ニューラルネットとルールベース手法を統合した品詞タグづけシステム

馬 青　内元 清貴　村田 真樹　井佐原 均
郵政省通信総合研究所
{qma, uchimoto, murata, isahara}@crl.go.jp

1 はじめに

これまで，できるだけ少量の訓練データで実用的な品詞タグづけシステムを構築する目的で著者らは複数のニューラルネットで構成する，情報量最大を考慮し最長文脈を優先するマルチニューロタガー（馬，井佐原，1999）を提案した．しかしながら，ニューラルネットを用いた手法は統計的手法と同様，「統計的」に解析を行なうもので，「確実」な規則を取り扱うことが困難である．例えばある単語の品詞が前の単語のみによって「確実」に決まると仮定しよう．この場合でも，ニューラルネットはあくまでも文脈全体の下での可能性に基づいて「統計的」に解析を行なう．その結果，前の単語が同じでも全体の文脈が変わると，タグづけ結果が変わってしまう可能性がある．また，ニューラルネットがコーパスから学習容易な規則は，基本的にその条件部が入力の各要素の論理積で構成されるもので，論理和，単項，或は単語で構成されるような条件部を持つ規則を取り扱うことが困難である．

本研究では，ニューラルネット手法のこのような弱点を補うために書き換え規則を後処理[1]に導入し，ニューラルネットとルールベースの統合システムを構築した．ここで用いるニューラルネットは，単一のニューラルネットでありながら，可変長の入力に対応するように構成した「伸縮型ニューロタガー」であり，従来のマルチニューロタガーの持つ，可変長文脈を取り扱えるという特徴をそのまま継承し，かつスリム化したものである．ニューロタガーの後処理に用いた書き換え規則は，ニューロタガーが学習困難とされるものを補うようなテンプレートを用いて誤り駆動型学習（Brill, 1994）により訓練データから自動獲得される．計算機実験の結果，小規模コーパスを訓練に用いた場合，伸縮型ニューロタガーはマルチニューロタガーと同等以上，また，HMM（89.1%）より遥かに高い精度（94.4%）で未訓練データ（複数品詞を持つ単語のみ）をタグづけできた．更に，書き換え規則を後処理に用いることによってタグづけのエラーは19.1%減少し，小規模コーパスを訓練に用いても全体の統合システムのタグづけ精度は95.5%まで向上した．

[1]書き換え規則を後処理に用いる考え方は（Brill, 1994）によって初めて提案されたもので，日本語の形態素処理においては最近ルールベース手法の後処理に書き換え規則を用いる研究（久光，丹羽，1998）がある．

2 品詞タグづけ問題

本稿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 $W = w_1 w_2 \cdots w_s$ が与えられた時，以下の手続き φ によって品詞列 $T = \tau_1 \tau_2 \cdots \tau_s$ を見つけることと考えられる．

$$\varphi : W^t \rightarrow \tau_t,\ t = 1, \cdots, s \qquad (1)$$

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，

$$W^t = w_{t-l} \cdots w_t \cdots w_{t+r} \qquad (2)$$

但し，$t-l \geq 1, t+r \leq s$．従って，品詞タグづけは品詞をクラスに置き換えたクラス分け問題として捉えることができ，ニューラルネットで取り扱うことができる．

3 統合システム

提案する統合システムの構成（概念図）を図1に示す．

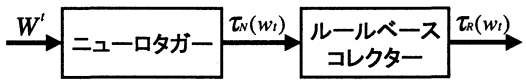

図1 統合システム

3.1 ニューロタガー

ニューロタガーは，図2のように単一の三層パーセプトロンで構成される．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，品詞タグづけを可変長文脈で行うことができる．

