NAIST-IS-MT9551119

修士論文

規則と確率モデルの統合による形態素解析

山下 達雄

1997年 2月 14日

奈良先端科学技術大学院大学
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

山下 達雄

指導教官： 松本 裕治 教授
西田 豊明 教授
伝 康晴 助教授

規則と確率モデルの統合による形態素解析*

山下 達雄

内容梗概

形態素解析は、構文解析・意味解析などの高度な自然言語処理の基盤として重要な技術である。今までに研究されてきた日本語形態素解析システムは大きく次の２つの流れに分けられる。一つは人手による優先規則を用いたシステムで、人間がさまざまな言語現象をおおまかに捉えて、規則などの形に抽象化した優先規則を用いるものである。これは今までの経験の蓄積であり、有効な資源といえる。しかし、このシステムには、例外的な規則を追加していくにつれ、保守・管理が人間の手には負えなくなってしまうという問題がある。もう一つは確率モデルに基づいた統計的手法により品詞タグ付きコーパスから学習されたパラメータを用いるシステムである。このシステムは、さまざまな言語現象を含む大規模な品詞タグ付きコーパスが存在すれば高精度の解析が可能である。しかし、実際にはそのような大規模コーパスはなかなか入手できない。

本研究では、貴重な資源である人手による優先規則を活かし、細かい言語現象を扱うのに適したコーパスからの学習による確率パラメータを補完するという手法を提案する。この手法により、これら二つの方法の、保守・管理、及び、コーパス不足の問題を克服し、形態素解析精度の向上を目指す。そして、この手法の有用性を、実験により示す。

キーワード

自然言語処理, 形態素解析, 優先規則, 確率モデル, コーパス

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 修士論文, NAIST-IS-MT9551119, 1997年 2月 14日.

Integration of Rule-Based and Stochastic Approaches to Morphological Analysis *

Tatuo YAMASITA

Abstract

Morphological analysis is a process that assigns a part-of-speech tag to each word in a sentence. It is an inevitable process in natural language processing. So far, two major approaches have been studied. One is the hand-crafted rule-based approach. Human observes various linguistic phenomena, and generalize them into the form of rules. However, it becomes harder and harder for human to maintain the whole rules as more and more exceptional rules are added. The other is the corpus-based probabilistic approach. This approach uses probabilities of part-of-speech tag sequences estimated from part-of-speech tagged corpus as parameters of morphological analyzer. This approach can achieve high accuracy, althogh it requires a large part-of-speech tagged corpus.

To solve these problems, I propose an approach that integrates the hand-crafted rules and the probabilities estimated from a small-size corpus. I show that the morphological analyzer employing this approach can achieve higher accuracy than those of previous approaches.

Keywords:

Natural Language Processing, Morphological Analysis, Preference Rule, Stochastic Model, Corpus

*Master's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9551119, February 14, 1997.

目 次

1. はじめに **1**

2. 形態素解析 **4**

2.1 優先規則による形態素解析 5

2.1.1 形態素の認識 5

2.1.2 形態素間の接続可能性の確認 5

2.1.3 優先規則による最適解の決定 6

2.2 確率モデルによる形態素解析 8

3. 形態素解析システム「茶筌」 **11**

3.1 システムの概要 11

3.1.1 形態素の構造 11

3.1.2 形態素解析処理に必要なデータ 13

3.1.3 解析アルゴリズム 16

3.2 「茶筌」への確率モデルの適用 16

3.2.1 品詞タグ付きコーパスからの確率値パラメータの学習 . . . 17

3.2.2 適用の際の問題点 19

3.2.3 確率値からコストへの変換 21

4. 規則と確率モデルの統合 **23**

4.1 統合処理の概要 23

4.2 人手によるコストの準確率値への変換 24

4.3 統合の際の問題点 26

5. 実験と考察 **27**

5.1 実験 27

5.1.1 実験に用いるデータ 27

5.1.2 評価尺度 28

5.1.3 予備実験 28

5.1.4 本実験 29
5.2 考察 29

6. 品詞タグ付きコーパス作成支援環境 **33**
6.1 グラフィカルユーザインターフェース 34
6.2 フィードバック 35
6.3 展望 36

7. まとめ **38**

謝辞 **39**

参考文献 **40**

図 目 次

1 形態素解析の簡単な例 4
2 形態素の認識 6
3 形態素解析におけるグラフ構造 7
4 最適解の選択 8
5 学習に用いる品詞パターンの例 18
6 品詞タグ付きコーパスの例 18
7 統合処理の流れ 24
8 形態素解析の例 31
9 実験結果 32
10 ViJUMAN 34
11 品詞タグ付けコーパス作成支援環境 37

1. はじめに

形態素解析は、構文解析・意味解析などの高度な自然言語処理の基盤として非常に重要な技術であり、従来盛んに研究がされてきた。しかし、形態素解析を高い精度で行うためには構文情報・意味情報の利用が不可欠であるため、近年、形態素解析システム単独での解析精度の向上よりも、構文・意味解析処理などとの統合といった研究に注目が集まりつつある [11][17]。

しかし、表層の情報のみを用いる形態素解析システムにもまだ改良の余地が残っている。限られた情報での形態素解析の限界を見極めることは重要である。さらに、情報検索や簡単な対話インターフェース、辞書の作成や検証のための品詞タグ付きコーパスの作成など、様々な分野において形態素解析システム単独での解析精度の向上が望まれている。

これらの理由により、本研究では、構文・意味情報を用いずに形態素解析システム単独での解析精度の向上を目指すことにする。

さて、今までに研究されてきた日本語形態素解析処理は大きく次の２つの流れに分けられる。

人手による優先規則を用いた形態素解析

> 優先規則というのは、人間がさまざまな言語現象をおおまかにとらえて、規則などの形に抽象化したものである。記述量や理解の容易さといった面において効率の良いものであるが、考慮外の現象がとらえきれていないことが多い。そのような例外的な現象を規則として追加・修正することによって、解析精度を向上させていくのが一般的である。しかし、それにつれて、規則は次第に複雑になり、保守・管理が人間の手には負えなくなってしまう。つまり、一つの規則を修正することによって他に悪影響が出るようになり、全体的な精度をあげていくのが難しくなる。

コーパスを用いた確率モデルによる形態素解析

> 近年大量のコーパスが整備されつつあり、確率モデルによる形態素解析も

盛んになっている [16][19][20][22][28]。さまざまな言語現象を含む大規模な品詞タグ付きコーパスがあれば、パラメータ推定によりそのコーパスに特化した高精度の形態素解析システムが容易に作成できる[1] 。

一般に形態素解析処理は分野依存性が高いといわれている。例えば、マニュアル、法律、医学などの専門分野では、語彙や文法がそれぞれに異なる。また、新聞、対話といった異なるメディアにおいても同じことが言える。故に、対象となる分野の品詞タグ付きコーパスからの学習結果を用いることにより、その分野における形態素解析精度を向上させることができる。

そのため、この方法は、学習対象として自分の扱いたい分野の品詞タグ付きコーパスが存在すれば非常に有用であるが、一般に、そのようなコーパスは入手することは困難である。

また、大規模なコーパスを学習に用いても、多くの言語現象の出現頻度には極端な偏りがあるため、低頻度の現象を捉えるためのパラメータを精度よく推定するのは難しいというデータスパースネス (data sparseness) の問題もある。

まとめると、以下のようになる。

	人手による優先規則を用いた形態素解析	コーパスを用いた確率モデルによる形態素解析
長所	人間の観点から言語現象をおおまかに抽象化できる。	実際のデータに基づき細かい言語現象に対処できる。
短所	規則の複雑化により、保守・管理が困難になる。	大規模な品詞タグ付きコーパスが必要。データスパースネスの問題。

本研究では、より高精度の形態素解析処理の実現を目指すために、これら二つの手法の長所を活かし、短所を補い合うための手法として、規則と確率モデルの統合を提案する。

[1]品詞タグのないコーパスでも、Baum-Welch アルゴリズム (Forward-Backward アルゴリズム)[2] を用いてパラメータ推定ができる。しかし、品詞タグ付きコーパスを用いた方法の方が、高い解析精度が得られる。

人間の観点からおおまかに捉えた言語現象 (人手による規則・コスト) と品詞タグ付きコーパスから学習される細かい言語現象 (確率パラメータ) を統合することにより、保守・管理の困難さ、コーパスの量、データスパースネスといった問題を克服できる。まず、人手による優先規則の調整は非常に困難であったが、この手法では、パラメータ学習に用いるコーパスの量や分野を変更するだけで良く、保守・管理の困難さの問題を克服することができる。そして、信頼性の低い低頻度の現象に関するパラメータは、人手による優先規則で補完することにより、データスパースネスの問題を克服することができる。また、学習に用いるコーパスが小さいために信頼性が低いパラメータも、人手による優先規則で補完すれば良く、コーパスの量に関する問題も克服することができる。

本論文の構成について説明する。

第 2章では、形態素解析一般、特に『人手による優先規則を用いた形態素解析』、『コーパスを用いた確率モデルによる形態素解析』について説明する。

第 3章では、本研究で用いる形態素解析システム「茶筌」[24] についての説明を行う。さらに、品詞タグ付きコーパスからのパラメータ学習の方法、「茶筌」への確率モデルの適用についても詳しく述べる。

第 4章では、人手による優先規則と、品詞タグ付きコーパスからの学習結果との統合の手法について説明する。

第 5章では、品詞タグつきコーパスを用いた実験と考察を行う。

第 6章では、応用として、本研究で提案する手法を用いた品詞タグ付きコーパス作成支援環境について述べる。

第 7章で、全体のまとめを行う。

2. 形態素解析

形態素解析とは、文字列として与えられた文から単語を認識し、それに品詞情報を付与するという処理である (図 1)。これは、構文解析・意味解析などの高度な自然言語処理の基盤として重要な技術である。

