自然言語処理　150－23
(2002. 7. 16)

中国語形態素解析に対するSVMとコスト最小法の比較実験

吉田　辰巳†　大竹　清敬††　山本　和英††

† 〒441-8580 豊橋市天伯町雲雀ヶ丘1-1，豊橋技術科学大学 知識情報工学系
†† 〒619-0288 京都府相楽郡精華町光台2-2-2，ATR音声言語コミュニケーション研究所
E-mail: †gaizi@smlab.tutkie.tut.ac.jp, ††{kiyonori.ohtake,kazuhide.yamamoto}@atr.co.jp

あらまし　現在入手可能なツールと言語資源を用いて中国語形態素解析を行った場合にどの程度の精度が得られるかを報告する．解析ツールにサポートベクトルマシン(SVM)を用いたYamCha，ならびにコスト最小法に基づく形態素解析器としてMOZを用いた．中国語コーパスとしては，最も一般的なPenn Chinese Treebank（10万語）を使用した．これらを組み合わせて，形態素解析実験を行った．この結果，YamChaによる形態素解析精度は約88%でMOZよりも4%以上高いが，実用的には計算時間に問題があることが分った．また，より大きなタグ付きコーパスとして人民日報タグ付きコーパス（110万語）を用いて解析実験を行ったところ，YamCha, MOZそれぞれの解析精度は92%, 89%となった．

キーワード　中国語形態素解析, SVM, YamCha, MOZ.

Comparative Experiments of Chinese Analyzers between Support Vector Machines and Minimum Connective Costs Method

Tatsumi YOSHIDA†, Kiyonori OHTAKE††, and Kazuhide YAMAMOTO††

† Dept. of Knowledge-based Information Engineering, Toyohashi University of Technology,
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi-shi, Aichi, 441-8580 Japan.
†† ATR Spoken Language Translation Research Laboratories,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan.
E-mail: †gaizi@smlab.tutkie.tut.ac.jp, ††{kiyonori.ohtake,kazuhide.yamamoto}@atr.co.jp

Abstract　We will report performances of the Chinese morphological analyzers using analysis tools and language resources, each of which is currently available to the public. We use YamCha, a tool based on Support Vector Machines, and MOZ, which based on minimum connective costs method. We employ the Penn Chinese Treebank (100 thousand words), known as the most common Chinese language resource. Combining these tools and the resource, we measure the performances of Chinese morphological analysis, i.e., word segmentation and part-of-speech tagging. We found that the accuracy using YamCha attains around 88%, which is over 4% higher than that of MOZ, although it is computationally very expensive. We also employ the tagged corpus of Renmin Ribao (1.1 million words) that is bigger than the Penn Chinese Treebank. We found that the accuracies of morphological analysis by YamCha and MOZ attain around 92% and 89%, respectively.

Key words　Chinese morphological analysis, SVM, YamCha, MOZ.

1. はじめに

自然言語処理を進める上で，形態素解析器をはじめとする言語解析器は，コーパスなどの言語資源と同様に最も重要な道具である．近年では，この重要性は研究者間でほぼ認識されており，英語や日本語に対する形態素解析器は多数作成，そして公開または市販され，我々研究者はその恩恵に預っている．

ところが，中国語に関しては状況が同じでない．我々の知る限り日本国内はもちろん，中国においても誰もが手軽に使える中国語解析器が研究者の間で広範に知られている，という状況にはなく，まだ十分にツールが整備されているとは言えない．

この背景の一つは，中国語解析の困難性であると考える．中国語では英語のような単語の分かち書きを行わない．また，日本語では文字種が単語分割のための大きな情報を持つが，中国

語はほぼ単一文字種（漢字）である．さらに，複数品詞を持つ語が多いため品詞付与も容易ではない．例えば，中国語の介詞（前置詞）のほとんどは動詞からの転成であるため，内容語と機能語との間で品詞付与の曖昧性が生じる．これは，日本語や英語ではほとんど生じない現象である．また，日本語における「－する」（動詞）「－い」（形容詞）などの明確な文法標識を持たないため，内容語間の曖昧性も比較的多い．

