Shift-Reduceと相対モデルの二側面を持つ日本語係り受け解析の対数線形モデルへの適用

山本 悠二　増山 繁　(豊橋技科大)

y_yamamoto@la.cs.tut.ac.jp, masuyama@tut.jp

1　はじめに

係り受け解析における n-best の出力は，rerankingによる係り受けの補正に有用であることが知られている (e.g. [1, 2])．日本語係り受け解析において，n-best 解を導出するためには相対モデルなどの係り先候補集合から係り先を選択する解析方法が使用される．この解析方法は，予め Shift-Reduce 法により，確実に係ると判断される文節対について部分係り受けを行うと，相対モデルにおける係り受け解析において係り先候補数を減らすことができ，n-best 解についても高速に導出することが見込まれる．しかしながら，二段階の解析方法による係り受け解析 [9] は確率値を出力する方法で定式化されていないため，信頼性のある n-best 解が得られない可能性がある．そこで本稿では，Shift-Reduce 法と相対モデルの二側面を持つ日本語係り受け解析について，対数線形モデルによる学習を行うことについての定式化を行い，実験で有効性を調査する．

2　対数線形モデルのオンライン学習

まず，係り受け解析の学習について示す前に，対数線形モデルについてのオンライン学習を示す．記号の定義として，入力 x_i と出力ラベル $y_i \in \mathcal{Y}$ から生成される素性ベクトルを $F(x_i, y_i)$，重みベクトルを $\mathbf{w}$ と表記する．このとき，(識別モデルとしての) 対数線形モデルは $p(y_i|x_i;\mathbf{w}) = \frac{\exp(\mathbf{w}\cdot F(x_i,y_i))}{\sum_{y\in\mathcal{Y}}\exp(\mathbf{w}\cdot F(x_i,y))}$ と表せる．そして，t 番目の更新において与えられる学習事例集合を s_t，t 番目の更新によって得られた重みベクトルを $\mathbf{w}_t$ (ただし，$\mathbf{w}_0 = \mathbf{0}$) と表記する．ここで，$s_{t+1}$ が与えられたときに，重みベクトル $\mathbf{w}_t$ を更新することを考える．このとき，単純に $t+1$ 番目で与えられた学習事例に対して最尤となるように重みベクトルを更新すると，過去の学習事例を無視して現時点の学習事例のみにオーバーフィットしてしまうという問題が生じる．そのため，更新前と更新後の重みベクトルの差分についての Gaussian Prior を設け，MAP 推定を行う[*1]．この最適化問題は以下のように記述することができる (ただし，D は定数)．

$$\mathbf{w}_{t+1} = \arg\max_{\mathbf{w}} \log \prod_{i\in s_{t+1}} p(y_i|x_i;\mathbf{w}) - \frac{D}{2}||\mathbf{w}-\mathbf{w}_t||^2$$

この双対問題は，$\psi_{i,y} = F(x_i, y_i) - F(x_i, y)$ と置いて，以下のように表せる．

$$\min_{\alpha} \sum_{i\in s_{t+1}}\sum_{y}\alpha_{i,y}(\mathbf{w}_t\cdot\psi_{i,y} + \log\alpha_{i,y}) + \frac{1}{2D}||\sum_{i\in s_{t+1}}\sum_{y}\alpha_{i,y}\psi_{i,y}||^2$$

subj. to

$$\sum_{y}\alpha_{i,y} = 1 \qquad \forall i \in s_{t+1} \text{ and } \alpha_{i,y} > 0$$

ただし，$\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{1}{D}\sum_{i\in s_{t+1}}\sum_y \alpha_{i,y}\psi_{i,y}$ である．この最適化問題は，Exponentiated Gradient[4, 5] を使用することで解くことができる．

3　Shift-Reduceと相対モデルの二側面を持つ日本語係り受け解析

以下では，対数線形モデルによる Shift-Reduce と相対モデルの二側面を持つ日本語係り受け解析 “bilateral parsing” を示す．

3.1　記号の定義

まず，N 個の文節列で構成される日本語文について，文節列を保持する配列を `B[]` とする．以降，配列は 0 から始まるものとする．そして，係り受け解析後の，推定された係り先文節の添字番号が格納されている配列を `est[]` とする．なお，係り先を持たない，もしくは，まだ係り先が決定していない場合には -1 が格納される．また，訓練データの文節列については，正しい係り先文節の添字番号が格納されている配列 `ans[]` が与えられる．