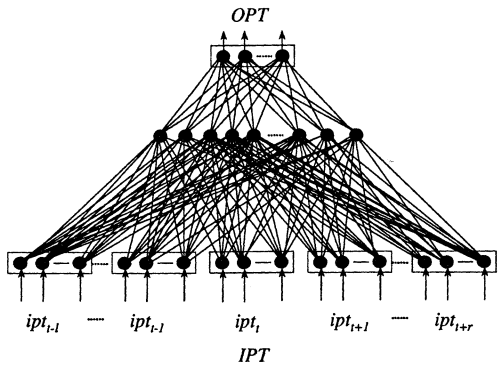

図2 ニューロタガー

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(2)]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．

$$IPT = (ipt_{t-l}, \cdots, ipt_t, \cdots, ipt_{t+r}) \tag{3}$$

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．

$$ipt_x = g_x \cdot (e_{w1}, e_{w2}, \cdots, e_{w\gamma}) \tag{4}$$

ここで，g_xはインフォメーションゲインで求められる重み，γは品詞の数，e_{wi}は単語wの品詞が$\tau^i (i = 1, \cdots, \gamma)$である事前確率で訓練コーパスから求められるものである（詳細は馬，井佐原，1999を参照）．出力OPTは以下のように定義されるパターンである．

$$OPT = (O_1, O_2, \cdots, O_\gamma) \tag{5}$$

但し，OPTは以下のようにデコードされるものとする．

$$\tau_N(w_t) = \begin{cases} \tau^i & O_i = 1, \text{かつ すべての} \\ & O_j = 0\ (j \neq i) \text{の場合} \\ Unknown & \text{その他} \end{cases} \tag{6}$$

ここで$\tau_N(w_t)$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, r)を最大に設定し，タグづけを行なう．タグづけの結果$\tau_N(w_t)$が$Unknown$ならば，入力IPTの長さ(l, 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$\tau_N(w_t)$が$Unknown$でなくなるか，$(l, r) = (0, 0)$（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．このように入力を縮めた場合に，各入力長に対して，それぞれの重み（ユニット間の結合強度）セットを学習することは効率的でない．そこで，ニューロタガーの訓練は，まず，最短の入力長に対応するパーセプトロンについて訓練することから始める．次に，この学習結果を初期値として，入力長を1段階増やしたパーセプトロンについて訓練を行う．これを繰り返し，最長の入力長に対応するものにまで，漸進的に訓練を進めていく．これにより，このニューロタガーにおいては，短い入力長に対応する部分の重みは，短い入力長での学習の影響を残しており，短い入力長の場合にもニューロタガーの一部を用いて，タグ付けを適切に行える．

ニューロタガーは以下のような特徴を持つ．例えば品詞が50種類ある言語をtri-gramベースの確率モデルでタグづけを行なう場合，$50^3 = 125,000$個のtri-gram（パラメータ）を同定しなければならない．それに対し，5-gramに相当するニューロタガーでも，その中間層のユニット数が入力層の半分とすれば，必要とされるパラメータ（ユニット間の結合強度）の数は37,500である．従って，限られたメモリの場合ニューロタガーのほうがより長い文脈を使用することが可能になる．また，一般的に，システムに必要とされるパラメータの数が少なければ，それらを正しく同定するのに必要な訓練データの数も少なくてよい．そのために，ニューラルネットのタグづけ性能は確率モデルのそれに比べ訓練データの数の少なさに影響されにくい（Schmid, 1994）．

3.2 ルールベースコレクター

前にも述べたように，ニューロタガーがコーパスからの学習で獲得しやすいのが基本的に論理積関係，即ち，$(ipt_{t-l}\ \&\ \cdots\ ipt_t\ \&\ \cdots\ ipt_{t+r} \rightarrow OPT)$のような規則であり，論理和関係の規則，例えば，$(ipt_x \parallel ipt_y \rightarrow OPT)$，単項式関係の規則[2]，例えば，$(ipt_x \rightarrow OPT)$，そして単語そのものを条件とする規則，例えば，$(w \rightarrow OPT)$や$(w_1\ \&\ w_2 \rightarrow OPT)$，を獲得するのが困難[3]である．ニューロタガーのこのような弱点を補うために，書き換え規則に基づく修正器を後処理として導入する．

書き換え規則はテンプレートを用いて訓練用コーパスから獲得する．テンプレートは前述した論理和関係などニューロタガーが獲得困難とされる規則を得られるように設計される．テンプレートの抜粋を表1，規則の学習

[2] 3.1節にも述べたように，(l, r)を(0,0)へ縮小する場合，ニューロタガーが単独の入力ipt_tでタグづけを行なうことになる．その意味では，ニューロタガーも単項式関係を扱える．しかしながら，ここで言う単項式関係はより一般的な場合，即ち，単独の入力は任意のipt_x $(x = t-l, \cdots, t+r)$である．

