論 文 Technical Papers

# 品詞自動タグづけシステム－伸縮性入力ニューロタガー

## Automatic Part-Of-Speech Tagging System - An Elastic Neuro Tagger

馬 青 * 内元 清貴 * 村田 真樹 * 井佐原 均 *
Qing Ma Kiyotaka Uchimoto Masaki Murata Hitoshi Isahara

* 郵政省通信総合研究所
Communications Research Laborotory, Ministry of Posts and Telecommunications, Japan

19YY 年 MM 月 DD 日 受理

Keywords: POS tagging, variable length of contexts, information gain, neural networks, elastic inputs.

Summary

A multi-neuro tagger that uses variable lengths of contexts and weighted inputs with information gains for part of speech tagging has been proposed by the authors [Ma 98, 馬 99]. It has been shown that the tagger has an accuracy higher than any of those obtained using the single neural networks with the fixed length of inputs, which indicates that the length of the context need not be chosen empirically; it can be selected dynamically instead. In this work, by introducing elastic inputs the tagger is slimed down into a single neural network which inherits the features that the multi-neuro tagger has. Computer experiments show that the new neuro tagger has an accuracy slightly higher than the original one instead. A series of comparative experiments for the new neuro tagger and various probability models, which include the frequency model (a base-line model), local n-gram model, and HMM, are further performed for evaluating the neuro tagger. These experimens show that the elastic neuro tagger is definitely far superior to these probability models.

## 1. ま え が き

品詞タグづけは，単語の品詞における曖昧性を文脈によって解消することであり，パーザや機械翻訳といった自然言語処理に欠かせない基本技術である．その技術は，また，音声合成の前処理，OCR や音声認識の後処理，そして情報検索などへの幅広い応用も考えられる．

これまで数多くの品詞タグづけシステム（例えば，ルールベース：[Garside 87, Hindle 89, Brill 92]，確率モデル：[Church 88, DeRose 88, Cutting 92, Charniak 93a, b, Weischedel 93, Merialdo 94, Schütze 94]，ニューラルネット：[Nakamura 90, Schmid 94, Ma 96]，最大エントロピー法：[Ratnaparkhi 96]，決定木：[Daelemans 96, Marquez 97]）が提案されてきた．それらのほとんどはタグづけに長さが固定の文脈を用いるものであり（HMM においても状態遷移を定義するのに固定された n-gram ベースのモデルを用いる），入力の各構成部分は同一の影響度を持つものとされていた．それらのシステムで高い精度を出すためには大量（英語の場合 1,000,000 オーダ）の訓練データを用いることを前提とする必要がある．しかしながら，実際，英語や日本語などを除いた数多くの言語（例えば本稿で取り上げたタイ語）に関しては，コーパス自体もまだ整備段階にあるのが現状で，予め大量の訓練データを得るのが困難である．

そのために，筆者らは少ない訓練データを用いても，十分実用的な品詞タグづけが可能なマルチニューロタガーを提案した[Ma 98, 馬 99]．マルチニューロタガーはそれぞれ長さの異なる文脈を入力とした複数のニューラルネットとそれらの出力を選別する最長文脈優先セレクターで構成され，主に以下の二つの特徴を有

する：（1）タグづけ結果の確信度を高めるために，タグづけはまずできるだけ長い文脈を用いて行ない，訓練データの不足から確定的な答えが出ない場合に順次文脈を短くするといったように長さ可変文脈を用いている；（2）タグづけにおいては，目標単語自身の影響が最も強く、前後の単語もそれぞれの位置に応じた影響度を持つことを反映させるために、入力の各構成部分は情報量最大を考慮して訓練データから得られるインフォメーションゲイン（略してIGと呼ぶ）[Daelemans 92, Quinlan 93]を影響度として重み付けられている。計算機実験の結果，マルチニューロタガーは，品詞の曖昧性のある22,311個の単語を含む小規模タイ語コーパスを訓練に用いることにより，テストデータを94.3%の精度（曖昧性のある単語のみで測定した場合）でタグづけすることができた．この結果は，どの固定長さの文脈を入力とした単一のニューラルネットを用いた場合よりも優れ，マルチニューロタガーを用いることによってよさそうな文脈の長さを先行実験で決めておく必要がなくタグづけ過程において動的に選択できることを示している．

本研究ではまず伸縮性入力を導入することによってマルチニューロタガーの構造の大幅なスリム化を行なった．新しく提案する伸縮性入力ニューロタガーはマルチニューロタガーの特徴をそのまま継承しながら単一のニューラルネットのみで構成される．計算機実験の結果，伸縮性入力ニューロタガーはマルチニューロタガーと同等以上の精度でタグづけできた．更に，伸縮性入力ニューロタガーの性能を他の手法との比較を通じ客観的に評価するために，頻度情報のみのベースラインモデル，局所的にn-gram確率を使う手法（これは本稿で比較のため特別に設定したもので局所n-gramモデルと呼ぶ），そしてコーパスの品詞を内部状態とする隠れマルコフモデル（HMM）などの確率モデルとの比較実験を行なった．実験によれば，それらの確率モデルよりも提案手法の方が品詞タグづけの精度が高かった．