ここで，`B[]` の i 番目と j 番目の文節対を表す素性ベクトルを $F(\langle i,j\rangle, \mathtt{B})$ と表記する．また，文節対の素性ベクトルに対して非線形写像を適用したものを $\phi(\langle \mathtt{i},\mathtt{j}\rangle, \mathtt{B})$ と表記する．そして，$\phi(\cdot)$ を単位ベクトルに正規化したものを $\omega(\langle \mathtt{i},\mathtt{j}\rangle, \mathtt{B}) = \phi(\langle \mathtt{i},\mathtt{j}\rangle, \mathtt{B})/||\phi(\langle \mathtt{i},\mathtt{j}\rangle, \mathtt{B})||$ と表記する．なお，非線形カーネルは特徴空間内での内積を，入力された素性ベクトル上の空間 (入力空間) 内での内積に対して非線形写像を適用したものとして計算することができる．例えば，統計的日本語係り受け解析

[*1] 論文 [3] においても，更新するパラメータ (2節における双対変数に相当) についての Gaussian Prior を設け，MAP 推定を行うことでオンライン学習を実現している．ただし，その最適化問題の Prior が重みベクトル更新についてどのような変化をもたらすのかについての解釈はできない．

でよく用いられる多項式カーネル (次元数を d とする) の場合，$\phi(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})\cdot\phi(\langle \mathtt{k},\mathtt{l}\rangle,\mathtt{B'}) = \{1+F(\langle \mathtt{i},\mathtt{j}\rangle,\mathtt{B})\cdot F(\langle \mathtt{k},\mathtt{l}\rangle,\mathtt{B'})\}^d$ で計算することができる．

次に，日本語係り受け解析で主に用いられる 2 つの識別モデルについて述べる [6]．一つ目は**絶対モデル**と呼ばれるもので，i 番目と j 番目の文節対が「係る」か「係らない」かを二値分類するものである．対数線形モデルにおいて，先の文節対が係る確率を以下のように表す．

$$p^{abs}(i{\rightarrow}j|\mathtt{B};\mathbf{w}) = \frac{\exp(\mathbf{w}\cdot\omega(\langle i,j\rangle,\mathtt{B}))}{\exp(\mathbf{w}\cdot\omega(\langle i,j\rangle,\mathtt{B}))+\exp(-\mathbf{w}\cdot\omega(\langle i,j\rangle,\mathtt{B}))}$$

二つ目は**相対モデル**と呼ばれるものである．これは，i 番目の文節について，係り先候補の添字番号集合 $C = \{i+1,\cdots,|\mathtt{B}|-1\}$ が与えられたもとで，j $(\in C)$ 番目の文節への係りやすさを求めるものである．対数線形モデルにおいて，先の文節対が係る確率を以下のように表す．

$$p^{rel}(i\rightarrow j|C,\mathtt{B};\mathbf{w}) = \frac{\exp(\mathbf{w}\cdot\omega(\langle i,j\rangle,\mathtt{B}))}{\sum_{k\in C}\exp(\mathbf{w}\cdot\omega(\langle i,k\rangle,\mathtt{B}))}$$

前者の絶対モデルは Shift-Reduce の係り受け解析に使用することができる．例えば，論文 [7] の場合，$p^{abs}(i\rightarrow j|\mathtt{B}) > 0.5$ のとき Reduce 操作，$p^{abs}(i\rightarrow j|\mathtt{B}) < 0.5$ のとき Shift 操作に結びつけることで解析ができる[*2]．

