分割候補の探索をともなうニューラル日本語単語分割

Neural Japanese Word Segmentation with Segmentation Candidate Search

山口 修平　三輪 誠　佐々木 裕
Shuhei Yamaguchi　Makoto Miwa　Yutaka Sasaki
豊田工業大学大学院工学研究科
Graduate School of Engineering, Toyota Technological Institute
{sd17440, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

1 はじめに

日本語の自然言語処理では，しばしば単語分割が最初に行われる．単語分割の精度が後に行われる処理に影響するため，単語分割を正確に行うことは重要である．近年ニューラルネットワークを用いた単語分割 (ニューラル単語分割) が盛んに研究されており，特に Long Short-Term Memory セルを用いたリカレントニューラルネットワーク (以降，LSTM) を用いた手法 [1] が高い精度を出している．LSTM は系列データを扱うことができるニューラルネットワークの 1 つで，LSTM を用いたニューラル単語分割では，入力文字列をすべて考慮し，各文字に対して単語の境界を表すラベルを順番に決定することで単語分割を行う．しかし，この手法は各文字の境界決定の際にこれまでの予測によってすでに分割された単語を考慮してはいないことや，単語の境界を表すラベルを文字ごとにまとめて扱っているなどの問題が存在する．一方，中国語単語分割において，既に決定した単語の情報を用いて境界決定を行う単語分割 [2] が高い精度を出しているが，日本語単語分割では行われていない．

本研究では，日本語単語分割において先に決まった単語を考慮する単語分割を目的として，探索中に単語単位の情報を用いるニューラル単語分割を提案する．

2 関連研究

2.1 LSTM を用いたニューラル単語分割

日本語の単語分割において，LSTM を用いた単語分割手法が提案されている [1]．LSTM を用いた単語分割の手法では，文字列 $c_1, c_2, \ldots, c_n$ に対して，単語の境界を表すラベル列 $t_1, t_2, \ldots, t_n$ を予測する系列ラベリング問題として単語分割を行う．ニューラルネットワークの全体像を図 1 に記す．具体的には，まず文字列をそれぞれ対応する実数値ベクトルに変換し，ベクトル列 $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ を作る．次に，LSTM を用いることで，入力文の文字すべてを考慮した隠れ層 $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n$ を作り，各 $\mathbf{h}_t$ で多値分類を行うことでラベル付けを行う．単語の境界を表すラベルは，単語の始まり，単語の途中，単語の終わり，1 文字からなる単語を表す B (begin), M (middle), E (end), S (single) を用いる．ラベルの予測分布 $\mathbf{y}_t$，損失関数は次式のように設定する．

$$\mathbf{y}_i = \mathrm{softmax}(\mathbf{W}\mathbf{h}_\mathrm{t} + \mathbf{b}) \quad (1)$$

$$\mathrm{loss} = \sum_{i=0}^{n} -\mathbf{t}_i \log \mathbf{y}_i \quad (2)$$

$\mathbf{t}_i$ は文字 c_i の正解ラベル分布，$\mathbf{W}, \mathbf{b}$ はそれぞれ重みとバイアス項である．この手法では，文字単位での B, M, E, S ラベルを分類問題として選択しているため，ラベル間に共通した境界の情報を別に扱っているという問題がある．例えば，B と S であれば，文字の左が単語の境界であるという点は同じである．

さらに，辞書を用いることで，単語単位での情報を付与し，精度向上している研究 [3] もあるが，入力文字列時点での単語単位の情報であり，境界決定によって確定した単語のみを考慮できておらず，関係のない単語情報を参照してしまう可能性がある．

