# マイクロブログに対する 文境界推定および係り受け解析

難波　悟史　　門内　健太　　但馬 康宏　　菊井 玄一郎
岡山県立大学大学院　情報系工学研究科
{cd25036z, cb24028x, tajima, kikui}@cse.oka-pu.ac.jp

## 1. はじめに

Twitter 等のマイクロブログに投稿されている膨大なテキストを分析して実世界の動向や人々の嗜好を探ることが活発に行われている．テキストを深く分析するためには係り受け解析が不可欠であるが，現状においてマイクロブログに対する係り受け解析の精度は必ずしも十分なものではない．

係り受け解析の精度低下の要因として大きく二つの問題が考えらえる．一つ目は，係り受け解析の各種パラメータが主に新聞コーパス等で調整されているため，言語的な性質の異なるマイクロブログに適応した場合精度が低下してしまうというモデルミスマッチの問題である．二つ目は，係り受け解析の前段で行う文境界の推定が不正確という問題である．文境界の推定は，新聞などの統制されたテキストであれば句点や疑問符など特定の記号を手掛かりに高精度に行える．しかし，マイクロブログではこのような規則に従わない例が多数存在するため，しばしば誤ってしまう．文境界推定を誤ると係り受け関係にある文節が別々の文に分かれたり，本来，別々の文に存在する文節が同一文に含まれたりするため，係り受けの精度を低下させる可能性が高い．文境界推定についてはマイクロブログと文体的に近い「話し言葉」の分野において，機械学習を用いた文分割の研究が行われているが（[1],[2]など），係り受け解析への影響を含めた効果は未評価である．

そこで，本稿では文境界推定，および，係り受け解析について，既存の統計的な手法を前提に素性の調整やマイクロブログを対象としたパラメータ学習により，係り受け解析の精度が程度改善できるかについて検討する．

## 2. 手法概要

### 2.1. 解析処理の概要

解析処理全体の流れを図 1 に示す．図において網掛けした所が本研究においてチューニングを試みた部分であり，３章以降で説明する．

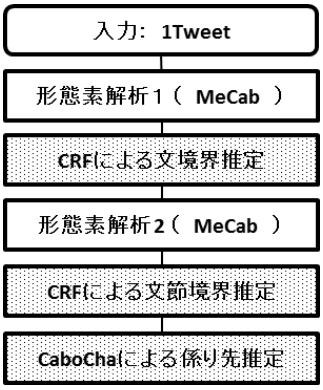


図 1 処理の流れ

最初の形態素解析 1 は文境界推定の入力を生成するための処理である．通常，形態素解析は文境界推定の後に行われるが，本研究では文境界推定の前に tweet 全体を一つの「文」として実行する．形態素解析には MeCab[3]を用いる．ただし，ユーザ辞書に渡邊ら[4]が作成した顔文字辞書中の出現回数が 100 回以上の顔文字 4199 種を登録した．さらに tweet 特有の非正規的な表現を扱うために，MeCab 適用の前に表 1 に示す文字列書き換えを行う．

形態素解析 1 に引き続いて CRF[5]による文境界

表 1　形態素解析 1 の前処理

| 変換前 | 変換後 |
|---|---|
| 全角スペースおよび、半角スペース | 半角スペース |
| 日本語に連続する1文字以上の「w」 | (笑) |
| 「・・・」や「…」「．．．」が続くとき | … |

推定を行う．その後，形態素解析 1 の結果を無視して，再度，形態素解析を行う．この形態素解析の際には全ての文字を全角に統一する処理を行う．これはテキスト中の半角文字による形態素解析結果の誤りにより，文節境界推定精度が低下することが予備実験で確認できたためである．次に CRF による文節境界の推定を行ったあと，係り先の推定を行う．

## 3. 文境界推定

系列ラベリング問題として文境界推定を行う．モデルとして CRF を用いる．

マイクロブログは口語的な表現や崩れた表現，断片的な表現など「用言文節の終止形で終わる」という原則に逸脱するものが多く，何をもって文とするかは難しい．話し言葉の文（節）境界を参考に，アノテータの判断によった．このため，例えば，名詞や接続詞のみの「文」の存在も許す．なお，Twitter の特有の表現であるハッシュタグ，RT，@(リプライ)や URL については，独立した文として扱うこととした．