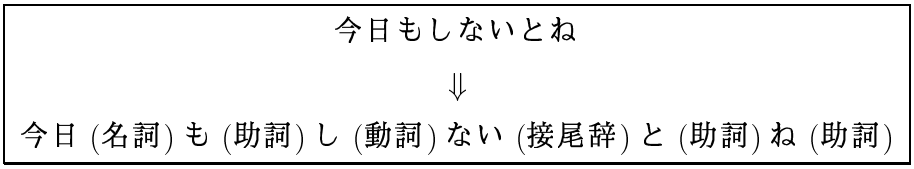


図 1 形態素解析の簡単な例

英語などのわかち書きされている言語の形態素解析において困難な点は品詞の曖昧性の解消、つまり、多品詞語に対していかに品詞を決定するかという点である[2]。英語では、品詞タグ付きコーパスが早い時期から存在したため、確率モデルに基づく方法が主流である。品詞タグ付きコーパスが与えられれば容易にパラメータ学習ができる。確率モデルに基づく方法として、Brown コーパスからの学習による Church のシステム [5] が知られている。Brown コーパスは 100 万語からなる品詞タグ付きコーパスである。これに対して Brill[3] は規則に基づく (rule-based) システムでの規則の適用順序を Penn Treebank[9] を用い学習している。Penn Treebank は 450 万語に及ぶタグ付きコーパスであり、前述の Church のシステムにより品詞タグが振られている。品詞タグだけでなく、構文情報も付与されている。また、Cutting ら [6] や Meriald[10] により、品詞タグなしコーパスからのパラメータ学習についての研究がなされているが、品詞タグ付きコーパスを用いた方法と比べて、解析精度の向上に困難を有する。

日本語の場合、もともとわかち書きされない言語であるため、品詞タグ付けだけでなく単語境界の確定も行う必要がある。つまり、品詞の曖昧性だけでなく、単語境界の曖昧性の解消も同時に行う必要がある。英語と比べ、日本語は品詞タ

[2]例えば、英語のほとんどの名詞は動詞としても用いることができる (book: 本、予約する など)。

グ付きコーパスの整備が遅れていたため、制約や優先規則を用いた方法が主流であった。しかし、最近ではコーパスもかなり整備されてきており、確率モデルに基づく手法も盛んに研究されている。永田 [19]、平沢ら [22] は確率モデルに基づき EDR 日本語コーパスからパラメータ学習を行っている。朴ら [20] は自ら作成した品詞タグ付きコーパスから学習を行っている。日本語の品詞タグなしコーパスからのパラメータ学習については、竹内ら [16] や山本 [28] の研究がある。

2.1 優先規則による形態素解析

処理はおおまかに分けて、**形態素の認識**、**形態素間の接続可能性の確認**、**優先規則による最適解の決定**、の三つの段階がある。効率化のためにこれらの処理を統合することが多いが、ここでは、これらの処理が完全に独立で順次実行されるものとして説明する。

2.1.1 形態素の認識

文の先頭から適当な長さの部分文字列を順次取り出し、形態素辞書を検索する。辞書に存在した部分文字列の次の位置から、さらに適当な長さの部分文字列を検索する。これを文の最後の文字に達するまで繰り返す。 例えば、『プログラムだよ』という文の場合、先頭から辞書を引くと『プログラム』と『プロ』という形態素があり、次は『グ』と『だ』の位置から辞書を引くことになるが、『ロ』の位置からは引く必要が無い。なぜなら、『ロ』から始まる形態素 (例えば『ログ』) が辞書にあったとしても、『プ』という形態素が存在しないので、形態素『ログ』が正しい解析結果に含まれることはあり得ないためである。

『今日もしないとね』という文から形態素を認識した例を図 2に示す[3]。

2.1.2 形態素間の接続可能性の確認

形態素認識の結果中の隣接する形態素同士が接続するか否かの検証を行う。連接可能性は、例えば、『名詞と動詞は連接できる』『動詞の基本形と格助詞は連接

[3]ただし、紙面の都合上形態素の認識の結果の一部だけを表示する。説明上これらが辞書引き結果の全てとしておく。なお括弧内の語はその形態素の「読み」または「基本形」を表す。

今/名詞 今日/名詞 (きょう) 今日/名詞 (こんにち)	日/名詞	も/副助詞 もし/動詞 (もす) もし/名詞	し/動詞 (する) しな/名詞 しない/名詞 しない/動詞 (しなう)
な/判定詞 な/終助詞 ない/動詞 (なう) ない/形容詞 ない/接尾辞	い/動詞 (いる) いと/名詞	と/引用助詞 と/格助詞 と/述語接続助詞 と/名詞接続助詞	ね/助詞

今日もしないとね

図 2 形態素の認識

できない』といった規則で検証される。

これにより図 3のようなグラフ構造を持つ解析結果が得られる。図中の線は、接続可能な形態素同士の接続を表す。

2.1.3 優先規則による最適解の決定

これまでの作業で求まったグラフ構造の解析結果だけでは不適切な解があまりにも多く含まれすぎている。そこで、何らかの優先規則を用いて、尤もらしい解だけを残していく。

多くの形態素解析システムで用いられている優先規則を以下に示す [15][18]。

左最長一致法 文を左から見て最も長い形態素から優先して切り出していく。

2 文節最長一致法 文を左から見て 2 文節毎の長さが長い解を優先する。

形態素数最小法 入力された文字列に対して可能な形態素の分割を列挙して、そのうち最も形態素数の少ない解を優先させる。

文節数最小法 入力された文字列に対して可能な形態素の分割を列挙して、そのうち最も文節数の少ない解を優先させる。

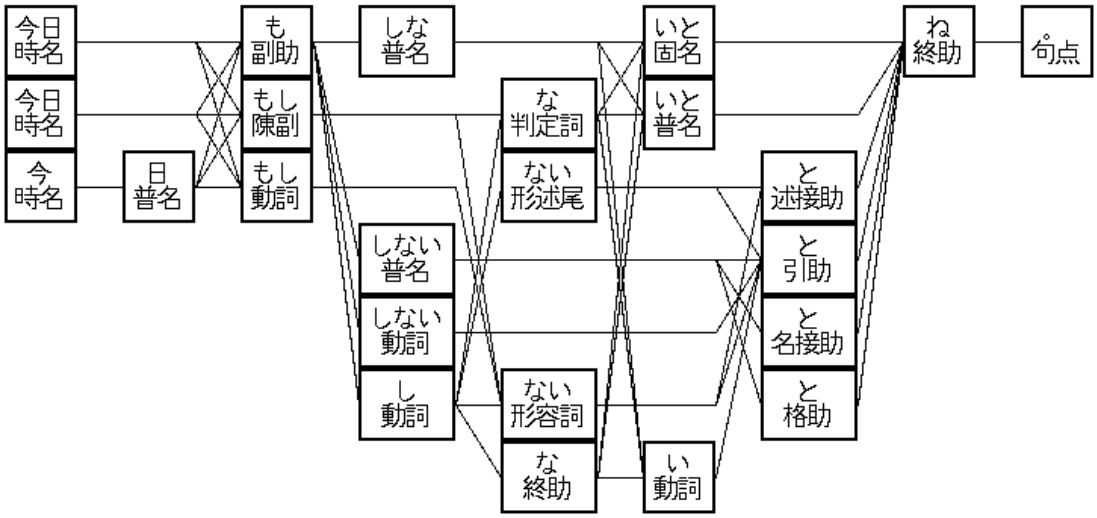


図 3 形態素解析におけるグラフ構造

接続コスト最小法 文節数最小法の拡張。分割された各形態素間の接続にコストを与えて、その合計が最小のものに高い評価を与える。久光・新田 [21] によって提唱されている。

コスト最小法 分割された各形態素間の接続と形態素そのものにコストを与えて、その合計が最小のものに高い評価を与える。

ここではコスト最小法について詳しく説明する。これは、ラティス状のグラフの全てのノード (語) とリンク (語と語の連接) に適当なコストを与え (前者を形態素コスト、後者を連接コストと呼ぶことにする)、コストの合計値が最小なパスを最適解として選択する方法である。

例えば、図 4では最適解としてコストの合計値が最小なパス

さかな — だ — よ

が選択されている。

コスト最小法におけるコストの意味付けとしては、コストをヒューリスティックとして捉えるモデルと、コストを遷移確率、出現確率と捉える確率モデルがある。

ヒューリスティックとして捉える場合、一般には人手によってコストを設定す

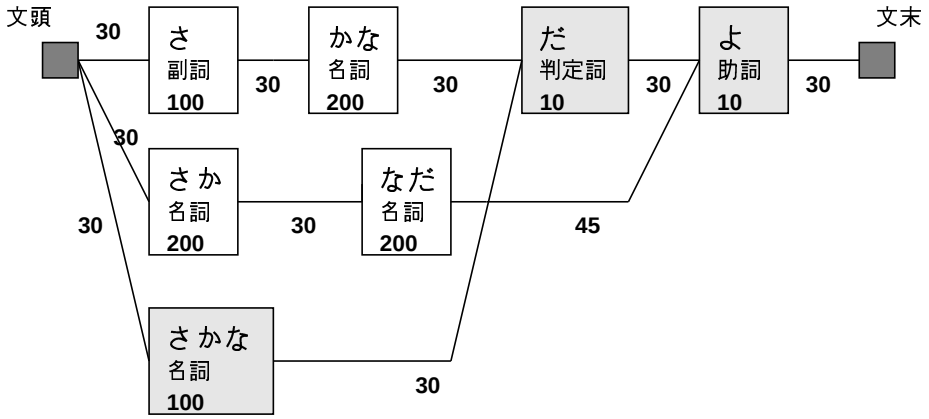


図 4 最適解の選択

ることが多い。これに対して小松ら [14] は、品詞タグ付きコーパスを評価用データとして用いて、コスト決定用のルールの確からしさを、統計的手法でなく数理計画法の手法で決定するという研究を行っている。

確率モデルとして捉える場合については次節で詳しく述べる。

2.2 確率モデルによる形態素解析

文が単語列 $W = w_1 \cdots w_n$、品詞列 $T = t_1 \cdots t_n$から構成されているものとすると、形態素解析は、単語列と品詞列の同時確率 $P(W,T)$ を最大化する品詞列$\hat{T}$を求める問題に帰着される [13]。

$$\hat{T} = \arg\max_{T} P(W,T)$$

$\arg\max_{x} f(x)$ は、$f(x)$ を最大にする x を返す関数である。

$P(W,T)$ を計算するための確率モデルを品詞付けモデルと呼ぶ。どのような品詞付けモデルを使用するかが問題となるが、一般に式 (1) に示すような品詞 N-gram モデルが用いられる。

$$P(W,T) \cong \prod_{i=1}^{n} p(w_i \mid t_1, ..., t_i) p(t_i \mid t_1, ..., t_{i-1}) \tag{1}$$