そこで，我々は，現在入手可能な解析器や言語資源を組み合わせて中国語の形態素解析を試みた．ここで，中国語コーパスとしては，現在一般的に用いられている Penn Chinese Treebank（以下，CTB とする）を使用した．一方，解析モデルとしてはサポートベクトルマシン（Support Vector Machine，以下 SVM とする）とコスト最小法の 2 つを用いて解析精度を比較し，考察を行った．実際には，SVM を用いたツールとして YamCha [1] (注1) を，連接コスト最小法に基づく形態素解析器として MOZ [2] (注2) を使用した．さらに，コーパスの大きさによる解析精度の違いを調べるために，人民日報タグ付きコーパスを用いての形態素解析実験も行った．

本報告の主な目的は，上記のツールと言語資源を用いて中国語の形態素解析器を構築した場合，どの程度の解析精度が得られるのかを報告することにある．すなわち，中国語処理に携わる研究者にとってこれらのツールがどの程度有用であり，使用の際にはどのような点に注意が必要か，などを報告することに主眼がある．

2. 中国語形態素解析

中国語形態素解析器として，我々は YamCha と MOZ を用いた．両者ともに一般公開され，学習用タグ付きコーパスがあればこれらのツールを用いることにより，容易に中国語形態素解析器を構築できる．

2.1 YamCha による中国語形態素解析

YamCha はすでに述べたように SVM に基づく解析器である．SVM は，d 次元の特徴ベクトル（パターン）$\mathbf{x}$ を定められた二つのクラス（A，B）のいずれかに識別する 2 値クラスの線形識別器である．カーネルトリックと呼ばれる計算技術によって非線形識別器を実現することもできる．従来の手法と比べて多くの面で優位性を示し，文字認識や画像認識など，様々な分野で応用されている．

2 値識別器である SVM を多値クラスの識別問題に適用するため，YamCha では *pairwise classification* [3]（一対比較分類）と呼ばれる手法を採用している．これは，K クラスの識別問題を解くために，各クラス 2 つの組合せを識別する $K\times(K-1)/2$ 種類の識別器を作成し，最終的にそれらの多数決でクラスを決定する手法である．

SVM を用いた自然言語解析の例として英文の基本句同定実験 [4] や，日本語の係り受け解析実験 [5] があり，従来手法と比較して高い解析結果を示している．また，平と春野は，SVM を用いた文書分類について，高い分類精度を得るためには品詞によるフィルタリングをした後，全単語を入力として用いればよいことを示している [6]．

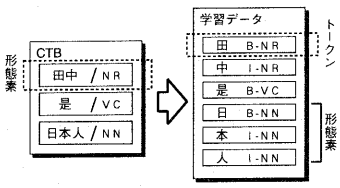

図 1 YamCha 用中国語形態素解析データ書式

YamCha で扱うデータ形式は，複数のトークンと複数のカラムから構成される．各行は入力のトークンに対応する．形態素解析を行う場合は，1 トークンが 1 文字に対応する．各カラムにはトークンに付与された属性が記述される．YamCha によって推定（学習）すべき属性は最後のカラムに与える．ここでは，形態素の要素である 1 文字を第一カラムに記述し，第 2 カラムに YamCha で推定する情報を記述する．この情報には，形態素の区切り位置を示す情報と，形態素に付与する品詞情報の両方が含まれる．

トークンが形態素に含まれるか否かの状態を示すために IOB2 モデルを用いた [7]．これは，あるトークンが形態素（一般的にはトークンをまとめあげる対象となるチャンク）の先頭ならば B タグを付与し，形態素に含まれる先頭以外のトークンならば I タグを付与し，形態素に含まれない場合には O タグを付与するモデルである．形態素解析では，すべてのトークンが何らかの形態素に含まれるため，結果的に O タグは用いられない．

