日本語学習者の作文の誤り訂正に向けた単語分割

藤野 拓也 †　水本 智也 †　小町 守 †　永田 昌明 ††　松本 裕治 †

† 奈良先端科学技術大学院大学　　††NTT コミュニケーション科学基礎研究所
†{takuya-fu, tomoya-m, komachi, matsu}@is.naist.jp
††nagata.masaaki@lab.ntt.co.jp

1 はじめに

国際交流基金の「2009年海外日本語教育機関調査」よると，海外の133の国と地域で，365万人の人々が日本語を学習している[*1]．これに対し，日本語教師の数は，学習者76人に対し1人と不足していることから，作文誤り検出・訂正などの自動添削を用いた日本語教師・学習者の支援が求められている．

自然言語処理を用いた作文誤り検出・訂正の前処理として，まず，単語分割を行う必要がある [7]．また水本ら [1] の統計的機械翻訳の手法を用いた自動誤り訂正においては，単語単位のアライメントを素性に用いた場合，正しく単語分割できた場合は訂正の精度が高くなることが述べられている．しかし，学習者の文は，ひらがなが多い，誤りを含む，などの理由から単語分割に失敗しやすい．これらのことから，学習者の文を誤り訂正に適した形に単語分割することは，誤り訂正の精度向上に有用であると考えられる．

本研究では日本語学習者の文を，誤り訂正を行うのに適した形に単語分割することを目的とし，日本語学習者の文とそれに対する添削文がペアになった添削コーパスから，単語境界がアノテーションされた学習者コーパスを自動的に作成する手法を提案する．学習者コーパスを用いることで学習者の文の誤りを含む箇所の単語分割精度が向上することを示した．

2 日本語学習者の文の単語分割

2.1 誤りを含んだ文の単語分割

現在，主として使われている日本語の単語分割の手法は，ルールベースのもの [3] や，機械学習に基づくもの [4][5][6] である．これらは，誤りを含んだ文に対しては，そうでない文を対象とした場合に比べて精度が落ちる傾向がある．理由として，誤りを含んだ文に対してはルールを定めることや，誤りを含んだ文の単語分割を学習させるための大量のコーパスを得ることが，困難であることが考えられる．

2.2 単語分割済みコーパスの作成

学習者の文は誤りを含んでいるため，単語分割に失敗する傾向がある．一方，学習者の文に対して誤り訂正などの添削が行われた文（添削文）は比較的，単語分割が容易である．このことから，提案手法では，まず添削文を単語分割し，その単語境界を学習者の文に反映させるという方法で，自動的に大量の単語分割コーパスを作成する．

単語分割コーパスの作成方法は，学習者の文と添削文で変更されていない単語や，添削文側で削除されている単語についてもアノテーションする**フルアノテーション**と，文対の前後で変更されている箇所のみをアノテーションする**部分的アノテーション**の2通りが考えられる．

フルアノテーションでは，文対の前後で変更のない単語や，添削後に削除されている単語については，再学習前のモデルの単語分割器による単語分割結果を用いる．このため，これらの部分については同じ素性を再度学習することになる．一方，部分的アノテーションでは文対の前後で変更されている箇所のみを，単語単位でアノテーションしていくため，誤りを含む箇所の単語分割精度を直接的に向上させることができると予想される．

アノテーションの例として，次の文の対を考える

学習者の文　でもじよずじゃりません
添削文　　でもじょうずじゃありません

文対間で，単語単位[*2]の対応を取ると以下のようになる．

学習者の文	で	も	じよず	じゃ		り	ま	せ	ん
添削文	で	も	じょうず	じゃ	あ	り	ま	せ	ん

この文対を提案手法でアノテーションすると以下のようになる．

フルアノテーション
で　も　じよず　じゃ　り　ま　せ　ん
部分的アノテーション
でも | じ-よ-ず | じゃりません

フルアノテーションではスペースが単語境界ありを表す．部分的アノテーションでは，'|' は単語境界あり，'-' は単語境界なしを表し，それ以外の部分についてはアノテーションなし，を表す．

同じ例を用いて，提案手法でアノテーションする手順の詳細を以下で述べる．

1. 学習者の文と添削文間での，文字の挿入・削除操作の箇所を，動的計画法を用いて求める．

[*1] http://www.jpf.go.jp/j/japanese/survey/result/

[*2] 本研究での誤り含む箇所以外の単語の分割基準は，単語分割器 KyTea (http://www.phontron.com/kytea/index-ja.html) で採用されている基準を用いる (http://plata.ar.media.kyoto-u.ac.jp/sasada/research/project/corpus/)
短単位基準に加えて，語幹と語尾を別単語として扱う．

