリー数	単語境界情報
AS-IS-log	32,119	39,708
CHUNK-log	6,572	63,144
MCONV-log	4,610	10,852
CHUNK-MCONV-log	1,218	14,242
テストコーパス		
記号	文数	単語数
BCCWJ-test	6,025	148,929
TWI-test	588	7,498

2.1.3 仮名漢字変換ログ

2014/04/13-2014/10/31に，5 人のユーザーに仮名漢字変換システムを提供し，実際にツイートを入力する際に使用してもらった．さらに，2014/11/01 から本システムを Web 上で公開，配布し，仮名漢字変換ログを継続して収集した[1]．ユーザー数は最大 17 名であった．2014/04/13-2015/01/01(263 日間) のログを使用する．

仮名漢字変換ログはノイズありの単語分割済みかつ読み付与済みの文断片である．このようなログを単語分割の学習コーパスとして利用するために次の方法を検討した．

AS-IS-log: 確定結果は単語境界情報が付与された部分的アノテーションコーパスと見なすことが出来る．このため，確定結果をそのままコーパスとして利用する．

CHUNK-log: 文断片の問題を回避するために，確定結果の時間を参照して連結する．確定時間と次の入力の開始時間の差が s 以下の場合，この確定結果を連結する．$s = 0.5$[s] とした．

MCONV-log: ユーザーが編集した文は，ノイズを含む可能性が低く，未知語を含む可能性が高い，と考えられる．このため，変換の操作が n 回以上のログのみを学習コーパスとして使用するフィルタリング処理を行う．$n = 2$ とした．

CHUNK-MCONV-log: MCONV-log に対して，CHUNK-log と同様に，時間を参照して確定結果を連結する．

[1] http://plata.ar.media.kyoto-u.ac.jp/takahasi/kagami/

表 2: TWI-test に対する単語分割精度

	再現率	適合率	F 値
BCCWJ-train	90.31	94.05	92.14
+ AS-IS-log	90.33	93.77	92.02
+ CHUNK-log	91.04	94.29	92.64
+ MCONV-log	90.62	94.09	92.32
+ CHUNK-MCONV-log	91.40	94.45	92.90

表 3: BCCWJ-test に対する単語分割精度

	再現率	適合率	F 値
BCCWJ-train	99.01	98.97	98.99
+ AS-IS-log	99.02	98.87	98.94
+ CHUNK-log	99.05	98.88	98.96
+ MCONV-log	98.98	98.91	98.95
+ CHUNK-MCONV-log	98.98	98.92	98.95

2.2 エラー分析

前節で説明したコーパスを用いて単語分割器 kytea[3] のモデルを学習し，その精度の評価を行った．解析結果とテストコーパスの単語単位のアライメントを取り，再現率，適合率，その調和平均 (F 値) で評価を行った (表 2 と表 3)．これらを比較すると，やはりツイートの単語分割が困難である事が分かる．表 2 では，MCONV-log は精度の向上が見られ，確定結果に含まれるノイズが変換操作回数によりフィルタリングされたことを示す．さらに，CHUNK-log を連結することによって得られる CHUNK-MCONV-log において，有意 ($p = 0.01$) に精度が向上した．表 3 では，ログを用いることによって多少精度は低下するものの，有意な精度の低下は見られなかった．以下ではこの実験結果をもとにエラーの分析を行う．

まず，分割を誤った単語の分析を行った．表 4 に，TWI-test において分割を誤った単語の数と，そのうち未知語である単語の数とその割合を示した．ここでは，BCCWJ-train と UniDic のどちらにも含まれない単語を未知語としている．AS-IS-log を用いると BCCWJ-train のみを用いた場合よりも誤り数が多い．しかし，AS-IS-log を含め，ログ由来のコーパスを用いた場合，誤り単語の未知語率が下がった．これは，ログを追加したことで未知語が解析できるようになったことを意味する．一方で AS-IS-log は，ノイズが多く含まれるため，既知語の解析誤りが多い．実際に，AS-IS-log の解析誤りに含まれるが，BCCWJ-train の解析誤りに含まれない単語は，97%が既知語であった．

