Support Vector Machines を用いた日本語書き言葉の文境界推定

福岡 健太　　松本 裕治

奈良先端科学技術大学院大学情報科学研究科
{kenta-fu,matsu}@is.naist.jp

1 はじめに

近年，日本語話し言葉における文境界推定の研究が行なわれている [5, 4]．特に，話し言葉においては，「文」の定義がはっきりしていないことから文境界推定が重要である．実際，話し言葉では「200ms 以上の空白など」で区切られる単位が主な解析単位と定義され，句点のないテキストから求められる単位に分割を行なう必要がある．

一方，書き言葉の文境界推定は，情報抽出などの分野で必要不可欠な要素技術である．英語の書き言葉に関して種々の研究が行なわれている [2, 3]．英語の書き言葉では，ピリオドが文境界になる場合と，“Mr.” の様に文境界にならない場合が存在し，この曖昧性解消が必要とされる．日本語の書き言葉に関しては，通常句点が文境界となる．句点は文境界以外に使われることはほとんどなく，英語のピリオド程の曖昧性は存在しない．しかし，Web 上の文章やアンケート結果などの自由記述型のテキストでは，文境界に句点が存在しない場合がある．話し言葉のように韻律の情報が手掛かりとならないため，これらの文境界推定は困難である．

そこで，本研究では，学習モデルとして Support Vector Machines(SVMs) を用いて書き言葉の文境界を自動的に推定する．また，書き言葉特有の改行を素性として利用する．その際，文境界を表さない改行を言語モデルを用いてあらかじめ削除する手法を提案する．

2 書き言葉の文境界

文を記述する場合，通常句点が文末として打たれるが，自由記述型の文章では，文末に句点が打たれない場合が存在する．この場合，句点の代わりとなる記号などが文境界を表すために使われるが，これらは全ての場合において文境界を表すとは限らない．例えば，

例文１ この値段は，買い！だと思います！

の場合の前者の「！」は文境界を表さないが，後者の「！」は文境界を表る．この様に，文境界を表すために使われるものでも，そこが文境界になるかならないかの曖昧性が存在する．また，書き言葉においては，改行が特に重要であると考える．以下に改行を含む 2 つの例を示す．

例文２ 昨日は雨だった（改行）今日は晴れるかな？

例文３ これにこだ（改行）わる必要は無いと思います。

例文２の改行は文境界を表しているが，**例文３**の改行は文境界を表していない．そればかりか，**例文３**の改行は「こだわる」という 1 つの形態素を分割する位置に挿入されている．形態素解析器は，改行までを 1 つの解析単位としているため，このままでは改行を削除しない限り正しく形態素解析すらできない．しかし，全ての改行を削除してしまえば，文境界を表す改行をも削除してしまう．この様に，改行は文境界になるかどうかの曖昧性を含むだけではなく，形態素解析の結果に影響を与えることもあり，どの様に扱うか重要であると考えられる．

さらに，書き言葉には括弧を含む文が存在する．括弧を含む文では，文の中に文が含まれる入れ子になる場合がある．例えば，

例文４ 彼は、「貴方は誰ですか？」と言った。

の場合，彼は、「貴方は誰ですか？」と言った。，で 1 文であるのは当然，貴方は誰ですか？，も 1 文となる．この様に，括弧を含む文章では入れ子構造が存在する．故に，文境界を推定する際，括弧の中は別に解析する必要があると考えられる．

なお，本研究で書き言葉と称しているものは，あらかじめテキストデータとなっているもの全般を示す．よって，くだけた表現を含むテキストデータも書き言葉として扱う．書き言葉では，句点，記号，改行，スペースなどの字種が文境界の始点，終点を同定する手掛かりとなる．よって，文境界は他の字種（ひらがな，カタカナ，漢字）の間に存在しないと仮定する．また，単文・重文・複文の区別は問題設定がより複雑になるため今回は行なわない．

3 文境界にならない改行の削除

書き言葉における改行は，文境界になるかならないかの曖昧性が存在する．そこで，言語モデルを用いて

文境界ではない改行を削除する手法を提案する．文境界にならない改行は，形態素解析の際に利用する文脈情報を分断することが知られている．このような改行をあらかじめ削除することにより，形態素解析の結果が向上し，SVMs に与える形態素情報がより正確なものとなる．