N = 2 のときを品詞 bigram モデル、N = 3 のときを品詞 trigram モデルと呼び、どちらもよく用いられる。$P(W,T)$ を品詞 bigram モデルで近似すると式 (2) のようになる。

$$P(W,T) \cong \prod_{i=1}^{n} p(w_i \mid t_i) p(t_i \mid t_{i-1}) \tag{2}$$

単語列を観測可能なシンボルの系列、品詞列を観測不能な状態系列と考えれば、$P(W,T)$ は隠れマルコフモデルにより定式化できる。品詞二つ組確率 $p(t_i \mid t_{i-1})$ と品詞別単語出現確率 $p(w_i \mid t_i)$ は、それぞれ、状態遷移確率とシンボル出力確率に相当する。

品詞タグ付きコーパスがあれば、品詞二つ組や単語の出現頻度を調べることにより、以下に示す式を用いて、これらの確率値を推定することができる。$C(x)$ は、x の出現頻度を表す。

$$\begin{aligned} p(w_i \mid t_i) &= \frac{C(w_i, t_i)}{C(t_i)} \\ p(t_i \mid t_{i-1}) &= \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \end{aligned}$$

例えば、ある品詞タグ付きコーパス中で、名詞が 100 回現れたとすると C(名詞) = 100 となる。そして、名詞の直後に動詞が 30 回現れたとすると C(名詞,動詞) = 30 となり、

$$p(\text{動詞} \mid \text{名詞}) = \frac{C(\text{名詞}, \text{動詞})}{C(\text{名詞})} = \frac{30}{100} = 0.3$$

と推定できる。また、名詞 100 回中、「さかな」が 5 回現れたとすると C(さかな、名詞) = 5 となり、

$$p(\text{さかな} \mid \text{名詞}) = \frac{C(\text{さかな}, \text{名詞})}{C(\text{名詞})} = \frac{5}{100} = 0.05$$

と推定できる。

品詞タグ付きコーパスが無い場合でも Baum-Welch アルゴリズム (Forward-Backward アルゴリズム)[2] を用いて確率値を推定することができる。品詞タグ付きコーパスを用いた方法は、品詞タグなしコーパスを用いた方法と比べて、高い解析精度が得られているが、大規模な品詞タグ付きコーパスが必要となる。

このようにして全ての確率値が求まれば、入力単語列に対して式 (2) の値を最大にする品詞列は、動的計画法の一種である Viterbi アルゴリズムによって求めることができる。Viterbi アルゴリズムとは、ラティス状のグラフの各ノードに、始めのノードからそのノードまでの最大確率を保存しておくという処理を順次行い、最後のノードにたどり着いたときに、最終的に最大確率を得ることのできるパスが判るというものである。形態素解析処理における Viterbi アルゴリズムの利用については文献 [1] に詳しい。

前節で述べたコスト最小法は、コスト計算に用いる目的関数 G を式 (2) の対数 (log) を取ったものとみなせば、確率モデルとみなすことができる。

$$\begin{aligned} G(W,T) &= -\log \prod_{i=1}^{n} p(w_i \mid t_i) p(t_i \mid t_{i-1}) \\ &= \sum_{i=1}^{n} \Big(-\log p(w_i \mid t_i) \Big) + \sum_{i=1}^{n} \Big(-\log p(t_i \mid t_{i-1}) \Big) \end{aligned}$$

$-\log p(w_i \mid t_i)$ が形態素コスト、$-\log p(t_i \mid t_{i-1})$ が連接コストに対応する。コスト最小のパスを求めることは、確率最大のパスを求めることに等しい。

コストを設定するには、先程述べたように、品詞タグ付きコーパスからパラメータ学習を行えばよい。

3. 形態素解析システム「茶筌」

ここでは、本研究で用いる形態素解析システム「茶筌」Version 1.0 [24] の概要と、品詞タグ付きコーパスからのパラメータ学習、及び、「茶筌」への確率モデルの適用について述べる。

3.1 システムの概要

「茶筌」とは奈良先端科学技術大学院大学松本研究室において開発された形態素解析システムである。「茶筌」の原形は。京都大学長尾研究室および奈良先端科学技術大学院大学松本研究室において開発された日本語形態素解析システム JUMAN[25] である。

本研究で「茶筌」を用いる主な理由を次に挙げる。

- 誰もが自由に利用できるパブリックドメインソフトである。
- 「茶筌」の原型である JUMAN は日本語形態素解析のツールとしてある程度流布しており、ユーザも多い。(「茶筌」Version 1.0 は JUMAN Version 2.0 と上位互換である)。
- 優先規則としてコスト最小法を採用しているので、確率モデルの適用が可能である。これにより品詞タグ付きコーパスからのパラメータ学習ができる。
- 使用する形態素文法をユーザが自由に定義できるため、様々な分野への適用が可能である[4]。

3.1.1 形態素の構造

「茶筌」システムで仮定している形態素の構造について説明する。まず形態素の構造を、品詞情報と活用情報の 2 つに分けて考える。

[4]多数提唱されている様々な日本語文法体系だけでなく、日常会話や法律文のための文法体系、さらには韓国語にも適用できる [8]。

品詞情報 品詞情報はさらに 2 つの範疇、**形態品詞**、**品詞細分類**に分けられる。例えば、形態品詞は「名詞」「動詞」「形容詞」などの基本的な分類に用いる。品詞細分類は形態品詞をさらに細かく分類するときに用いる。例えば、「名詞」をさらに「普通名詞」「サ変名詞」「固有名詞」などに細分類する場合これらは品詞細分類の範疇に入る。また、ある形態品詞を細分類したくない場合は品詞細分類を指定しなくてもよい。

活用情報 活用情報も 2 つの範疇、**活用型**、**活用形**に分けられる。大部分の活用は規則的である。その規則性に従って分類した活用の型を活用型と呼ぶ。そして、連接に応じて実際にとり得る表層的な個々の形態を活用形と呼ぶ。活用型の例として、「子音動詞ガ行」「イ形容詞イ段」などが、活用形の例として、「基本形」「未然形」「命令形」などがあげられる。

形態品詞、品詞細分類、活用型、活用形、見出し語をそれぞれ H1, H2, K1, K2, M で表す。また、5 項組 (H1 H2 K1 K2 M) を、**形態素構造**と呼ぶ。見出し語 M は、活用する語の場合はそれの基本形を用いる。

各形態素構造の記述には特別なシンボル “$*$” を使用することができる。“$*$” は、その項を考慮しない (未指定、“don't care”) ことを表わす。たとえば、形態素構造

$$\alpha_1 = (* * * * *)$$

は任意の形態素を表わす集合とみなすことができる。同様に、形態素構造

$$\alpha_2 = (\text{名詞} * * * *)$$

は ‘名詞’ という形態品詞に分類された全ての形態素の集合を表わす。また、形態素構造

$$\alpha_3 = (* \ * \ * \ \text{未然形} \ *)$$

は ‘未然形’ という活用形を取る全ての形態素の集合を表わす。また、

$$\alpha_4 = (\text{助詞 格助詞} \ * \ * \ \text{が})$$

のように具体的な形態素を指定することも許される。

実際に連接規則等を記述する際には、形態素構造を表わすリストのある要素以降が全て未指定でよい場合、形態品詞の部分以外はそれらを省略してもよい。

3.1.2 形態素解析処理に必要なデータ

形態素解析処理を行うためには以下にあげるデータが必要である。

- **形態品詞分類ファイル** (cf. `JUMAN.grammar`)

  システムで用いる形態品詞およびその品詞細分類の名称を定義する。

- **活用関係ファイル** (cf. `JUMAN.kankei`)

  どの品詞がどのような活用をするかということを定義する。品詞情報と活用型の対応が記述されている。

- **活用ファイル** (cf. `JUMAN.katuyou`)

  各活用型とそれの持つ全ての活用形を定義する。

- **形態素辞書** (cf. `*.dic`)

  個々の形態素を定義する。

- **品詞コスト**

  各品詞にあたえられるコスト。リソースファイルで定義する。

- **連接規則ファイル** (cf. `JUMAN.connect.c`)

  連接規則の集合である。

形態素解析の精度に直接関わる「形態素辞書」「品詞コスト」「連接規則ファイル」の 3 つのデータについて、詳しく述べる。

形態素辞書

S 式を用いて記述する。以下に形態素定義の記述方法を BNF で示す。

```
〈形態素定義〉          ::=  (〈#形態品詞名〉〈形態素情報の並び〉) |
                             (〈#形態品詞名〉(〈#品詞細分類名〉〈形態素情報の並び〉))
〈形態素情報の並び〉    ::=  〈形態素情報〉 | 〈形態素情報〉〈形態素情報の並び〉
〈形態素情報〉          ::=  (〈見出し語情報〉〈読み情報〉〈活用型情報〉〈意味情報〉)
〈見出し語情報〉        ::=  (見出し語 〈見出し語内容の並び〉)
〈見出し語内容の並び〉  ::=  〈見出し語内容〉 | 〈見出し語内容の並び〉
〈見出し語内容〉        ::=  〈#見出し語表記〉 | (〈#見出し語表記〉) |
                             (〈#見出し語表記〉〈#数値〉)
〈読み情報〉            ::=  (読み 〈#読み表記〉)
〈活用型情報〉          ::=  (活用型 〈#活用型名〉) | NIL
〈意味情報〉            ::=  (意味情報 〈#意味記述〉) | NIL
```

各形態素に、品詞コストに対する相対的な重みを与えることができる。<#数値>で指定する。省略されている場合は 1 とみなす。この値と後に述べる品詞コストとの積が、その形態素自体が持つコストとなる。

記述例：

```
(動詞 ((見出し語 (騙す) (だます 1.5)) (読み だます) (活用型 子音動詞サ行)))
```

品詞コスト

解析処理中、ある形態素のコストを知るためには、この形態素に属する品詞の品詞コストと形態素コスト重みの積を計算する。各個人が解析環境をカスタマイズするためのファイルであるリソースファイル内に記述される。

記述例：

```
(品詞コスト
        ((*)              10)
        ((未定義語)     5000)
        ((特殊 *)        100)
        ((動詞)          100)
        ((形容詞)        100)
        ((判定詞)         10)
```

```
((助動詞)          10)
((名詞 *)         100)
((名詞 固有名詞) 110)
((名詞 形式名詞) 10)
...
```

例えば、

```
(動詞 ((見出し語 (騙す) (だます 1.5)) (読み だます) (活用型 子音動詞サ行)))
```

の場合、「動詞」の品詞コストは 100 なので、『騙す』の形態素コストは $1 \times 100 = \underline{100}$、『だます』の形態素コストは $1.5 \times 100 = \underline{150}$ となる。

