E-011

長さ制限のない未知語形態素候補の自動生成

Automatic Generation of Unknown Morpheme Candidates without Length Restriction

後藤 功雄 †　　田中 英輝 †
Isao Goto　　Hideki Tanaka

1 はじめに

日本語の形態素解析は，日本語文を形態素へ分割し，各形態素の文法的属性（品詞や活用型・活用形など）を決定する処理である．一般にこの処理は，形態素の辞書を用いて行う．解析したい日本語文（入力文）中で，形態素辞書の見出しと一致する全ての表現を形態素の候補とし，この候補をグラフ構造で表現して，このグラフ構造中で最適な形態素の並びと文法的属性の並びを決定する．ただし，新しい語（特に固有名詞や専門用語など）が出現するため，あらかじめ全ての形態素を辞書に登録しておくことは，解析対象を限定しない限り困難である．

そこで形態素解析では，辞書に登録されていない語である未知語の形態素を解析する未知語処理が必要である．未知語処理では，辞書に登録されていない表現で，形態素の可能性があるものを入力文中から推定し，それを形態素の候補としてグラフ構造に追加する．この未知語処理で，正しい形態素が形態素候補としてグラフ構造に追加されなければ，正しく解析することはできないため，正しい未知語形態素を生成して形態素候補とすることが重要である．

なお，未知語の解析手法として，品詞を考慮せずに形態素への分割のみを先に決定する手法がある [1] が，我々は，形態素の決定は品詞などの文法的属性も同時に考慮して行った方がより多くの情報を考慮できるので有利だと考えている．

本稿では，未知語の形態素候補を自動生成する手法を提案する．文字単位で計算した条件付き確率に基づいて候補を生成することで，長さの制限なく未知語の形態素候補を生成できる．さらに，任意の文字列表現を候補として用いる場合より候補数を削減できる．また，確率を計算することで極端に不適切な候補を排除できる．

以下，2 章で従来の未知語候補生成の問題について述べ，3 章で提案手法について説明し，4 章で評価実験と考察について述べ，5 章でまとめる．

2 従来の未知語候補生成の問題

2.1 長さヒューリスティクスの問題

任意の文字列表現の全てを形態素候補として扱うと，候補数のオーダが n^2（入力文の文字数を n とする）と多くなってしまう．候補数を削減するためにヒューリスティクスにより形態素候補の文字列長を制限する手法がある [2]．5 文字以下の全ての文字列と連続するカタカナの文字列を未知語の候補としている．しかし，文字数を制限すると，それ以上長い未知語を解析できないという問題がある．

2.2 文字種ヒューリスティクスの問題

文字種に基づいたヒューリスティクスにより未知語候補を生成する手法が一般に用いられている．この場合，ルールに適合する文字列はすべて形態素候補となる．ところが，学習データに出現しない特徴を持つ文字列を形態素候補とすると，識別モデルでは適切に識別できず，解析誤りの原因となるという問題がある．例えば，文字種などの情報を利用してヒューリスティクスにより未知語候補を生成し，識別モデルである CRF に基づいて形態素解析する MeCab[*1] [3] で「震度 2 が福井県三国町岐阜県久瀬村名古屋市西区愛知県三好町などです。」を解析すると，「震度/2/が/福井県/三国町岐阜県久瀬村名古屋市西区愛知県三好町/など/です/。」と解析された．この解析結果には，文字種のヒューリスティクスで形態素候補と認定された不適切な形態素「三国町岐阜県久瀬村名古屋市西区愛知県三好町」が含まれている．このような特徴を持つ負例の形態素候補が学習データに存在しなかったため，適切に識別できなかったことが解析誤りの原因だと考えられる．そのため，入力文から形態素候補を生成する段階で，形態素となる可能性を計算して，極端に不適切な形態素候補を生成しないようにすることが重要である．

また，従来手法の最長一致法や分割数最小法を用いる場合も，不適切な未知語候補は解析精度の低下を引き起こす．

2.3 n-gram 生成モデルの問題

未知語の生成確率を文字 n-gram の生成モデルを用いて計算し，未知語候補の生成に利用する手法がある [4]．学習データに全くまたはほとんど出現しない文字列の生成確率は小さくなる．未知語は学習データに全くまたはほとんど出現しない場合が多いと考えられるため，未知語の文字列の生成確率は小さくなりやすく，適切に未知語形態素候補を生成することは困難である．

3 提案手法

提案手法は，文字列が与えられた条件のもとで，それらの文字列がどのような形態素の並びに分割されるかという条件付き確率を文字単位で計算する．これまで出現したことがない文字列が入力文に含まれている場合，2.3 節の生成モデルではそのような文字列の生成確率は必ず低くなってしまうが，文字列の分割を計算する条件付き確率では，そのような問題は起きない．