以下，2節では品詞タグづけ問題の定式化，3節では伸縮性入力ニューロタガー，4節では確率モデル，そして，5節では計算機実験の結果について述べる．

## 2. 品詞タグづけ問題

本研究では，単語の取りうる品詞がリストアップされている辞書

$$V = (w^1, w^2, \cdots, w^v) \tag{1}$$

と品詞セット

$$\Gamma = (\tau^1, \tau^2, \cdots, \tau^\gamma) \tag{2}$$

が既に用意されているものと仮定する（但し，$v$は登録された単語の総数で，$\gamma$は品詞の数である）．つまり，本研究では（辞書に存在しない）未知語は取り扱わない．従って，品詞タグづけ問題は，任意の文$W = w_1 w_2 \cdots w_s$（$w_i \in V,\ i = 1, \cdots, s$）が与えられた時，以下のように単語レベル(3)と文レベル(4)のいずれかの手続き（$\varphi_1,\ \varphi_2$）によって品詞列$T = \tau_1 \tau_2 \cdots \tau_s$（$\tau_i \in \Gamma, i = 1, \cdots, s$）を見つけることである．

$$\varphi_1 : W^t \rightarrow \tau_t,\ t = 1, \cdots, s \tag{3}$$

$$\varphi_2 : W \rightarrow T \tag{4}$$

但し，$t$は品詞を定めようとする目標単語のインデックスを表し，$W^t$は目標単語$w_t$を中心とした長さ$l+1+r$の単語列である．即ち，

$$W^t = w_{t-l} \cdots w_t \cdots w_{t+r} \tag{5}$$

但し，$t-l \geq 1, t+r \leq s$．従って，単語レベルと文レベルでのタグづけはそれぞれ（品詞をクラスに置き換えた）クラス分け問題と（品詞を状態に置き換えた）最適状態遷移系列の探索問題としてとらえることができる．

## 3. 伸縮性入力ニューロタガー

新しく提案するニューロタガーは，単一の三層パーセプトロン（図1）で構成され，単語レベル[式(3)]で品詞タグづけを行なうシステムである．但し，ニューロタガーはその入力が伸縮可能なものとされているため，前に提案したマルチニューロタガーと同様，品詞タグづけを長さ可変文脈で行うことができる．この節では，順に入出力構成，入力の重みづけに用いるインフォメーションゲイン，三層パーセプトロンのアーキテクチャ，伸縮性入力に適した訓練方法，そして伸縮性入力ニューロタガーとマルチニューロタガーとの比較について述べる．

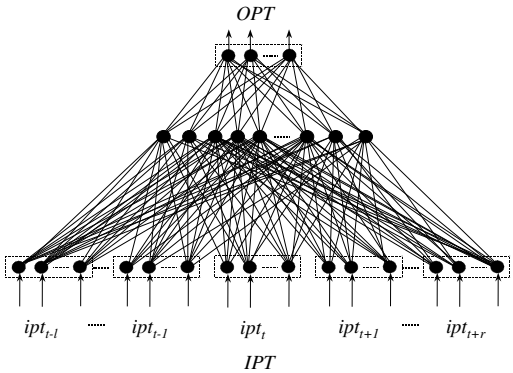


図1 伸縮性入力ニューロタガー

### 3・1 入出力構成

入力$IPT$は目標単語$w_t$を中心とした長さ$l+1+r$の単語列$W^t$[式(5)]から得られた情報で構成されるもので，以下のように表す．

$$IPT = (ipt_{t-l}, \cdots, ipt_t, \cdots, ipt_{t+r}) \qquad (6)$$

但し，入力の長さ$l+1+r$は固定なものではなく，後で述べるように伸縮性をもつものとされる．具体的に，単語$w$が入力の位置$x$ $(x = t-l, \cdots, t+r)$に与えられた時，$IPT$の構成部分である$ipt_x$は以下のように重み付けされたパターンで定義される．

$$\begin{aligned} ipt_x &= g_x \cdot (e_{w1}, e_{w2}, \cdots, e_{w\gamma}) \\ &= (I_{w1}, I_{w2}, \cdots, I_{w\gamma}) \end{aligned} \qquad (7)$$

ここで，$g_x$は次節で述べるインフォメーションゲインで[(17)-(20)を用いて]求められる重み，$\gamma$は品詞の数，$I_{wi} = g_x \cdot e_{wi}$ $(i = 1, \cdots, \gamma)$である．もし単語$w$が既知のもの，即ち，訓練データに出現するならば，各要素$e_{wi}$は以下のように得られる．

$$e_{wi} = Prob(\tau^i|w) \qquad (8)$$

但し，$Prob(\tau^i|w)$は単語$w$の品詞が$\tau^i$である確率で，訓練データから以下のように推定される．

$$Prob(\tau^i|w) = \frac{C(\tau^i, w)}{C(w)} \qquad (9)$$

ここで，$C(\tau^i, w)$は全訓練データを通じ，$w$が品詞$\tau^i$を取る回数で，$C(w)$は$w$が出現する回数である．一方，もし単語$w$が未知のもの，即ち、訓練データに出現しないならば，各要素$e_{wi}$は以下のように得られる．

$$e_{wi} = \begin{cases} \frac{1}{\gamma_w} & \tau^i \text{が} w \text{の取りうる品詞の場合} \\ 0 & \text{その他} \end{cases} \qquad (10)$$

ここで，$\gamma_w$は単語$w$が持ちうる品詞の数である．出力$OPT$は以下のように定義されるパターンである．

$$OPT = (O_1, O_2, \cdots, O_\gamma) \qquad (11)$$

但し，$OPT$は以下のようにデコードされるものとする．

$$\tau(w_t) = \begin{cases} \tau^i & O_i = 1, \text{かつ} \\ & \text{すべての} O_j = 0 \ (j \neq i) \text{の場合} \\ \text{Unknown} & \text{その他} \end{cases} \qquad (12)$$

ここで$\tau(w_t)$は単語$w_t$へのタグづけ結果を表す．

新しいニューロタガーに前に提案したマルチニューロタガーの機能を継承させるために，ニューロタガーの入力に伸縮性を持たせた．具体的には，はじめに入力$IPT$の長さ$l+1+r$を最大に設定し，タグづけを行なう．タグづけの結果$\tau(w_t)$がUnknownならば，入力$IPT$の長さ$l+1+r$を一定の間隔で縮小してタグづけ処理をもう一度行なう．この処理は$\tau(w_t)$がUnknownでなくなるか，入力の長さ$l+1+r$が1（即ち，入力は目標単語のみ）に縮むか，になるまで繰り返される．このような入力の伸縮性の導入は，マルチモジュールシステムの使用と同様，結果の確信度を高めるために，タグづけをまずできるだけ長い文脈を用いて行ない，訓練データの不足から確定的な答えが出ない場合に順次文脈を短くするといったように長さ優先可変文脈を用いることを意味する．

