Shift-Reduce 操作に基づく未知語を考慮した形態素解析

岡野原 大輔 † 辻井 潤一 †‡§
† 東京大学情報理工学系研究科コンピュータ科学専攻
‡School of Computer Science, University of Manchester
§NaCTeM (National Center for Text Mining)
{ hillbig, tsujii }@is.s.u-tokyo.ac.jp

1 はじめに

形態素解析とは，与えられたテキスト中の単語（形態素）を同定し，品詞を付与するタスクである．日本語の形態素解析の特徴として明示的な単語境界が存在しないことが挙げられ，単語境界の同定も重要なタスクとなる．これを機械学習を用いて実現する手法には大きく分けて二つある．一つ目は大規模な辞書を利用しテキスト上に形態素ラティスを構成し，ラティス上のパスを求めて解析を行う手法であり，二つ目は文字単位でのラベリングから解析を行う手法である．

前者の形態素ラティスを利用する手法では，不必要な単語候補を探す必要が無いので高速であり，かつ辞書情報を利用することができるため高精度の解析を達成できる．これらを用いた方法では隠れマルコフモデルを利用した方法 [1]，最大エントロピーモデルを利用した方法 [2]，条件付確率場を利用した方法 [3, 4] が挙げられる．しかし，出現する形態素が辞書に含まれていない場合、つまり未知語がある場合は，辞書にヒットしない部分文字列も形態素の候補として調べる必要がある．例えば一定長以下の全ての部分文字列を単語候補だとし，それらでラティスを構築する方法も考えられるが，形態素候補の数は非常に多く、非常に遅くなってしまう問題がある．

後者の手法は，各文字に対し，そこが形態素の開始であるか，そうであるならばどの品詞であるかのラベルを付与する手法である．この手法では，既知語，未知語を統一的に扱えるが，文字単位で処理するため辞書情報を統合しにくいことに加え，各文字について学習器による推定が必要なため，前者の手法に比べて低速である問題点があった．このように未知語を考慮した解析では精度のみならず速度の面でも考慮しない解析に比べて難しい問題となる．

未知語は，情報抽出をはじめますます重要となっており，特に最近の流行語を調べるタスクなどでは，抽出される多くの流行語が新しく生まれた未知語であるから，未知語抽出の精度は特に重要となる．一般に未知語がどの程度存在するかの推定は難しい問題だが，一つの指標として，IPA 辞書に登録されている名詞の件数は約 23 万件であるが，はてなダイアリーでの「はてなキーワード」[1]が約 22 万種類，Wikipedia のタイトル中でノイズを簡単な前処理で除いたものが約 68 万種類であり，IPA 辞書とそれらの間で共通して出現しているキーワード数はそれぞれ約 13 万件，約 6 万件であった[2]．このように大規模な辞書においても，現状使われているキーワードを全て網羅することは難しく，また新たに出現する語に対応することはできない．

本稿では，形態素ラティスの手法と文字ラベリングの手法の中間にあたる手法を提案する．また単純な識別器によるフィルタリングを用いて計算量を減らすことも提案する．[5] では本手法と同様の方法を提案して中国語の単語分割を提案しているが，本稿ではそれに加え品詞付与、フィルタリングによる高速化を行い拡張している．

2 Shift-Reduce に基づく形態素解析

本稿では，Shift と Reduce 操作を繰り返し適用していくことで形態素解析を行う方法を提案する．本提案手法では，入力文字列を前から順に 1 文字ずつ順に読み込み，現在保持している各解析候補の最後にその文字をつなげた上で Shift 操作，Reduce 操作を適用し新しい解析候補集合を作成する．Shift 操作は現在保持している解析候補をそのままに次の候補にする操作であり，Reduce 操作は現在の位置が形態素境界だとして，現在の末尾の部分列を形態素だとし，品詞を付与する操作である．IPA 品詞体系など，品詞が階層を持っている場合，Reduce 操作は品詞の階層の回数だけ適用される．Shift 操作では候補数は増えないのに対し，Reduce 操作では品詞の種類数分候補数が増える．