以下，提案手法について説明する．入力文字列 S の前後に端記号 "#" を追加し，w_i を形態素として，$S = w_1 w_2 .. w_i .. w_n = w_1^n$, $(w_1 = w_n =$"#"$)$ とする．n は形態素の数である．すると，形態素列への分割確率は $P(w_1^n|S)$ となる．さらに，各形態素が他の

† NHK 放送技術研究所, NHK

[*1] vesion 0.95 + ipadic-2.7.0-20060707

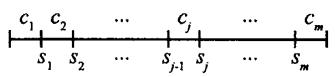

図 1　文字間の分割を示す s_j

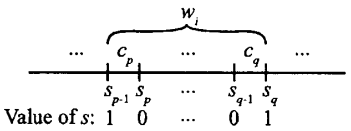

図 2　w_i と s_j との関係

形態素に依存しないように近似すると，

$$P(w_1^n|S) = \prod_{i=2}^{n} P(w_i|w_1^{i-1}, S) \approx \prod_{i=2}^{n-1} P(w_i|S) \quad (1)$$

となる．この $P(w_i|S)$ は，S 中のある文字列が形態素である確率を示している．

ここで，$P(w_i|S)$ を，文字単位の処理に変形する．文字を c_j として，$S = c_1c_2..c_j..c_m = c_1^m$ とする．m は文字数である．S をどのような w_1^n へ分割するかは，全ての文字 c_1^m の間が分割点（delimiter）か分割点でない（non-delimiter）かによって表現できる．そこで，図 1 に示すように，文字 c_j と c_{j+1} との間が delimiter (1) か non-delimiter (0) であるかを $s_j \in \{0,1\}$ で表す．

$w_i = c_p^q$ とすると，図 2 に示すように，w_i は $S = c_1^m$ と s_{p-1}^q で表現できる．そこで，

$$P(w_i|S) = P(s_{p-1}^q|S),\ s_{p-1} = s_q = 1,\ s_j = 0,\ \{j|p \le j < q\} \quad (2)$$

となる．$P(s_{p-1}^q|S)$ を近似して，s_j が他の s に依存しないようにし，さらに条件の $S = c_1^m$ を s_j の直前の a 文字と直後の a 文字に制限する．すると，

$$P(s_{p-1}^q|S) = P(s_{p-1}|S)\prod_{j=p}^{j=q} P(s_j|s_q^{j-1}, S) \approx \prod_{j=p-1}^{j=q} P(s_j|S)$$

$$\approx \prod_{j=p-1}^{j=q} P(s_j|c_{j-a+1}^{j+a}),\ s_{p-1} = s_q = 1,\ s_j = 0,\ \{j|p \le j < q\} \quad (3)$$

となる．

提案手法は，この 3 式の値を，入力文 S 中の文字列が形態素である確率とし，この確率が閾値以上の文字列を形態素候補とする．

入力文 S 中から，3 式の値が閾値以上となる文字列を探索する形態素候補生成アルゴリズムを図 3 に示す．始めに，各文字間が分割される確率を計算する（1〜2 行目）．次に，入力文字列の各文字位置を形態素候補の最初の文字位置として（3 行目），後続の文字を接続していき（5 行目），その文字列が 1 つの形態素となる確率を計算する（6 行目）．11 行目の e が閾値より小さくなった段階で，p から $q+1$ 以降の文字までの文字列の確率は閾値より小さくなることが明らかなので，5 行目から始まるのループを終了する（11〜13 行目）．これによって，計算量は全ての候補の確率を計算する場合の $O(n^2)$ より少ない $O(n)$ になっている．

$P(s_j|c_{j-a+1}^{j+a})$ を計算する確率モデルには，最大エントロピー法を用い，モデルのパラメータ推定は，Gaussian prior を用いて

```
1  d_j ← 1    (j = 1, j = m − 1)
2  d_j ← P(s_j = 1|c_{j−a+1}^{j+a})    (1 < j < m − 1)
3  for p ← 2 to m − 1 do
4    e ← d_{p−1}
5    for q ← p to m − 1 do
6      f ← e d_q
7      if f ≥ threshold then
8        add c_p^q to morphological candidate
9      endif
10     e ← e(1 − d_q)
11     if e < threshold then
12       break
13     endif
14   end
15 end
```