文の各単語を左から右へ順にタグづけしていくとき，左側の単語はつねにタグづけ済みと考えられるため，それらの単語に関する入力を構成するとき，より多くの情報が活用できる．具体的には，(7)-(10)を用いる代わりに，入力は次のように構成される：

$$ipt_{t-i} = g_{t-i} \cdot OPT(-i) \qquad (13)$$

但し，$i = 1, \cdots, l$．ここで，$OPT(-i)$は$i$個前の単語に対してタガーが出した出力を表す．しかしながら，訓練過程においてはタガーの出力はまだ正確ではないため，それらを直接入力にフィードバックして使うことができない．そのために，訓練過程における入力は以下のように実際の出力と目標出力の重みつき平均を用いて構成する：

$$ipt_{t-i} = g_{t-i} \cdot (w_{OPT} \cdot OPT(-i) + w_{DES} \cdot DES) \qquad (14)$$

ここで，$DES$は目標出力で，$w_{OPT}$と$w_{DES}$はそれぞれ次のように定義される：

$$w_{OPT} = \frac{E_{OBJ}}{E_{ACT}} \qquad (15)$$

$$w_{DES} = 1 - w_{OPT} \qquad (16)$$

ここで，$E_{OBJ}$と$E_{ACT}$はそれぞれ目標誤差と実際の誤差（詳細は3.3節で述べる）を表す．従って，訓練の始めの入力構成では目標出力の比重が大きく，時間が立つにつれゼロへ減っていく．逆に，実際の出力の比重は最初小さく，時間が立つにつれて大きくなっていく．

### 3・2 インフォメーションゲイン（IG）

インフォメーションゲイン（IG）は，特徴ベクトルで定義されるデータセットの情報量がある特定の特徴の値を知ることによってどれだけ増えるかを表す量である[Daelemans 92, Quinlan 93]．より具体的に言えば，ある特徴のIGとはその特徴がデータのクラス同定にどれだけ重要かを反映する量である．ここで，特徴を入力

の構成部分，特徴の値をその構成部分の取りうる品詞，データの属するクラスを目標単語の取りうる品詞にそれぞれ置き換えてやれば，各構成部分のIGはその構成部分の品詞タグづけへの影響度として考えることができる．

従って，式(6)における入力の各構成部分$ipt_x$（$x = t-l, \cdots, t, \cdots, t+r$）は式(7)にあるようなそれぞれタグづけへの影響度に応じた重み$g_x$を持つと仮定すれば，その重みは以下のように求められる．ここで全訓練データのセットを$S$，$i$番目のクラス，即ち，$i$番目の品詞（$i = 1, \cdots, \gamma$）を$C_i$，$C_i$に属するデータのセットを$S_{C_i}$で表す．セット$S$のエントロピー，即ち，$S$の中の一つのデータのクラス（品詞）を同定するのに必要とされる情報の平均量は

$$info(S) = -\sum_{i=1}^{\gamma} \frac{C(S_{C_i})}{C(S)} \times \ln \frac{C(S_{C_i})}{C(S)} \qquad (17)$$

である．但し，$C(X)$はセット$X$の中のデータの数を表す．セット$S$が構成部分$ipt_x$の持ちうる$h$個の品詞によって$h$個のサブセット$S_i$ $(i = 1, \cdots, h)$に分割されたとき，新しいエントロピーはこれらのサブセットのエントロピーの重みつき総和で求められる．即ち，

$$info_x(S) = \sum_{i=1}^{h} \frac{C(S_i)}{C(S)} \times info(S_i) \qquad (18)$$

この分割（即ち，構成部分$ipt_x$の品詞を知ること）による情報の増益（IG）は以下になる．

$$gain(x) = info(S) - info_x(S) \qquad (19)$$

従って，構成部分$ipt_x$のタグづけへの影響度に応じた重みは以下のように設定できる．

$$g_x = gain(x) \qquad (20)$$

### 3・3 三層パーセプトロン

伸縮性入力ニューロタガーを構成する三層パーセプトロン（図1，詳細は[Haykin 94]を参照）は品詞のタグづけ済みの訓練用データを誤差逆伝播学習アルゴリズム[Rumelhart 84]を用いて学習することによって品詞タグづけ能力を身につけることができる．

訓練段階においては，各訓練データのペア$(IPT^{(b)}, DES^{(b)})$は順番にネットワークに与えられる．但し，上つき記号$b$はデータの番号を表し，$b = 1, \cdots, P$，$P$は訓練データの数である．$b$番目の訓練データ$IPT^{(b)}$と$DES^{(b)}$は次のようなパターンとされる．

$$\begin{aligned} IPT^{(b)} &= (ipt_{t-l}^{(b)}, \cdots, ipt_t^{(b)}, \cdots, ipt_{t+r}^{(b)}) \\ &= (I_1^{(b)}, \cdots, I_p^{(b)}) \\ DES^{(b)} &= (D_1^{(b)}, \cdots, D_\gamma^{(b)}) \end{aligned} \qquad (21)$$

但し，$p = (l+1+r) \cdot \gamma$である．$IPT^{(b)}$の各要素$I_i^{(b)}$は(7)-(10)（或は(13)-(16)）及び(17)-(20)を用いて得られる．$DES^{(b)}$の各要素$D_i^{(b)}$は次のように与えられる．

$$D_i^{(b)} = \begin{cases} 1 & \tau^i \text{が正解の場合} \\ 0 & \text{その他} \end{cases} \qquad (22)$$

