研究のツールボックス 【1】

茶筌と南瓜による日本語解析
—構文情報を用いた文の役割分類—

Japanese Sentence Analysis by ChaSen and Cabocha
— Using Syntactic Information for Sentence Role Classification —

松本 裕治 Yuji Matsumoto
奈良先端科学技術大学院大学
Nara Institute of Science and Technology.
matsu@is.naist.jp, http://cl.naist.jp/

高岡 一馬 Kazuma Takaoka
株式会社ジャストシステム
Justsystem Corporation.
kazuma_takaoka@justsystem.co.jp

浅原 正幸 Masayuki Asahara
奈良先端科学技術大学院大学
Nara Institute of Science and Technology.
masayu-a@is.naist.jp

工藤 拓 Taku Kudo
NTTコミュニケーション科学基礎研究所
NTT Communication Science Laboratories.
taku@cslab.kecl.ntt.co.jp

Keywords: morphological analysis, dependency parsing, unknown word detection, named entity extraction, sentence role classification.

1. はじめに

大規模な電子化言語データ（コーパス）が利用可能になり，コーパスからの機械学習を用いた言語処理システムの性能が実用レベルに達するまで進展している．本稿では，我々が開発している次の三つの日本語処理システムを紹介する．日本語形態素解析システム「茶筌」は，日本語文を単語に分割し，品詞，読み，活用形などの情報を単語に付与する．日本語は単語を分かち書きせずに記述されるため，これは日本語文には避けられない処理である．現実的な文解析には，未知語（辞書に未登録の単語）の出現を無視するわけにはいかない．また，固有名詞，日付，数値表現など，特殊な役割をもつまとまった表現が数多く出現し，これらの適切な前処理が，さまざまな言語処理応用にとって重要である．「bar」というシステムは，文字単位のまとめ上げをSupport Vector Machinesに基づく学習によって行う．日本語文の表層的な統語処理として文節係り受けがよく利用される．「南瓜」は，日本語の文節間の係り受け解析を行うシステムであり，係り受け解析されたコーパスから係り受け規則をSupport Vector Machinesによって学習することによって高精度の日本語係り受け解析を実現している．

文書の大まかなトピックを分類する文書分類（text classification）の研究では，文書を単語の集まり（bag of words）と見て，分類タスクを行うのが普通であり，文や文章の構造を考慮に入れることは，分類性能にさほど貢献しないとみなされてきた．近年，アンケート結果の解析やWeb上の評判情報の検索を目的として，文の主観性・客観性の判定や，文がある事柄の良否のいずれを評価しているかなど，より深い意味，あるいは著者の意図に踏み込んだ文の分類が注目を集めつつある．文書や文を単語の集まりとして捉えるだけでは，このような目的を達成することは困難であり，文のより詳細な解析が必要となる．本稿の後半では，前半で紹介した言語処理ツールを用いることによって，このような文分類のタスクを行う事例を紹介する．

2. 使ってみよう日本語処理ツール

本章では，我々が開発している基本的な日本語解析ツールを紹介する．日本語処理では，文を単語に分割し，必要に応じて語尾処理を行い，個々の単語の品詞を推定する形態素解析を避けて通ることができない．「茶筌」は日本語の形態素解析を行うツールである．ここでは，従来の茶筌の機能，および，最近実装された制約付き解析機能を紹介する．Web上のページや現実的な文書には，既存の辞書には登録されていない新しい語，通常とは異なる表記をもつ語，専門用語や固有名詞など新しく生まれる語などがあり，閉じた辞書に基づくシステムでは解決

できない問題がある．文字列を対象としてまとめ上げ処理を行う「bar」は，その学習モデルを変えることによってさまざまな言語表現を抽出することができる．ここでは，未知語および固有表現の自動抽出について説明する．日本語文は，文節という最小の統語単位に分割し，それらの間の係り受け構造として解析することができる．「南瓜」は，文節間の係り受け関係を同定し，係り受け解析木を生成するツールである．

これらのいずれのツールもタグ付き事例からの機械学習に基づいている．茶筌は，可変長の隠れマルコフモデルに基づいており，barおよび南瓜は，Support Vector Machinesを学習に用いている．これらのシステムは，日本語解析において現状の最高水準の精度を達成している．