図 3　形態素候補生成アルゴリズム

MAP 推定する．最大エントロピー法で利用する素性には文字以外にカタカナ・漢字・ひらがなといった文字種も用いる．

具体的には，モデルの確率は次のように与えられる．

$$P(s_j|c_{j-a+1}^{j+a}) = \frac{1}{Z}\exp(\Lambda \cdot \mathrm{F}(c_{j-a+1}^{j+a}, s_j)) \quad (4)$$

ここで，$\mathrm{F}(c_{j-a+1}^{j+a}, s_j)$ は，c_{j-a+1}^{j+a} と s_j を特徴づける，0 または 1 を成分の値とする素性ベクトルを示す．Λ は，素性ベクトルの各成分に対応した重みのパラメータのベクトルである．"·" は内積を表す．モデルの正規化項 Z は，$s_j \in \{0,1\}$ であるので，

$$Z = \sum_{s' \in \{0,1\}} \Lambda \cdot \mathrm{F}(c_{j-a+1}^{j+a}, s') \quad (5)$$

となる．モデルのパラメータは，最尤推定により推定することができる．学習データ全体の対数尤度 $\mathcal{L}_\Lambda$ を最大化するようにパラメータを推定する．全学習データの文字間の位置を j で番号付けすると，

$$\Lambda = \underset{\Lambda}{\mathrm{argmax}}\ \mathcal{L}_\Lambda \quad (6)$$

$$\mathcal{L}_\Lambda = \sum_j \log\left(\frac{1}{Z}\exp(\Lambda \cdot \mathrm{F}(c_{j-a+1}^{j+a}, s_j)\right) - \frac{1}{2\sigma^2}\|\Lambda\|^2 \quad (7)$$

となる．σ^2 はハイパーパラメータである．最適解は，準ニュートン法である L-BFGS[5] を用いて求める．

4　実験と考察

4.1　実験設定

提案手法の有効性を示すために，京都コーパス ver. 4.0 と JUMAN ver. 5.1 の辞書を用いて，コーパス中で辞書に登録されていない形態素を形態素候補として推定する実験を行った．データの詳細を表 1 に示す．表 2 に使用した素性のテンプレートを示す．ただし，文字種を用いたテンプレートは省略している．文字種を用いたテンプレートは，表 2 の文字部分を文字種に置き換えたものを使用した．このテンプレートを用いて次のように素性ベクトルを構築した．例えば，テンプレート<c_j^{j+1}, s_j>から<c_j^{j+1}="値が", s_j=1>といった素性が生成されたとする．その素性の有無を返す素性関数 f は，

$$f(c_j^{j+1}, s_j) = \begin{cases} 1 & c_j^{j+1} = \text{“値が”},\ s_j = 1 \\ 0 & \text{otherwise} \end{cases}$$

表 1　実験データの詳細

コーパス	毎日新聞 ('95) & 京都コーパス ver.4.0
辞書	JUMAN ver.5.1
訓練データ	1 月 1〜10 日の記事，1〜6 月の社説
形態素数 (文字数)	501,306 (862,799)
開発データ	1 月 11〜12 日の記事，7〜8 月の社説
形態素数 (文字数)	155,528 (268,524)
テストデータ	1 月 13〜17 日の記事，9〜12 月の社説
形態素数 (文字数)	316,060 (543,534)
未知語数 (文字数)	7,032 (23,674)

表 2　素性テンプレート

文脈	テンプレート
前後	$<c_j^{j+1}, s_j>$, $<c_j^{j+2}, s_j>$, $<c_j^{j+3}, s_j>$, $<c_j^{j+4}, s_j>$, $<c_{j-1}^{j+1}, s_j>$, $<c_{j-2}^{j+1}, s_j>$, $<c_{j-3}^{j+1}, s_j>$, $<c_{j-1}^{j+2}, s_j>$, $<c_{j-1}^{j+3}, s_j>$, $<c_{j-2}^{j+2}, s_j>$
前	$<c_j, s_j>$, $<c_{j-1}^{j}, s_j>$, $<c_{j-2}^{j}, s_j>$, $<c_{j-3}^{j}, s_j>$
後	$<c_{j+1}, s_j>$, $<c_{j+1}^{j+2}, s_j>$, $<c_{j+1}^{j+3}, s_j>$, $<c_{j+1}^{j+4}, s_j>$

表 3　実験結果（リコールと候補数）

リコール (正解数)	提案手法	ヒューリスティクス	生成モデル	未知語全候補
0.964 (6779)	50,201	1,833,075	1,575,373	-
0.999 (7025)	872,318	-	3,058,581	-
1.0 (7032)	2,645,163	-	3,922,453	14,865,057