### 3.1. 文境界判別ラベルの定義

通常，文境界推定は形態素解析に先立って文字列処理として行われるため，系列ラベリングにおける系列の要素は「文字」とするのが自然である．しかしながら，本研究ではこれとは別に系列の要素として「形態素」(形態素解析 1 の結果) の利用も試みた．いずれの場合も文末の要素には BR ラベル，文末以外の要素には TEXT のラベルを付与する．

### 3.2. 素性

#### 3.3.1 系列の要素が形態素単位の場合

CRF で扱う素性は MeCab の出力(表層形，品詞，品詞細分類 I )，表層形の文字種，形態素の直後のスペースの有無である．文字種は，ひらがな，カタカナ，漢字，大文字英字，小文字英字，英字キャピタライズ(大文字英字 1 文字+小文字英字列)，数字，記号，その他の 9 種類に分類した．学習で使用する形態素数は実験的に求めた結果，前後 2 文字を学習する．図 2 に素性例を示す．

| 表層形 | 主品詞 | 品詞細分類 | 文字種 | 直後のスペースの有無 | ラベル |
|---|---|---|---|---|---|
| なんか | フィラー | * | ひらがな | 無 | TEXT |
| 最近 | 名詞 | 副詞可能 | 漢字 | 無 | TEXT |
| めちゃくちゃ | 名詞 | 形容動詞語幹 | ひらがな | 無 | TEXT |
| ハード | 名詞 | 一般 | カタカナ | 無 | TEXT |
| 。 | 記号 | 句点 | 記号 | 無 | BR |

図 2　文境界推定の素性例(系列要素が形態素)

| 文字 | 表層系 | 主品詞 | 品詞細分類 | 文字種 | ラベル |
|---|---|---|---|---|---|
| ハ | B-ハード | B-名詞 | B-名詞 | カタカナ | TEXT |
| ー | I-ハード | I-名詞 | I-名詞 | 記号 | TEXT |
| ド | E-ハード | E-名詞 | E-名詞 | カタカナ | TEXT |
| 。 | S-。 | S-記号 | S-句点 | 記号 | BR |

図 3 文境界推定の素性例(系列要素が文字)

#### 3.3.2 系列の要素が文字の場合

CRF で扱う素性は文字と MeCab の出力(表層形，品詞，品詞細分類 I )，表層形の文字種である．表層形，主品詞，品詞細分類にはその文字の単語中の位置に応じて Start-End 法(以下 SE 法)に基づくチャンクタグを付与する[6]．SE 法では形態素の先頭文字に対し B タグ，末尾に E タグ，内部に I タグ，1 文字からなる形態素に対しては S タグが付与される．文字種は，ひらがな，カタカナ，漢字，大文字英字，小文字英字，数字，記号，スペースの 8 種類に分類した．学習で使用する形態素数は実験的に求めた結果，前後 5 文字を学習する．図 3 に素性例を示す．

## 4. 文節境界推定

### 4.1. 文節境界判別ラベルの定義

本研究では CRF を用いた系列ラベリングにより文節境界を推定する．系列要素の単位としては文字と形態素の双方を試みる．いずれの場合もラベルは文節境界を表す BS とそれ以外の部分を表す TEXT である．

### 4.2. 素性

#### 4.2.1 系列の要素が形態素単位の場合

CRF で扱う素性は単語と品詞と文字種である．形態素解析には MeCab を使用した．文字種は HIRA(ひらがな)，KATA(カタカナ)，KAN(漢字)，NUM(数字)，ALPL(半角英字)，ALPU(全角英字)，SPACE(空白)，SYMBOL(これら以外の文字)の 8 種類に分類した．CRF の学習には，前後 2 形態素

を用いる．図 4 に素性例を示す．

#### 4.2.2　系列の要素が文字単位の場合

CRF で扱う素性は文字と単語と品詞と文字種である．形態素解析には MeCab を使用した．単語および品詞には，SE 法に基づくチャンクタグを付与する．文字種の種類は形態素単位と同様とした. CRF の学習には，前後 2 列文字を用いる．図 5 に素性例を示す．