これにより形態素辞書内部を書き換えなくても、人手によるコスト修正が容易に行える。

連接規則ファイル

一つの連接規則は形態素構造の対と連接コストからなる 3 要素のリストによって表現される。第一要素に含まれる形態素と第二要素に含まれる形態素の連接コストを第三要素で指定する。連接コストの値は、0 から 255 までの整数値でなければならない。省略されている場合は 10 とみなす。また、連接コストを 0 に指定することによって、二つの形態素が連接不可能であること表わす。

連接規則の定義の記述方法を BNF で示す。

```
〈連接規則〉          ::=  ( (〈形態素構造の並び〉) (〈形態素構造の並び〉) ) |
                           ( (〈形態素構造の並び〉) (〈形態素構造の並び〉) 〈#連接コスト〉)
〈形態素構造の並び〉  ::=  〈形態素構造〉| 〈形態素構造〉〈形態素構造の並び〉
〈形態素構造〉        ::=  (〈#形態品詞名〉) |(〈#形態品詞名〉〈#品詞細分類名〉) |
                          (〈#形態品詞名〉〈#品詞細分類名〉〈#活用型名〉) |
                          (〈#形態品詞名〉〈#品詞細分類名〉〈#活用型名〉〈#活用形名〉) |
                          (〈#形態品詞名〉〈#品詞細分類名〉〈#活用型名〉〈#活用形名〉〈#見出し語〉)
```

記述例：

```
(((名詞)                          ; 名詞、指示詞の名詞形態指示詞などは
  (指示詞 名詞形態指示詞)         ; 判定詞および特殊の読点と連接可能
  (接尾辞 名詞性述語接尾辞)
```

```
  (接尾辞 名詞性名詞接尾辞)
  (接尾辞 名詞性名詞助数辞))
        ((判定詞)
         (特殊 読点))
)

(((名詞 サ変名詞)                          ; サ変名詞と接尾辞の「化」には、
  (接尾辞 名詞性名詞接尾辞 * * 化))        ; 「する」「できる」などが
        ((動詞 * サ変動詞 * する)           ; 後接可能である。
         (動詞 * 母音動詞 * できる)         ; また、その連接のコストは 5 であり
         (動詞 * 母音動詞 * 出来る))        ; 結合度が強い。
        5 )
```

3.1.3 解析アルゴリズム

「茶筌」は、コスト最小法に基づき、最適解を求める。具体的には、2.1.3節に述べたように、Viterbi アルゴリズムを用い、ラティス状のグラフ構造をもつ解析結果の形態素コストと連接コストの合計が最小になるような経路を選ぶ。その経路上の形態素列を最適解として出力する。このときに用いられる形態素コストとは、各形態素の品詞の「品詞コスト」と形態素辞書中に各形態素情報とともに記述されている「相対的な重み」の積である。

3.2 「茶筌」への確率モデルの適用

まず、品詞タグ付きコーパスからの確率パラメータの学習について述べる。そして、「茶筌」上へ確率モデルを適用し、学習したパラメータに基づき形態素解析処理を行う方法について述べる。

品詞付けモデルには、式 (3) に示す品詞 bigram モデルを用いる。これは、「茶筌」がコスト最小法を採用しており、品詞二つ組で連接コスト、形態素と品詞の二つ組で形態素コストを規定しているので、品詞 bigram モデルに適合しやすいからである。

$$p(W,T) = \prod_{i=1}^{n} p(w_i \mid t_i) p(t_i \mid t_{i-1}) \tag{3}$$

3.2.1 品詞タグ付きコーパスからの確率値パラメータの学習

品詞タグ付きコーパスから、品詞別単語出現確率 $p(w_i \mid t_i)$ と二つ組確率 $p(t_i \mid t_{i-1})$ を学習することを考える。以下の式に従い、$C(w_i, t_i), C(t_i), C(t_{i-1}, t_i)$ といった出現頻度を求めれば良い。

$$
\begin{aligned}
p(w_i \mid t_i) &= \frac{C(w_i, t_i)}{C(t_i)} \\
p(t_i \mid t_{i-1}) &= \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}
\end{aligned}
$$

単語にあたる w_iは見出し語、品詞にあたる t_iの構造には、品詞情報全て (H1 H2 K1 K2 $*$) や形態品詞のみ (H1 $*$ $*$ $*$ $*$) を用いるのが一般的である。しかし、t_iの構造に、品詞情報全てを用いる場合は、細かすぎてデータスパースネスの問題が起こりやすくなるし、形態品詞のみを用いる場合は荒すぎて細かい言語現象を捉えきれない。実際には、細かく見たい品詞もあれば、荒く見れば十分な品詞もあるので、人手による連接規則の記述に用いられている形態素構造 (p.16の連接規則の記述例を見よ) を t_iの構造として利用することにする。この形態素構造を品詞パターンと呼ぶことにする。品詞パターンの例を図 5に示す。

品詞タグ付きコーパスからの学習の際に、コーパスから形態素が与えられたとき、複数の品詞パターンに適合してしまうここがある。そのような場合、「より右側の項目が具体化されているものが優先される」という基準を用いて、唯一の品詞パターンを決定することにする。

例えば、学習時に、以下のような形態素が与えられたとする。

可能な かのうな 可能だ 形容詞 $*$ ナ形容詞 ダ列基本連体形

図 5を見ると、この形態素は以下に示すような複数の品詞パターンに適合する。

形容詞
形容詞　$*$　ナ形容詞
形容詞　$*$　ナ形容詞　ダ列基本連体形
形容詞　$*$　ナ形容詞　$*$　　可能だ

H1	H2	K1	K2	M
名詞				
名詞	形式名詞			
形容詞				
形容詞	*	*	基本形	
形容詞	*	イ形容詞アウオ段		
形容詞	*	イ形容詞イ段		
形容詞	*	ナ形容詞		
形容詞	*	ナ形容詞	語幹	
形容詞	*	ナ形容詞	ダ列基本連体形	
形容詞	*	ナ形容詞	*	可能だ
形容詞	*	ナ形容詞	*	不可能だ

図 5 学習に用いる品詞パターンの例

しかし、先程の基準により、

形容詞 * ナ形容詞 * 可能だ

が選択されることになる。

品詞パターンの採用による問題点については次節で述べる。

確率値パラメータの学習に用いる品詞タグ付きコーパスの例を図 6に示す。このような品詞タグ付きパラメータから全ての i について、出現頻度 $C(w_i, t_i), C(t_i), C(t_{i-1}, t_i)$ を求め、確率値 $p(w_i \mid t_i), p(t_i \mid t_{i-1})$ を計算する。

	表層形	読み	M	H1	H2	K1	K2
1	これ	これ	これ	指示詞	名詞形態指示詞	*	*
2	は	は	は	助詞	副助詞	*	*
3	可能な	かのうな	可能だ	形容詞	*	ナ形容詞	ダ列基本連体形
4	こと	こと	こと	名詞	形式名詞	*	*
5	だ	だ	だ	判定詞	*	判定詞	基本形

図 6 品詞タグ付きコーパスの例

例えば $i = 4$ とすると、

$$
\begin{aligned}
w_4 &= \text{「こと」} \\
t_4 &= (\text{名詞 形式名詞 } * \ * \ *) \\
t_3 &= (\text{形容詞 } * \text{ ナ形容詞 } * \text{ 可能だ})
\end{aligned}
$$

が求まるので、$C(w_4, t_4), C(t_4), C(t_3, t_4)$ にそれぞれ 1 を加えれば良い。

3.2.2 適用の際の問題点

「茶筌」への実装上、考慮すべき点について述べる。

形態素辞書と活用形の問題

「茶筌」の形態素辞書には活用形 (K2) は記述できない[5]。例えば、

> ナ形容詞型活用の形容詞『きれいだ』の形態素コストは 100 です。

という情報は辞書に記述できるが、

> ナ形容詞型活用の形容詞『きれいだ』のダ列基本連体形『きれいな』の形態素コストは 130 です。

という情報は記述できない。

そこで、品詞別単語出現確率 $p(w_i \mid t_i)$ の条件部 (t_i) の形態素構造に活用形が含まれている場合は、以下に示すように、t_iから活用形を除いた品詞パターン t'_iを用いて、確率値 $p(w_i \mid t_i)$ を近似することにする。

$$
\begin{aligned}
p(w_i \mid t_i) &= \frac{C(w_i, t_i)}{C(t_i)} \\
&\cong \frac{C(w_i, t'_i) \times \dfrac{C(t_i)}{C(t'_i)}}{C(t_i)} \\
&= \frac{C(w_i, t'_i)}{C(t'_i)}
\end{aligned}
$$

[5] p.14の形態素辞書の記述方法を参照。辞書に活用形が記述できるようにシステムを改良することは可能だが、辞書エントリが増大し、あまり実用的とは言えない。

$$= \quad p(w_i \mid t'_i)$$

これは、あらゆる活用形の形態素の出現確率を見出し語の出現確率で代表することに相当する。

品詞パターンの制約

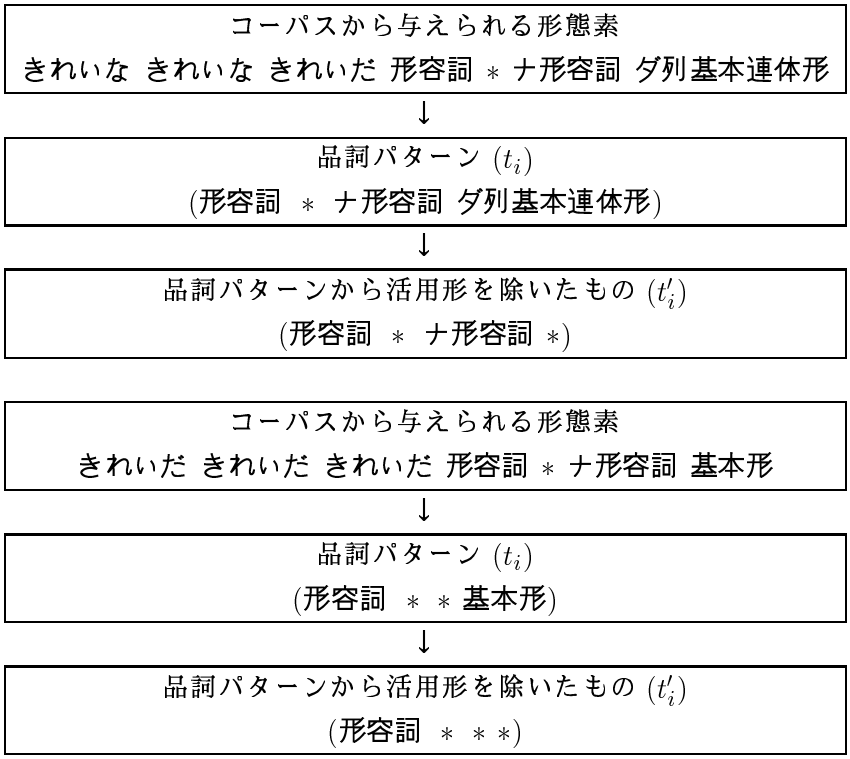
学習時に、品詞タグ付きコーパス中に現れる同じ辞書エントリの形態素から、t'_iが一意に求まらないことがある。すると、その形態素は、複数の品詞別単語生成確率 $p(w_i \mid t'_i)$ を持ってしまうことになる。コーパスからの学習結果を解析に反映させるには、品詞別単語生成確率から計算されるコスト[6]をその形態素の辞書エントリに書き込む必要があるので、その形態素の品詞別単語生成確率が複数存在すると非常に困る。