訓練は全訓練データに対し平均出力誤差が目標値以下になるまで繰り返して行なわれる．ここで$k$回目の繰り返し訓練において$b$番目の訓練データのペアが提示されたとする．その時，ネットワークは，入力層に与えられた入力パターン$IPT^{(b)}$を以下の(23)-(27)を用いて出力層へ前向きに伝播しながら変換する．入力層はまず以下のようにセットされる．

$$y_i(k) = I_i^{(b)} \text{ and } y_0(k) = 1 \qquad (23)$$

但し，$i = 1, \cdots, p$, $y_i(k)$は入力層のユニット$i$の出力，$i = 0$はバイアスユニットを表す．ここで，次の層（中間層或は出力層）のユニット$j$の出力は次のように得られる．

$$y_j(k) = \phi(v_j(k)) \qquad (24)$$

但し，$v_j(k)$はユニット$j$の内部活動度と呼ばれるもので次のように得られる．

$$v_j(k) = \sum_{i=0}^{q} w_{ji}(k) y_i(k) \qquad (25)$$

ここで，$y_i(k)$と$q$はそれぞれ前の層のユニット$i$の出力とユニットの総数で（入力層においては$q = p$），$w_{ji}$は（前の層の）ユニット$i$から（次の層の）ユニット$j$への結合強度である．また，$\phi(x)$は次のように定義される．

$$\phi(x) = \frac{1}{1 + exp(-x)} \qquad (26)$$

出力層のユニット$j$の出力には別の記号$O_j^{(b)}$を用いる，即ち，

$$O_j^{(b)}(k) = y_j(k) \qquad (27)$$

出力$O_j^{(b)}(k)$が得られた後，その出力と目標出力間の二乗誤差$E^{(b)}$は次のように計算される．

$$E^{(b)}(k) = \frac{1}{2} \sum_{j=1}^{\gamma} (D_j^{(b)} - O_j^{(b)}(k))^2 \qquad (28)$$

そこで，その誤差を出力層から入力層へ逆伝播して誤差を減らすようにネットワークの重みを次のように修正する．

$$w_{ji}(k+1) = w_{ji}(k) + \eta \Delta w_{ji}(k) + \alpha \Delta w_{ji}(k-1) \qquad (29)$$

但し，$\eta$ は重みの更新量を決める学習率で，$\alpha$ は慣性率である．$\Delta w_{ji}(k)$ は最急降下法で次のように計算される．

$$\Delta w_{ji}(k) = -\frac{\partial E^{(b)}(k)}{\partial w_{ji}(k)} \tag{30}$$

このように式(23)-(30)を通じて $k$ 回目の繰り返し訓練においての $b$ 番目のデータの処理が終る．訓練は，下の条件が満足されるまで，即ち，各訓練データと各出力ユニットに対する平均誤差 $E_{ACT}(k)$ が目標誤差 $E_{OBJ}$ 以下になるまで，全訓練データを通じて繰り返して行なわれる:

$$E_{ACT}(k) = \frac{\sum_{b=1}^{P}\sum_{j=1}^{\gamma}|D_j^{(b)} - O_j^{(b)}(k)|}{\gamma P} \leq E_{OBJ} \tag{31}$$

品詞タグづけ段階においては，入力 $IPT = (I_1, \cdots, I_p)$ が与えられた時，ネットワークはその入力パターンを式 $(23) - (27)$ を用いて入力層から出力層へ前向きに伝播しながら変換する．ここで $k$ は 1 にセットされ，上つき記号 $b$ が取り除かれる．最終的に，出力 $OPT = (O_1, \cdots, O_\gamma)$ が (27) の代わりに次のように得られる．

$$O_j = \psi(y_j(1) - \theta) \tag{32}$$

ここで $\theta$ は出力の閾値で，$\psi(x)$ は以下のように定義される．

$$\psi(x) = \begin{cases} 1 & x > 0 \text{ の場合} \\ 0 & \text{その他} \end{cases} \tag{33}$$

### 3・4 訓　　練

入力を縮めてもタグづけに同じパラメータ（ユニット間の結合強度: $w_{ij}$）セットを用いられるようにするために，以下のような訓練方法を取った．ここで，伸縮性入力を持つニューロタガーは，最小入力を持つネットワーク（入力層ー中間層ー出力層にそれぞれ $N_{Imin} - N_{Hmin} - N_O$ 個のユニットを持つ三層パーセプトロン）から成長してきたもの（入力層ー中間層ー出力層にそれぞれ $N_{Imax} - N_{Hmax} - N_O$ 個のユニットを持つ三層パーセプトロン）としてとらえる．但し，$N_{Imin}$ と $N_{Imax}$ はそれぞれ最小入力と最大入力を持つネットワークの入力層のユニット数，$N_{Hmin}$ と $N_{Hmax}$ はそれぞれ最小入力と最大入力を持つネットワークの中間層のユニット数，そして $N_O$ はすべてのネットワークの出力層のユニット数である．従って，伸縮性入力ニューロタガーの訓練は小さいパーセプトロンの訓練から最大パーセプトロンの訓練への漸進型過程として考える．具体的には，まず最小入力を持つパーセプトロン（例えば図 2 の実線の部分）を訓練する．訓練し終えた後，それを段階的に成長させ，新しいパーセプトロンを形成する（例えば図 2 の点線を含む全体）．それから，その新しいパーセプトロンの訓練を行なう．そのとき，前の訓練で得たネットワークの結合強度のセット（例えば図 2 の $w_1$ と $w_2$）はそのまま初期値として用いられる[*1]．