付与する品詞タグセットは CTB のタグセットと同一である．また，CTB において品詞が “-NONE-” の形態素は構文構造上形式的に配置され，実体を持たないため対象外とする．最終的にトークンに付与されるタグは B/I タグと品詞タグを “–” で結んだものとなる．CTB から YamCha で中国語解析を行うための書式へ変換する概要を図 1 に示す．

以上の処理で得られた学習データを YamCha に与え，SVM のモデルを作成する．その際に，素性として使用したデータは YamCha の標準設定に従った．すなわち，推定するトークンとその前後 2 トークンの合計 5 トークンにおける文字データと，前方 2 トークンにおける推定タグとを素性として学習した．解析方向は前方からである．これは，使用する素性の数，および解析の方向を変化させた予備実験の結果から，YamCha の標準設定が最も高い精度であったためである．

また，YamCha で学習を行うために用いた SVM の実装は同じく工藤が公開している TinySVM 0.08 (注3) である．

2.2 MOZ による中国語形態素解析

本節では，MOZ による中国語形態素解析の概略を述べる．

MOZ はコスト最小法に基づく解析器である．そのため，コストを与えた形態素辞書と接続表が必要である．ここでは，CTB から得られる情報（品詞 2 つ組の頻度や形態素の頻度など）か

(注1)：http://cl.aist-nara.ac.jp/~taku-ku/software/yamcha/
(注2)：http://cl.aist-nara.ac.jp/student/tatuo-y/ma/
(注3)：http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM/

ら形態素辞書と接続表，ならびにそれらのコストを求める．なお，中国語には活用がないためタグ付きコーパスからこれらの辞書を作成するのは非常に容易である．形態素辞書は形態素とその出現確率から，品詞接続表は品詞 bi-gram によって与える．MOZ では，品詞接続表に tri-gram 以上のデータを用いることができるが，データ過疎性 (data sparseness) による精度低下を避けるために，本実験では品詞 bi-gram のみを用いた．

形態素を w_i，品詞を POS_i，x の頻度を $C(x)$ と表記すると品詞が POS_i である形態素 w_i の出現確率を式 (1) で与える．ここで，$C(w_i,\ POS_i)$ は形態素 w_i，かつその品詞が POS_i である頻度を示している．

$$p(w_i \mid POS_i) = \frac{C(w_i,\ POS_i)}{C(POS_i)} \tag{1}$$

また，品詞接続表の確率は式 (2) で与える．ここで $C(POS_i,\ POS_j)$ は品詞 POS_i のあとに品詞 POS_j が出現した頻度である．

$$p(POS_j \mid POS_i) = \frac{C(POS_i,\ POS_j)}{C(POS_i)} \tag{2}$$

システムで扱う最高コストを 128 として，コスト化係数を求める．コスト化係数は式 (3) により与えられる．ここで最小確率は，すべての $p(w_i \mid POS_i)$ および $p(POS_j \mid POS_i)$ における最小値である．

$$\text{コスト化係数} = |\ \text{最高コスト} / \log(\text{最小確率})| \tag{3}$$

形態素辞書ならびに接続表のコストはそれぞれの確率から式 (4) により与えられる．

$$\text{コスト} = \lceil |\log(\text{確率}) \times \text{コスト化係数}\ | \rceil \tag{4}$$

以上の方法により形態素辞書ならびに接続表のコストを計算する．

3. 実　　験

中国語形態素解析に対する YamCha と MOZ の精度を比較するために学習文テスト (closed test)，未知文テスト (open test)，ならびに未知語に対する両解析器の振舞いを調べる実験を行った．さらに，これらの実験結果で示された精度が何に起因するものかを調べるために，より大きなコーパスを用いた実験，日本語の形態素解析に対する YamCha の性能を調べる実験を行った．以下，これらの順に述べる．