次に，文献 [6] を参考に未知語を分類した．次のような各項目に，未知語が含まれる文を見ながら，項目の重複を許して分類を行った．

- **表記揺れ**：既知語と読みが同じだが表記が異なる単語，または正式名称のある未知語と表記の異なる単語 (例：キオク).
- **連濁**：2 つの語が結合して複合語をつくるとき，後ろの語頭の清音が濁音に変わった単語 (例：(掘り) ごたつ).
- **長音化**：既知語，または既知語が派生した未知語の母音が，長音に変わった単語 (例：たのしー (たのしい)，たけー (たけえ，たかい)).
- **小文字化**：既知語，または既知語が派生した未知語，の文字を小書き文字で置き換えて表記した単語 (例：ぁなた).
- **記号化**：文字に形が似た記号で置き換えた単語 (例：あや $_{\iota}$ い).
- **話し言葉・方言**：話し言葉や方言に特有な単語．言い淀みの表現や最後まで言わない表現を含む (例：やん).
- **オノマトペ**：擬音語や擬態語 (例：だだん).
- **感動詞**：感動，呼びかけ，応答などを表す，活用しない自立語のうち主語や修飾語にならず主として独立した文節を構成する単語 (例：イヤッホーゥ).
- **顔文字・アスキーアート**：視覚的表現技法．単語を強調するためのものも含む (例 :! だ！れ！).
- **固有名詞**：同じ種類に属する事物から一つの事物を区別するために，それのみに与えられた名称を表す単語．人名・ユーザー名・ID・商品名・キャラクター名・サービス名・地名・企業名・ブランド名などの正式名称を含むが，その略称や通称は含まない (例：スパイダーマン).
- **挿入**：既知語に長音記号，小書き文字，母音字，促音文字のいずれかが挿入された単語 (例：どこー).
- **他言語**：日本語以外の言語に含まれる語の表記と音の片仮名表記．固有名詞は含まない (例：ＡＱＵＡＲＩＵＭ).
- **誤り入力**：表記揺れに当てはまらない単語 (例：よろすく (よろしく)).
- **新語・低頻度語**：既知語の複合語や，商品名・キャラクター名・サービス名・地名・企業名などの略称や通称，国語辞書に含まれる単語．数字を含む (例：セルカ，ニコ生).

表 5 に，未知語の頻度，割合，1 種類あたりの頻度を示した．Twitter の未知語は，新語・低頻度語，固有名詞が多かった．Twitter では多様な分野の話題が扱われいるためこのような結果になったと考えられる．また公的な文書と異なるため，表記揺れが多く見られた．表記揺れでは，既知語を全て平仮名で表記する例が多かった．一方で誤り入力は少ないため，ユーザは意図して表記揺れの表現を用いていると考えられる．顔文字・アスキーアートは 13%程度あった．1 種類あたりの頻度が小さいため，多様な表現があることがわかる．従ってコーパスや辞書による追加よりも，個別のモデルによる対応が効果的であると考えられる．今回の未知語には，連濁と記号化は含まれず，長音化と小文字化は低頻度であった．

表 6 に，実験で収集したログ (LOG)，実験で用いた仮名漢字変換システムの語彙 (TWI-pstc)，ツイートの学習コーパス (TWI-train) に含まれる未知語の頻度と，各項目ごとに TWI-test に対する適合率を示した．ログには，ユーザーに選択された仮名漢字変換システムの語彙が記録されるため，ログを無限に集めると TWI-pstc に近づく．今回の収集したログでは約 5%の未知語が記録されており，TWI-pstc の適合率は約 39%だった．これに加えて，TWI-train の適合率は約 41%であることから，ログを収集するし続けることで TWI-train 程度まで精度が向上できると考えられる．小文字化，記号化，顔文字は，読み推定候補に正しい読みが出ないので，正しい読みが擬似確率的タグ付与コーパスに現れず，本論文で提案する方法では改善できない．小文字化，記号化の改善方法としては，置き換わる文字を規則に当てはめることが出来るので，すべての可能な未知語候補を元の単語の変換候補として提示すれば変換ログとして獲得が可能である．しかし，出現頻度が低いにもかかわらず，煩雑な変換候補が大量に列挙される事になり，実用する際に弊害があると考えられる．顔文字は，文献 [4] の方法を用いて抽出可能だが，構成される文字の読みとインプットメソッド利用者の考える入力に違いがあるため変換ログとしての獲得が困難である．