2・1　茶筌を用いた形態素解析

ここでは茶筌[松本 00, 松本 03]を用いた日本語の形態素解析とそれに付随する処理について解説する．形態素解析は入力を単語（形態素）単位に分かち書きし，それぞれに対して品詞情報などを付与することをいう．茶筌では文脈長可変の隠れマルコフモデルに基づく言語モデルをコスト最小法の枠組みに置き換えて処理を行っている．

例として，茶筌は「すももももももものうちだ.」という文を入力すると次のような解析結果を出力する（空白区切りはタブ記号である）．

```
すもも  スモモ  すもも  名詞-一般
も      モ      も      助詞-係助詞
もも    モモ    もも    名詞-一般
も      モ      も      助詞-係助詞
もも    モモ    もも    名詞-一般
の      ノ      の      助詞-連体化
うち    ウチ    うち    名詞-非自立-副詞可能
だ      ダ      だ      助動詞特殊・ダ 基本形
.       .       .       記号-句点
EOS
```

各行が一つの形態素を表し，それぞれの列は，入力文での形，読み，基本形，活用型，活用形となっている．EOSは文末（End Of Sentence）を表す．

茶筌が出力できる情報はほかにもいくつかあり，-Fオプションを用いることで出力する情報や出力のフォーマットを制御することができる．詳細はマニュアルを参照されたい．

また，デフォルトで出力されるのは最尤（コスト最小）の解のみだが，-pオプションを用いることですべての解を得ることが可能である．このとき-wオプションを用いて解探索のビーム幅を指定すれば解の生成が制御できる．

§1　前処理の必要性

茶筌は強力な形態素解析ツールであるが，より精度の良い解析結果を得るためには入力をあらかじめ加工しておくことが必要な場合がある．

まず，入力から言語以外のテキストを取り除く．HTMLやXMLのタグ，レイアウトのための空白や改行，罫線，絵文字などの視覚的な効果のために付加されている部分が残っていると，その部分も何らかの形態素として解析されるために意図しない結果となるためである．特にいわゆる半角スペースは形態素の区切りとして認識されるため，HTMLやXML文書においては注意が必要である．

また，茶筌の形態素解析は文を単位に行われるが，茶筌自体はテキストから文を抽出する機能をもっておらず，前処理として入力を1文1行に加工する必要がある．近似的な処理として茶筌に-jオプションを用いることで特定の文字（デフォルトでは「. . ?!」の4文字）を文末とみなして処理を行うこともできるが，引用文や見出しの部分で文の範囲を誤ってしまうことが多い．元のテキストがもつレイアウトやタグ情報を利用して，より確実な文分割処理を行っておくのが望ましい．

§2　制約付き解析

現在の茶筌の最新版は，バージョン2.3.3*1であるが，茶筌バージョン2.4系列からの新機能として制約付き解析がある．この機能を用いると，入力文の一部の形態素情報が既知である，あるいは形態素の境界がわかっているときに，それを満たすように解析を行うことができる．

例として以下のような入力を考える（ここではタブを“\t”によって明示した）．

```
すも
もも \t モモ \t もも \tUNSPEC
も
もも
も \t モ \t も \t 名詞-一般
のうちだ.
```

この例は「酢も桃も桃も藻の内だ.」という文を意図している．

入力の各行をセグメントと呼び，セグメントの境界は必ず形態素の境界となるよう解析され，この境界をまたぐような形態素は解の候補として生成されない．この例では「すも」と「もも」をまたぐ「すもも」という形態素候補の生成は禁止されることになる．

2列目以降の形態素情報が与えられているセグメントは，その部分が必ずその形態素となる制約を表す*2．品詞情報は茶筌のデフォルトの出力と同じく各細分類を‘-’でつないで表すが，辞書の品詞定義（grammar.chaに記述される）にないものは書くことができない．

また，形態素情報が「UNSPEC」である場合はこのセグメントが一つの形態素であるという制約はあるが実際にどのような形態素であるか同定はしないという指示となる．解析時にはこのセグメントの見出しと一致するものを辞書から検索し，複数の品詞の可能性があれば，前後文

*1 2004年2月現在.
*2 読み，基本形の情報は茶筌の通常の解析結果との互換性をとるためにあって，実際には解析に使用されない.