各ステップでは，解析候補数が一定以上になったら解析候補を，スコアが高い上位のみを保存していく．各候補のスコアの計算方法については後述する．そして，全

[1] http://d.hatena.ne.jp/images/keyword/keywordlist_furigana.csv
[2] キーワードの単位と形態素の単位は異なる場合も多く，直接的な比較は難しい

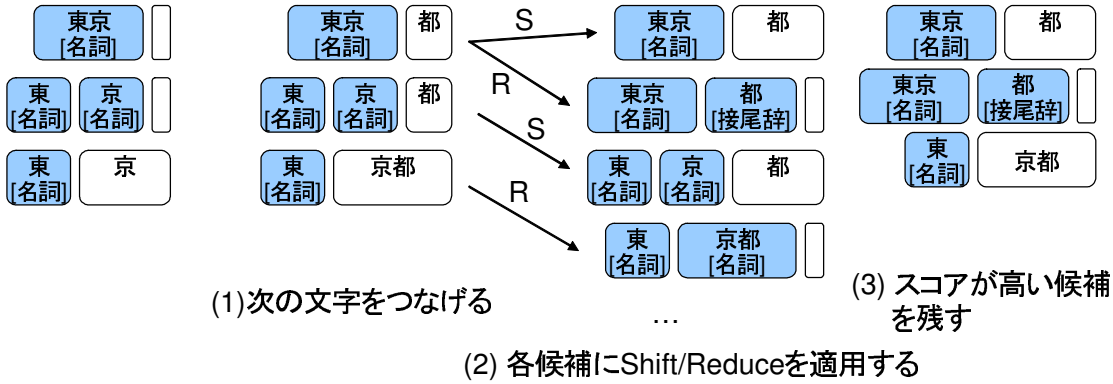

図 1: Shift/Reduce 操作による形態素解析の例．既に形態素が同定された部分は影付きの四角で表され，同定されてなく次の文字がつながる可能性がある部分は白の四角で表されている．各候補は次の 1 文字を読み込み，それぞれ独立に Shift/Reduce 操作（図中の S と R) を行い，次候補を生成し，スコアが高い候補を残す．

ての文字を読み終えた後に最もスコアが高かった解析候補を解析結果とする．この Shift/Reduce 法による解析は，ありうる全ての形態素候補列を前から順に Greedy に展開しビームサーチを行っていると考えられる．

図 1 に Shift/Reduce 操作による形態素解析での 1 文字を読み込んだ際の例を示す．最初に，三つの解析候補があり，一つ目と二つ目の解析候補とは末尾の文字列は空文字列であり，三つ目は “京” が品詞がついていない文字列となる (図 1(1))．次の文字 “都” を読み込み，それぞれの末尾につなげた後それぞれに Shift/Reduce 操作を適用する (図 1(2))．図では Reduce 操作は 1 個しか書かれていないが実際には品詞の種類数分適用される．実装では品詞の階層毎に上位の候補だけを残すようにした．そして，生成した候補の中からスコアが高い上位のみを残す (図 1(3))．

各ステップで保持する解析候補は，それまでに同定した形態素列からなるが，以降の解析に利用する情報が限られている場合は，同じ情報を持っているものをまとめ効率的に解析することができる．例えば HMM や CRF と同様に，直前の品詞情報しか使わない場合は，直前の品詞が同じだった解析候補の中で最もスコアが高いもののみを残しておけばよい．また，HMM や CRF 等とは違い遠距離の情報を自由に使うこともできる．従来手法では，使える情報をローカルなものに限ることにより，正規化項の効率的な計算を可能としているのに対し、提案手法では必ずしも正規化項を求める必要が無いので遠距離の情報も自由に使える．この場合，解析候補がまとめられないため計算量が増加する．利用する素性関数の自由度と計算の効率性はトレードオフの関係にある．

次に，スコアがどのように決定されるかについて述べる．各解析候補の結果は入力 x（文字列）と出力 y（形態素列）から決定される素性関数からなる素性ベクトル $\phi(x, y) \in R^m$ によって表され，各解析候補のスコアは素性ベクトルと重みベクトル $\mathbf{w} \in R^m$ との内積 $\langle \phi(x, y), \mathbf{w} \rangle$ により決定される．素性関数に利用できる情報としては，辞書情報の他，接頭辞/接尾辞の情報など従来の MEMM や CRF と同様の情報が利用できる他，前述のように必ずしも隣接する情報のみならず，遠距離の情報も利用できる．また，文字単位では無く単語単位で Reduce 操作を行うために，文字単位のラベリングとは違い，形態素にマッチする辞書情報等が利用可能である．