TWI-pstc は，話し言葉・方言，固有名詞，新語・低頻度語を広く網羅していた．実際にログに含まれている未知語もこれらのものが多かった．一方で TWI-pstc では表記揺れの適合率が低い．これは，擬似確率的タ

表 7: 辞書拡張による解析結果の変化.

		辞書拡張後	
		正解	不正解
辞書拡張前	正解	11,641	154
	不正解	1,312	1,217

グ付与で表記揺れを正しく単語分割するのが困難であることが原因であると考えられる．しかし，表記揺れは平仮名のみで表記しているものが多く，変換せずとも入力できるので，ログに記録されやすい．

3 辞書拡張法のエラー分析

辞書に新しい形態素を追加登録することは，形態素解析器の精度を高めるための最も簡便な方法であり，これまでにも広く用いられてきた [7]．しかし，辞書を拡張することによって，それまで正しく解析できていた箇所にどのくらい副作用が発生するのか，辞書を拡張しても正しく解析できない形態素はどのような種類のものなのかなど，その誤りに関する調査はこれまで十分に行われてきたとは言い難い．

そこで，テストコーパスに出現する未知語をあらかじめ追加し拡張した辞書を用いた形態素解析器と，元の辞書を用いた形態素解析器の比較を行った．テストコーパスにはマイクロブログのテキスト (1831 文，14,324 形態素) に人手で形態素情報を付与したものを用いた [1]．形態素解析器は MeCab (ver. 0.996)，解析用辞書は JUMAN (ver. 7.1) を用いた[2]．未知語を MeCab 用解析辞書に追加する際には，その「コスト」を設定する必要があるが，本実験では，辞書に既に登録されている同一品詞形態素のコストの平均値を未知語のコスト値とした．なお，辞書に追加された形態素の異なり数は 2471 であった．

表 7 に，辞書拡張を行う前後での正解と不正解の形態素数の変化を示す．ここでは，形態素の区切りと品詞大分類を正しく出力できた形態素を正解に数えている．品詞細分類を正解の基準として考慮していないのは，名詞の品詞細分類の曖昧性解消を行うためには (例えば「豊田」が地名なのか人名なのか)，辞書の拡張を行うだけではそもそも不十分な場合が多く，それらを分析の対象から外すためである．

表 7 から，全 14,324 形態素中，1312 形態素に対して改善が見られ，その結果として正解形態素の数が大きく増加していることが確認できる．また，単語単位の F 値を計算したところ，76.6 から 90.0 へと向上していることも確認することができた．しかしながら，辞書の拡張を行った副作用として，拡張前は正解していたにも関わらず，拡張後は不正解となってしまった事例が 154 例見られた．さらに，辞書の拡張を行う行わないに関わらず，不正解であった形態素も 1217 例存在していた．以下では，これら 2 種類の誤り事例の分析を行う．

[2]https://code.google.com/p/mecab

3.1 正解から不正解に変化した事例

まず，辞書拡張によって正解から不正解へと変化した 154 の事例の分析を行う．基本的に，これらの解析誤りの原因は，辞書の拡張により誤った解析候補が新たに生成されるようになり，そのコストが適切に調整されていなかったため，適切な解析結果よりもコストが低くなってしまったためであると考えられる．

こうした誤りがどのような箇所に発生しやすいのかを調べるため，解析に失敗した 154 の形態素を集計し，そこに頻度する品詞を調べた (表 8)．この結果から，接尾辞や助詞など，機能語に関連する箇所に誤りが集まっていることが分かる．