例を用いて説明する。形態素辞書上では同じエントリである「ナ形容詞型活用の形容詞『きれいだ』」が、以下のように異なる活用形で、品詞タグ付きコーパスに現れたとする。

きれいな きれいな きれいだ 形容詞 * ナ形容詞 ダ列基本連体形
きれいだ きれいだ きれいだ 形容詞 * ナ形容詞 基本形

それぞれ、図 5の品詞パターンを用い、t'_iを求めることにする。

[6]確率値からコストへの変換については次節で述べる。



それぞれ異なる t'_iが求まってしまう。すると、p(きれいだ | (形容詞 * ナ形容詞 *))、p(きれいだ | (形容詞 * * *)) という複数の品詞別単語生成確率が存在することになってしまう。

この問題を解決するため「活用する品詞の品詞パターンには活用型 (K1) は必須」という制約を課した[7]。この制約により、図 5の品詞パターン (形容詞 * * 基本形) は破棄され、かわりに (形容詞 * ナ形容詞 基本形) や (形容詞 * イ形容詞イ段 基本形) のように活用型が具体化されたものが使用されることになる。

3.2.3 確率値からコストへの変換

「茶筌」ではコスト最小法を採用しているので、実装上、学習による確率値 $p(w_i \mid t_i)$、$p(t_i \mid t_{i-1})$ をコストへ変換する必要がある。この場合のコスト計算に用いる目的関数 G は、式 (3) の log を取ったものとみなせる。

[7]学習の際に活用型を考慮しないのならば、「品詞パターンの活用型 (K1) は必ず “*”」という制約でも良い。

$$
\begin{aligned}
G(W,T) &= -\log \prod_{i=1}^{n} p(w_i \mid t_i) p(t_i \mid t_{i-1}) \\
&= \sum_{i=1}^{n} \Big(-\log p(w_i \mid t_i) \Big) + \sum_{i=1}^{n} \Big(-\log p(t_i \mid t_{i-1}) \Big)
\end{aligned}
$$

$-\log p(w_i \mid t_i)$ が形態素コスト、$-\log p(t_i \mid t_{i-1})$ が連接コストに対応する。しかし、「茶筌」で利用できるコスト値は 0〜255 の整数に限られているので、スケール変換を行う必要がある。そのために用いる正の係数αをコスト化係数と呼ぶ。以下の式で確率値をコストに変換する。

$$
\text{形態素コスト } c(w_i \mid t_i) = -\alpha \log p(w_i \mid t_i) \tag{4}
$$

$$
\text{連接コスト } c(t_i \mid t_{i-1}) = -\alpha \log p(t_i \mid t_{i-1}) \tag{5}
$$

コスト化係数αは、以下の式から求めることができる。C_{max}をコストの上限 (255 以下の値) とする。

$$
\alpha = \frac{C_{max}}{\max\{-\log p(w_i \mid t_i), -\log p(t_i \mid t_{i-1})\}}
$$

以上の作業により、品詞タグ付きコーパスから学習した確率値を用いた確率モデルによる形態素解析が「茶筌」上に実装できる。

本研究で提案する手法 (次章に詳しく述べる) と比較するために、「茶筌」を用いた確率モデルによる形態素解析の実験を行った。結果等については第 5章に述べる。

4. 規則と確率モデルの統合

今までに研究されてきた日本語形態素解析システムは大きく次の 2 つの流れに分けられる。

人手による優先規則を用いた形態素解析

人間の観点から言語現象をおおまかに抽象化できるが、規則の複雑化により、保守・管理が困難になる。

コーパスを用いた確率モデルによる形態素解析

実際のデータに基づき細かい言語現象に対処できるが、大規模な品詞タグ付きコーパスが必要となる。また、データスパースネスの問題もある。

本研究では、これら 2 つの方法の欠点を補い合い、より高精度な形態素解析処理の実現を目指す。

具体的には、コスト最小法に基づく形態素解析システム「茶筌」[24] に与えられた人手によるコスト (形態素コスト、連接コスト) と、3.2.1節で述べた方法により品詞タグ付きコーパスから学習された**確率値** (品詞別単語出現確率、品詞二つ組確率) を統合し、新たなコストを作成するという作業を行う。

4.1 統合処理の概要

人手によるコストを指数関数を用いて $[0,1]$ の値に変換した確率値的なもの (以降、**準確率値**[8] と呼ぶことにする) と、品詞タグ付きコーパスから学習した確率値を、式 (6) に従い、ある比率 λ $(0 \leq \lambda \leq 1)$ で足し合わせる。λ を**統合比率**と呼ぶことにする。P'_{hand} は人手によるコストを変換した準確率値、P_{corpus} は品詞タグ付きコーパスから学習した確率値、P'_{new} はこれらを混ぜ合わせた準確率値を表す。

$$P'_{new} = \lambda P'_{hand} + (1-\lambda) P_{corpus} \tag{6}$$

[8] 準確率値とは、0 以上 1 以下の値であり、かつ確率値でないにも関わらず確率値として用いられるものと定義する。

そして、P'_{new}を変換し新たなコストとして解析に利用する。

処理の流れを図 7に示す。4.2節で述べる方法で、人手によるコストを変換プログラムにより準確率値に変換する。また、3.2.1節で述べた方法で品詞タグ付きコーパスから確率値を学習プログラムにより学習する。統合プログラムが、これらの準確率値と確率値を式 (6) を用いて統合し、求まった準確率値 P'_{new}をコストに変換する[9]。

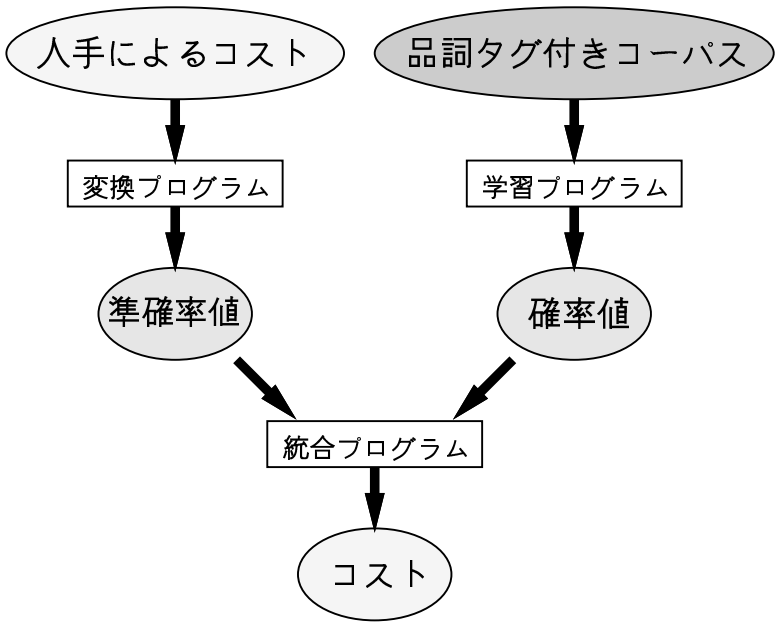


図 7 統合処理の流れ

4.2 人手によるコストの準確率値への変換

ここで、人手によるコストの準確率値への変換について説明する。

3.2.3節の式 (4)(5) より導き出された式 (7) により、人手によるコスト Cを指数関数を用いて $[0, 1]$ の値へ変換することができる。α_1はコスト化係数である[10]。

[9] P'_{new}は厳密な確率値ではないが、コスト最小法の性質上、使用するコストは厳密な確率の値を反映したものである必要はないので問題は無い。

[10] コスト化係数とは確率値をコストに変換する際に用いる係数である。3.2.3節を参照のこと。

$$P'_{hand} \quad = \quad \exp\left(-\frac{1}{\alpha_1} \times C\right) \tag{7}$$

しかし、P'_{hand}は確率の公理をみたさないので、厳密には確率値ではない。

人手によるコストをコーパスからの学習結果である確率値と統合させるには、P'_{hand} を確率値として扱いたい。そのための一番素直な方法は、P'_{hand} を確率値に正規化させることである。形態素コスト (品詞別単語出現確率) について考えてみる。p'を正規化前 (準確率値)、p を正規化後 (確率値) とする。正規化するには、全ての品詞に対して、その品詞に属する語について、品詞別単語出現確率の和が 1 になるようにする必要がある。つまり、t を品詞としたとき、式 (8) が成り立つようする必要がある。

$$\sum_{w_i} p(w_i \mid t) = 1 \tag{8}$$

正規化は式 (9)(10) により行う。

$$q_t \quad = \quad \sum_{w_i} p'(w_i \mid t) \tag{9}$$

$$p(w_i \mid t) \quad = \quad \frac{p'(w_i \mid t)}{q_t} \tag{10}$$

$-\log p'(w_i \mid t)$ に対応する形態素コストは、ほとんど品詞コスト (3.1.2節を参照) により決定され、その品詞に属する形態素には、ほぼ一様なコストが与えられる。例えば、「動詞」に対する品詞コストが 100 ならば、「動詞」に属する形態素「歩く」「走る」「笑う」などには、それぞれ 100 のコストが与えられる。コストは個々の形態素の出現頻度を考慮して与えられている訳ではないので、式 (9) による q_tの値は、その品詞に属する形態素の辞書エントリー数に大きく依存することになる。

しかし、人手によるコスト Cは、確率値から変換されたものではないので、α_1を 3.2.3節の方法で求めることはできない。求め方については節末に述べる。

すると、たとえ、$p'(\text{本} \mid \text{名詞}) < p'(\text{本} \mid \text{接尾辞})$ という関係が成り立っていたとしても、式 (10) により、$q_{\text{名詞}}, q_{\text{接尾辞}}$ の値、つまり名詞・接尾辞のエントリー数によっては、$p(\text{本} \mid \text{名詞}) > p(\text{本} \mid \text{接尾辞})$ となりうる。