3.1 学習文テスト

まず，CTB 全体を学習データ（4181 文(注4)）とし，この中から無作為に抽出した 1 割の文（418 文）を解析する学習文テストを行った．結果を CTB の正解と比較し，再現率（recall）と適合率（precision）を算出した．再現率と適合率はそれぞれ式（5）および式（6）とした．

$$\text{再現率} = \frac{\text{解析結果中の正解形態素数}}{\text{正解形態素数}} \times 100(\%) \tag{5}$$

(注4)：CTB の説明には 4185 文とあるが，我々が発見した明らかな誤り，たとえば句点のみを 1 文とするなど，を除くと 4181 文となった．

表 1　学習文テストの正解率

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	99.91%	99.93%	99.58%	99.60%
MOZ	97.78%	98.82%	93.74%	94.73%

表 2　未知文テスト結果

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	93.04%	93.71%	87.58%	88.20%
MOZ	92.19%	85.89%	86.32%	80.42%

表 3　YamCha での処理時間

	学習	解析
タグの種類	53	—
文数	約 3700	約 400
トークン数	約 15 万 8 千	約 1 万 7 千
処理時間	約 6 時間	約 35 分

$$\text{適合率} = \frac{\text{解析結果中の正解形態素数}}{\text{解析結果形態素数}} \times 100(\%) \tag{6}$$

この結果を表 1 に示す．ただし，本実験では正解形態素数を求める場合に形態素分割のみ正解の場合と，分割ならびに品詞の両方が正解の場合の 2 つの条件を設けて再現率と適合率を求めた．これは，以下の実験でも同様である．

両解析器の品詞誤りに関して分析したところ，それらの上位は本質的に困難な，解析誤りによって占められた．たとえば，動詞と名詞，名詞と固有名詞の間での品詞付与誤りが両解析器ともに目立ち，解析器による大きな差異は見られなかった．

3.2 未知文テスト

次に，CTB 全体を母集団とする 10 分割交差検定（cross validation）による未知文テストを行った．

まず，CTB 全体（4181 文）を無作為に 10 等分し，1 つをテストデータ，残りの 9 つを学習データとする．これを 10 回繰り返し 10 個のテストセットを作成した．

YamCha と MOZ をそれぞれに対して，これら 10 個のテストセットを用いて実験した．

実験結果を表 2 に示す．

未知文テストにおいても，両解析器の品詞誤りに関して分析した結果，それらの上位には，たとえば，動詞と名詞，名詞と固有名詞の間での品詞付与誤りが目立ち，解析器による大きな差異は見られなかった．

未知文テストにおいて YamCha が 1 学習データを学習するために要した処理時間と 1 テストデータを解析するために要した処理時間などを表 3 に示す．測定時は，CPU: Pentium III 600 MHz，メモリ: 256 MB，OS: Linux の計算機を用いた．

一方，MOZ が学習データ（約 3700 文）から形態素辞書と接続表のコストを求めるために必要とした時間は約 3 秒であり，1 テストデータ（約 400 文）を解析するために必要とした時間は約 1 秒であった．

3.3 未知語の性質

次に，テストデータに含まれる形態素のうち，学習データに

表 4　未知文テストにおける形態素数と未知語数

	学習データ	テストデータ	未知語	未知語率
形態素 (のべ)	89748.0	9972.0	702.4	7.05%
形態素 (異なり)	11412.5	3069.2	667.5	21.74%

表 5　未知語と解析精度に関する実験におけるテストセットの形態素数

	形態素数		未知語数		未知語率 (%)	
テストセット	異なり	のべ	異なり	のべ	異なり	のべ
0	2809	10819	572	770	51.49	26.07
1	2892	11402	489	639	44.01	21.63
2	2921	11512	460	594	41.40	20.11
3	3044	11929	337	442	30.33	14.96
4	3075	12134	306	406	27.54	13.74
5	3127	12402	254	337	22.86	11.41
6	3190	12764	191	252	17.19	8.53
7	3265	12993	116	150	10.44	5.08
8	3293	13274	88	115	7.92	3.89
9	3319	13393	62	75	5.58	2.54
10	3381	13773	0	0	0.00	0.00