3.2 学習アルゴリズム

擬似コードの学習アルゴリズムを示す前に，入力として与える引数について説明する．まず，T は訓練データの文集合である．T の要素は，「訓練データの文節列，正しい係り先文節の添字番号の配列」の対で構成される．D は，2 節で示したオンライン学習において，どれだけ積極的に重みベクトルを更新するかについて決めるパラメータである．D が小さければ，与えられた事例について大きな補正がかかる．I は，訓練データセット単位で何回学習を繰り返すかについて指定するパラメータである．擬似コードを図 1 に示す．このコードでは，訓練データから文を取り出し，Shift-Reduce の係り受け解析で部分的な係り受けができるように学習し，その後で残りの文節において，相対モデルによる係り受けができるように学習する．なお，4 節での実験で用いた学習アルゴリズムは，論文 [10] に掲載されている重みベクトルの平均化を行った．

```
function train(T, D, I)
w ← 0;
for (iter = 1; iter <= I; iter++) {
  foreach ( ⟨B,ans⟩ ∈ T ) {
    N = |B|;  //  |B| は文節数を表す
    est = [-1] * N;
    ⟨w,est⟩ = bil_sr_train(w,D,B,ans,est);
    ⟨w,est⟩ = bil_rel_train(w,D,B,ans,est);
  }
}
return w;
```

図 1　擬似コード - Bilateral Parsing の学習アルゴリズム

擬似コードで示した Shift-Reduce での部分係り受け解析の学習アルゴリズムを図 2 に示す．なお，擬似コード中の関数 pop は第 1 引数のスタックから，先頭の要素を取り除き，その要素を返す機能を持ち，関数 push は第 1 引数のスタックの先頭に第 2 引数の値を追加する機能を持つ．このアルゴリズムは，基にした解析アルゴリ

```
function bil_sr_train(w,D,B[],ans[],est[])
N = |B|;
push(stack,-1);  // −1 は番兵
push(stack, 0);
for (j = 1; j < N; j++) {
  i = pop(stack);
  while (i != -1) {
    if ( (i == N-2) && (j == N-1) )
      est[i] = j;
    else {
      pr = p^abs(i → j|B; w) ;
      if (j != ans[i]) {
        w ← arg max_w' log{1 − p^abs(i → j|B; w')}
                              − (D/2)||w' − w||^2  ;
        break;
      }
      else if ( (j == ans[i]) && (pr >= 0.5) ) {
        est[i] = j;
        w ← arg max_w' log p^abs(i → j|B; w')
                              − (D/2)||w' − w||^2  ;
      }
      else  //  (j == ans[i]) && (pr < 0.5)
        break;
    }
    i = pop(stack);
  }
  push(stack,i);
  push(stack,j);
}
return ⟨w, est⟩;
```

図 2　擬似コード - Bilateral Parsing の Shift-Reduce 解析側の学習アルゴリズム

ズムである颯々野の係り受け解析 [7] とは，末尾まで探索しても係り先が見つからなかった場合は，末尾を係り先と定めていない (つまり，`estlink[]` の要素が -1 のまま変わらない) 点で異なる．これにより部分係り受け解析を実現している．このアルゴリズムの重みベクトルの更新について見てみる．ここで，ある文節対 $\langle i,j\rangle$ が与えられた場合に，更新前の重みベクトルを $\mathbf{w}_t$，更新後の重みベクトルを $\mathbf{w}_{t+1}$ と表記する．更新則 $\mathbf{w}_{t+1} = \arg\max_{\mathbf{w}}\log\{1-p^{abs}(i\rightarrow j|\mathtt{B};\mathbf{w}')\} - \frac{D}{2}||\mathbf{w}'-\mathbf{w}_t||^2$ を式変形すると，$\mathbf{w}_{t+1} = \arg\max_{\mathbf{w}} - \log\{1+\exp(2\mathbf{w}'\cdot\omega(\langle i,j\rangle,\mathtt{B}))\} - \frac{D}{2}||\mathbf{w}'-\mathbf{w}_t||^2$ となる．この式を最大化するためには，式中の exp 内の値ができるだけ負数を取るように $\mathbf{w}'$ が定められる．つまり，多くの場合，次の関係が成立する．

1. $j \neq \mathtt{ans}[i]$ (i 番目の文節は j に係らない) 場合:
   $p^{abs}(i\rightarrow j|\mathtt{B};\mathbf{w}_{t+1}) < 0.5$

同様にもう一つの更新則についても調べると，2 節の双対問題から以下の不等式が成立する[*3]．

2. $j = \mathtt{ans}[i]$ (i 番目の文節は j に係る) 場合:

[*2] 等式が成立する場合にどちらの操作を取るかは任意である．

[*3] $||\omega(\langle i,j\rangle,\mathtt{B})|| > 0$ という自明な前提を置く．

```
function bil_rel_train(w,D,B[],ans[],est[])
N = |B|;
for (i = N - 3; i >= 0; i--) {
  if (est[i] != -1) { //  すでに係り先が決定している
    continue;
  }
  C ← {};  // 非交差を満たす係り先添字番号集合
  j = i + 1;
  while (j != -1) {
    C ← C ∪ {j};
    j = est[j];
  }
  w ← arg max_w' log p^rel(i → j|C,B;w')
                        − (D/2)||w' − w||^2  ;
  est[i] = ans[i];
}
return ⟨w, est⟩;
```

図 3　擬似コード - Bilateral Parsing の 相対的な比較による解析側の学習アルゴリズム

(a) $p^{abs}(i \to j|\mathtt{B};\mathbf{w}_t) \geq 0.5$ の場合:
$p^{abs}(i \to j|\mathtt{B};\mathbf{w}_{t+1}) > p^{abs}(i \to j|\mathtt{B};\mathbf{w}_t)$

2-(a) の条件では，この時点で係り先が決定できる．一方，「2-(b): $j = \mathtt{ans}[i]$ かつ $p^{abs}(i \to j|\mathtt{B};\mathbf{w}_t) < 0.5$」の場合, 重みベクトルの更新は行わない．加えて，この時点で係り先が決定されない．これは，文節対に係り受けの曖昧性があることを考慮しているためである．仮に，文節対に係り受けの曖昧性がある場合に正例として学習すると，よく似た形式の未知の文節対に対して Reduce 操作を行い，長距離依存の係り先同定に誤りが生じる可能性があるためである．

関数 bil_rel_train (図 3) は，論文 [8, 6] を基にして，決定的な解析で係り先が同定できなかった係り元について優先度学習を用いて重みベクトル $\mathbf{w}$ を更新するように変更を加えたものである．なお，擬似コード中の添字番号集合 C は，3.1 節の相対モデルの定義に正確に従うならば係り元より後方にある文節の添字番号集合を取るべきであるが，係り受けに影響が少ないと考えられる交差を満たす係り先文節は省くことにした．このアルゴリズムの重みベクトルの更新について見てみる．係り元である i 番目の文節について，非交差を満たす係り先添字番号集合を C とする．そして，更新前の重みベクトルを $\mathbf{w}_t$，更新後の重みベクトルを $\mathbf{w}_{t+1}$ と表記する．更新則を式変形すると，$\mathbf{w}_{t+1} = \arg\max_{\mathbf{w}'} -\log \sum_{k \in C} \exp[\mathbf{w}' \cdot \omega(\langle i,k\rangle, \mathtt{B}) - \mathbf{w}' \cdot \omega(\langle i, \mathtt{ans}[\mathtt{i}]\rangle, \mathtt{B})] - \frac{D}{2}||\mathbf{w}' - \mathbf{w}_t||^2$ が成立するため，最大化するために式中の exp 内の値ができるだけ負数を取るような $\mathbf{w}'$ が定められる．ゆえに重みベクトルの更新により，$p^{rel}(i \to \mathtt{ans}[i]|C,\mathtt{B};\mathbf{w}_{t+1}) > p^{rel}(i \to l|C,\mathtt{B};\mathbf{w}_{t+1})$ (ただし，l は $l \in C$，かつ，$l \neq \mathtt{ans}[\mathtt{i}]$) なる関係が多くの場合成立する．また，2 節の双対問題から，以下の関係が成立する[*4][9].

$$p^{abs}(i \to j|\mathtt{B};\mathbf{w}_{t+1}) \geq p^{abs}(i \to j|\mathtt{B};\mathbf{w}_t)$$

[*4] 正確には動的素性がある場合は，この不等式が成立しない場合がある．

この性質は次のことを意味する．まず，ある文節対が一貫して正しい係り受けになる場合，ある程度学習が進むと $p^{abs}(i \to j|\mathtt{B})$ の値が高くなる．この場合，Shift-Reduce の 係り受け解析の 2-(a) の条件で重みベクトルが更新されるため，Shift-Reduce 側の係り受けの同定が行われる．