具体的には，改行を削除して形態素解析器である茶筌で解析した場合と改行を削除せずに茶筌で解析した場合の 2 つの出力を比較することで，文境界を表さない改行を削除する．この際，出力される形態素列の変化とパスコストの変化の 2 つを相補的に利用する．

3.1 形態素の変化

改行を含む文章では，本来 1 つの形態素である文字列の間に改行が挟まれて書かれている場合が存在する．この様な例は，特にメールで良く見られる．メールでは，視覚的には 1 行の文字数がある程度そろっていた方が見やすいとされているため，文境界でない箇所に改行が挿入され，またそのとき 1 つの形態素の文字列の間に改行が含まれてしまう場合が多い．そのような改行は当然文境界を表さない．そこで，改行を削除した場合，改行前後の形態素に変化があるようならその改行は削除する．具体的には，以下の手順で改行を削除するかどうかの判定を行なう．

1. 改行を削除する前の改行直前の形態素と改行直後の形態素を茶筌を用いて出力
2. 改行を削除した後，改行を削除する前の 2 つの形態素がどのようになっているかを出力
3. 改行を削除した後，改行直前，直後の形態素の表層表現に変化があれば改行は削除する

これを文章の終りまで繰り返す．

3.2 パスコストの低下

茶筌は，正しい形態素の並びに対して低いパスコストを出力するという特徴がある．日本語話し言葉の文境界推定においても，パスコストを利用する手法が提案されている [5]．この手法では，句点を挿入した場合の出力コストと挿入しない場合の出力コストの比較を行ない，挿入した場合の方がコストが低下すればその箇所を文境界候補としている．本研究では，改行を削除する前のパスコストの合計と，改行を削除した後のパスコストを比較することで文境界を表さない改行を削除する．具体的には，以下の手順で改行を削除するかどうかの判定を行なう．

1. 改行を削除する前の改行前後の 2 行の文字列のパスコストの合計を求める
2. 改行を削除した後のパスコストを求める
3. 改行を削除した後のパスコストが改行削除前のパスコストの合計より低下していれば改行は削除する（ただし，改行直前，もしくは直後の形態素が，記号，括弧，アルファベット，未知語のいずれかであれば改行は削除しない）

これを文章の終りまで繰り返す．手順 3. で，改行直前，もしくは直後の形態素が記号，括弧，アルファベット，未知語のいずれかであれば改行を削除しないのは，これらの場合，改行が文境界になる可能性が高いにもかかわらず，改行を削除した後のパスコストの方が低い値になる場合が多々見られるからである．

3.3 2 手法の相補的な利用

形態素の変化を見るだけでは，文境界となる改行を削除してしまう場合がある．

例文５ ペンを買っていた（改行）なんであれにしたんだろう？

この場合，改行を削除する前の形態素は，改行直前が「た（助動詞）」，改行直後が「なんで（副詞-一般）」になる．一方，改行を削除した後の形態素は「た（助動詞）」と「な（助動詞）」になるため，この改行は削除される．しかし，この改行は文境界を表するものである．この例の様に，改行を削除した場合の形態素の変化を見るだけでは文境界を表す改行を削除してしまう場合がある．

また，パスコストの低下を考慮するだけでも，文境界となる改行を削除してしまう場合が存在する．

例文６ 花は美しい（改行）鳥は美味しそう

この場合，改行を削除する前のパスコストは改行以前の文字列で 6253，改行以後の文字列で 7488 となり合計 13741 となる．一方，改行を削除した後のパスコストは 13470 となり，改行を削除した後ではパスコストが低下するため，この改行は削除される．しかし，実際にはこの改行は文境界を表すものである．

このように，形態素の変化を見るだけでも，あるいはパスコストの比較を行なうだけでも文境界を表す改行を削除してしまう恐れがある．そこで，出力される形態素列の変化とパスコストの変化の 2 つを相補的に利用する手法を提案する．

4 SVMs を用いた文境界推定

文境界を自動的に推定するために SVMs を用いる．理由は，入力次元数に依存しない高い汎化能力を持っているためである．文境界推定の流れは，図 1 に示す．本研究では，文境界推定の問題を，形態素列に対する文をチャンクと考えたチャンキング問題と考える．チャ

図 1: 文境界推定の流れ

ンカーとして SVMs ベースの YamCha [1] を用いる．YamCha では，カーネル関数として多項式カーネルが用いられている．また，ラベリングスキームは，I：チャンク内（文内），O：チャンク外（文外），B：チャンクの先頭（文頭），E：チャンクの末尾（文末），S：その要素だけで 1 つのチャンク（文），の 5 種類のタグを用いる．ラベルは形態素解析された各形態素に付与する．