さらに，どのような未知語を辞書に追加した時に，こうした誤りが発生しやすいのかを調査した．辞書拡張後の出力に含まれる不正解の形態素のうち，154 の形態素と文字列の範囲が重複している形態素は，誤りを生じさせた直接的な原因になっていると考えられる．例えば，表 8 の 1 行目の出力例では，名詞「いま」と動詞「す」の 2 つが，解析誤りの直接的な原因になっていると考えられる．そこで，それらの中から，今回の実験によって辞書に追加された形態素の頻度を集計した (表 9)．その結果，頻度の上位は主にくだけた表記の機能語であり，平仮名表記でなおかつ文字数の少ないものが多いことが分かった．このような形態素は，テキスト (とりわけ機能語の部分) と頻繁にマッチするため，コストの調整に失敗したときの影響が大きいものと考えられる．

3.2 不正解のまま変化しなかった事例

次に，辞書の拡張を行ったにも関わらず，不正解のままであった 1217 事例の分析を行う．表 10 に誤りの多かった上位 5 つの品詞と，その出力例を示す．名詞や特殊 (記号など) といった品詞が上位となっており，表 8 とは大きく傾向が異なっていることが分かる．

表 4: ツイートの分割誤りの分布

	誤り単語数 (E)	誤り未知語数 (U)	割合 (U/E)
BCCWJ-train	446	103	23.09
+ AS-IS-log	467	89	19.06
+ CHUNK-log	428	81	18.93
+ MCONV-log	443	88	19.86
+ CHUNK-MCONV-log	413	74	17.79

表 5: ツイートに含まれる未知語の分布

分類	頻度	割合	頻度/種類
表記揺れ	99	21.85	4.13
連濁	0	0.00	-
長音化	6	1.32	1.50
小文字化	3	0.66	1.00
記号化	0	0.00	-
話し言葉・方言	31	6.84	1.07
オノマトペ	19	4.19	1.06
感動詞	11	2.43	1.00
顔文字・アスキーアート	58	12.80	1.05
固有名詞	92	20.31	1.15
挿入	53	11.70	1.29
他言語	18	3.97	1.00
誤り入力	5	1.10	1.25
新語・低頻度語	104	22.96	1.20
合計	453	100.00	1.36

これらの誤りの原因を分析したところ，未知語処理に起因するものが多く見られた．例えば，表 10 の 1 行目と 2 行目では，正解形態素が辞書に登録されているにも関わらず，デートパッチなどの辞書にない形態素を使った解析結果のほうが優先されてしまっている．このように，未知語の解釈が優先されてしまった事例は 1217 中 382 件と，多くの割合を占めていることが分かった．なかでも名詞と特殊が大部分を占めており，それぞれ 152，148 事例が見られた．

4 おわりに

本稿では，仮名漢字変換ログの利用，辞書の拡張という二つの手法を対象として，形態素解析のエラー分析を行った．今後は，これらの結果を踏まえて，解析技術の改良を行っていきたい．

参考文献

[1] Nobuhiro Kaji and Masaru Kitsuregawa. Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 99–109, Doha, Qatar, October 2014. Association for Computational Linguistics.

[2] Kikuo Maekawa. Balanced corpus of contemporary written japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*, pp. 101–102, 2008.

[3] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings*

表 6: 各コーパスに含まれる未知語の分布

分類	LOG		TWI-pstc		TWI-train	
	頻度	適合率	頻度	適合率	頻度	適合率
表記揺れ	2	2	11	11	80	80
連濁	0	-	0	-	0	-
長音化	0	0	3	50	3	50
小文字化	0	0	0	0	0	0
記号化	0	-	0	-	0	-
話し言葉・方言	1	3	17	54	6	19
オノマトペ	0	0	8	42	0	0
感動詞	0	0	5	45	0	0
顔文字・アスキーアート	0	0	1	1	15	25
固有名詞	2	2	46	50	22	23
挿入	0	0	17	32	15	28
他言語	0	0	8	44	1	5
誤り入力	0	0	4	80	0	0
新語・低頻度語	19	18	78	75	52	50
すべて	21	4	176	38	184	40

of the 49th Annual Meeting of the Association for Computational Linguistics, pp. 529–533, 2011.