3.3 解析アルゴリズム

解析アルゴリズムについては，[9] のように，Shift-Reduce の係り受け解析で部分的な係り受け解析を行い，その後で残りの文節について相対モデルによる係り受けを行う．このとき，相対モデルによる係り受け解析については論文 [8] のようにビームサーチによる n-best 解を求めることができる．例えば，図 4 のように，2 番目の文節以外の係り先が Shift-Reduce の係り受け解析で既に決定しているとする．このとき，文全体の確率を以下

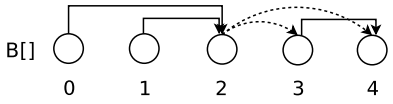

図 4　部分係り受けの例

のように定める．

- 2 番目の文節が 3 番目の文節に係る場合:
  $p^{abs}(0{\to}2|\mathtt{B})p^{abs}(1{\to}2|\mathtt{B})p^{rel}(2{\to}3|\{3,4\},\mathtt{B})$
- 2 番目の文節が 4 番目の文節に係る場合:
  $p^{abs}(0{\to}2|\mathtt{B})p^{abs}(1{\to}2|\mathtt{B})p^{rel}(2{\to}4|\{3,4\},\mathtt{B})$

4 実験

3 節で示した，対数線形モデルによる Shift-Reduce と相対モデルの二側面を持つ日本語係り受け解析の評価を行う．京都テキストコーパス 4.0[*5]を以下の 3 つに分けて実験を行った．

- 訓練データ: 一般記事[*6] 1 月 1，3-11 日，社説 1-8 月，合計 24,280 文，234,639 文節
- 開発データ: 一般記事 1 月 12，13 日，社説 9 月，合計 4,833 文，47,571 文節
- 評価データ: 一般記事 1 月 14-17 日，社説 10-12 月，合計 9,284 文，89,874 文節

これらの記事の分け方は，論文 [6] と同じである．

次に提案手法の学習についての詳細について述べる．実験では 3 次の多項式カーネルを用いた．また，指定すべきパラメータ D と I については，$D = \{0.1, 1, 10\}$, $I = \{1, \cdots, 20\}$ の組合せでモデルをそれぞれ生成し，開発データの係り受け正解率が最も高くなる値 ($D = 1$, $I = 14$) を使用した．なお，学習によって得られたモデル (対数線形モデル) は，多項式カーネルの計算に時間が掛かるため，論文 [11] に示されている素性の組合せを陽に展開する手法 (Polynomial Kernel Expanded; PKE)

[*5] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/corpus.html

[*6] ただし，以下の ID を持つ文は文節番号とその文節の係り先番号が同一であるというタグ付けの誤りがあったため，訓練データから除外した; 950101159-010, 950106177-017, 950106192-002.

により高速化を施した．PKE で用いたパラメータ [11] は $\sigma = 0.0005$，$\xi = 1$ とした．

学習に用いた素性は，CaboCha 0.53 中にある素性抽出プログラム selector.pl の出力を使用した．ただし，動的素性については使用していない．

4.1 実験結果

まず，ビーム幅を 1 としたときの提案手法の係り受け正解率は 91.12 % (73437 / 80590), 文正解率は 55.08 % (5114 / 9284) であった．なお，ここでの係り受け正解率は，文末の一文節を除くすべての文節に対して正しく係り先が同定できたものの割合，文正解率は，文単位で全体の文節の係り先が正しく同定できたものの割合を示す．参考として，論文 [7] で示されている Shift-Reduce 法による日本語係り受け解析について，同一の素性集合 (動的素性を含む) で SVM ($C = 0.0001$) を用いて学習した場合の結果を述べると，係り受け正解率 90.74%，文正解率 54.06% であった．提案手法の係り受け正解率，文正解率は，有意水準 5% (両側) の二項検定で，Shift-Reduce 法による日本語係り受け解析の正解率に対して有意差が認められた．

次に，ビーム幅を変化させた場合の係り受け，文正解率を表 1 に示す．係り受け正解率はビーム幅 2，3 のときに，文正解率はビーム幅 2 のときに最大になっている．このようにビームサーチにおいて各解析段階である程度正解を絞るほうが正解率が高くなる現象は，他の日本語係り受け解析で n-best を求める研究 [8, 12] の知見と一致する．一方，他の研究と異なる現象として，ビーム幅を大きくしても係り受け正解率や文正解率が急激に下がっていないことが挙げられる．提案手法は，最初に Shift-Reduce 法により確実に係ると判断される文節の係り先を定める．この係り先は以降の相対的な比較による解析においても固定のままであるため，ビームサーチに使用するコストには影響を及ぼさない．したがって，確実に係ると判断される文節対において適切なコストが推定されていない場合であっても，頑健に解析が行える．実際に，提案手法の Shift-Reduce の解析のみを行った結果を用いた適合率・被覆率[*7]はそれぞれ 0.9711，0.7410 であった．