となる．このような素性関数を素性ベクトルの成分とすることで素性ベクトルを構築する．学習データに 3 回以上出現したテンプレートの s_j 以外の部分と $s_j = 0$ との素性と，$s_j = 1$ との素性を用いた．ハイパーパラメータは開発データを用いて設定した．

4.2　実験結果

テストデータから生成した，辞書に登録がない形態素候補を確率で順位付けした実験結果を図 4 と表 3 に示す．図 4 と表 3 は同一の実験結果を示している．図 4 は，横軸が辞書に登録がなかった形態素候補の順位，縦軸が辞書に登録がなかった未知語の正解形態素の累積数である．表 3 は，3 つのリコールポイントでの候補数である．“ヒューリスティクス” は，辞書に登録がない 5 文字以下の文字列と連続するカタカナ文字列を形態素候補として生成した場合である．“生成モデル” は，形態素候補 w の前後に端記号を付与し，$w = c_1 c_2 .. c_j .. c_m$, ($c_1$ = <BOS>, c_m = <EOS>) とし，文字バイグラムモデル[*2]による生成確率 $P(w) = \prod_{j=2}^{m} P(c_j|c_{j-1})$ により順位付けした結果である．

4.3　考察

表 3 より提案手法は，“ヒューリスティクス” より少ない候補で，“ヒューリスティクス” のリコール 0.964 よりも高いリコール 0.999 を達成している．“ヒューリスティクス” で候補に含まれなかった正解形態素は，長さが 6 文字以上のものである．そのため，提案手法は長い未知語形態素候補を生成できていることが分かる．正解未知語形態素の 99.9% が候補に含まれるということは，未知語でないものも含めた全正解形態素の 99.998% が候補に含まれることになる．また，図 4 と表 3 より，提案手法は，“生成モデル” と比べて同じリコールを少ない候補で達成していることが分かる．

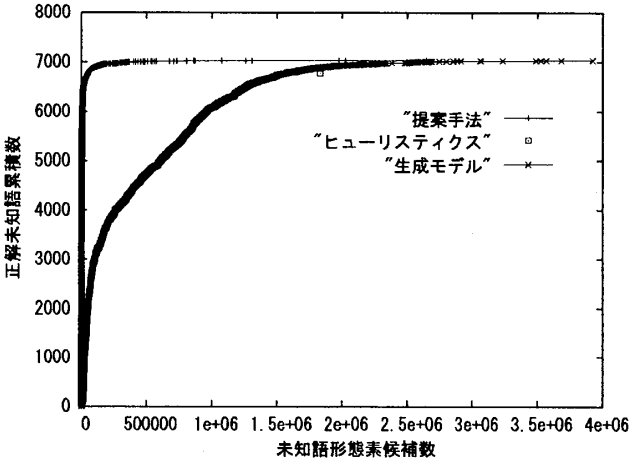

図 4　実験結果（正解未知語累積数と候補数）

さらに，2.2 節の例を提案手法で解析したところ，「三国町岐阜県久瀬村名古屋市西区愛知県三好町」の確率は，テストデータの正解未知語形態素の最小確率 (3.9×10^{-16}) よりも小さな値 (8.4×10^{-22}) となり，形態素候補として不適切なこの表現を排除できることが分かった．提案手法では，形態素の境界となりそうな表現を多く含む形態素候補の確率は小さくなるので，多くの形態素を含む極端に不適切な候補を排除できるといえる．

5　おわりに

形態素解析の未知語処理での形態素候補生成を自動で行う手法を提案した．提案手法は，候補とする形態素に長さの制限がなく，極端に不適切な候補を排除できるという特徴を持つ．今後は，この手法を利用した形態素解析の評価を行いたい．

参考文献

[1] 中川 哲治，松本 裕治，単語レベルと文字レベルの情報を用いた中国語・日本語単語分割，情報処理学会論文誌，Vol.46, No.11, pp.2714-2727, Nov. 2005.

[2] 内元 清貴，関根 聡，井佐原 均，最大エントロピーモデルに基づく形態素解析—未知語の問題の解決策—，自然言語処理，Vol.8, No.1, pp.127-141, Jan. 2001.

[3] 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，SIG-NL-161(13), pp.89-96, 2004.

[4] 永田 昌明，統計的言語モデルと N-best 探索を用いた日本語形態素解析法，情報処理学会論文誌，Vol.40, No.9, pp.3420-3431, Sep. 1999.

[5] D.C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization.” Math. Program., Vol.45, Issue 3, pp.503-528, December 1989.

[*2] The CMU-Cambridge Statistical Language Modeling Toolkit v2 の標準設定を用い，グッド・チューリング法で頻度をディスカウンティングした．