括弧を含む文章は入れ子構造になっている場合があるため，括弧内は別に解析する．具体的には，括弧で囲まれている文字列は別に取り出し解析を行なう．取り出した部分は，1 つにまとめて形態素情報として SVMs に与える．その際，品詞情報と活用形の情報は括弧内の最後の形態素のものを用いる．

5 実験

5.1 実験方法

実験に用いたデータは Web 及びニュースグループから集めたものを使用し，データの文頭，及び文末は人手でタグを付与した．また，評価の際，括弧内に含まれる文章は文境界推定の対象外とした．今回の実験では，括弧は「」( )【】『』に限定した．用いたデータは 15 種類で，括弧内の文を除くと計 1328 文．これを 3 種類ずつ，5 つに分割し 5 交差検定により評価を行なう．実験では，簡単なルールで文境界を推定したものと，SVMs を用いたものを比較した．ここで，ルールは “。. !? !?” の 6 種類を文末とするというものであり，これらの記号が 2 つ以上続く場合は最後の記号のみを文末と判断する．SVMs を用いる際は，文章に含まれる改行に対して，

(a) 全て残す
(b) 全て削除
(c) 形態素に変化があれば削除
(d) パスコストが低下すれば削除
(e) 形態素に変化がありかつパスコストが低下すれば削除

の 5 通りの結果を示す．

SVMs に与える素性としては，形態素情報（表層表現，品詞情報，活用形）を与える．さらに，改行，及び半角スペースの情報も素性として与えた．形態素情報を与える際，茶筌で形態素解析を行なっているが，茶筌では半角スペースが無視されてしまうため半角スペースの情報が入るよう配慮している．現在位置の形態素のタグを推定するのに，前後 2 形態素ずつの形態素情報を静的素性，前 3 形態素のタグを動的素性として用いた．カーネル関数は 2 次の多項式カーネルを用いた．多値クラス識別には Pairwise 法を用いた．

表 1: 文境界推定精度 (文末のみ)

	再現率	適合率	F 値
ルール	74.85 % (994/1328)	91.87 % (994/1082)	82.49
SVMs (a)	90.81 % (1206/1328)	95.56 % (1206/1262)	93.13
SVMs (b)	85.02 % (1129/1328)	96.33 % (1129/1172)	90.32
SVMs (c)	90.59 % (1203/1328)	95.70 % (1203/1257)	93.08
SVMs (d)	91.11 % (1210/1328)	95.96 % (1210/1261)	93.47
SVMs (e)	91.04 % (1209/1328)	96.03 % (1209/1259)	93.47

5.2 実験結果

文境界推定精度は，以下の式で評価する．なお，ルールを用いた場合，文末（タグ “E,S”）しか同定できない．よって，条件を合わせるため，文末のみを評価対象とする．

$$\text{再現率} = \frac{\text{正しく } E,S \text{ と判定されたラベル数}}{\text{正しい } E,S \text{ のラベル数}}$$

$$\text{適合率} = \frac{\text{正しく } E,S \text{ と判定されたラベル数}}{\text{システムが } E,S \text{ と判定したラベル数}}$$

$$F \text{ 値} = \frac{2 \times \text{再現率} \times \text{適合率}}{\text{再現率} + \text{適合率}}$$

表 1 から，簡単なルールを用いた手法より，SVMs を用いた手法の方が優れていることがわかる．また SVMs を用いた際，(b) の文章に含まれる改行を全て削除した場合，他に比べて再現率が 5 %以上，F 値で約 3 低い値となった．このことから，文章に含まれる改行は文境界推定の際に重要な情報であるといえる．また，言語モデルを用いて不適切な改行の削除を行なった場合，(d) と (e) の手法が F 値で最も高い結果となった．評価の際，“E,S” だけでなく，“B,E,S” を対象とした場合は，形態素の変化とパスコストの低下を共に考慮した場合の F 値が 93.44 で最も高い値となった．

5.3 考察

今回扱ったデータの大部分は Web のデータ（ウェブログ, レビュー記事）だったため，メールに多く見られるような 1 つの形態素の間に改行が入っているという事例はあまり見られなかった．表 2 に，言語モデル