これでは、人手による規則本来の解析精度が保持できないことになる。

以上のことから厳密な確率値への変換は諦め、P'_{hand}の値を確率値の近似値としてそのまま利用することにする。

人手によるコストをなるべく確率値に近い値に変換するために、コスト化係数α_1の決め方が重要になる。コスト化係数は、さまざまなコスト体系ごとに、決まった値が存在すると思われる。本研究ではこれを実験により経験的に求めることにする。

4.3 統合の際の問題点

人手によるコストと学習による確率値を式 (6) により統合するする際に、考慮すべき点について述べる。

まずは、統合比率λ をどのように決定するかという点である。統合比率とは、人手によるコストと学習による確率値を足し合わせるときの比率であり、大きければ人手によるコストを、小さければ学習による確率値を重要視することになる。これは、実験により経験的に求めることにする。

もう一つは、式 (6) により計算された準確率値 P'_{new}をコストに変換するときに用いるコスト化係数α_2 をどのように決定するかという点である。これは、3.2.3 節で述べた方法を用いる。これは、式 (11) に示すように、$-\log P'_{new}$の最大値とコストの上限 C_{max}から求める方法である。

$$\alpha_2 = \frac{C_{max}}{\max\{-\log P'_{new}\}} \tag{11}$$

5. 実験と考察

本章では、本研究で提案した手法を検証するために、品詞タグ付きコーパスを用いて実験を行い、それに基づいて考察を行う。

まず予備実験により、コスト化係数、統合比率の最適値を求める。そして、それを基に行った実験について述べる。

5.1 実験

5.1.1 実験に用いるデータ

実験に用いた品詞タグ付きコーパス k1000 について解説する。

これは、日経新聞 CD-ROM 94 年版から無作為に取り出した 1000 文を、形態素解析システム「茶筌」[24] で解析し、人手で修正したものである。品詞体系は、「茶筌」のデフォルトのものを採用している。この作業には、品詞タグ付きコーパス作成支援システム Vi JUMAN[26] が用いられている[11]。

特徴を以下に示す。

- 一人の人間により作成されたため品詞タグ付けに一貫性がある。
- 量が少ないため、全体に対する修正作業が容易である。
- 「茶筌」の解析結果との差分による度重なる品詞タグ修正作業が行われている。これにより、品詞の付与間違い、表層語と活用形の不一致などの、現存する他の日本語品詞タグ付きコーパスに多く見られる誤りは、ほとんど無いといっても良い。
- コーパス修正作業の際に、「茶筌」の規則 (コスト) も同時に人手で修正しているので、「茶筌」の人手によるコスト体系と非常に相性が良い。

現在の「茶筌」の品詞体系から見て、非常に質の高いコーパスであるといえる。

[11]品詞タグ付きコーパス作成支援環境については、第 6章に詳しく述べた。

5.1.2 評価尺度

評価尺度として再現率 (recall)・適合率 (precision) を用いる。正解データ (テストデータ) に含まれる形態素の数 (Std)、システムの出力に含まれる形態素の数 (Sys)、照合した形態素 (システムの出力中の正しい形態素) の数 (M) を数え、以下の式で求める。

$$\begin{aligned} recall &= \frac{M}{Std} \\ precision &= \frac{M}{Sys} \end{aligned}$$

2つの形態素を照合する際に、「形態素の持つ全ての情報が等しい」という基準を採用した。これは、単語分割、読み、品詞情報の全てが一致していなければ、誤りとみなすという、大変厳しい基準である。

図 8を用いて説明する。入力文「前年度に比べ、八割増となった。」に対し、正解データに含まれる形態素の数 (Std) は 10 個、システムの出力に含まれる形態素の数 (Sys) は 9 個である。両者の中で、形態素の持つ全ての情報が等しいものは 7 個である。従って、再現率と適合率は 7/10 と 7/9 である。

5.1.3 予備実験

まずこれらの品詞タグ付きデータを用いて、予備実験を行った。目的は、人手によるコストを準確率値に変換する際のコスト化係数α_1と統合比率λの最適値を求めることである。実験の手順は次節に示すものと同様であるが、その際、α_1とλを様々な値に変更させ、最適値を探す。

予備実験の結果、一番良い精度を出すα_1の値は、13〜15 であることが判った。これは人手によるコスト体系の持つ固有の値であると思われるが、現時点では断定できない。また、一番良い精度を出すλの値は、0.1〜0.3 であることが判った。この値は、学習に用いる品詞タグ付きコーパスの質と量に依存する可能性があるが、今回利用したコーパスだけからでは断定できない。

以降では、$\alpha_1 = 14$ 、 $\lambda = 0.15$ として、実験を行う。

5.1.4 本実験

実験の手順について述べる。実験は 10-fold cross validation により行った。

1. 1000 文からなる品詞タグつきコーパス k1000 を、機械的に 100 文ずつ 10 個に分割し、それぞれを正解データとする。

2. 各正解データ毎に、以下の処理を行う。

   (a) その正解データ以外の残り 900 文の中から、無作為に文を選択し、100 〜900 文からなる 9 個の学習データを作成する。

   (b) 各学習データ毎に以下の処理を行う。

      i. そのデータをもとに確率パラメータの学習を行う。

      ii. 正解データを対象に、学習結果を用いた確率モデル形態素解析システムと統合による形態素解析システムで解析を行う。

      iii. 形態素解析結果を正解データと比較し、適合率·再現率を計算する。

3. 上記の 1, 2 の処理を行った後、適合率・再現率の平均を計算する。

また、人手によるコストを用いた形態素解析処理の適合率·再現率も各正解データ毎に求めて、平均を取った。

実験結果を図 9 に示す。図中の “rule” は人手によるコストを用いた場合、“probablistic” は品詞タグ付きコーパスから統計的学習を行った場合、“integration” はこれら二つを統合した場合を表す。横軸は、学習に用いた文の数を表す。

5.2 考察

この実験により、統合による手法は、人手により調整されたコストや、コーパスから学習された確率パラメータを用いた既存の手法よりも、解析精度において優位であることが示された。特に、学習に用いる品詞タグ付きコーパスが小規模の場合において、他の手法と比べて、著しく解析精度が高くなることが確認された。これにより、小規模な品詞タグ付きコーパスであっても、解析精度の向上には有効であると言うことができる。

現時点のデータからでは、統合比率λの最適値は断定できないが、0.5以下の値ならば、統合による解析精度が他と比べて高くなることは予備実験から確かめられた。これは、人手によるコストよりもコーパスからの学習による確率値を重要視する方が有利であるということを意味するものである。つまり、本手法に関しては、品詞タグ付きコーパスから学習される細かい言語現象 (確率パラメータ) の不完全な部分を、人間の観点からおおまかに捉えた言語現象 (人手による規則・コスト) により補完するという方法により高精度の解析結果が得られると考えるべきである。

統合比率λの最適値は、人手によるコストでの解析とコーパスからの学習による解析の精度の差と関係があると思われる。これは、学習に用いるコーパスの量にも関係する。学習に用いるコーパスの量が多くなると、人手によるコストでの解析と比べ、コーパスからの学習による解析の方が精度が高くなる。それらの精度の差を考えると、λはより小さい値の方が良いと推定できる。すると、学習に用いるコーパスの量が多くなるに連れて、λは小さい値に設定され、統合による手法とコーパスからの学習による手法の解析精度は漸近的に近付いていくということは、簡単に予想できる。このことは実験結果からも予想できる。

しかし、実験に用いた品詞タグ付きコーパス k1000 は、小規模すぎて、λとコーパスの量との関係を実験的に明らかにすることができなかった。このことを検証するためには、大規模で高品質の品詞タグ付きコーパスが必要になる。

正解データ
前年度 ぜんねんど 前年度 名詞 普通名詞 * *
に に に 助詞 格助詞 * *
＋ 比べ くらべ 比べる 動詞 * 母音動詞 基本連用形
、 、 、 特殊 読点 * *
八 はち 八 名詞 数詞 * *
＋ 割 わり 割 接尾辞 名詞性名詞助数辞 * *
＋ 増 ぞう 増 名詞 普通名詞 * *
と と と 助詞 格助詞 * *
なった なった なる 動詞 * 子音動詞ラ行 タ形
。 。 。 特殊 句点 * *

システムの出力
前年度 ぜんねんど 前年度 名詞 普通名詞 * *
に に に 助詞 格助詞 * *
＋ 比べ くらべ 比べ 名詞 普通名詞 * *
、 、 、 特殊 読点 * *
八 はち 八 名詞 数詞 * *
＋ 割増 わりまし 割増 名詞 普通名詞 * *
と と と 助詞 格助詞 * *
なった なった なる 動詞 * 子音動詞ラ行 タ形
。 。 。 特殊 句点 * *