含まれていないものを未知語と定義し，その性質を調べた．未知文テストにおける学習データとテストデータの形態素数，ならびにその未知語数を表 4 に示す．ここで，表において各語数に端数があるのは，10 回の交差検定の平均を取っているためである．また，未知語率とはテストデータの形態素数に占める未知語数の割合を指す．平均未知語率とはテストセット全体での未知語率の平均を示している．

未知語に対する解析器の性質をより詳しく調べるため，以下の実験を行った．まず，学習データとして 33 記事，テストデータとして 10 記事を，いずれも無作為に CTB から選択する．次にテストデータから順に 1 記事ずつ学習データに加えていき，計 11 個の学習データを作成する．それぞれの学習データに基づく解析器で，同一のテストデータ 10 記事を解析した．

テストデータに含まれる形態素の異なり数は 1111，のべ数は 2954 である．11 のテストセットにおける学習データの形態素，未知語数ならびに未知語率を表 5 に示す．

各テストセットにおける未知語率（異なり）と解析精度の関係を図 2 に示す．図では，解析精度を F 値 (F-measure) で示す．F 値は再現率と適合率の調和平均である．なお，未知語率 (のべ) と精度の関係についても図 2 とほぼ同一の傾向であった．

次に，未知語がある場合の解析結果を調査した．テストセット t_i における未知語の集合を $UK(t_i)$，$w \in UK(t_i)$ のうち形態素分割に成功した形態素の集合を $USeg(t_i)$ とする．さらに，$w \in USeg(t_i)$ のうち品詞も正しく解析された形態素の集合を $USP(t_i)$ とする．これらの集計結果を表 6 に示す．また，MOZ では入力に未知語が含まれる場合，解析不能で停止することはないが，最終的に未知語と判断された文字列を 1 文字ずつ，null という品詞を与えて出力する．したがって，MOZ での解析で $|USP(t_i)|$ を示していないのは，MOZ では未知語が null と解析されるため，$USP(t_i)$ は空集合となるためである．一方，MOZ での解析において $USeg(t_i)$ が得られるのは，正解

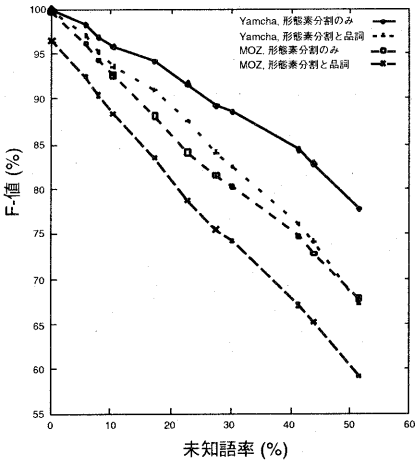

図 2　未知語率 [異なり] と解析精度

表 6　未知語とその解析結果

		YamCha		MOZ
t_i	$\|UK(t_i)\|$	$\|USeg(t_i)\|$	$\|USP(t_i)\|$	$\|USeg(t_i)\|$
0	572	376	249	87
1	489	319	212	78
2	460	299	199	77
3	337	225	157	56
4	306	202	147	47
5	254	174	127	35
6	191	135	95	27
7	116	82	59	18
8	88	60	43	15
9	62	47	35	10

が 1 文字の形態素である場合に形態素分割が成功したと見なすからである．

3.4　コーパスの大きさと言語依存性

これまで，中国語の形態素解析について YamCha と MOZ を比較してきた．しかし，その未知文テストの結果は，表 2 に示す通り，これまでに報告されている日本語の形態素解析器の精度より低い．ここでは，その原因がコーパスの量にあるのか，中国語の言語としての解析の難しさにあるのかを検討する．