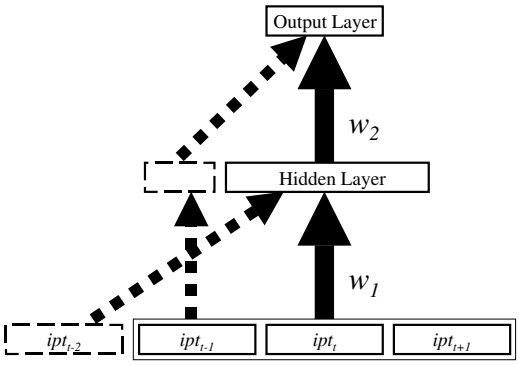


図 2 伸縮性入力ニューロタガーの訓練

### 3・5 比　　較

伸縮性入力ニューロタガーは，前に提案したマルチニューロタガーに比べその構造が複数のニューラルネットから単一のニューラルネットへ大幅にスリム化された．それに伴い必要とされる記憶量が大きく減少した．例えば品詞が 50 種類ある言語を最大左右それぞれ三つの単語の情報を文脈としてタグづけを行なう場合を考えよう．ここですべてのニューラルネットは中間層のユニット数が入力層の半分であるような三層パーセプトロンで構成されると仮定する．マルチニューロタガーに必要とされるパラメータ（ユニット間の結合強度）の数はそれを構成する五つの長さ $l + 1 + r$ がそれぞれ 3 から 7 まで固定長入力のニューラルネットのそれの総和，即ち，$\sum_{i=1}^{5} N_I^{(i)} \cdot N_H^{(i)} + N_H^{(i)} \cdot N_O^{(i)} = \sum_{i=1}^{5}[(i+2) \times 50 \times \frac{(i+2)\times 50}{2} + \frac{(i+2)\times 50}{2} \times 50] = 200,000$ である．ここで，$N_I^{(i)}$，$N_H^{(i)}$，と $N_O^{(i)}$ はそれぞれ $i$ 番目のニューロタガーの入力層，中間層，及び出力層のユニットの数，$(i+2)$ は $i$ 番目のニューラルネットの入力の長さ $l + 1 + r$，$N_H = \frac{N_I}{2}$ である（詳細は [馬, 99] を参照）．一方，伸縮性入力ニューロタガーに必要とされるパラメータの数は長さ $l + 1 + r$ が 7 の固定長入力のニューラルネットのそれと同様，$N_I \cdot N_H + N_H \cdot N_O = (7 \times 50) \times \frac{7\times 50}{2} + \frac{7\times 50}{2} \times 50 = 70,000$ である．

*1 この方法を用いてもパーセプトロンの性質から，$w_1$ と $w_2$ は新しい学習によって多少修正されるが，最初から最大入力のパーセプトロンを訓練する場合に比べその変動の程度は小さいと考えられる．

即ち，伸縮性入力ニューロタガーはそのユニット間の結線の数がマルチニューロタガーのそれの約1/3になる．ニューロタガーの構造のスリム化とユニット間の結線の数の大幅な減少は一般的な意味での記憶量の改善だけでなく，ニューロタガーの（学習の高速化・実用化のための）チップ化の上で特に重要な意味を持つと考えられる．

学習速度においては，マルチニューロタガーにおいても個々のニューラルネットの訓練はそれぞれ独立に行なわれるのではなく，短い文脈での訓練結果（訓練で獲得したユニット間の結合強度）を長い文脈での初期値として使うため，両者の学習速度はほとんど変わらない．処理速度においては，マルチニューロタガーの中の個々のニューラルネットをそれぞれ独立にタグづけを行なわせる代わりに，まずは長い入力のほうから試み，必要に応じ次に長い入力に移るといったように逐次的に行なわせれば，両者の処理速度もほとんど変わらない．

## 4. 確率モデル

品詞タグづけのメインアプローチとして，HMMを用いた手法がよく知られている．本節では，伸縮性入力ニューロタガーとの比較実験に用いたベースラインモデル，局所n-gramモデル，及びHMMについて述べる．

### 4・1 ベースラインモデル

頻度情報のみを用いたベースラインモデルは単語レベル[式(3)]で品詞タグづけを行なう手法である．目標単語$w_t$への品詞タグづけは以下のように表される．

$$\tau(w_t) = \arg\max_{\tau_t} Prob(\tau_t|w_t) \tag{34}$$

但し，$\tau_t$は目標単語の品詞候補である．もし単語$w_t$が既知のもの，即ち，訓練データに出現するならば$Prob(\tau_t|w_t)$は訓練データから，スムージングを行なう式(35)に示す加算法によって得られる．

$$Prob(\tau_t|w_t) = \frac{C(\tau_t, w_t)+1}{C(w_t)+\gamma} \tag{35}$$

但し，$\gamma$は品詞の数である．もし単語$w_t$が未知のもの，即ち，訓練データに出現しないなら$Prob(\tau_t|w_t)$は式(10)と同様以下のように得られる：

$$Prob(\tau_t|w_t) = \begin{cases} \frac{1}{\gamma_{wt}} & \tau_t \text{が} w_t \text{の取りうる品詞の場合} \\ 0 & \text{その他} \end{cases} \tag{36}$$

ここで，$\gamma_{wt}$は単語$w_t$が持ちうる品詞の数である．

### 4・2 局所n-gramモデル

提案手法との比較のために，本稿では局所的にn-gram確率を使う品詞タグづけモデルを特別に設定した．これはニューロタガーと同様，単語レベル[式(3)]で品詞タグづけを行なう手法である．n-gramテーブルの膨大さなどの問題から本稿ではn=3のtri-gramモデルを用いた．局所tri-gramモデルによる目標単語$w_t$への品詞タグづけは以下のように表される：

$$\tau(w_t) = \arg\max_{\tau_t} Prob(\tau_t|w_t)Prob(\tau_t|\tau_{t-1}, \tau_{t+1}) \tag{37}$$