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

挿入を **T** タグ，削除を **S** タグ，操作なしを **N** タグで表すと以下のようになる．

学習者の文	で	も	じ	よ			ず	じ	ゃ		り	ま	せ	ん
添削文			で	も	じ		ょ	う	ず	じ	ゃ	あ	り	ま	せ	ん
文字操作タグ	N	N	N	S	T	T	N	N	N	T	N	N	N	N

2. 学習者の文および，添削文を単語分割器で分割する．

単語の先頭の文字を **B** タグ，先頭以外の文字を **I** タグで表すと以下のようになる．以下の例では，一行目の単語分割タグが学習者の文の単語分割を，5 行目の単語分割タグが添削文の単語分割を表す．

単語分割 (学)	B	B	I	B			B	B	I		B	B	B	B
学習者の文	で	も	じ	よ			ず	じ	ゃ		り	ま	せ	ん
添削文			で	も	じ		ょ	う	ず	じ	ゃ	あ	り	ま	せ	ん
文字操作タグ	N	N	N	S	T	T	N	N	N	T	N	N	N	N
単語分割 (添)	B	B	B		I	I	I	B	I	B	B	B	B	B

3. 添削文の単語分割箇所を学習者の文に反映させる．

添削前後で，単語を構成する文字の変更がない単語は，添削文の単語分割を用いる．例では「**で**」「**も**」「**じゃ**」などである．

文字が挿入された箇所は，添削文において挿入文字の前または後の文字と同じ単語を構成していた場合，学習者の文の側も単語として分割する．そうでない場合は挿入箇所前後に単語境界があるものとする．例では，「ありません」の「**あ**」が該当する．

文字の削除と挿入が連続している箇所は，挿入の場合と同様に，挿入文字の前または後ろの文字が挿入文字前後の文字が一単語であった場合，学習者の文の側も一単語として分割する．例では「**じよず**」と「**じょうず**」が該当する．学習者の文の「**よ**」は，添削文側では単語分割タグが無い．しかし，添削文の「じょうず」の「**じ**」から「**ず**」までが一単語であるので，学習者の文の「**じよず**」も一単語とする．

文字が削除され，対応する単語が存在しない箇所の単語分割は学習者の文のものを用いる．

例での学習者の文は以下のように単語分割される．

単語分割 (学)	B	B	B	I	I	B	I	B	B	B	B
学習者の文	で	も	じ	よ	ず	じ	ゃ	り	ま	せ	ん

4. 学習者の文をアノテーションする．

フルアノテーションの場合　学習者の文の単語境界すべてをアノテーションする．

部分的アノテーションの場合　文対前後で文字の変更があった単語の単語境界のみをアノテーションする．例では「**じょず**」のみが該当する．

3 提案手法のコーパスの評価実験

提案手法で作成したコーパスによる単語分割器の学習が，単語分割に与える影響を評価するための実験を行った．

3.1 単語分割器とベースライン

学習者の文を単語分割するための単語分割器として，分野適応に適しているとされる点予測 [4] を用いた単語分割器である KyTea-0.3.2 を使用する．

ベースラインは KyTea と共に配布されているモデル[*3]のうち，以下の 2 つを用いた．

高性能 SVM モデル：(Baseline)　BCCWJ と UniDic などの言語資源を用いて学習されたモデル．一般文書に対する単語分割について，配布されているモデルの中で最も高い性能を持つ．

ひらがなもでる（Baseline(H)）　ひらがなで書かれている単語の多い文章を解析できるモデル．日本語学習者の作文はひらがなが多い傾向があるので，比較のため用いる．

単語分割器の学習にあたって，KyTea の素性ファイル kytea-0.3.2.feat[*4]を使用した．これはモデル学習の際に用いると，KyTea と共に配布されているモデルと同等のモデル (Baseline) を学習できるもので，他の言語資源を追加することでモデルを再学習させる事ができる．今回の評価実験では，このモデルに提案手法による学習者コーパスを追加して，再学習した結果を評価した．

学習プログラム train-kytea の学習器は線形 SVM を用いた．学習の素性は文字 3-gram，文字窓幅 3，文字種 3-gram，文字種窓幅 3 を用いた．