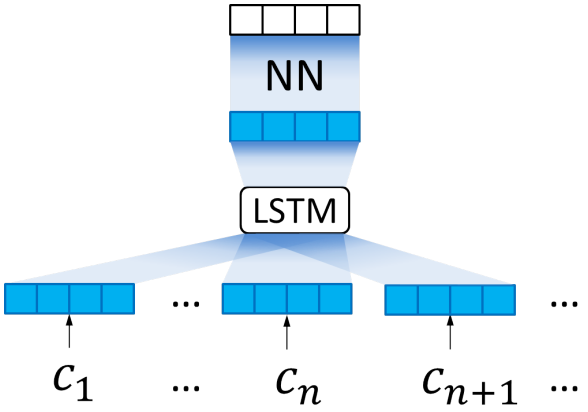

図 1: LSTM を用いたニューラル単語分割

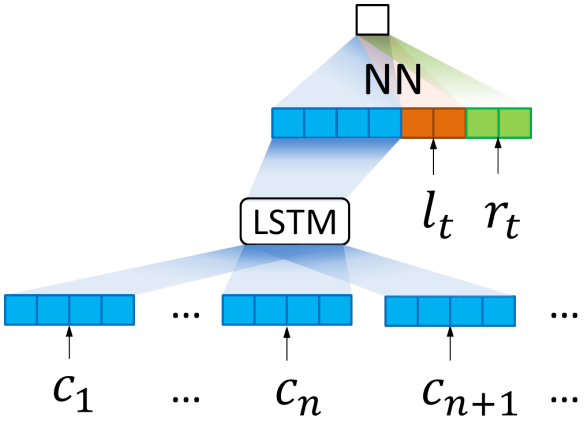

図 2: 文字スコアを計算するモデル

2.2 構造学習を用いたニューラル単語分割

中国語単語分割において，入力文の文字だけでなく境界決定によって生じた単語の情報を考慮した単語分割が提案されている．この手法では，前節のように文字毎に B, M, E, S の分類問題を解くのではなく，各文字に対して新しい単語の始まりかそうでないかを表す，SEP (separate) と APP (append) を用いて単語分割の候補を作成し，その中から一番単語分割として正しい文を選ぶことにより単語分割を決定している．単語分割として正しい文は SEP と APP の選択と，それによって生じた単語列から計算される文スコア S_s によって考えられる．文スコアによって，境界決定で生じた単語単位の情報を考慮した単語分割を決定する．損失関数は次式のように設定する．

$$\text{loss} = \max(0, S_s^{\text{p}} - S_s^{\text{g}}) \tag{3}$$

S_s^p, S_s^g はそれぞれ予測の文スコアと正解の文スコアであり，loss を最小化することで，正解の文スコアが大きくなるように学習する．

3 提案手法

本論文では，境界決定によって生じた単語の情報を考慮し，新しいラベルの表現方法を採用したニューラル単語分割の手法を提案する．提案手法では，文頭から単語の境界を探索し，文の境界決定の仕方毎に単語分割の正しさを表す文スコアを計算し，正解の単語分割が一番高い文スコアを持つように学習する．文スコアは文字スコアと単語スコアの合計から計算される．

3.1 ラベル付け

従来の B, M, E, S ラベル付けでは共通した境界に関する情報を別に扱っているため、提案手法では境界とそうでないラベル S, A の 2 つのみを用いることで，文字の単位を保ちつつ，境界に関して共通に扱う新たな表現方法を提案する．各文字に対して S, A ラベルを両端に考えることで B, M, E, S ラベルを表現する．対応を表 1 に記す．

3.2 文字スコア

文字スコアは各文字に対して，文字の両側の境界に関する情報を付与し，スコアを計算する．文字スコアを計算するネットワークのモデルを図 2 に記す．入力文字から双方向 LSTM の出力である隠れ層 $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n$ を得るまでは従来の LSTM を用いたニューラル単語分割と同じである．隠れ層 $\mathbf{h}_t$ に，両側の境界を表すベクトル $\mathbf{l}_t$, $\mathbf{r}_t$ を結合したものをニューラルネットワークに通すことで c_t がそれらの境界を持つ時の文字スコアが計算される．境界を表すベクトルは，左側・右側それぞれについて境界を持つ (S) か持たないか (A) に対応する 2 つの境界ベクトルを，表 1 の組み合わせで結合し生成する．そのため，各 c_t について，それぞれの境界情報を含んだベクトルが作られ，それぞれの