図 8 形態素解析の例

“+” は正解データとシステムの出力とで異なる部分を示している。

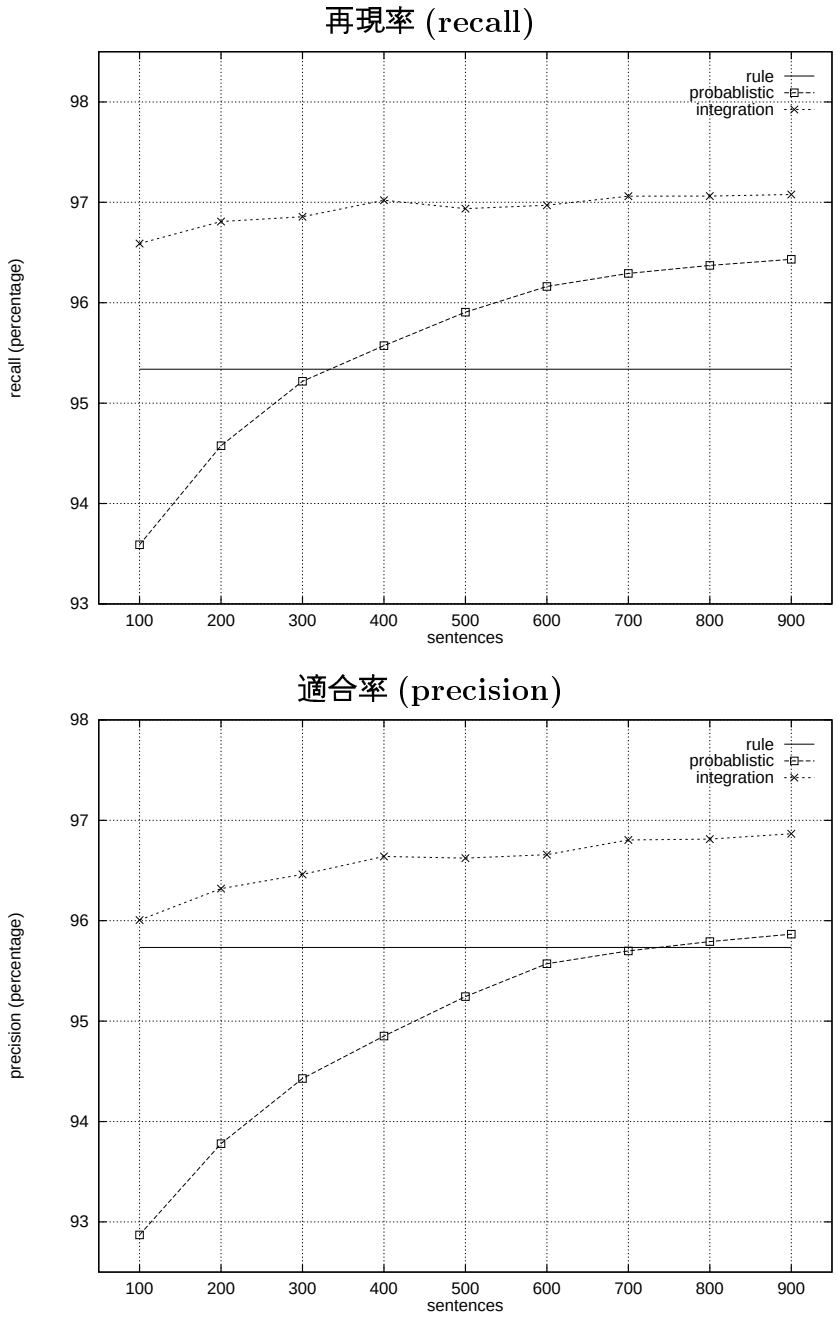

再現率 (recall)
recall (percentage)
sentences
rule
probablistic
integration
適合率 (precision)
precision (percentage)
sentences
rule
probablistic
integration

図 9 実験結果

6. 品詞タグ付きコーパス作成支援環境

この章では、品詞タグ付けコーパス作成支援環境について説明する。これは、本研究で提案した手法も含めた統合的な環境である。

近年、様々な分野で品詞タグ付きコーパスが整備されつつある。大規模な品詞タグ付きコーパスを効率良く作成するためには、形態素解析システムや修正ツール等の作業環境が重要となる。

例えば、RWC の品詞タグ付きコーパス [12] は、日本 IBM の形態素解析システム JMA を用いて自動解析したものを、後処理フィルタで RWC の採用した品詞体系へ置き換え、それをテキストエディタ上で動く編集ツールを用い人手により修正するという方法で整備されている。また、Penn Treebank[9] 作成時には、テキストエディタをベースとしたマウス操作によるユーザインターフェースが用いられている。これは一通りの形態素解析結果を表示し、品詞の誤りがあれば作業者がその単語をマウスで選び正しい品詞を打ち込み修正するいう単純なものである。

しかし、このような方法は決して最適のものとは言えない。理想的な作業環境の実現を目指すためには、以下に示すような問題を解決する必要がある。

- テキストエディタベースの作業環境では、解析結果が容易に認識できず、作業効率が悪い。
- フィードバックの欠如により、同じような誤りが繰り返し現れ、作業者に心理的負担を与える。
- 複数の人間が作業に従事する際に、品詞タグ付与の一貫性を維持するのが困難である。
- 作業者は、誤りの含まれない大量の解析結果にも目を通さなければならない。

第一の問題点に対処するために、形態素解析システムのグラフィカルユーザインターフェースとして ViJUMAN[26] を構築した。これにより快適な作業環境が提供される。詳細については、6.1節で述べる。

第二の問題点に関しては、文献 [27] でその解決のための一手法を提案したが、本研究で提案した手法も有用である。これについては、6.2節で述べる。

第三の問題点に関しては、品詞タグ付け作業の際に、過去に蓄積した品詞タグ付きコーパスを参照することにより解決できる。第四の問題点に関しては、Dagan ら [7] の Committee-Based Sampling の考え方を用いることにより解決できる。これらを含め、6.3節で、統合的な品詞タグ付けコーパス作成支援環境についての展望を述べる。

6.1 グラフィカルユーザインターフェース

ここでは、形態素解析システムのグラフィカルユーザインターフェースである ViJUMAN について解説する。図 10に実行画面を示す。

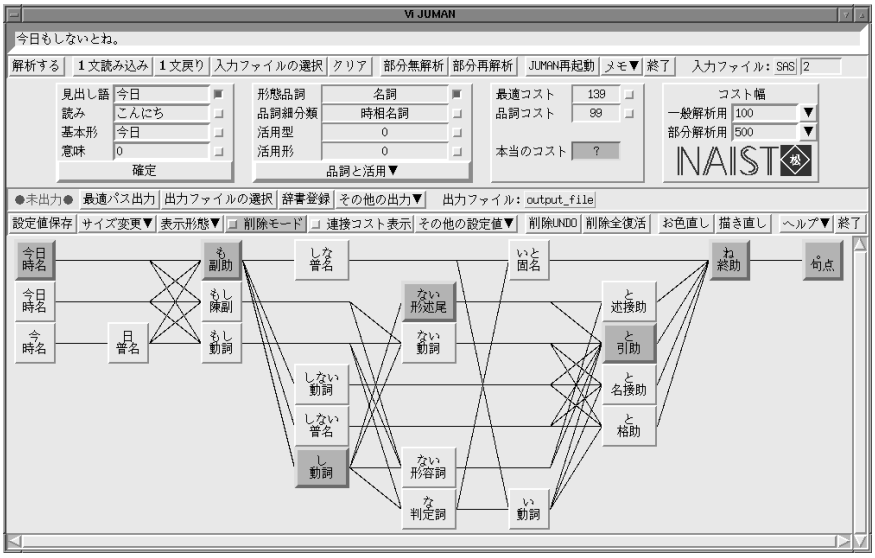


図 10 ViJUMAN

これは、形態素解析結果をラティス状のグラフとして図示し、そのグラフに対してマウスで操作を行うという方針により、品詞タグ付きコーパス作成作業を簡易化したものである。

システムの特徴を以下に挙げる。

- 形態素解析の結果をグラフ状に図示することにより曖昧性が視覚的に理解できる。
- グラフ状に図示された解析結果からマウス操作のみで、任意にパスを選ぶことができる。さらに文の一部だけ制約を緩めて再び解析する機能も実装されており、最初の解析結果で得られなかった形態素を出現させることもできる。
- メニュー選択方式により、システムの品詞体系に基づき矛盾なく品詞情報を選択することができる。これにより新単語の登録、品詞や活用の修正などが容易に行うことができる。

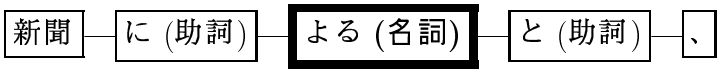
従来のテキストエディタベースの環境と比べ、理解容易性、操作性などの点で非常に優れている。

6.2 フィードバック

同じような誤りが繰り返し現れ、作業者に心理的負担を与えるという問題は、蓄積される品詞タグ付きコーパスの増加に伴い、形態素解析システムの精度を漸近的に向上させることにより解決できる。

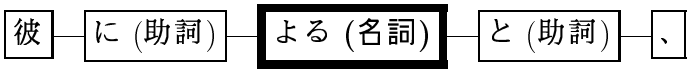
これについては、形態素解析結果データを修正する度に修正規則を学習し次の解析から即座に活用するという方法を文献 [27] で提案した。

例えば、「新聞によると、」の形態素解析結果が、



であった場合、Vi JUMAN 上で、

に修正しても、似た句「彼によると、」を含む文を解析すると、やはり、



となってしまい再度同様の修正をする必要がある。これは作業者にかなりの心理的負担を与える。そこで、直前の形態素が「に (助詞)」で、直後の形態素が「と (助詞)」のとき、「よる」は必ず「動詞」にするというような修正規則を蓄積していき、次の解析から即座に活用する。

また、本研究で提案した手法は、小規模コーパスからの学習結果でも解析精度の向上に有効に利用でき、形態素解析システムの漸化的精度向上に適している。修正規則による方法と異なり学習結果は即座に反映される訳ではないが、ある程度の量の品詞タグ付きコーパスが整備される度に、そのコーパスを対象として学習していけば、解析精度が全体的に上がっていくという利点がある。先の修正規則によるフィードバックと組み合わせることにより、さらなる作業効率の向上が期待できる。

6.3 展望

Vi JUMAN を中心に、本研究で提案した手法などを統合した品詞タグ付けコーパス作成支援環境の実現を目指す。

図 11に概観を示す。以下、この図について解説する。

ViCha 今後公開を予定している Vi JUMAN の「茶筌」対応版である。

タグ付きコーパス検索システム 大規模品詞タグ付きコーパスの作成では、複数の人数が作業に従事する。ここで、作業者間で、品詞付与の一貫性が無いという問題が起こり得る。これは、一般に、品詞付与に関して明確化された信頼できる基準が存在しないためである。このような状況で、作業者が品詞付与に迷った場合、過去に蓄積された品詞タグ付きコーパスを規範とするのが一番確実な方法である。そこで、過去に蓄積した品詞タグ付きコーパスを柔軟に検索できるシステムが必要となる。

フィルタ 品詞タグ付きコーパスの作成の際、全ての文の解析結果を人手で確認するのは、大変な労力を要する。そこで、解析結果に誤りが含まれている可能性のある文のみを人手で確認・修正を行うようにし、解析結果に誤りが含まれている可能性のない文に関しては、形態素解析システムの解析結果