3 フィルタリングによる高速化

本手法では，文字単位のラベリングと同様に，1 文字ずつ処理し，その度にスコアを計算する必要があり、辞書情報を用いてラティスを構築する手法に比べ計算量が多くなってしまう．本稿では大量のコーパスを元に不必要な候補を除去するためのフィルタリングを利用する手法を提案する．

フィルタリングには，生コーパスを既存の形態素解析器で解析した結果を利用し，文字 Bi-gram による単純なモデルによる識別を行う．文字 Bi-gram による識別の理由は，解析が高速に行えるという点と，未知語に対してもロバストに動くという点である．

はじめに生コーパスに対し形態素解析を行い，全ての 2 文字間 c_1, c_2 に対し，その間に形態素境界があった割合を調べた．この形態素境界があった割合を $P(c_1, c_2)$ とする．次に閾値 T^+ と T^- を設定し，解析の際に $P(c_1, c_2) \geq T^+$ である候補については高い確率で形態素境界があることから全ての候補に対し Reduce 操作のみを行い，$P(c_1, c_2) \leq T^-$ である候補については全ての候補に対し Shift 操作のみを行うようにする．このフィルタリングの目的は，正解となる解析候補を減らさずに，候補数を

```
function analysis (s, len, k)
# s:  解析対象の文字列
# len:  s の長さ
# k:  ビーム幅.
# cand:  現在の候補集合
for i=1 to len
   cand 中の全ての候補の末尾に s[i] を付加する
      if P(s[i], s[i+1]) ≥ T+ then
         cand 中の全ての候補に対し Reduce を適用
      else P(s[i], s[i+1]) ≤ T− then
         cand 中の全ての候補に対し Shift を適用
      else
         cand 中の全ての候補に対し Reduce と Shift を適用
   cand をスコアによってソートし，上位 k 件のみを残す
cand 中のスコアが最大のものを結果として返す
```

減らすことであるので再現率が高くなるように閾値 T^+, T^- を選んだ．本手法と同様なフィルタリングを用いた高速化については条件付確率場など他の学習器でもその有用性が示されている [6, 7]．図 3 にフィルタリングも含めた Shift/Reduce 操作による形態素解析の擬似コードを示す．

4 オンライン学習

本章では解析済みの訓練データが与えられた時，重みベクトル $\mathbf{w} \in R^m$ をどのように学習するかについてを述べる．今回は重みベクトルを求める際に正規化項は計算できないので，マージンベースの学習法を用いた．今回のように出力同士が相関を持つ、いわゆる構造付き出力のマージンベースの学習方法としては，Structured Perceptron [8] が知られている．この方法では，各学習データに付き，正解データ y' と学習器が予測した現在のパラメータの中で最もスコアが高い候補 $y*$,

$$y* = \arg\max_{\mathbf{w}} \langle \phi(x), \mathbf{w} \rangle \quad (1)$$

が与えられた時，重みベクトルを次のように更新する．

$$\mathbf{w} = \mathbf{w} + \phi(x, y') - \phi(x, y^*) \quad (2)$$

この更新により，重みベクトルは正解データに対しより大きいスコアを与えるようになり，誤った出力に対しては小さいスコアを与えるようになる．さらに最終的に得られた重みベクトルではなく，全てのステップで得られた重みベクトルの平均を利用する Average Perceptron が精度が高いことが報告されており，今回はそれを利用した [8].