3.4.1　コーパスの大きさと解析精度

CTB は既に述べた通り約 10 万語からなるコーパスだが，品詞タグ付きコーパスとしては大きいとは言えない．そのため，10 分割交差検定を行っても，テストセットにおける未知語率が非常に大きくなり，精度が低くなる．高価なデータに対する評価手法として，1 つ取って置き法（Leave One Out，以下，LOO）と呼ばれる方法も考えられる．この方法では，n 個のデータがあった場合に，1 つを除いて $n-1$ 個を用いて学習を行い，残りの 1 つをテストデータとして用いる操作を n 回繰り返す．解析器を評価するために，文を単位とした LOO も考えられるが，YamCha の学習時間を考慮すると非現実的である．

一方，品詞タグ付けされたコーパスであれば，CTB のよう

表 7　PKU での未知文テスト (10 万語)

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	86.67%	87.52%	80.19%	80.99%
MOZ	90.05%	80.58%	84.57%	75.67%

表 8　CTB と PKU の未知語率

	平均未知語率 [異なり/のべ](%)	平均文長	平均形態素長
CTB (10 分割)	21.74 / 7.05	41.10 字	1.72 字
PKU′	26.85 / 10.87	40.71 字	1.64 字
PKU (11 分割)	15.79 / 2.84	41.85 字	1.64 字

ここで，PKU′とは PKU から任意に抽出した 10 万語を 10 分割した場合を示す．

に構文木を備えていなくても，形態素解析のために利用することができる．我々は，CTB よりも大きく，同じ新聞記事である人民日報タグ付きコーパス(注5) のうち，無償公開されている 1 ヶ月分のデータを用いた．人民日報タグ付きコーパス 1ヶ月分（以下，PKU とする）は，43913 文，1118794 形態素からなるコーパスである(注6)．定義されているタグセット(注7)の大きさは 39 である．

PKU は CTB の約 11 倍の大きさを持つ．そこで，まずコーパスの大きさを CTB と同程度とした場合の精度を検証した．PKU をランダムに文単位で 11 等分し，そのうちの 1 つ（約 10 万語）を用いて 10 分割交差検定を行った．結果を表 7 に示す．

PKU の 10 万語を用いての未知文テストと CTB での未知文テストでの精度の違いは，両者のコーパスの違いに起因する．表 8 に平均未知語率などを示すが，この表から，PKU の方が未知語が多く，かつ 1 文あたりの平均形態素数が CTB より約 1 多くなる．したがって，CTB と比較して，PKU 10 万語での結果は低くなった．また，YamCha と MOZ の違いは，タグセットが増えて，タグの推定がより難しくなっていることに加えて，未知語率が大きくなったことにより，YamCha の精度が大きく低下したと考える．MOZ が再現率の点で，YamCha を上回るのは，辞書を用いる利点が活かされているからだと考える．

次に，PKU 全てを学習コーパスとし，学習文テストを行った．テストデータとして PKU から無作為に抽出した 3993 文，101218 形態素を解析した．なお，学習に用いるコーパスが大きくなることから，より大きな分解能が必要になると考え，MOZ のコスト化係数を 128 から 1024 へと変更した．実験結果を表 9 に示す．この結果から，学習コーパスが 100 万語を越えても，YamCha は変わらず高い性能を示していることがわかる．

学習に用いるコーパスの大きさが非常に大きくなった場合の 2 つの解析の振舞いを検討するために，11 等分したデータを用いて 11 分割交差検定を行った．結果を表 10 に示す．

(注5)：http://www.fujitsu.com.cn/support/

(注6)：本来は，44011 文，1121447 形態素からなるコーパスであるが，未定義タグが極少量含まれており，未定義タグを含む文は除いた．

(注7)：http://www.icl.pku.edu.cn/research/corpus/addition.htm

表 9　PKU での学習文テスト

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	99.95%	99.94%	99.76%	99.75%
MOZ	98.72%	99.17%	94.71%	95.14%