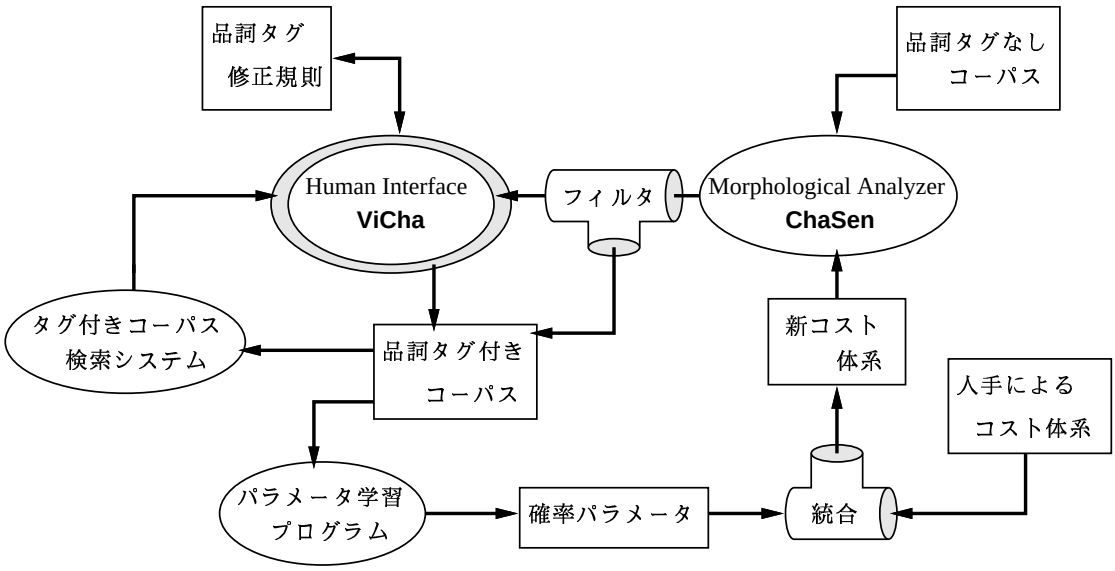


図 11 品詞タグ付けコーパス作成支援環境

をそのまま品詞タグ付きコーパスのデータとすることを考える。誤りが含まれている可能性は、Dagan ら [7] の Committee-Based Sampling の考え方を用いて調べることができる。この論文において提案されているのは、品詞タグ付きコーパスから Markov 学習を行い HMM による英語の構文解析システム獲得する際、そのコーパス全部を学習に使ってしまうのではなく、学習効果の大きい文のみを選択し学習を行う方法である。学習効果が大きいということは、誤りが含まれている可能性が高いと見なせるので、人間が確認すべき文の自動選択に利用できる。「フィルタ」は、これを実現する機能である。

統合 本研究で提案した手法に基づき、コストと確率パラメータの統合を行うシステムである。

7. まとめ

本研究では、形態素解析における規則と確率モデルの統合手法を提案した。これは、人間の観点からおおまかに捉えた言語現象 (人手による規則・コスト) と品詞タグ付きコーパスから学習される細かい言語現象 (確率パラメータ) を統合することにより、保守・管理の困難さ、コーパスの量、データスパースネスといった既存の形態素解析システムの問題を克服する手法である。

この手法を形態素解析システム「茶筌」[24] 上に実装し、実験を行った。その結果、この手法は、人手により調整されたコストや、コーパスから学習された確率パラメータを用いた既存の手法よりも、解析精度において優位であることが示された。特に、学習に用いる品詞タグ付きコーパスが少規模の場合において、他の手法と比べて、著しく解析精度が高くなることが確認された。

しかし、この手法の重要なパラメータである統合比率[12] の最適値をどのように求めるかという問題が残されている。コーパスの量と関係すると予想されるので、大規模で良質の品詞タグ付きコーパスを用いた実験が今後の課題となる。

現在さまざまな日本語品詞タグ付きコーパスが公開されており、量の点では問題は無いが、品質という点で満足のいくものは入手困難である。高品質な品詞タグ付きコーパスの整備は、形態素解析の分野において今後一番重要な課題となるであろう。その点を踏まえて、第 6章では、この手法を用いた統合的品詞タグ付きコーパス作成支援環境の提案を行った。この環境の整備と高品質な品詞タグ付きコーパスの作成も今後の課題となる。

[12]人手によるコストと品詞タグ付きコーパスから学習された確率値を統合する際の比率。

謝辞

主指導教官である松本裕治教授には、研究内容など多くの点で有益な御意見、御助言を頂きました。先生の熱心な御指導がなければ、本研究を進めることはできなかったと思います。ここに心から深く感謝致します。

西田豊明教授には、お忙しい中、副指導教官になって頂き、感謝致します。また、研究計画についての有益な御意見、御助言を頂き、心から感謝しています。

伝康晴助教授には、さまざまな議論に付き合って頂きました。そこで得た知識は、本研究を進める上で非常に有益でした。心から感謝しています。

宇津呂武仁助手には、日頃から関連研究の文献などを数多く紹介して頂き、たいへん感謝しています。

宮田高志助手には、忙しい中、論文の校正をして頂き、有難うございました。

これまで一緒に研究してきた松本研究室の皆さんには、公私に渡り、とてもお世話になりました。研究会や勉強会のみならず普段の何気ない会話からも、さまざまな知識を得ることができました。

博士後期課程の今一修さんには、日頃から研究に必要な基礎的知識を教えて頂いたり、論文の校正をして頂き、心から感謝しています。

博士前期課程の今村友明さんには、日頃から計算機環境についての適切な助言を頂き深く感謝しています。

博士後期課程の山本靖さん、中山拓也さん、玉野健一さん、博士前期課程の岩田真琴さん、平尾努さん、平野善隆さん、北内啓さんには、研究の方針、論文の構成などに関する貴重な意見を頂き非常に感謝しています。

秘書の藤本知子さんには、研究生活において非常にお世話になりました。ここに感謝します。

本研究では、日経新聞 CD-ROM 94 年版を利用させて頂きました。新聞記事データの研究利用許諾を頂きました日経新聞社に感謝します。

最後に、御意見、御質問を頂いた全ての方々に感謝致します。

参考文献

[1] James Allen. “Natural Language Understanding (Second Edition)”, Benjamin/Cummings Publishing, 1995

[2] L. E. Baum. “An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Function of a Markov Process”, Ineqealities, Vol.3, pp.1-8, 1972

[3] Eric Brill. “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, December 1995

[4] Eugene Charniak. “Statistical Language Learning”, MIT Press, 1993

[5] Kenneth W. Church. “A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text”, Second Conference on Applied Natural Language Processing, pp.136-143, February 1988.

[6] D. Cutting, J. Kupiec, J. Pedersen, P. Sibun. “A Practical Part-of-Speech Tagger”, Third Conference on Applied Natural Language Processing, pp.133-140, April 1992

[7] Ido Dagan, Sean P.Engelson. “Committee-Based Sampling For Training Probabilistic Classifiers”, Proceedings of the 12th International Conference on Machine Learning, pp.150-157, 1995

[8] Yoshitaka Hirano, Yuji Matsumoto. “A Proposal of Korean Conjugation System and Its Application to Morphological Analysis”, PACLIC 11: Proceedings of The 11th Pacific Asia Conference on Language, Information and Computation, pp.229-236, December 1996

[9] Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz. “Building a Large Annotated Corpus of English:The Penn Treebank”, Computational

Linguistics, Vol.19, No.2, pp.313-330, June 1993

[10] Bernard Merialdo. “Tagging English Text with a Probabilistic Model”, Computational Linguistics, Vol.20, No.2, pp.155-171, June 1994

[11] 相場徹, 奥村学. “構文・意味解析と統合した形態素解析に関する研究”, EDR 電子化辞書利用シンポジウム論文集, pp. 41-48, July 1995

[12] 井佐原均, 元吉文男, 徳永健伸, 橋本三奈子, 荻野紫穂, 豊浦潤, 岡隆一. “RWC における品詞情報付きテキストデータベースの作成”, 言語処理学会第 1 回年次大会発表論文集, pp.181-184, March 1995

[13] 北研二, 中村哲, 永田昌明. “音声言語処理 —コーパスに基づくアプローチ—”, 森北出版, 1996

[14] 小松英二, 安原宏. “コスト最小法形態素解析のコストルール作成実験”, 情報処理学会研究報告, 95-NL-105, pp.1-6, Janualy 1995

[15] 人工知能学会編. “人工知能ハンドブック”, p.226-227, オーム社, 1990

[16] 竹内孔一, 松本裕治. “HMM による日本語形態素解析システムのパラメータ学習”, 情報処理学会研究報告, 95-NL-108, pp.13-19, July 1995

[17] 田中穂積. “パージング —制約統合型モデルの提案—”, 人工知能学会誌, Vol.11, No.4, pp. 507-513, July 1996

[18] 長尾真編. “自然言語処理”, p.117-137, 岩波書店, 1996

[19] 永田昌明. “EDR コーパスを用いた確率的日本語形態素解析”, EDR 電子化辞書利用シンポジウム論文集, pp. 49-56, July 1995

[20] 朴哲済, 季鐘赫, 季根培. “統計モデルによる日本語の形態素解析手法”, 情報処理学会研究報告, 95-NL-109, pp.19-26, September 1995

[21] 久光徹, 新田義彦. “接続コスト最小法による形態素解析の提案と計算量の評価について”, 信学技法, Vol.90, No.116, pp.17-21, 1990

[22] 平沢克宏, 吉田敬一. “確率モデルを用いた日本語形態素解析”, 情報処理学会第 53 回全国大会 講演論文集 (2), pp.5-6, September 1996

[23] 益岡隆志, 田窪行則. “基礎日本語文法 —改定版—”, くろしお出版, 1992

[24] 松本裕治, 北内啓, 山下達雄, 今一修, 今村友明. “日本語形態素解析システム『茶筌』 version1.0 使用説明書”, NAIST Technical Report, NAIST-IS-TR97007, February 1997

[25] 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾真. “日本語形態素解析システム JUMAN 使用説明書 version 2.0”, NAIST Technical Report, NAIST-IS-TR94025, July 1994

[26] 山下達雄, 松本裕治. “形態素解析視覚化システム ViJUMAN 使用説明書 version 1.0”, NAIST Technical Report, NAIST-IS-TR96005, February 1996

[27] 山下達雄, 松本裕治. “形態素解析結果の視覚化システム ViJUMAN とその学習機能”, 情報処理学会研究報告, 95-NL-110, pp. 71-78, September 1996

[28] 山本幹雄. “Untagged-corpus を用いた形態素解析用 HMM パラメータの一推定法”, 言語処理学会第 2 回年次大会発表論文集, pp.61-64, March 1996

参考文献の入手先

- NAIST Technical Report [24] [26] [25] は http://isw3.aist-nara.ac.jp/IS/TechReport2/ で入手できる。
- 形態素解析システム「茶筌」のマニュアル [24] は、配布パッケージにも含まれている。「茶筌」に関する情報は http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html を参照のこと。