脈から最適な品詞が選択されることになる.

形態素情報が与えられていないセグメントの内部は,まったく制約のない通常の解析と同様に処理される.したがって,「ももも」というセグメントからは「もも/も」「も/もも」「も/も/も」という候補が生成される.

この入力を-sオプション(制約付き解析モード)を用いて処理すると次のような出力が得られる.

```
す      ス      す      名詞-一般
も      モ      も      助詞-係助詞
もも    モモ    もも    名詞-一般
も      モ      も      助詞-係助詞
もも    モモ    もも    名詞-一般
も      モ      も      助詞-係助詞
も      モ      も      名詞-一般
の      ノ      の      助詞-連体化
うち    ウチ    うち    名詞-非自立-副詞可能
だ      ダ      だ      助動詞特殊・ダ 基本形
.       .       .       記号-句点
EOS
```

これは上で述べたような制約を満たす最尤解として出力されており,意図したとおりの結果となっている.

この制約付き解析機能は次節で述べる未知語や固有表現抽出,あるいは統語処理との連携の目的で導入されたものである.これらの処理で同定された部分を制約として再度形態素解析を行うことでより高精度の解析が可能となると思われる.

また,ごく簡易な固有表現抽出処理としてRubyやPerlなどのスクリプト言語の正規表現を用いてURLやE-mailアドレスを事前に括り出し,それらを固定した形で制約付き解析を行うといった使い方も考えられる.

あるいは別の応用例として,ある品詞体系で作成されたコーパスを別の品詞体系へ変換することにも利用可能であろう[松田 99].

2・2 チャンキングによる未知語および固有表現抽出

前節では,日本語の文を単語に分割する形態素解析器「茶筌」について解説した.現在の茶筌は,カタカナ語などに対する,字種による簡単な連結以外には,辞書に登録されていない語(未知語)をうまく解析することができない.茶筌の辞書であるipadicには約25万語の単語が登録されているが,新聞記事をもとにして構成されているため,ひらがな語が少なかったり,新しい固有名詞が含まれていなかったりする.テキストによっては,現在の辞書には登録されていないため形態素解析器にとっての未知語が多く出現したり,固有名詞や数量表現などが多く出現したりする.これらをカバーするために,本節ではチャンキングという技術を使ってテキスト中から未知語や固有表現を抽出するツールについて解説する.

チャンキングとは,トークン列からトークンのかたまり(チャンク)を抽出する手法のことをいう.主として,各トークンにチャンクの開始位置や終了位置を付与するポジショニングという手法が用いられ,隠れマルコフモデルをはじめとして多くの手法が提案されている.このチャンキングの技術を利用して,未知語抽出器や固有表現抽出器を構成することができる.未知語抽出器は,形態素解析器が未知語に遭遇した場合に出力するパターンを,チャンカーが学習し未知語出現箇所を推定する.同様に固有表現抽出器は,固有表現出現箇所を推定する.いずれの場合も,未知語や固有表現が正しく同定されたタグ付きデータが学習のために必要である.未知語については,辞書の一部を削除することによって擬似的に未知語をつくり,その解析パターンを学習データとして用いる.

bar [Asahara 03]は,テキスト中に出現する未知語や固有表現を抽出するツールである.内部では前節で説明した茶筌と,Support Vector Machinesにより決定的にポジショニングを付与する汎用チャンカーYamCha [Kudo 01]を用いている.barはYamCha用のモデルを含んでおり,ユーザは自分でモデルを構成する必要はない.

未知語抽出の実例を見てみよう.下に示すのが,barによる未知語抽出結果の例である.この例では「手ほどき」が茶筌辞書に登録されていない語として抽出される.

英語の<UNKNOWN> 手ほどき </UNKNOWN>を教えられた.