## 5.　係り先推定

本研究では係り先の推定に CaboCha[7]の係り先推定部分を用いる．

### 5.1.　CaboCha の入出力

CaboCha には解析レイヤという概念があり形態素解析，文節の区切り情報を与えたデータ(CaboCha の文節区切りレイヤ)を入力として係り先推定のみを行うことができる. 本研究ではこの機能を利用して，4 章の文節境界推定結果に対し係り先を推定させた．なお，文節境界推定の際に文字単位の系列ラベリングを行うと形態素境界でない位置に文節境界が推定されることがある，これに対処するため，MeCab の機能を用いて文節境界が形態素の切れ目となるように再度形態素解析を行った．

### 5.2.　モデル学習

CaboCha では，正解データを用意すれば係り受け解析，文節区切りの各モデルのパラメータを学習させることができる．CaboCha 添付のモデルは新聞記事を対象に学習されているため，Tweet を対象に解析を行う場合には精度が低下するのではないかと考えられる．そこで，本研究では Tweet を対象に図 6 に示すような形で学習データを構築し，新しいモデルを構築した．

## 6.　実験

### 6.1.　利用データ

実験には 2012 年 2 月～2012 年 7 月の間に Twitter に投稿された Tweet からランダムで集めた 7000Tweet に人手で文境界タグ，文節境界タグを振ったデータを使用した．7000Tweet の内，6500Tweet を学習に使用し，500Tweet を評価に使用し．交差検定(14 分割)を行った．

| 単語 | 品詞 | 文字種 | ラベル |
|---|---|---|---|
| なんか | フィラー-* | HIRA | BS |
| 最近 | 名詞-副詞可能 | KAN | BS |
| めちゃくちゃ | 名詞-形容動詞語幹 | HIRA | BS |
| ハード | 名詞-一般 | OTHER | TEXT |
| 。 | 記号-句点 | SYMBOL | TEXT |

図 4　文節境界推定の素性例(系列要素が形態素)

| 文字 | 単語 | 品詞 | 文字種 | ラベル |
|---|---|---|---|---|
| な | B-なんか | B-フィラー-* | HIRA | TEXT |
| ん | I-なんか | I-フィラー-* | HIRA | TEXT |
| か | E-なんか | E-フィラー-* | HIRA | BS |
| 最 | B-最近 | B-名詞-副詞可能 | KAN | TEXT |
| 近 | E-最近 | E-名詞-副詞可能 | KAN | BS |

図 5　文節境界推定の素性例（系列要素が文字）

### 6.2.　評価方法

#### 6.2.1 文境界推定の評価尺度

評価尺度として，以下の式で与えられる適合率，再現率，およびこれらから計算される F 値を用いる．

$$\text{適合率} = \frac{\text{出力された文末の正解数}}{\text{出力された文末の数}}$$

$$\text{再現率} = \frac{\text{出力された文末の正解数}}{\text{評価データ内に存在する文末の数}}$$

ただし，評価を行う文末は Tweet の途中に出現する文末のみとし，Tweet の最後に出現する文末は対象としない．

ベースラインとして，「。」「 .」「!」「 ?」「♪」「・」「…」「スペース」を文境界とする方法と比較する，

#### 6.2.2 文節境界推定の評価尺度

評価尺度として，以下の式で与えられる適合率，再現率，およびこれから計算される F 値を用いる．

$$\text{適合率} = \frac{\text{出力された文節の正解数}}{\text{出力された文節の数}}$$

$$\text{再現率} = \frac{\text{出力された文節の正解数}}{\text{評価データ内に存在する文節の数}}$$

#### 6.2.3 係り先推定の評価尺度

係り先推定の評価は，出力の文節と正解の文節で同じ文字位置で始まり同じ文字位置で終わるものに対して，係り先の文節が一致するかを評価する．これは再現率に相当すると考えられる．

$$\text{正解率} = \frac{\text{出力と正解の文節の係り先が一致した数}}{\text{正解の文節の数} - \text{正解の最終文節の数}}$$

たたし，係り元が文末となる文節は評価の対象外と

する.