表 1　ビーム幅と係り受け正解率，文正解率との関係

ビーム幅	係り受け正解率	文正解率
1	91.12 %	55.08 %
2	91.19 %	55.22 %
3	91.19 %	55.16 %
10	91.18 %	55.16 %
20	91.18 %	55.15 %

[*7] それぞれの定義は，(適合率) = (解析器が出力した文節のうち正解した数) / (解析器が係り先を出力した文節数)，(被覆率) = (解析器が係り先を出力した文節数) / (末尾の文節を除いた総文節数) である．

5 おわりに

本稿では，統計的日本語係り受け解析の n-best 解を高速に求める方法として，対数線形モデルで定式化した Shift-Reduce 法と相対モデルの二側面を持つ日本語係り受け解析手法を示した．実験より，提案手法の Shift-Reduce の解析のみを行った結果を用いた適合率・被覆率はそれぞれ 0.9711，0.7410 となり，7 割程度の文節の係り先が Shift-Reduce の解析の時点で正確に同定できていることが確認された．また，ビーム幅を 2 以上の値にしたときの係り受け正解率・文正解率が決定的 (ビーム幅 1) の正解率よりも向上していることが確認された．今後の課題として，Conditional Random Fields のような構造を持つ対数線形モデルについて，Shift-Reduce 法と相対モデルの二側面を持つような解析方法が適用できるかについて検討していきたい．

謝辞

本研究は文部科学省グローバル COE プログラム「インテリジェント　センシングのフロンティア」，日本学術振興会科研費 (C)22500129, 総務省戦略的情報通信研究開発推進制度（SCOPE）地域 ICT 振興型の支援を受けた．

参考文献

[1] David McClosky, Eugene Charniak, Johnson, Mark: Reranking and Self-Training for Parser Adaptation, *Proc. ACL 2006*, pp. 337-344 (2006).
[2] 阿辺川武, 奥村学 : 共起情報及び複数格の組み合わせを 考慮した係り受け解析, 自然言語処理, Vol.13, No.2, pp.43-62 (2006).
[3] Richard Johansson: Logistic Online Learning Methods and Their Application to Incremental Dependency Parsing, *Proc. ACL 2007*, pp.49-54 (2007).
[4] Jyrki Kivinen, Manfred K. Warmuth: Exponentiated Gradient versus Gradient Descent for Linear Predictors, *Journal of Information and Computation*, Vol. 132, Issue 1, pp. 1-63 (1997).
[5] Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, Peter L. Bartlett: Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks, *Journal of Machine Learning Research*, Vol. 9, pp. 1775-1822 (2008).
[6] 工藤 拓, 松本 裕治 : 相対的な係りやすさを考慮した日本語係り受け解析モデル, 情報処理学会論文誌, Vol.46, No.4, pp.1082-1092 (2005).
[7] 颯々野 学 : 日本語係り受け解析の線形時間アルゴリズム，自然言語処理，Vol.14，No.1, pp.3-18 (2007).
[8] 関根 聡，内元 清貴，井佐原 均 : 文末から解析する統計的係り受け解析アルゴリズム，自然言語処理，Vol.6，No.3，pp.59-73 (1999).
[9] 山本悠二，増山繁 : 決定的な解析と相対的な比較による解析の二側面を持つ日本語係り受け解析，情報処理学会研究報告 自然言語処理研究会報告 (2010).
[10] Hal Daumé III: Practical Structured Learning Techniques for Natural Language Processing, PhD Thesis, University of Southern California (2006).
[11] 工藤 拓，松本 裕治 : カーネル法を用いた言語処理における高速化手法，情報処理学会論文誌，Vol.45，No.9，pp.2177-2185 (2004).
[12] Taku Kudo, Yuji Matsumoto: Japanese Dependency Structure Analysis Based on Support Vector Machines, *Proc.* EMNLP/VLC 2000, pp. 18-25 (2000).