抽出された未知語の利用法であるが,前節で述べた制約付き解析を用いて,あらかじめ未知語の分かち書き箇所を指定して解析することができる.また,抽出された語を再利用するために,茶筌の辞書に対してエントリを追加することも可能である.図1のようなエントリを用意して辞書に単語を追加し,辞書を再コンパイルすることにより解析が可能になる.登録語の右にある数値は,茶筌が解析時に用いるコストであり,この値が小さいほど,登録語が出現しやすいことを表す.最も頻度が低い語に対し4 000をコストとして設定しているため,未知語を辞書に登録する際も4 000として登録し,解析がうまくいかない場合には,この値を小さくしていけばよい.

```
(品詞 (名詞一般)) ((見出し語 (手ほどき 4 000)) (読み テホドキ) (発音テホドキ))
```

図1 茶筌辞書へのエントリの追加

また,形態素解析結果の品詞情報を利用しない場合,barには未知語を考慮した分かち書きを行うオプションがある.単純に分かち書きのみを行いたい場合には,このオプションを利用するのがよい.

次に,固有表現抽出の実例を見てみよう.表1が,現在barが抽出できる8種類の固有表現・数値表現である.これらはIREXのワークショップ[IREX 99]により定義されたものに準拠している.図2に固有表現抽出結果を示す.

表1　固有表現・数値表現

固有表現の種類	意味	例
ARTIFACT	固有物名	ノーベル化学賞
DATE	日付表現	五月五日
LOCATION	地名	日本，韓国
MONEY	金額表現	２０００万ドル
ORGANIZATION	組織名	社会党
PERCENT	割合表現	二〇%，三割
PERSON	人名	村山富市
TIME	時間表現	午前五時

<DATE>１０月１日</DATE>に<PERSON>太郎</PERSON>は<LOCATION>元町</LOCATION>の<ORGANIZATION>大丸</ORGANIZATION>でハンカチを<MONEY>１０００円</MONEY>で買った.

図2　固有表現抽出結果

表2　barのオプション

オプション	機能
-unk	未知語抽出
-unkseg	未知語を考慮した分かち書き
-ne	固有表現抽出
-filfil	フィラーフィルタ

固有表現抽出システムは，情報抽出や質問応答システムなどのキーワード抽出に用いることができる．実際，検索時に対象となるキーワードは形態素解析結果より長い単位であり，一般名詞よりも固有表現のほうが多いため，固有表現抽出は多くの応用分野に対し有用だろう．

なお，barには，上記を含む四つのモデルファイル（固有表現抽出，未知語抽出，未知語を考慮した単語分かち書き，話し言葉の書き起こし文中のフィラーや言い淀み*3の検出）が用意されており，オプションとして指定することによりモデルを切り替えることができる．表2に，barの全オプションを示す．

2・3　南瓜による係り受け解析

日本語では，統語解析の一つとして文節係り受け解析がよく用いられる．英語などで用いられる句構造解析に比べると，統語的な固まりである句に名前をもたないことや文節内の構造が未確定であるなど，荒い解析であるが，表層的な構文構造として日本語話者にはわかりやすく，かつ，話者間の解析の揺れも少ないことから，広く用いられている．

南瓜[工藤 02]は，二つの文節の間に係り受け関係があるかどうかの判断を，前後文脈を考慮しながらSupport Vector Machinesによって学習し，文節係り受け解析を行うシステムである．京大コーパス*4の約4万文を学習データとして用い，文節係り受け精度約91%で日本語文の係り受け解析を行うことができる．品詞体系としては，京大コーパスで用いられている益岡・田窪文法に基づくものと，茶筌が用いているipadic体系に基づくものの両方に対応している．ipadicに対応するバージョンでは，茶筌による形態素解析，単語を単位とするチャンキングによる固有表現同定，文節へのまとめ上げ，および文節間の係り受け解析を連続的に行う．これらの処理は，レイヤとしてUNIXのパイプのようにつながれており，利用者は任意のレイヤの入力と出力を指定して実行することができる．例えば，すでに品詞情報を付加したファイル（茶筌の出力形式になっているもの）を利用者がもっており，文節分かち書きの結果だけを得たい場合には，入力を形態素解析結果，出力を分かち書きとして南瓜を実行すればよい．

<DATE> 10月1日</DATE>に————D
<PERSON>太郎</PERSON>は————D
<LOCATION>元町</LOCATION>の-D　|
<ORGANIZATION>大丸</ORGANIZATION>で——D
ハンカチを—D
<MONEY>1000円</MONEY>で-D
買った.