表 10　PKU での未知文テスト

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	95.19%	95.19%	91.72%	91.72%
MOZ	95.68%	93.42%	89.87%	87.75%

表 11　京大コーパスの未知語率

平均未知語率 [異なり/のべ](%)	平均文長	平均形態素長
24.18/ 8.38	43.91 字	1.77 字

表 12　京大コーパス未知文テスト結果

	分割のみ		分割と品詞付与	
	再現率	適合率	再現率	適合率
YamCha	92.02%	93.23%	88.17%	89.33%
JUMAN	98.97%	98.65%	93.49%	93.19%

3.4.2　日本語形態素解析における YamCha

CTB で用いられているのは，新華社通信の新聞記事である．そこで，日本語での SVM を用いた形態素解析器の精度を検証するために我々は，京都大学テキストコーパス Version 3.0（以下，京大コーパスとする）(注8) を用いて実験を行った．

CTB の大きさが約 10 万語であるところから，我々は，京大コーパスのうち 1 月 1, 3, 4, 5 日の記事，4117 文，102310 形態素を用いることにした．CTB 全体では，4181 文，99720 形態素である．

我々が選択した京大コーパスの一部について CTB に対する実験と同様に，10 分割交差検定を実施した．この検定における平均未知語率などを表 11 に示す．

京大コーパスを用いた実験における品詞は，JUMAN(注9)が定義する品詞のうち品詞細分類までを含めたものとした．この結果，タグセットの大きさは，41 となり，CTB の 33 より大きい．

実験結果を表 12 に示す．参考までに我々が選択した京大コーパスの一部を JUMAN で形態素解析した結果もあわせて示す．

日本語を対象とした実験でも，同程度の大きさのコーパスでは、同程度の精度となった．

4. 考　　察

以上得られた形態素解析に関する実験結果について考察する．まず，YamCha と CTB を使用した場合の未知文に対する形態素解析（分割と品詞付与）の精度（F 値）は 87.9% であった．同条件で MOZ が 83.3% であることを考えると，言語資源として CTB しか得られない条件下においては YamCha を使用したほうが高精度な解析器を実現できる．

(注8)：http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/corpus.html

(注9)：http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/juman.html

次に，解析時間については YamCha が極端に遅い．学習時間も同様である．よって解析時に実時間性を問われる状況においては MOZ を使用すべきである．YamCha では，既に述べたように，一対比較分類に基づき品詞付与を行うため，品詞数が大きくなると，その 2 乗に比例する SVM が必要となる．そのため，品詞数の増加とともに，学習，解析時間が増大する．

品詞付与誤りの傾向では，1 節で述べたように中国語において本質的に解析の難しいと予想される箇所で両解析器共に誤っている．解析器としての誤りのくせはあまり見受けられない．

未知語に対する頑健性については，原理的に YamCha のほうが明らかによい．実験では，YamCha は未知語の約 4 割を正しく解析しており，頑健性を確かめられた．その割合は，未知語率が変化しても，大きく変化することはなく，実験した範囲の未知語率（51.1% から 6.5% ）で，40% から 45% 程度であった．このことから，未知語率が大きくなったからといってそれに影響されて極端に精度が低下することはないと予想する．一方，MOZ は，未知語に対して 1 文字ずつ null という品詞を付与して出力するのみであるため，何らかの拡張を行わない限り品詞の推定を行えない．したがって，再現率に対して適合率が低くなる傾向がある．また，YamCha にはこのような傾向はなく，若干適合率が再現率を上回るという傾向を示す．これらのことから，入力文中に多くの未知語の存在が予想される場合，あるいは学習データの語彙傾向と異なる入力文を解析する場合は YamCha を用いたほうがよい．

ただし，一般的な状況としてコーパスとは別個に単語集合を入手できる場合がある．この場合には MOZ を使うべきであろう．YamCha では単語集合があってもこの情報を学習に反映させることができず，コーパス中の出現単語のみが学習対象であるためである．