但し，$\tau_t$と$\tau_{t+1}$はそれぞれ目標単語$w_t$と右の単語$w_{t+1}$の品詞候補で，$\tau_{t-1}$は（ニューロタガーと同様，左側の単語はつねにタグづけ済みと考えられるため）$w_{t-1}$へのタグづけされた品詞である．また，$Prob(\tau_t|\tau_{t-1}, \tau_{t+1})$は前後の単語$w_{t-1}$と$w_{t+1}$がそれぞれ品詞$\tau_{t-1}$と$\tau_{t+1}$を取った場合，目標単語$w_t$が品詞$\tau_t$を取る確率である．$Prob(\tau_t|w_t)$は訓練データから式(35)或は(36)を用いて得られるもので，$Prob(\tau_t|\tau_{t-1}, \tau_{t+1})$はスムージングを行なう式(38)に示す線形補間法[北 96]によって推定される．

$$Prob(\tau_t|\tau_{t-1}, \tau_{t+1}) = (1-\varepsilon)\frac{C(\tau_t, \tau_{t-1}, \tau_{t+1})}{C(\tau_{t-1}, \tau_{t+1})} + \frac{\varepsilon}{2}Prob(\tau_t|\tau_{t-1}) + \frac{\varepsilon}{2}Prob(\tau_t|\tau_{t+1}) \tag{38}$$

ここで，$C(\tau_t, \tau_{t-1}, \tau_{t+1})$は全訓練データを通じ，左右の単語$w_{t-1}$と$w_{t+1}$がそれぞれ品詞$\tau_{t-1}$と$\tau_{t+1}$を取った場合，目標単語$w_t$が品詞$\tau_t$を取る回数で，$C(\tau_{t-1}, \tau_{t+1})$は左右の単語$w_{t-1}$と$w_{t+1}$がそれぞれ品詞$\tau_{t-1}$と$\tau_{t+1}$を取る回数，$\varepsilon$はスムージングを行なうために導入した微小の定数でその最適な値を訓練データから自動的に求めることも可能ある．また，$Prob(\tau_t|\tau_{t-1})$と$Prob(\tau_t|\tau_{t+1})$はそれぞれ左右の単語$w_{t-1}$と$w_{t+1}$が品詞$\tau_{t-1}$と$\tau_{t+1}$を取った場合目標単語$w_t$が品詞$\tau_t$を取る確率で訓練データから以下のスムージング化の式で推定される．

$$Prob(\tau_t|\tau_{t-1}) = (1-\varepsilon)\frac{C(\tau_t, \tau_{t-1})}{C(\tau_{t-1})} + \varepsilon \tag{39}$$

或は

$$Prob(\tau_t|\tau_{t+1}) = (1-\varepsilon)\frac{C(\tau_t, \tau_{t+1})}{C(\tau_{t+1})} + \varepsilon \tag{40}$$

### 4・3 HMM

HMMは文レベル[式(4)]で品詞タグづけを行なう手法である．状態遷移をbi-gramを用いて定義したHMM[*2]による文$W = w_1 w_2 \cdots w_s$への品詞タグづけ

*2 状態遷移の定義にtri-gramを用いるほうがより一般的で，bi-gramより性能がよいという結果も報告されている[Nagata 94]．しかし，本稿のような訓練データが少ないタスクにおいては，データスパースネスの問題からbi-gramを用いた

は以下のように表される（詳細は [Charniak 93a, b] を参照）.

$$T(W) = \arg\max_{T} \prod_{i=1}^{s} Prob(\tau_i|\tau_{i-1}) Prob(w_i|\tau_i) \quad (41)$$

但し，$T(W) = \tau_1\tau_2\cdots\tau_s$. ベイズ定理で上式を（定数である $Prob(w_i)$ を省略して）次のように変換できる：

$$T(W) = \arg\max_{T} \prod_{i=1}^{s} Prob(\tau_i|\tau_{i-1}) \frac{Prob(\tau_i|w_i)}{Prob(\tau_i)} \quad (42)$$

但し，$Prob(\tau_i)$ は品詞 $\tau_i$ の頻度を表すもので訓練データから以下のように推定される.

$$Prob(\tau_i) = \frac{C(\tau_i)}{C(\Gamma)} \quad (43)$$

ここで $C(\tau_i)$ は全訓練データを通じ，品詞 $\tau_i$ が出現する回数，$C(\Gamma)$ は品詞セットの各品詞の出現する回数の総和である．また，$Prob(\tau_i|w_i)$ と $Prob(\tau_i|\tau_{i-1})$ は訓練データからそれぞれ式 (35)(36) と (39) を用いて推定される．実際，HMM による品詞タグづけ [式 (41)] は出力系列 $W$ が与えられた時，最適状態遷移系列 $T(W)$ を探索することである．このような探索はよく知られている Viterbi algorithm[Viterbi 67, Forney 73] を用いて行なわれる.

## 5. 実 験 結 果

実験用データはすでに品詞のタグづけがされたタイ語コーパスから得られた 10,452 の文であった．それを無作為に 8,322 文と 2,130 文に分けてそれぞれ訓練とテストに使った．訓練文においては 124,331 個の単語（うち，22,311 個の単語が複数の品詞を持つ），テスト文において 34,544 個の単語（うち，6,717 個の単語が複数の品詞を持つ）を有する．各種ニューロタガーと n-gram モデルの訓練には訓練文の中の品詞の曖昧性のある単語のみを用いた．HMM の訓練には訓練文の全単語を用いた．但し，いずれの場合でも各単語がそれぞれの品詞を取る頻度 $Prob(\tau^i|w)$ の推定 [式 (9) と (35)] には訓練文の全単語を用いた．ニューロタガーの訓練に曖昧性のある単語のみを用いた理由は訓練コストを考慮した結果で，局所 n-gram モデルの訓練に曖昧性のある単語のみを用いた理由はそのほうが結果がよかったからである.

場合と tri-gram を用いた場合の性能は同じぐらいであると推測される．簡単のため，本稿では bi-gram を用いることにした.