[3] ニューラルネットは理論的に任意の関係が学習可能なので，ここで「困難」という言葉を用いる．

手続きを表2，そして，実際に獲得された規則の一部を表3に示す．

表1 書き換え規則のテンプレートの抜粋

タグ τ^a をタグ τ^b へ変更する，もし
（単項入力）
（入力は品詞）
1．左（右）の単語のタグが τ である
⋮
（入力は単語）
4．左（右）の単語が w である
5．二つ左（右）の単語が w である
（論理和入力）
（入力は品詞）
6．左の単語のタグが τ_1，或は，右の単語のタグが τ_2 である
⋮
（単語の論理積入力）
13．左（右）の単語が w_1 で，二つ左（右）の単語が w_2 である
14．左の単語が w_1 で，右の単語が w_2 である
（品詞と単語の論理積入力）
15．目標単語が w で，左（右）の単語のタグが τ である
16．左（右）の単語が w で，左（右）の単語のタグが τ である

表2 書き換え規則の学習手続き

1．ニューロタガーで訓練用コーパスをタグづけし，コーパスを更新する
2．タグづけた結果と正解を比較し，エラー箇所を見つける
3．個々のエラー箇所において，テンプレートとの照合を行ない，規則群を得る
4．最適な規則を $(cnt\_good - h \cdot cnt\_bad)$ が最大であるように選ぶ．但し， cnt_good: 間違ったタグを正しい方へ変更する数 cnt_bad: 正しいタグを間違った方へ変更する数 h: 規則を生成する厳格さを制御する重み
5．最適な規則を訓練コーパスへ適用し，コーパスを更新する
6．最適な規則を順序付書き換え規則のリストに付け加える
7．最適な規則がなくなる（$cnt\_good - h \cdot cnt\_bad \leq 0$）まで手順2から6まで繰り返す

タグづけしようとするコーパスが与えられたとき，まずそれをニューロタガーによってタグづけする．そしてタグづけされたコーパスが表2に示している学習手続きで獲得した順序付書き換え規則で修正される．その修正は個々の規則を順番にコーパスに適用してはコーパスを更新するといった繰り返し過程である．

4 実験結果

実験用データは（馬，井佐原，1999）と同様，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22,311個の単語が複数の品詞を持ち，テスト文においては6,717個の単語が複数の品詞を持ちえた．タイ語には47種類の品詞（Charoenporn et al., 1997）が定義されている．

ニューロタガー 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，$\gamma = 47$，$p = \gamma \cdot (l + 1 + r)$．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[4]．

ルールベースコレクター 表2の学習手続きに用いた規則生成の評価関数 $cnt\_good - h \cdot cnt\_bad$ のパラメター h を100に設定した．h は規則生成の厳しさを制御するものである[5]．h を大きく設定すると，cnt_bad の影響が大きくなり，少しでも間違いを生じるような規則は生成されにくくなる．本稿では，重み h を大きく設定（即ち，規則の生成を厳しく）したのは，ニューロタガーはすでに高い精度を持ち，ルールベースコレクターはあくまでも微調整のチューナという位置づけで用いられているからである．学習によって計524個の順序つき書き換え規則が得られた．表3はその最初の十個の規則を示す．

品詞タグづけの精度は以下のように品詞の曖昧性のある単語のみを対象として測定したもの（"*accuracy1*" と記す）とすべての単語を対象として測定したもの（"*accuracy2*" と記す）に分けて定義される．

$$accuracy1 = \frac{C(ambiguous\ words\ tagged\ correctly)}{C(all\ ambiguous\ words)}$$

$$accuracy2 = \frac{C(all\ words\ tagged\ correctly)}{C(all\ words)}$$

表4はテストデータへのタグづけ結果を示している．表には伸縮型ニューロタガーと統合システムの精度を示している他，これらとの比較のため，ベースラインモデル，HMM，そしてマルチニューロタガーのそれぞれの精度も示している．ここで言うベースラインモデルとは，文脈情報を使わず訓練コーパスから得られた個々の単語が取る品詞の頻度情報のみを用いてタグづけを行なうものである．