言語資源をより活用しているのは YamCha であるが，辞書を用いないことから言語的整備ができない．また，人間の内省による知見を反映させにくい．したがって，既に大量のタグ付きコーパスが存在する状況では，MOZ のような，接続コストを統計的言語モデルに基づいて推定する手法が頑健で，整備しやすい道具となると考える．逆に，タグ付きコーパスが充分に整備されていない言語の解析器を必要とする，あるいは，新たに品詞を定義したが，その品詞で解析されたコーパスが充分に存在しない状況にもかかわらず，その解析器を必要とする場合には，YamCha が有効である．

一方，より大きなコーパスを用いることにより，より高度な解析器が実現可能であることが，表 7 および表 10 からわかる．また，表 9 に示した学習文テストの結果から，学習コーパスをさらに大きくすると YamCha はさらに精度を向上させる可能性があることがわかる．それに対し，MOZ は現状の枠組のままでは，分割と品詞付与の F 値で 95% 程度がその性能の限界だと考える．これをさらに向上させるためには，接続表への tri-gram 規則の適用ならびにその補完などが可能である．しかし，浅原らは，中国語の場合には tri-gram の規則自体があまり有効ではなく，品詞体系の詳細化が精度の向上に寄与することを実験結果から予測している [8].

また，中国語に固有の解析の難しさが考えられるが，日本語を対象とした実験の結果から，コーパスの大きさが同程度の場合は，顕著な差が現れなかった（表 2 と表 12）．しかし，京大コーパスの平均未知語率が CTB のそれと比較して大きいことや（表 11 と表 8），京大コーパスのタグセット（41）が CTB のそれ（33）より大きいことを考慮すると，中国語解析が日本語解析に比べて難しいと判断する．さらに，表 2 と 12 を比較すると，形態素分割のみと，分割および品詞付与との間に逆転現象がある．これは，中国語解析の困難な点が品詞付与にあるという我々の予見を裏付ける結果と考える．

5. むすび

本報告では，Penn Chinese Treebank を言語資源とした時に，サポートベクトルマシン（SVM）とコスト最小法を用いてどの程度の中国語解析精度が得られるのかを報告した．

SVM による形態素解析精度は形態素単位で約 88% であり，MOZ よりも 4% 以上高い．ただし，計算量に問題があり，解析時間，学習時間共に非常に長い．

一方，大量のタグ付きコーパスが入手できる場合は，YamCha，MOZ のいずれを用いても，さらに高精度の解析器（110 万語のコーパスの場合，F 値でそれぞれ約 92%，89%）を実現できる．

本研究は通信・放送機構の研究委託により実施したものである．

文　献

[1] T. Kudo and Y. Matsumoto: “Chunking with support vector machines”, Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (2001).
[2] 山下, 松本：“言語に依存しない形態素解析処理の枠組”, 自然言語処理, **7**, 3, pp. 39–56 (2000).
[3] U. H.-G. Kreßel: “Pairwise classification and support vector machines”, Advances in Kernel Methods (Eds. by B. Schölkopf, C. J. Burges and A. J. Smola), The MIT Press, pp. 255–268 (1999).
[4] 工藤, 松本：“Support vector machine を用いた chunk 同定”, 情報処理学会研究報告 2000-NL-140, pp. 9–16 (2000).
[5] 工藤, 松本：“チャンキングの段階適用による係り受け解析”, 情報処理学会研究報告 2001-NL-142, pp. 97–104 (2001).
[6] 平, 春野：“Support vector machine によるテキスト分類における属性選択”, 情報処理学会論文誌, **41**, 4, pp. 1113–1123 (2000).
[7] E. F. T. K. Sang and J. Veenstra: “Representing text chunks”, Proceedings of EACL'99, pp. 173–179 (1999).
[8] 浅原, 松本：“形態素解析のための拡張統計モデル”, 情報処理学会論文誌, **43**, 3, pp. 685–695 (2002).