タイ語には 47 種類の品詞が定義されている [Charoenporn 97] ため，$\gamma$ [式 (2)] は 47 にセットされる．伸縮性入力ニューロタガーは入力層－中間層－出力層に $p - \frac{p}{2} - \gamma$ 個のユニットを持つ三層パーセプトロンで構成される．ここで，$p = \gamma\cdot(l+1+r)$．但し，$(l+1+r)$ は以下のように伸縮性をもつ．訓練段階においては，$(l,r)$ を (1,1) → (2,1) → (2,2) → (3,2) → (3,3) のように段階的に増加させ，小さいネットワークから大きいネットワークへ漸進的に訓練を行なう．一方，タグづけにおいては，逆に必要に応じて $(l,r)$ を (3,3) → (3,2) → (2,2) → (2,1) → (1,1) → (1,0) → (0,0) のように段階的に縮小していく．但し，タグづけにおいては，中間層のユニット数を最大のまま（即ち，$(l,r) = (3,3)$ に対応したもの）に固定した*3．各種ニューロタガーの出力の閾値 $\theta$[式 (32)] は 0.5 に設定した．また，重みの更新量を決める学習率 $\eta$ と慣性率 $\alpha$[式 (29)] はそれぞれ 0.1 と 0.9 に，訓練を止める基準である目標誤差 $E_{OBJ}$[式 (31)] は 0.005 に設定した．訓練セットから得られた各入力部分の重み [式 (7)] は $(g_{t-3},\ g_{t-2},\ g_{t-1},\ g_t,\ g_{t+1},\ g_{t+2},\ g_{t+3}) = (0.575,\ 0.524,\ 0.749,\ 2.667,\ 0.801,\ 0.575,\ 0.649)$ であった．確率モデルに用いた定数 $\varepsilon$[式 (38)-(40)] は 0.0001 にセットした．品詞タグづけの精度は以下のように定義される.

$$accuracy = \frac{C(words\ tagged\ correctly)}{C(all\ ambiguous\ words)}$$

即ち，ここでいう精度は品詞に曖昧性のある単語のみを対象に測定されるものである.

表 1 は各種ニューロタガーによる品詞タグづけの精度を示す*4．マルチニューロタガーと同様，伸縮性入力ニューロタガーは IG の有無とは関係なく，その精度はどの固定長入力のニューロタガーのそれよりも高かった．従って，この実験結果から見れば伸縮性入力ニューロタガーを用いることによっても，よさそうな文脈の長さを事前に経験的に選ぶ必要がなく，タグづけ過程において動的に選択できることを示している．この表はまた，伸縮性入力ニューロタガーはマルチニューロタガーと同等以上のタグづけ精度を持っていることを示した．そして，マルチニューロタガーと同様，IG を用いる場合，伸縮性入力ニューロタガーの精度は IG を用いない場合と比べほんの僅かしか上がらなかった．しかし，IG を

*3 実際，中間層のユニット数を入力の長さに応じて変化させる方法を用いてもほぼ同じ実験結果が得られた.

*4 この表に示している固定長入力ニューロタガーの精度は 3.5 節で述べた新しい訓練方法を用いて得られたもので，個々のタガーを本来の方法で独立に訓練した場合，長い入力のニューロタガーの精度はもう少し低下する．詳細については [馬 99] を参照されたい.

表1 各種ニューロタガーによる品詞タグづけ精度

| | 固定長入力ニューロタガー | | | | | マルチニューロタガー | 伸縮性入力ニューロタガー |
|---|---|---|---|---|---|---|---|
| 入力の長さ | 3 | 4 | 5 | 6 | 7 | | 7 → 1 |
| IG あり | 0.915 | 0.920 | 0.929 | 0.930 | 0.933 | 0.943 | 0.944 |
| IG なし | 0.924 | 0.927 | 0.922 | 0.926 | 0.926 | 0.941 | 0.943 |

表2 伸縮性入力ニューロタガーと確率モデルとの比較実験結果

| ベースラインモデル | 局所 tri-gram モデル | HMM | 伸縮性入力ニューロタガー |
|---|---|---|---|
| 0.836 | 0.893 | 0.891 | 0.944 |

用いることによって伸縮性入力ニューロタガーの訓練時間は大幅に短縮された．例えば最大入力（入力長さ7）のパーセプトロンへの逐次訓練は，IG を用いない場合30回の繰り返し訓練が必要だったが，IG を用いた場合は僅か13回で済んだ．

表2は伸縮性入力ニューロタガーと種々の確率モデルとの比較実験結果を示している．この表によって提案するニューロタガーは局所 tri-gram モデルや HMM を用いた確率手法より優れていることが確認された．厳密にいえば，この結果はニューラルネットと確率モデルを同一条件（即ち，同じ長さの入力を用いた場合）で比較して得られたものではない．従って，表2に示しているのはニューラルネットと確率モデル一般についてのものではなく，あくまでも提案するニューロタガーと従来の（一般的な品詞タグづけ用）確率モデルとの比較結果に過ぎない．ここでもう一つ注意されたいのは，HMM の状態遷移確率は本来 Baum-Welch algorithm [Baum 72] を用いて繰り返し訓練で推定されるものである．しかし，実験の結果，繰り返し訓練なし，即ち，初期値（即ち，訓練データから得られた bi-gram テーブル）をそのまま用いたほうがタグづけの精度が一番高かった[*5]ため，それを用いた．

最後に注意されたいのは，これまでの文献では品詞タグづけの精度としてよくテストデータの全単語（品詞の曖昧性のありなし問わず）を対象に測定して得たものを用いている．もしこのような定義を用いるならば，伸縮性入力ニューロタガーの精度は 98.9% に達する．

## 6. 結　　び

情報量最大を考慮し最長文脈優先に基づいて長さ可変文脈で品詞タグづけを行うマルチニューロタガーの特徴をそのまま継承する伸縮性入力を持つ単一のパーセプトロンで構成するニューロタガーを提案した．計算機実験で新しいタガーはマルチニューロタガーと同等以上のタグづけ精度を持つことが確認された．伸縮性入力ニューロタガーの性能を他の手法との比較実験を通じ系統的に評価を行なった．その結果，小規模コーパスを用いた場合，伸縮性入力ニューロタガーが局所 tri-gram モデルや HMM より優れていることが確認された．

但し，本稿で取り上げたタイ語は日本語と同様に，わかち書きされないため，品詞タグづけと共に単語分割も重要な問題である．HMM は容易にこの問題に応用できる [Nagata 94] のに対し，提案する手法では対処できていないのが現状である．従って，如何にして同じニューラルネットの枠組で単語分割問題や本稿では取り扱っていない未知語問題をも有効に取り扱えるようにするかは今後の研究課題となる．

*5　この結果は英語の場合 [Merialdo 94] と一致している．

## ◇ 参 考 文 献 ◇


[Baum 72] L. E. Baum: "An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process." *Inequalities* No. 3, pp. 1-8 (1972).