まず，品詞の曖昧性のある単語のみを対象として測定した精度（即ち，*accuracy1*）に注目する．書き換え

[4] 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．

[5] (Brill, 1994) では h のようなパラメターを用いず $cnt\_good - cnt\_bad$ を評価関数とした．

表3 最初の十個の書き換え規則

No.	From	To	Condition
1	Unknown	ADVN	左の単語のタグがXVAE、 或は、二つ左の単語のタグが NCMN、 或は、三つ左の単語の品詞がVACTである
2	PREL	RPRE	左の単語が<space>で、 右の単語がระดับである
3	PREL	RPRE	左の単語がอยู่ で、或は、 右の単語がเก่าである
4	XVMM	XVBM	左の単語がหรือである
5	VATT	ADVN	左の単語がถึงである
6	NCMN	RPRE	左の単語がผลで、或は、 右の単語がจอภาพである
7	VATT	VSTA	左の単語がสามัญである
8	PREL	RPRE	右の単語がระดับで、 二つ右の単語がความである
9	VATT	ADVN	目標単語がสูงสุดで、 左の単語のタグがNCMNである
10	NCMN	RPRE	左の単語がในで、 左の単語のタグがNCMNである

但し、 ADVN: Adverb with normal form, PREL: Relative pronoun, RPRE: Preposition, ...

表4 テストデータへの品詞タグづけ結果

	ベースラインモデル	HMM	マルチニューロタガー	伸縮型ニューロタガー	統合システム
*accuracy*1	0.836	0.891	0.943	0.944	0.955
*accuracy*2	0.969	0.979	0.989	0.989	0.991

規則を後処理に導入した統合システムの精度は95.5%で，伸縮型ニューロタガーのみを用いる場合（94.4%）より1.1%高く，統計モデルよりは遥かに高かった．書き換え規則は，実際，訓練データとテストデータに対し，ニューロタガーが出したそれぞれのエラー（計733個と376個）の87.6%と19.1%を修正することができた．また，伸縮型ニューロタガーの精度（94.4%）は僅かながらマルチニューロタガーのそれ（94.3%）より高かった．その精度はどの固定長入力のニューロタガーのそれ（詳細は馬，井佐原，1999を参照）よりも高い．従って，伸縮型ニューロタガーも，マルチニューロタガーと同様，文脈の長さを事前に経験的に選ぶ必要がなく，いつも状況に応じて適切な長さの文脈を自動的に選べる．一方，品詞タグづけの精度としてテストデータの全単語（品詞の曖昧性のありなし問わず）を対象として得たもの（即ち，*accuracy*2）を用いるならば，統合システムの精度は99.1%にも達した．

5 結び

伸縮型ニューロタガーとルールベースコレクターで構成される統合システムを提案した．伸縮型ニューロタガーはマルチニューロタガーの特徴を継承して情報量最大を考慮し動的に適切な長さの文脈でタグづけができる．一方，ルールベースコレクターはニューロタガーが獲得困難な規則を誤り駆動型学習で自動獲得し，ニューロタガーが生じたエラーを修正する．計算機実験の結果，伸縮型ニューロタガーはマルチニューロタガーと同等以上，また，HMMより遥かに高い精度でタグづけできた．さらに，書き換え規則を後処理に導入したことによりタグづけのエラーは19.1%減少し，小規模コーパスを訓練に用いても全体の統合システムのタグづけ精度は95.5%まで向上した．

参考文献

[1] 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.

[2] Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging, *Computational Linguistics*, Vol. 21, No. 4, pp. 543-565, 1994.

[3] 久光，丹羽：書き換え規則と文脈情報を用いた形態素解析後処理，NL研126-8, pp. 55-62, 1998-7.

[4] Schmid, H.: Part-of-speech tagging with neural networks, *Proc. of the Int. Conf. on Computational Linguistics*, pp. 172-176, 1994.

[5] Charoenporn, T., Sornlertlamvanich, V., and Isahara, H.: Building a large Thai text corpus - part of speech tagged corpus: ORCHID, Proc. Natural Language Processing Pacific Rim Symposium 1997, Thailand 1997.