3.2 データセット

日本語学習者の文と添削文がペアになった添削コーパスとして，言語学習者の相互添削型 SNS「Lang-8」[*5]をクロールして取得した 2010 年 12 月までのデータを用いた．このうち，日本語学習者の添削文対 1,361,086 文を使用した．テストデータは添削コーパスからランダムに取り出した 500 文とし，残りを訓練データとした．訓練データは，フルアノテーションコーパスと部分的アノテーションコーパスの両方を作成し，それぞれで実験を行った．添削コーパスは挿入数・削除数が少ないものが多く，また，コメントを付加するのみの添削など特殊なものもあり（参照 [1]），その影響も比較するため，訓練データを以下のように設定した．

All：　全訓練データ 1,360,586 文

Cut5：　訓練データのうち，削除数・挿入数がともに 5 以下のもの 753,355 文

Cut5Sub3：　Cut5 のうち，さらに削除数・挿入数の差が 3 以下のもの 46,627 文

更にそれぞれのデータについて，訓練データの大きさの影響を比較するため，10 万文，30 万文，50 万文，全てのデータでトレーニングを行った．

[*3] http://www.phontron.com/kytea/model-ja.html

[*4] http://www.phontron.com/kytea/download/kytea-0.3.2.feat.gz

[*5] http://lang-8.com/

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

表 1: 各文字境界の種類と比率

文字境界	個数	全体に対する比率 [%]
正解箇所	11,912	92.45
誤り箇所	259	2.01
交差箇所	615	4.77
欠落箇所	99	0.77

本研究では，誤りを含む文での「単語」の定義を，「正しい単語に入れ替えると，正しい文になる文字の列」とした．正解データはこの定義に従って人手で単語分割し，誤り箇所にタグ付したテストデータである．付与するタグは，誤った単語を<e>誤った単語</e>のタグで，単語欠落誤りの箇所を<d />のタグで表現する．

3.3 評価尺度

単語境界推定精度と再現率と適合率と F 値 [4]，および，誤りを含む箇所を考慮した単語境界推定精度で評価した．

単語境界推定精度は各文字間で自動単語分割結果の判断が一致した割合である．再学習が文中の誤り箇所の単語分割にどう影響するかを調べるために，単語境界推定精度を以下の 5 通りで評価した．

全境界: 文字間の全ての境界．

正解箇所: 正しい単語を構成する文字間の境界，および，正しい単語が連続する部分の境界．

誤り箇所: 誤った単語を構成する文字間の境界，および，誤った単語が連続する箇所の境界．

交差箇所: 誤った単語と正しい単語の境界．

欠落箇所: 単語が抜け落ちている，欠落誤りの箇所の境界．

単語境界の例として，以下のような場合を考える．

タグ付き正解コーパス：

で も <e>じよず</e> じゃ <d />り ま せ ん

このようにタグが付いている場合，文字間の各境界は，正解箇所を C，誤り箇所を E，交差箇所を X，欠落箇所を D で表すと，以下のようになる．

正解コーパスの文字境界：

でCもXじEょEずXじCゃDりCまCせCん

テストデータ内での各境界の個数と比率は，**表 1** の通りである．

また，再現率と適合率の定義は次のとおりである．正解文に含まれる延べ単語数を N_{REF}，自動単語分割結果に含まれる延べ単語数を N_{SYS}，双方で分割が一致した延べ単語数を N_{COR} とすると，再現率は N_{COR}/N_{REF} と定義され，適合率は N_{COR}/N_{SYS} と定義される．F 値は再現率と適合率の調和平均である．以降，再現率，適合率，F 値を総して単語認識精度と呼ぶ．

実際の評価の例として，このコーパスに対する単語分割結果が以下のようになったとする．

正解コーパス で も じよず じゃ り ま せ ん
単語分割結果 で も じよ ず じゃり ま せん

この例文の場合，11 文字あり，単語境界か否かの判断をすべき文字間の境界は 10 箇所ある．このうち，単語分割結果が正解コーパスに一致した箇所は 7 箇所なので，全ての文字間での単語境界推定精度は 7/10 である．また，正解箇所では 4/5，誤り箇所では 1/2，交差箇所では 2/2，欠落箇所では 0/1 である．次に単語認識精度は，分割が一致した単語は「で」「も」「り」「ま」の 4 つであるので，$N_{COR} = 4$ となる．正解コーパスの単語数は 8，単語分割結果の単語数は 7 であるので，$N_{REF} = 8$，$N_{SYS} = 7$ である．したがって，再現率は $N_{COR}/N_{REF} = 4/8$，適合率は $N_{COR}/N_{SYS} = 4/7$，F 値は $[\{(4/8)^{-1} + (4/7)^{-1}\}/2]^{-1} = 8/15$ である．