[Brill 92] E. Brill: "A simple rule-based part-of-speech tagger." *Proc. 3rd ACL Applied NLP*, Trento, Italy, pp. 152-155 (1992).

[Charniak 93a] E. Charnik, c. Hendrickson, N. Jacobson, and M. Perkowitz: "Equations for part-of-speech tagging." In *Proc. 11th National Conference on Artificial Intelligence*. AAAI Press/MIT Press, Menlo Park, pp. 784-789 (1993).

[Charniak 93b] E. Charniak: *Statistical language learning*, The MIT Press (1993).

[Charoenporn 97] T. Charoenporn, V. Sornlertlamvanich, and H. Isahara: "Building a large Thai text corpus - part of speech tagged corpus: ORCHID." *Proc. Natural Language Processing Pacific Rim Symposium 1997*, Phuket, Thailand, pp. 509-512 (1997).

[Church 88] K. Church: "A stochastic parts program and noun phrase parser for unrestricted text." *Proc. 2nd ACL Applied NLP*, Austin, Texas, pp. 136-143 (1988).

[Cutting 92] D. Cutting, J. Kupiec, J. Pederson, and P. Sibun: "A practical part of speech tagger." *Proc. 3rd ACL Applied NLP*, Trento, Italy, pp. 133-140 (1992).

[Daelemans 92] W. Daelemans and A. Van den Bosch: "Generalisation performance of backpropagation learn-

ing on a syllabification task." In M. Drossaers & A. Nijholt (Eds.), *TWLT3: Connectionism and Natural Language Processing*. Enschede: Twente University, pp. 27-38 (1992).

[Daelemans 96] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis: "MBT: A memory-based part of speech tagger-generator." *Proc. 4th Workshop on Very Large Corpora*, Copenhagen, Denmark, pp. 1-14 (1996).

[DeRose 88] S. DeRose: "Grammatical category disambiguation by statistical optimization." *Computational Linguistics*, Vol. 14, No. 1, pp. 31-39 (1988).

[Forney 73] G. D. Forney, Jr: "The viterbi algorithm." *Proceedings of The IEEE*, Vol. 61, No. 3, pp. 268-278 (1973).

[Garside 87] R. Garside, G. Leech, and G. Sampson: *The computational analysis of English: A corpus-based approach*, London: Longman (1987).

[Haykin 94] S. Haykin: *Neural Networks*, Macmillan College Publishing Company, Inc. (1994).

[Hindle 89] D. Hindle: "Acquiring disambiguation rules from text." *Proc. ACL'89*, Vancouver BC, pp. 118-125 (1989).

[北 96] 北，中村，永田：“音声言語処理－コーパスに基づくアプローチ－”，森北出版，1996.

[Ma 96] Q. Ma, H. Isahara, and H. Ozaku: "Automatic part-of-speech tagging of Thai corpus using neural networks", *Artificial Neural Networks – ICANN'96*, Lecture Notes in Computer Science 1112, C. von der Malsburg et al (Eds.), Springer, pp. 275-280 (1996).

[Ma 98] Q. Ma and H. Isahara: "A multi-neuro tagger using variable lengths of contexts", *Proc. COLING-ACL'98*, Montreal, pp. 802-806 (1998).

[馬 98] 馬，井佐原：“長さ可変文脈を用いたマルチニューロタガー”，自然言語処理，vol. 6, No. 1. pp. 29-42, 1999.

[Marquez 97] L. Marquez and L. Padro: "A flexible POS tagger using an automatically acquired language model." *Proc. ACL-EACL'97*, Madrid, Spain, pp. 238-252 (1997).

[Merialdo 94] B. Merialdo: "Tagging English text with a probabilistic model." *Computational Linguistics*, Vol. 20, No. 2, pp.155-171 (1994).

[Nakamura 90] M. Nakamura, K. Maruyama, T. Kawabata, and K. Shikano: "Neural network approach to word category prediction for English texts." *Proc. COLING'90*, Helsinki University, pp. 213-218 (1990).

[Nagata 94] M. Nagata: "A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm," *Proc. COLING'94*, pp. 201-207, 1994.

[Quinlan 93] J. Quinlan: *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann (1993).

[Ratnaparkhi 96] A. Ratnaparkhi: " A maximum entropy model for part-of-speech tagging." In *Proc. Conf. on Emprical Methods in Natural Language Processing*. University of pennsylvania, pp. 133-142 (1996).

[Rumelhart 84] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group: *Parallel Distributed Processing*, the MIT Press (1984).

[Schmid 94] H. Schmid: "Part-of-speech tagging with neural networks." *Proc. COLING'94*, Japan, pp. 172-176 (1994).

[Schütze, 94] H. Schütze and Y. Singer: "Part-of-speech tagging using a variable memory markov model." *Proc. ACL'94*, Las Cruces, New Mexico, pp. 181-187 (1994).

[Viterbi 67] A. J. Viterbi: "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm." *IEEE Transactions on Information Theory*, Vol. 13, No. 2, pp. 260-269 (1967).

[Weischedel 93] R. Weischedel, et al.: "Coping with ambiguity and unknown words through probabilistic models." *Computational Linguistics*, Vol. 19, No. 2, pp. 359-382 (1993).

〔担当編集委員：××○○，査読者：○○××〕