図3　南瓜による日本語文の係り受け解析例

図3に，南瓜の係り受け解析出力例を図示する．各行が一つの文節に対応し，それぞれがどの文節に係るかが，“D”という記号によって示されている．この例では，「10月1日に」や「太郎は」が文末の「買った」に係り，「元町の」は「大丸で」に係ることなどがわかる．

日本語の文節は，必ず前から後ろに係り，異なる係り受けは互いに交差しない（話し言葉などで例外はあるが）と仮定しているので，図に示したのような木構造で表示することができる．このような解析を行うことより，文中では遠く離れた「10月1日」という日付が「買う」という用言を直接修飾していることなどを知ることができる．

3.　文の役割分類への応用

文書分類では，文書を単語の集まり（bag-of-words: BOW）によって表現し，カテゴリー分類のためのさまざまな手法が適用され，成功を収めてきた．文書分類は，一般に，政治，経済，スポーツなどの文書の内容を大きな分類クラスに分けるタスクであり，個々の単語がこれらのカテゴリーを特徴づけるのに有効な意味情報を与えるた

*3 フィラーとは，「あのー」，「えーと」など話し言葉の合間に挿入される間を取るための発言をいう．言い淀みは，「わ，私は...」のように語の一部を言い直したりする際に混ざる語の断片のことである．

*4 www.kc.t.u-tokyo.ac.jp/nl-resource/corpus.html

め，BOWのような単純な属性を用いるモデルによっても高い精度を達成することができる．

一方で，テキストマイニングの分野では，Web上の製品レビューサイトやアンケート結果などから製品に対する要望や不満などの有用な情報を効率良く入手する要素技術が求められている．このようなタスクでは，意見が主観的に述べられているのか客観的に述べられているのか，あるいは，ある製品をほめているのかけなしているのかなど，書き手の意図に関する分類が求められる．分類の単位も文書ではなく，文のような小さな単位になる．つまり，このようなタスクは，分類するのは文が漠然と表す意味内容ではなく，文そのものがもつクラスであったり，文書中での文の役割についての分類になる．

図4に，想定するタスクで分類すべき文例を示す．“PHS”は，PHSユーザに良い点・悪い点を区別してレビューを投稿するよう指示した掲示板のデータである．“Eval”は，車のレビューサイトにおいて投稿者が車に対して下している評価のうち，それが投稿者自身の主観的な評価かそうでないかを分類するタスクの具体例である．「パワーが足りないっていう人もいます.」という文は，製品の評価にはなっているが，本人の評価ではないため，主観的な評価文とは判断していない．“MOD”は，新聞の社説からランダムに選んだ記事中の文のモダリティを分類した例であり，「断定」，「意見」，「叙述」を表す文を抽出するタスクである．

PHS　良い点：メールを送受信した日付，時間が表示されるのも結構ありがたいです.
悪い点：なんとなく，レスポンスが悪いように思います.

Eval　主観的評価表現：エンジンパーツが豊富で安い.
主観的でない評価表現：パワーが足りないっていう人もいます.

MOD　断定：「ポケモン」の米国での成功を単純に喜んではいけない.
意見：その議論を詰め，国民に青写真を示す時期ではないのか.
叙述：バブル崩壊で会社神話が崩れ，教育を取り巻く環境も変わった.

図4　分類対象文の例

文の意味内容だけでなく発話の意図を量るこのようなタスクでは，単純なBOWではなく，単語のつながりや文体を見る必要があるのが予測されるだろう．これを確認するために，文を茶筌と南瓜によって解析し，BOW以外に，単語の可変長N-gram（長さを限定しない任意長の単語列），および，係り受け木の部分構造（大きさを固定せず，任意の大きさの部分木を許す）を基本属性として，分類学習を行った．実験方法の詳細は，[工藤03]に譲る．簡単に説明すれば，木構造データを効率的にマイニングするアルゴリズムによって特定のクラスの文をほかのクラスの文と最もよく分離する部分構造を求め，それを単独の属性として文分類を行う学習器をつくる．これを弱学習器としてBoostingアルゴリズムを実行することによって，それまでの学習器にとって分離困難な事例に集中して学習した弱学習器が次々につくられる．最終的な文分類は，これらの弱学習器の重み付き多数決によって行われる．この方法では，精度の高い統合的な分類器が得られるだけでなく，各弱学習器が用いている属性を直接観察することができるので，どのような部分構造が文分類に有効に働いたかを知ることができるという利点がある．