表 1: B,M,E,S ラベルの新しい表現方法

位置	従来	提案
単語の始まり	B	S, A
単語の途中	M	A, A
単語の終わり	E	A, S
1 文字の単語	S	S, S

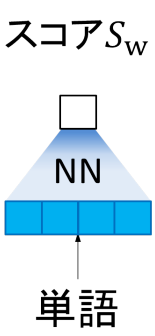

図 3: 単語スコアを計算するモデル

ラベルについての文字スコアが計算される．この文字スコアは，入力文中の文字すべてを考慮し，各文字に対してどのように境界決定をするかを表す値であり，探索を行う前にすべて以下の式で計算される．

$$S_c(c_t, l_t r_t) = \mathbf{w}_c \tanh(\mathbf{W}[\mathbf{h}_t; \mathbf{l}_t; \mathbf{r}_t] + \mathbf{b}) + \mathrm{h}_c \quad (4)$$

$[\mathbf{a}; \mathbf{b}]$ はベクトル $\mathbf{a}, \mathbf{b}$ の結合，$\mathbf{W}, \mathbf{b}$ はそれぞれ重みとバイアス項，$\mathbf{w}_c, \mathrm{h}_t$ はそれぞれベクトルと閾値，$\mathbf{l}_t, \mathbf{r}_t$ は c_t の両側の境界ベクトルを表す．

3.3 単語スコア

単語スコアは探索中の境界決定によって生じた単語に対して作られるスコアである．単語スコアを計算するネットワークのモデルを図 3 に記す．探索中に決定した単語に対して，対応したベクトルに埋め込み，それをニューラルネットワークに通して得られたスカラー値を単語スコアとする．単語スコアは文字スコアとは異なり，探索中に以下の式で計算される．

$$S_w = \mathbf{w}_w \mathbf{e}_w + \mathrm{h}_w \quad (5)$$

$\mathbf{w}_w, \mathrm{h}_w$ はそれぞれベクトルと閾値，$\mathbf{e}_w$ は単語 w に対応した単語ベクトルを表す．

3.4 文スコア

文スコアは，文字スコアと単語スコアの合計と定義し，以下の式で計算される．

$$S_s = \sum_t S_c(c_t, l_t r_t) + \sum S_w(w) \quad (6)$$

S_s, S_c, S_w はそれぞれ，文スコア，文字スコア，単語スコアであり，$l_t r_t$ は文字 c_t の両端の境界ラベルを表す．

Algorithm 1: 探索によるニューラル単語分割

入力：文字列：$(c_1, \ldots, c_n)$，ラベル列：$(l_1 r_t^g, \ldots, l_n r_n^g)$
出力：ラベル列：$(l_1 r_t, \ldots, l_n r_n)$
初期化：候補群 $\leftarrow (S_s = 0, l_0 r_0 = ss)$
文字スコア：$4 \times n$ 個の計算
for i **in** $1, \ldots, n$ **do**
  $beam \leftarrow []$
  for 候補 **in** 候補群 **do**
    if 候補 $.r_{i-1} = a$ **then**
      ADDITEM($beam$, 候補 $.S_s + S_c(c_i, sa)$)
      ADDITEM($beam$, 候補 $.S_s + S_c(c_i, ss) + S_w$)
    else
      ADDITEM($beam$, 候補 $.S_s + S_c(c_i, aa)$)
      ADDITEM($beam$, 候補 $.S_s + S_c(c_i, as) + S_w$)
  候補群 $\leftarrow$TOP-B($beam, B$)
  if $(l_1 r_1^g, \ldots, l_i r_i^g) \notin$ 候補群 **then**
    $S_s^p \leftarrow$best(候補群)
    UPDATE(S_s^g, S_s)
    return
$S_s^p \leftarrow$best(候補群)
if $S_s^g \neq S_s^p$ **then**
  UPDATE(S_s^g, S_s)
return

3.5 単語分割の予測

単語の境界決定は文頭から文スコアを計算し，一番高い文スコアを予測とする．文スコアは 1 つ前のステップの文スコアに文字スコアと，加えた文字スコアが右に境界を持つもので単語が確定すれば，その単語スコアを加えることによって計算する．文字スコアは前のステップの文字スコアが右に境界を持つものと同じものを左に境界を持つもののみが選択される．そのため，1 つの単語分割の候補につき，1 ステップで 2 つの単語分割の候補が増える．探索では，各ステップでスコアの高いもののみを残し，次のステップの文スコアを計算するビームサーチを行う．入力文字列すべてを探索した時，単語分割の候補の中で一番高い文スコアを持っている候補を単語分割の予測とする．