表 2: 削除された改行の数

手法	文中	文境界
(c)	34	17
(d)	195	9
(e)	32	0

表 3: 文境界の分布（SVMs(e)）

文境界の種類	文境界の正解率	タグ I の正解率
。	797/798	10/18
！	89/94	36/38
？	38/42	17/17
.	54/57	7/10
記号-一般	70/82	203/206
アルファベット	13/18	894/899
未知語	44/56	448/452
EOF	0/1	0/0
括弧閉	48/70	179/184
括弧開	8/24	141/147
スペース	0/6	55/58
改行	48/79	231/254
その他	0/1	18960/19029
合計	1209/1328	21181/21312

を用いて削除した文境界を表さない（文中の）改行と，文境界を表す改行の数を示す．実験に用いたデータでは，全ての改行数が 1655 に対して，文中の改行数は 305 であった．手法 (d) では約 3 分の 1 の不適切な改行を削除できているが，文境界を表す改行を 9 つ削除してしまっている．ここで削除された文境界を表す改行の箇所は，SVMs で文境界と特定することが困難になるため，問題となる．一方，手法 (e) では，不適切な改行は 10 分の 1 くらいしか削除できていないが，文境界を表す改行は 1 つも削除していない．

文境界の分布を表 3 に示す．この表での “括弧開”，“スペース”，“改行” は，その直前の形態素（ひらがな，カタカナ，漢字のいずれか）が文末となっている数を指す．“EOF” はテキストの最後の形態素が文末となっている数を指す．他のものは，それ自身が文末となった数を指している．これより，事前に文境界を表さない改行を削除しているにもかかわらず，改行が文境界となる箇所を推定するのが困難な場合が多いことが見てとれる．これは，現在，表題なども 1 文としていることが主な要因として挙げられる．表題の文末に来る改行では，改行の直前と直後が共に名詞となる場合が多く見られた．しかし，文境界を表さない改行でもそのような例は多くみられるため，この区別が難しくなっていると考えられる．よって，表題は別の枠組で特定すべきであると考える．

タグの数は，“B”:1323，“E”:1323，“S”:6 だった．これに対し，“I” は約 21000 もあった（”I” の数は改行の削除方法によって若干異なる）．実験では，6 つのタグ”S” を全く推定することができなかった．これは，タグが”S” となる訓練事例数が極端に少ないことが原因と考えられる．また，“B” や”E” も”I” の 15 分の 1 以下しかなく，今後，このバランスを考慮した学習モデルを考える必要がある．また，データ数を約 3 分の 2（858 文）にして実験した場合，手法 (e) で F 値が 90.53 と最も高く，1328 文を用いた場合の F 値が 93.47 だったことから，データ数を増やせば精度が向上する余地はあると考えられる．

6 おわりに

本研究では，SVMs を用いて日本語書き言葉の文境界を自動的に推定する手法を提案した．評価実験を行ない，書き言葉における改行の重要性を示し，さらに言語モデルを用いることで，文境界を表さない不適切な改行を SVMs に素性として与える前に削除する手法を提案し，その有効性を示した．

今後の課題としては，今回の実験では事例数が少なかったため，データ量を増やして実験する必要がある．また，テキストの種類には，箇条書きの文や話し言葉の様な文，メールなどいくつかの種類が存在する．これらの文境界を 1 つのモデルで推定するには限界があると考えられるため，異なる種類の文境界を推定する際には，異なるモデルを用意し文境界を推定する必要がある．その際，どのモデルを用いるべきかは，テキスト分類の技術を用いて最適なモデルを選択することが考えられる．

参考文献

[1] Taku Kudo and Yuji Matsumoto. Chunking with Support Vector Machines. In *2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001), Proceedings of the Conference*, pp. 192–199, 2001.

[2] Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Fifth Conference on Applied Natural Language Processing*, pp. 16–19, 1997.

[3] Mark Stevenson and Robert Gaizauskas. Experiments on sentence boundary detection. In *Association for Computational Linguistics: 6th Applied Natural Language Processing Conference (ANLP-2000)*, pp. 84–89, 2000.

[4] 下岡和也, 内元清貴, 河原達也, 井佐原均. 話し言葉の係り受け解析と文境界推定の相互作用による高精度化. 話し言葉の科学と工学ワークショップ, pp. 119–126, 2004.

[5] 田島幸恵, 難波英嗣, 奥村学. 形態素解析器を利用した講演書き起こしの文境界検出について. 情報科学技術フォーラム (FIT 2003), 2003.