表3に，実験結果を示す．PHSでは約5 700文，Evalでは約7 500文，MODでは約1 700文のタグ付きデータを用い，5分割交差検定によって得られたF値を示している．F値とは，精度と再現率の調和平均であり，数値が高いほど分類性能が優れていることになる．bowは個々の単語を個別属性として用い，ngramは可変長のN-gramを個別属性として用い，depは係り受け木構造内の任意の部分木構造をそれぞれ個別属性とする弱学習器をつくり，Boostingアルゴリズムによって学習を行った結果である．この結果より，単語個別の情報では，明らかに文分類のタスクが十分な精度で行えないことがわかる．PHSでは，単独の単語で評価の良否を表現するものも多くあるため，その差が大きくはない．が，MODではその差が特に顕著である．ngramとdepには大きな差がないが，Evalや，MODの「意見」のように複雑な意図内容の分類においては，depが若干良い精度を示している．

表3　文分類実験の結果（F値）

	PHS	Eval	MOD		
			断定	意見	叙述
bow	76.6	77.4	71.2	62.1	83.0
ngram	79.3	80.6	87.6	78.4	91.9
dep	79.0	81.6	87.5	80.5	91.9

係り受け木を用いた実験（dep）によって得られた属性の例を図5に示す．数値は，各属性が分類にどのように寄与しているかを表す重みである．正の数値が「良い点」の分類に肯定的に寄与する属性，負の数値はその逆を示す．例えば，「切れる」が「にくい」に係る構造は「良い」カテゴリーにプラスに働いているのに対し，「にくい」を含むその他の表現（「にくくなった」，「読みにくい」など）は，「悪い」カテゴリーを分類するのに寄与している．「使う」を含む構造では，「使いたい」，「使ってる」，「使いやすい」などが正の重みをもつのに対し，「使いやすかった」，「を使ってた」のように過去形になったり，「方が使いやすい」のように比較になると負の重みをもつことがわかる．また，同じ「充電時間」を含む場合でも，これが「短い」に係るか「長い」に係るかで，評価が正反対になっている．このような属性は，BOWに基づくモデルでは用いることのできない情報である．

A.「にくい」を含む素性

```
0.004024  切れる にくい
-0.000177 にくい EOS.
-0.000552 にくい なる た
-0.000566 にくい.
-0.000696 読む にくい
-0.000738 にくい なる
-0.000760 使う にくい
-0.001702 にくい
```

B.「使う」を含む素性

```
0.00273  使う たい
0.00015  使う
0.00013  使う てる
0.00007  使う やすい
-0.00010 使う やすい た
-0.00076 使う にくい
-0.00085 は 使う づらい
-0.00188 方 が 使う やすい
-0.00233 を 使う てる た
```

C.「充電」を含む素性

```
0.0028  充電 時間 が 短い
-0.0041 充電 時間 が 長い
```

PHSデータ
単語単位の係り受け(dep)
(ルール中の素性は係り受けのパス)

図5　抽出された属性の例

4. ま　と　め

我々のグループで開発し，公開している日本語解析ツール「茶筌」,「bar」,「南瓜」を紹介し，その応用として文の構造を考慮した文の役割分類タスクについて述べた．

なお，本稿で紹介した言語処理ツール「茶筌」,「YamCha」,「bar」,「南瓜」は，すべてフリーソフトウェアであり，以下のURLからリンクされている「自然言語処理のためのツール」のページから入手可能である．また，各ツールのページには，入力文を入れることによってシステムの解析結果を表示するサービスがあり，簡単に試してみることができる．

http://cl.naist.jp/

さらに，関連したツールとして，茶筌の内部で行われている解析を曖昧性込みで（単語よりなるグラフとして）表示する「VisualMorphs」や，茶筌の解析結果に対して，品詞，単語，活用情報などを用いて全文検索を行い，結果をKWIC（Keyword in Context）形式で表示する「茶器」というツールがあり，それらの情報も上記ページにリンクされているので，興味ある人は参照されたい．