3.6 学習

提案手法のニューラル単語分割のモデルを Algorithm1 に記されたもので学習する．探索は単語分割の予測時と同じように行い，学習は入力文字列のすべてを探索した場合と，候補内に正解の単語分割が無くなった場合に行う [4]．これにより，探索が大きく外れて進んでしまったとき，文全体で更新するのではなく，間違った単語分割の箇所に焦点を当てて更新することができる．損失関数はどのタイミングで学習した場合でも，次式のように設定する．

$$\mathrm{loss} = \max(0, S_{\mathrm{s}}^{\mathrm{p}} - S_{\mathrm{s}}^{\mathrm{g}} + \sum_{\mathrm{t}} \eta \cdot \delta(\mathrm{l_t r_t^p}, \mathrm{l_t r_t^g})) \quad (7)$$

$\delta(a, b)$ は，$a = b$ の時 0，$a \neq b$ の時 1，η はマージンのパラメータである．loss を最小化することで，正解の文スコアが大きくなるように学習する．マージン項を設けることで，単語の境界選択が間違っているものが多いほど，文スコアが正解と離れて学習されるようにする．

4 実験設定

本研究では京都大学テキストコーパスを用いて，単語分割の学習，評価を行った．学習には 35,900 文，開発に 500 文，テストに 2,000 文用いた．単語スコアの計算に用いた単語ベクトルは日本語版 Wikipedia で学習した単語ベクトル 938,947 単語 [5] を用いて初期化した．学習には Adam を用いて最適化を行った．実装には Python 3.5.2, Chainer 3.0.0 を用いた．ニューラルネットワークの詳細なパラメータは表 2 に記す．dropout は LSTM 層に入力する前に行った．比較手法として，双方向 LSTM を用いた単語分割の実装を行ったが，共通している部分のパラメータは同じ設定にした．

5 結果と考察

京都大学テキストコーパスを用いて，単語分割の F 値で評価した結果を表 3 に示した．提案手法の比較として，ルールベースの形態素解析器である JUMAN [6] と LSTM を用いて単語分割を行った手法で同様の評価を行った．表 3 から，提案手法の先に単語の境界として決定した単語の情報を用いる手法が一番高い精度を記録した．

表 2: ハイパーパラメータの設定

パラメータ	値
文字ベクトルの次元	100
隠れ層の次元	150
境界ベクトルの次元	50
単語ベクトルの次元	100
学習率	0.001
dropout	0.2
η	0.2

また，提案手法から単語スコアを計算しない場合の評価も行ったところ，単語スコアがある場合と比べて大きく精度が下がっているため，先に決まった単語の情報が単語分割に重要な影響を与えるとわかる．

6 おわりに

本研究では，日本語単語分割における予測により先に決まった単語を考慮した単語分割を目的として，探索中に単語単位の情報を用いたニューラル単語分割を提案した．結果として，単語単位の情報を用いることで，単語分割の精度向上が確認できた．

今後は辞書から得られる情報を単語単位の情報に利用するとともに，品詞推定，読み推定など形態素解析への拡張をしていきたい．

参考文献

[1] 北川善彬ら. 深層ニューラルネットワークを利用した日本語単語分割. In *NLP2016*, pages 933–936, 2016.

[2] Meishan Zhang et al. Transition-based neural word segmentation. In *ACL2016*, pages 421–431, 2016.

[3] 池田大志ら. 辞書情報と単語分散表現を組み込んだリカレントニューラルネットワークによる日本語単語分割. In *NLP2017*, pages 879–882, 2017.

[4] Michael et al. Collins. Incremental parsing with the perceptron algorithm. In *ACL2004*, 2004.

[5] 鈴木正敏ら. Wikipedia 記事に対する拡張固有表現ラベルの多重付与. In *NLP2016*, pages 797–800, 2016.

[6] Sadao Kurohashi and Makoto Nagao. A method of case structure analysis for japanese sentences based on examples in case frame dictionary. *IEICE*, 77(2):227–239, 1994.

表 3: 既存手法と提案手法の F 値での比較 [%]

モデル	F 値
JUMAN	97.68
LSTM を用いた単語分割	98.08
提案手法	**98.43**
提案手法 (S_w 無)	97.85