研究のツールボックス [1]

茶釜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,
matsumoto@ias.naist.jp, http://el.naist.jp/

髙岡 一馬
Kazuma Takaoka
株式会社ジャストシステム
Justystem Corporation.
kazuma_takaoka@justystem.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,
amayuki-ai@ias.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で進展している。本稿では、我々が開発している次の二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「茶釜」は、日本語文を単語に分割し、品詞、短絡、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、これでは日本語文には解析されない処理である。実際の文解析には、未知語（辞書に未登録の単語）の出現を無視するわけではない。また、固有名詞、日付、数値表現など、特殊な役割をもつとまとまった表現が数多く出現し、これらの適切な処理が、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。日本語文の表現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割する文書分類（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には、分類性能に著しく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 上の評価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

本稿では、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。日本語解析では、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。「茶釜」は日本語の形態素解析を行うツールである。ここでは、従来の茶釜の機能、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言葉、通常とは異なる表現をもつ語、専門用語や固有名詞など新しく生まられる語などがあり、読む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
tさまざまな言語表現を抽出することができる。ここで
は、未知数および固定表現の自動抽出について説明する。
日本語文は、文節という最小の構文単位に分割し、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、文節内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ブ付き例文からの機械学習
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、入力文単語（形態素）単位に分かち書きし、それ
ぞれに対して文法情報など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文脈モデル
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か」「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 も 助詞-係助詞
も も も 助詞-係助詞
も も も 助詞-係助詞
も も も 助詞-係助詞
ノ の 助詞-連体化
うち ウチ うち 名詞-非自立-副詞
だ タ だ 助動詞特殊・ダ 基本形
EOS

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、活用形、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大（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
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、異
綴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
筌自体はテキストを文出力を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-p オプションを用いることで定
位の文字（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
もスライドアウトやタブ情報を利用して、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 番号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
素情報が既定である、ある特定の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文を入力すると（ここではタブを
"\t" によって明示した）。
すももももももももももももももももも
ももももももももももももももももも
ももももももももももももももももも
うち ウチ うち 名詞-非自立-副詞
だ タ だ 助動詞特殊・ダ 基本形
EOS

EOS

この例は「酢で桃が桃を藻の内だ。」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な解析され、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る。この例で
は「すもも」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**。品
詞情報は茶筌のデフォルトの出力と同じく細分類を「*」
でないで表すが、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予定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同一性をとる
ためにあって、実際には解析に使用されない。
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89】

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「茶釜」について説明した。現在の茶釜は、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。茶釜の辞書である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から未知語や固有表現を抽出するツールについて説明する。

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ンタグリ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する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築することができる。未知語抽出器は、形態素解析器が未知語を検出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表現箇所を推定する。同様に固有表現抽出器は、固有表現表現箇所を推定する。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単純化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釜と、Support Vector Machinesにより決定的にポジションタグを付与する利用チャンナー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」が茶釜辞書に登録されていない語として抽出される。英語のUNKNOWN は手ほどきUNKNOWN を教える。

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、茶釜の辞書に対してエンティ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、茶釜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て4000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4000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れば。
表1 固有表現・数値表現

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

表2 barのオプション

<table>
<thead>
<tr>
<th>オプション</th>
<th>機能</th>
</tr>
</thead>
<tbody>
<tr>
<td>-unk</td>
<td>未知語抽出</td>
</tr>
<tr>
<td>-unkseg</td>
<td>未知語を考慮した分から書き</td>
</tr>
<tr>
<td>-ne</td>
<td>固有表現抽出</td>
</tr>
<tr>
<td>-fillfil</td>
<td>フィラーフィルタ</td>
</tr>
</tbody>
</table>

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。実際に、検索時に対象となるキーワードは事象表現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から書き、話し言葉の書き起こし文中のフィラーおよび言い換え*3が検出）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、モードを切り替えることができる。表2に、barのオプションを示す。

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、統語的な係りである句名前を含むことや文節構造の構造が未確定である事実、文脈解析であるが、一定の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

*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言葉を抜くための発音をいう。言い換えは、「わ、私は…」のように話の一部を言い直したりする際に語る区の断片のことである。

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

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・田畑文法に基づくものと、茶釜が用いている ipadic 体系に基づくものの両方に対応している。ipadic に対応するベージではないが、茶釜による形態素解析、単語を単位とするチャキングに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、すでに品詞情報が付加したファイル（茶釜の出力形式になっているもの）を利用者がもっとおり、文節分から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、出力を分から書きとして南瓜を実行すればよい。

図3に、南瓜の係り受け解析出力例を図示する。各文の文節に対し、それぞれがどの文節に係るか、"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」や「太郎」が文末の「買った」に係り、「元町」は「大丸」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があるが）と仮定しているので、図に示したような文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により、文試ではより離れた「10月1日」という日付が「買う」という用言を直接接続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

文書分析では、文書を単語の集合（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、一般的に、政治、経済、スポーツなどの文書の内容を大小に分類クラスに分けるタスクであり、個々の単語がこれら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
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满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か客観的に述べられているか、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。分類の単位も文ではなく、文のようないい単位になる。つまり、このようなタスクは、分類するものは文が単純に表す意味内容ではなく、文のものかもがクラスであった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ユーザの良い点・悪い点を記述に対するレビューを投稿するように指示された掲示板のデータである。「Eval」は、レビューのレビュータイプにおいて投稿者が文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どう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：なんとか、レスポンスが悪いかんように思います。

Eval 主観的評価表現：エンジンパワーやが豊富で安い。
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：「ボケモノ」の米国での成功を単純に喜んでいない。
意見：その議論を読め、国民に書写真を示す時期ではないのか。
叙述：パ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を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脈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、文を英語と日本語によって解析し、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含む）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93]に読む。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る。それを単純の属性として文分類を行う学習器をつく。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よって分離困難な事例に集中し、学習した弱学習器が次々とつくられる。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は、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5700文、Evalでは約5750文、MODでは約1700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、精密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、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あり、その差が大きくなる。一方、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が若干高い精度を示している。

<table>
<thead>
<tr>
<th></th>
<th>PHS</th>
<th>Eval</th>
<th>MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>断定</td>
<td>見解</td>
<td>叙述</td>
</tr>
<tr>
<td>bow</td>
<td>76.6</td>
<td>77.4</td>
<td>71.2</td>
</tr>
<tr>
<td>ngram</td>
<td>79.3</td>
<td>80.6</td>
<td>87.6</td>
</tr>
<tr>
<td>dep</td>
<td>79.0</td>
<td>81.8</td>
<td>87.5</td>
</tr>
</tbody>
</table>

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」カテゴリにプラスに働いているもののに対し、「にくい」を含む他の表現（「にくかった」、「読みにくい」など）は「悪い」カテゴリを分類するのに寄与している。「使う」を含む構造では、「使いやすい」、「使っつる」、「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」、「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。
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、その運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 からダウンロードされている「自然言語処理のためのツール」のページから入手可能です。また、各ツールのページには、入力文を入力す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暗喻の形で（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、用法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「茶筌」というツールがある、それらの情報も上記ページにリンクされているので、興味が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ーに感謝します。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、産総研の神崎敏弘氏には、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


松本茂：茶筌、農忙時、野菜大車、松田寛、高岡一馬、浅原正幸：自然言語解析システム「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)