謝　辞

ここで紹介したさまざまな言語処理ツールの開発に協力していただいた奈良先端科学技術大学院大学自然言語処理学講座のメンバに感謝します．また，これらのツールに関するエラー報告や要望を寄せていただいた多くの利用者の皆様に感謝します．また，産総研の神嶌敏弘氏には，本稿の草稿に目を通していただき，さまざまなコメントをいただきました．ここに感謝します．

◇ 参 考 文 献 ◇

[Asahara 03] Asahara, M. and Matsumoto, Y.: Japanese Named Entity Extraction with Redundant Morphological Analysis, *Proc. of 4th Meeting of North American Chapter of Association for Computational Linguistics*, pp. 8-15 (2003)

[IREX 99] IREX: *Proc. of IREX workshop* (1999)

[Kudo 01] Kudo, T. and Matsumoto, Y.: Chunking with Support Vector Machines, *Proc.of 2nd Meeting of North American Chapter of Association for Computational Linguistics*, pp. 192-199 (2001)

[工藤 02] 工藤 拓, 松本裕治：チャンキングの段階適用による日本語係り受け解析,情報処理学会論文誌, Vol. 43, No. 6, pp. 1834-1842 (2002)

[工藤 03] 工藤 拓, 松本裕治：部分木を素性とするDecision StumpsとBoosting Algorithmの適応, 情報処理学会研究報告, 2003-NL-158, pp. 55-62 (2003)

[松田 99] 松田 寛, 桐山和久, 山田悟史, 吉野圭一, 松本裕治：情報処理学会研究報告, 99-NL-134, pp. 23-30 (1999)

[松本 00] 松本裕治：形態素解析システム「茶筌」, 情報処理, Vol. 41, No. 11, pp. 1208-1214 (2000)

[松本 03] 松本裕治, 北内 啓, 山下達雄, 平野善隆, 松田 寛, 高岡一馬, 浅原正幸：日本語形態素解析システム『茶筌』version 2.3.3 使用説明書, 奈良先端科学技術大学院大学 (2003)

2004年2月29日　受理

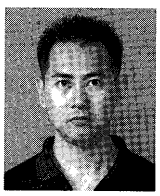
著　者　紹　介

松本　裕治（正会員）

1979年京都大学大学院工学研究科修士課程情報工学専攻修了．同年，電子技術総合研究所入所．1984～85年英国インペリアルカレッジ客員研究員．1985～87年（財）新世代コンピュータ技術開発機構に出向．京都大学助教授を経て，1993年奈良先端科学技術大学院大学教授，現在に至る．工学博士．専門は自然言語処理．情報処理学会，日本ソフトウェア科学会，言語処理学会，日本認知科学会，AAAI, ACL, ACM各会員．

高岡　一馬

1998年京都大学理学部卒業（化学専攻），2000年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，同年同大学院大学博士後期課程に進学，2004年同指導認定退学．同年株式会社ジャストシステム入社．自然言語処理，テキストマイニングに興味をもつ．



浅原　正幸

1998年京都大学総合人間学部基礎科学科卒業，2001年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，同年同大学院大学博士後期課程に進学．同年より日本学術振興会特別研究員（DC）．2003年同大学院博士後期課程修了．2004年奈良先端科学技術大学院大学助手，現在に至る．博士（工学）．自然言語処理の研究に従事．情報処理学会，言語処理学会各会員．

工藤　　拓

1999年京都大学工学部電気電子工学科卒業，2001年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，同年同大学院大学博士後期課程に進学．日本学術振興会特別研究員（DC）．2004年同大学院大学博士後期課程修了．博士（工学）．同年NTTコミュニケーション科学基礎研究所リサーチアソシエイト．統計的自然言語処理，テキストマイニング，機械学習に興味をもつ．情報処理学会会員．