### 6.3. 結果

#### 6.3.1 文末境界推定

推定結果を表 2 に示す．形態素単位での CRF による推定精度が最も高かった．間違えた箇所は，読点などの一般的には文末境界にならない記号が句点のように扱われた箇所や文境界にもかかわらず記号やスペース等の手がかりがない箇所であった，後者は数も多かった.

#### 6.3.2 文節境界推定の結果

正しい文境界を与えた場合の文節境界推定の結果を表 3 に示す. 文字単位での学習の方が良い結果となった．間違えた箇所は，固有表現中の一般名詞や助詞で区切ってしまう例や名詞が連続している例が目立った. これらの対策としては固有表現抽出や助詞補完などが必要だと考えられる.

#### 6.3.3 係り先推定の結果

文境界推定, 文節境界推定を変えた場合の係り先推定の評価結果を表 4 に示す. 表中の記号の意味は次の通りである.

文境界推定：

・CRF-MBM：形態素単位の CRF（マイクロブログモデル),

・baseline：記号を手掛かりにした文境界推定

文節境界推定：

・CRF-MBM：文字単位の CRF（マイクロブログモデル)

・Cab-ORG ：CaboCha（添付モデル）

係り先推定：

・Cab-MBM：CaboCha(マイクロブログモデル),

・Cab-ORG：CaboCha（添付モデル）

従来の手法・パラメータの組み合わせ（１行目）に比べて今回パラメータを調整した処理の組み合わせの方が約 10 ポイント(0.10)の精度向上があった．特に文節境界推定の影響が大きいことが分かる. 文境界, 文節境界が全て正解の場合の係り先正解率を調べると 0.88 となり，これらの処理の精度が重要だと言える．また，文境界推定結果でも精度の向上が見られる．文境界が正しく推定できることで，文を飛び越えるような係り関係がなくなったのではないかと考えられる.

表 2 文末境界推定の結果

| | 適合率 | 再現率 | F値 |
|---|---|---|---|
| 形態素単位での学習 | 0.913 | 0.832 | 0.870 |
| 文字単位での学習 | 0.906 | 0.800 | 0.849 |
| 記号区切り | 0.645 | 0.787 | 0.708 |

表 3 文節境界推定結果の結果

| | 適合率 | 再現率 | F値 |
|---|---|---|---|
| 形態素単位での学習 | 93.25 | 91.34 | 92.28 |
| 文字単位での学習 | 93.74 | 92.21 | 92.97 |
| CaboChaによる文節区切り | 83.42 | 84.65 | 84.03 |

表 4　係り先推定の結果

| 文境界推定 | 文節境界推定 | 係り先推定 | 正解率 |
|---|---|---|---|
| baseline | Cab-ORG | Cab-ORG | 0.617 |
| CRF-MBM | Cab-ORG | Cab-ORG | 0.643 |
| CRF-MBM | CRF-MBM | Cab-ORG | 0.711 |
| CRF-MBM | CRF-MBM | Cab-MBM | 0.716 |

## 7. おわりに

本稿では，マイクロブログコーパスを用いた，文境界推定・文節境界推定・係り先推定の評価を行った．係り先の精度は 0.716 という結果であった. 今後は, 係り先の推定精度の向上を目指すほか, 各実験においても，更なる精度向上を目指す.

謝辞：本研究の一部は JSPS 科研費 24500296 の助成を受けた

### 参考文献

[1] 祖父江翔，山本けい子，田村哲嗣，速水悟，“音声人結果の文末境界推定における識別モデルの評価”言語処理学会第 15 年次大会, pp582-585 , Mar.2009

[2] 下岡 和也，内元 清貴，河原 達也，井佐原 均，“日本語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”，in 自然言語処，Vol.12, No. 3，P3-17

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

[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, pp.866-869, Mar.2013.

[5] http://crfpp.googlecode.com/svn/trunk/doc/index.html?source=navbar

[6] 中野桂吾，平井有三，“日本語固有表現抽出における文節情報の利用，” in 情報処理学会論文誌, Vol.45, No.3，Mar.2004

[7] https://code.google.com/p/cabocha/