[4] 渡邉謙一, 高橋寛幸, 但馬康宏, 菊井玄一郎. 系列ラベリングによる顔文字の自動抽出と顔文字辞書の構築. 言語処理学会第 19 回年次大会発表論文集, 2013.

[5] 高橋文彦, 森信介. 仮名漢字変換ログを用いた単語分割の精度向上. 言語処理学会第 21 回年次大会発表論文集, 2015.

[6] 笹野遼平, 黒橋禎夫, 奥村学. 日本語形態素解析における未知語処理の一手法-既知語から派生した表記と未知オノマトペの処理-. 自然言語処理, Vol. 21, No. 6, pp. 1183–1205, 2014.

[7] 鍜治伸裕. 日本語形態素解析とその周辺領域における最近の研究動向. 日本知能情報ファジィ学会誌, Vol. 25, No. 6, pp. 174–183, 2013.

表 8: 辞書拡張によって出力が正解から不正解に変化した上位 5 品詞の頻度と出力例．下線は辞書拡張により追加された形態素を表す．

品詞　頻度	解析結果の変化例 (上: 辞書拡張前，下: 辞書拡張後)
接尾辞　34	紹介/名詞 を/助詞 して/動詞 **い/接尾辞 ます/接尾辞** ので/助動詞 紹介/名詞 を/助詞 して/動詞 いま/名詞 す/動詞 ので/助動詞
助詞　26	僕/名詞 は/助詞 **ね/助詞** 、/特殊 君/名詞 が/助詞 僕/名詞 は/助詞 ね/接尾辞 、/特殊 君/名詞 が/助詞
名詞　25	冤罪/名詞 も/助詞 **くそ/名詞** も/助詞 ない/形容詞 やろ/助動詞 冤罪/名詞 も/助詞 くそ/副詞 も/助詞 ない/形容詞 やろ/助動詞
動詞　24	先/名詞 に/助詞 **見ちゃ/動詞** う/接尾辞 んです/助動詞 先/名詞 に/助詞 見/動詞 ちゃう/動詞 んです/助動詞
助動詞　13	食べて/動詞 ん/助動詞 **のに/助動詞** 太/形容詞 ん/助動詞 ない/接尾辞 食べて/動詞 ん/接尾辞 の/名詞 に/助詞 太ん/動詞 ない/接尾辞

表 9: 誤りを引き起こした直接的な要因と考えられる未知語の例．

誤り要因となった回数	未知語 表層形　品詞	評価コーパスでの出現例
15	て　助詞	国会/名詞 議員/名詞 **て/助詞** バカだ/形容詞 な/助詞
10	ね　接尾辞	店/名詞 ない/形容詞 んじゃ/助動詞 **ね/接尾辞** ？/特殊
5	ど　指示詞	**ど/指示詞** や/判定詞
5	ん　接尾辞	何/名詞 の/助詞 仕事/名詞 して/動詞 **ん/接尾辞** の/助詞
5	ある　助詞	鈴茂/名詞 買い戻した/動詞 **ある/助詞**

表 10: 辞書拡張を行う前後で出力が不正解のまま変化しなかった事例の品詞ごとの分布．枠線で囲まれた形態素は，未知語処理により動的に生成されたものを表す．

品詞　頻度	解析誤りの例 (上: 正解，下: システム出力)
名詞　454	大型/名詞 **アップデート/名詞 パッチ/名詞** ２．１/名詞 大型/名詞 アップ/名詞 デートパッチ/名詞 ２．１/名詞
特殊　242	逮捕/名詞 者/接尾辞 きたー/動詞 **ｗｗｗｗｗ/特殊** 逮捕/名詞 者/接尾辞 きたー/動詞 ｗｗｗｗｗ/名詞
助詞　112	また/副詞 **普通/名詞 に/助詞** 戻ろう/動詞 また/副詞 普通に/形容詞 戻ろう/動詞
形容詞　75	**めんどくさい/形容詞** めん/名詞 どくさい/名詞
接尾辞　71	ラーメン/名詞 来て/動詞 **る/接尾辞** し/助詞 ラーメン/名詞 来/名詞 てる/動詞 し/助詞