4 実験結果

4.1 単語分割の精度の比較

表 2 は提案手法およびベースラインの，単語境界推定精度と単語認識精度である．表中の「フル」はフルアノテーションされた訓練データを，「部分」は部分的アノテーションされた訓練データを指す．

提案手法とベースラインの単語境界推定精度を比較すると，全体的に誤り箇所で提案手法が **Baseline** より精度が高い．逆に全ての文字境界および，正解箇所では **Baseline** が最も高い精度を示している．**Baseline (H)** の精度は **All（フル）**に近い傾向があるが，欠落誤り箇所では精度が低下した．また単語認識精度は，いずれも **Baseline** が高い．理由として，正解箇所は誤り箇所に比べて数が多いため，正解箇所の精度が高いことの影響が考えられる．

フルアノテーションと部分的アノテーションを比較すると，フルアノテーションは比較的 **Baseline** に近い結果なのに対し，部分的アノテーションは，より誤り箇所の精度が高く，それ以外の箇所の精度が低い．これは，仮定した通り，部分的アノテーションでは再学習前のモデルと同じ素性は学習せず，変更があった単語のみを学習したことが理由だと考えられる．また，変更のあった単語は誤りを含む単語が多かったことが予想できる．

提案手法 **All**，**Cut5**，**Cut5Sub3** を比較すると，**Cut5Sub3** はフルアノテーション，部分的アノテーションともに，**Baseline** からやや精度が下がるのみの結果となった．フルアノテーションでは **All** のみ誤り箇所の精度が高くなり，**Cut5** は **Baseline** より全体的に精度が下がるにとどまった．一方，部分的アノテーションでは **Cut5** がすべての項目で **All** より高い精度を示している．これらから，編集距離を用いたデータの足切りは精度向上に効果があることと，大きく足切りし過ぎると学習の効果が小さくなることが分かる．

訓練データの量を増やすと，**Cut5（部分）**のみ，誤り部分の精度向上が見られた．その他の訓練データや，単語境界についてはデータ量を増やすと精度が低下する，または変化しない結果となった．

4.2 提案手法による分割箇所への影響

表 3，**表 4** は，提案手法によって単語分割がうまくいくようになった例と，うまくいかなくなった例である．下線部が単語分割に成功している箇所，波線部が失敗して

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved

表 2: 単語分割の実験結果．もっとも性能が高いシステムを太字で示す．

訓練データ	文数	単語境界推定精度 [%] 全境界	正解箇所	誤り箇所	交差箇所	欠落箇所	単語認識精度 再現率 [%]	適合率 [%]	F 値
All（フル）	10 万	98.64	99.06	80.69	97.89	**100.00**	97.73	97.34	97.54
	30 万	98.84	99.29	80.31	97.72	98.99	98.09	97.63	97.86
	50 万	98.77	99.16	81.85	**98.54**	97.98	97.98	97.52	97.75
	全て	98.89	99.38	79.15	97.56	**100.00**	98.19	97.67	97.93
Cut5（フル）	10 万	98.91	99.45	76.83	97.89	97.98	98.23	97.81	98.02
	30 万	98.95	99.52	76.83	97.56	96.97	98.36	97.89	98.12
	50 万	98.91	99.47	76.83	97.72	96.97	98.23	97.82	98.03
	全て	98.95	99.51	76.45	97.89	96.97	98.33	97.86	98.09
Cut5Sub3（フル）	全て	98.92	99.46	76.83	97.89	97.98	98.33	97.78	98.05
All（部分）	10 万	94.71	95.15	82.63	91.22	95.96	88.37	93.00	90.63
	30 万	92.11	92.41	83.01	89.92	92.93	82.79	90.18	86.33
	50 万	91.59	91.85	85.71	88.46	94.95	81.38	89.51	85.25
	全て	89.43	89.50	82.63	90.57	91.92	77.25	87.13	81.90
Cut5（部分）	10 万	97.18	97.58	83.40	95.28	96.97	94.18	95.57	94.87
	30 万	95.93	96.21	85.33	94.80	95.96	91.31	94.08	92.67
	50 万	95.28	95.49	**86.10**	94.80	96.97	89.74	93.31	91.49
	全て	94.73	94.86	**86.10**	95.61	95.96	88.34	92.25	90.26
Cut5Sub3（部分）	全て	98.91	99.45	76.83	98.21	95.96	98.24	97.72	97.98
Baseline (KyTea)		**98.98**	**99.54**	76.45	97.89	96.97	**98.37**	**97.87**	**98.12**
Baseline (H)		98.76	99.26	81.08	97.40	92.93	97.89	97.71	97.80

表 3: 単語分割がうまくいくようになった例（文の一部分）

テストデータ	Baseline の出力	提案手法の出力
お <e> ばあ </e> のことを	お ばあの こと を	お ばあ の こと を
なにも <e> がんがえ </e> <e> ら </e> ない。	なに も がん が え ら な い 。	なに も がんがえ ら な い 。
<e> やばり </e> 一人でいいから	やば り 一人 で い い から	やばり 一人 で い い から
泣きたい時 <d/ > 雨の中を歩く	泣 き た い 時雨 の 中 を 歩 く	泣 き た い 時 雨 の 中 を 歩 く

表 4: 単語分割がうまくいかなくなった例（文の一部分）

テストデータ	Baseline の出力	提案手法の出力
最近、仕事 <e> の </e> 原因で	最近 、 仕事 の 原因 で	最近 、 仕事 の 原因で
海水浴をして	海水 浴 を し て	海水 浴 を して
ご飯を作る	ご飯 を 作 る	ご飯 を 作る
近頃、人間の体の	近頃 、 人間の 体 の	近 頃 、 人間 の 体 の

いる箇所である．

うまくいった例では，まず，学習者の誤り箇所で，ベースラインでは本来一単語として意図されていたものが複数に分割されることと，提案手法ではそれらが一単語として分割されるようになることが分かる．**表 3** の 4 行目のように，逆に，助詞が欠落していることで一単語として分割されていたものが，提案手法では正しく別の単語として分割されている例もある．

また，うまく行かなくなった例では，正解箇所で，ベースラインでは正しく別々の単語として分割されていたものが，提案手法では一単語として分割されるようになっている．一方，**表 4** の 4 行目のように，ベースラインでは正しく一単語として分割されていたものが，提案手法では別の単語として分割されている例もある．

全体では，ベースラインでは別々の単語に分割されていた単語が，提案手法では一単語として分割される傾向がある．理由として，提案手法で学習した，変更があった箇所の単語の文字境界を求める際は，文字が大きく削除されている場合でも，変更後の単語の対であるとみなされれば，一単語として扱われる．これにより，単語を長めに切る傾向が学習されたと考えられる．

5 おわりに

本研究では日本語学習者の作文を，誤り訂正に適した形に単語分割することを目的とし，その手法を検討した．提案手法は，文中の誤りを含む箇所の単語分割の精度を高め，正解箇所の精度を低めることが分かった．今後の課題として，訓練データをフルアノテーションした場合と，部分的アノテーションをした場合の，2 つを組み合わせて利用することが考えられる．

また，誤り訂正タスクにおいて，提案手法で再学習した単語分割器を用いて，結果を評価することが挙げられる．

提案手法は誤り検出にも応用できる．誤り検出においては，文中のひらがなの箇所について，文字 n-gram，単語 n-gram などの言語モデルを用いて，確率値の小さい箇所を誤りとみなす手法が提案されている [2]．単語分割においても，確率付きのスコアを出すことで，確率値の小さい箇所を誤り箇所とみなすことができると考えられる．

参考文献

[1] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proc. of IJCNLP*, pp. 147–155, 2011.

[2] 新納浩幸. 平仮名 n-gram による平仮名列の誤り検出とその修正. 情報処理学会論文誌, Vol. 40, No. 6, pp. 2690–2698, 1999.

[3] 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾真. 日本語形態素解析システム JUMAN 使用説明書 version 2.0. NAIST Technical Report, NAIST-IS-TR94025, 1994.

[4] 森信介, Neubig Graham, 坪井祐太. 点予測による単語分割. 情報処理学会論文誌, Vol. 52, No. 10, pp. 2944–2952, 2011.

[5] 工藤拓, 山本薫, 松本裕治. Conditional Random Fields を用いた日本語形態素解析 (解析). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2004, No. 47, pp. 89–96, 2004.

[6] 松本裕治. 「形態素解析システム『茶筌』」. 情報処理, Vol. 41, No. 11, pp. 1208–1214, 2000.

[7] 南保亮太, 乙武北斗, 荒木健治. 文節内の特徴を用いた日本語助詞誤りの自動検出・校正 (語学学習支援・自動校正). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2007, No. 94, pp. 107–112, 2007.

Copyright(C) 2012 The Association for Natural Language Processing.
All Rights Reserved