本提案手法では，途中でスコアが低いものを枝刈りするビームサーチを行うことから，最もスコアが高い候補が必ずしも得られないが，今回は [5] と同様に最終的に得られた結果をそのまま最もスコアが高い候補の代わりに利用することにした．

本タスクで利用した素性関数は以下の通りである．

- 現在の形態素の prefix, suffix, 全体の文字列, 文字種
- 直前の形態素の prefix, suffix, 全体の文字列, 文字種
- 現在の品詞
- 直前の品詞

解析精度を上げるのに有効だと思われる非局所的な素性（同じ形態素の繰り返しなど）の効果については今後の課題とする．

5 実験

はじめに 3 章で述べたフィルタリングの実験を行った．日本語 Wikipedia[3]に対し Mecab (ver 0.96)[4]を用いて形態素解析を行い, 文字間分割確率を求めた．閾値 T^+ と閾値 $^-$ はそれぞれ, 0.999, 0.05 を用いた．これを京大コーパスの 1 月 1〜8 日分に対し適用した結果は表 1 に示した．Shift の精度は $P(c_1, c_2) \geq T^+$ であった候補の中で実際に Shift 操作が正解だったものの割合，Reduce の精度は $P(c_1, c_2) \leq T^-$ の中で実際に Reduce 操作が正解だったものの割合，候補数の削減割合はフィルタリングされた後に残った候補数の元の候補数の割合である．この結果

[3] http://download.wikimedia.org/jawiki/
[4] http://mecab.sourceforge.net/

表 1: フィルタリングの精度．括弧内の数字は（正解数/問題数) を示す．

Shift の精度 (%)	Reduce の精度	候補数の削減割合
99.0% (134599/134607)	99.0% (98507/99498)	28.5% (93389/327494)

表 2: 形態素解析の精度

	単語区切り (%)	品詞	細品詞
精度	83.4	78.5	74.8
再現率	83.9	79.0	75.3
F 値	83.6	78.7	75.0

からフィルタリングほぼ間違えずに解析候補を約 1/4 にできていた．

次に、京大コーパスの 1 月 1〜8 日分を訓練データとして利用し，本手法を学習し、1 月 9 日分でテストデータとして用いた。品詞体系には，JUMAN 品詞体系を用いた．ビーム幅は 4 である．結果を表 2 に載せる．これらの結果が従来手法 [3, 1, 2] と比較し，精度が低かったのは，テストコーパスにおいて出現する語は殆ど既知語であり, 既知語のみを集中して処理手法と比べ，未知語を考慮している分精度が低くなってしまったのと考えられる．また，今回は辞書情報を利用していないため，これらの情報を統合することにより精度が上げられると考えられる．補足実験として後ろから 1 文字ずつ読み込む，後ろ向きの解析も行ったが精度は前向きの場合とほぼ同じかわずかに低かった（単語区切りの精度は F 値 80%). 今回は素性関数が単純なものしか用いていないということもあり，今後，要因を詳細に調べていく予定である．

6 まとめ

本稿では，Shift/Reduce 型の形態素解析を提案し、フィルタリングを用いることで効率的に行うことができることを示した．今回の実験結果は従来の既知語をベースとした形態素解析手法と比較し精度が低く，今後辞書情報の有効活用による改良が必要と考えられる．また，本手法は Shift/Reduce 型の係り受け解析 [9] と自然に統合することができる．つまり，Reduce 操作の際にスタックに積み，係るか係らないかの操作を調べることにより係り受け解析も同時に行うことができる．その他，本手法は部分文字列に対する解析もインクリメンタルに与えることができるため日本語入力インターフェースなどでの利用が考えられる．今後は実装・実験を進め本手法の評価を進めるとともに，ブログなど実際に未知語が多いデータ上で精度測定を行う予定である。

参考文献

[1] 浅原 正幸 and 松本 裕治. 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):685–695, 2002.

[2] 内元 清貴, 関根 聡, and 井佐原 均. 最大エントロピーに基づく形態素解析 未知語の問題の解決策. **自然言語処理**, 8(1):127–141, 2001.

[3] 工藤 拓, 山本 薫, and 松本 裕治. Conditional random fields を用いた日本語形態素解析. In **情報処理学会研究報告** *2004-NL-161*, 2004.

[4] 東 藍, 浅原 正幸, and 松本 裕治. 条件付確率場による日本語未知語処理. In **情報処理学会研究報告** *2006-NL-173*, 2006.

[5] Yue Zhang and Stephen Clark. Chinese segmentation with a word-based perceptron algorithm. In *Proc. of ACL*, pages 840–847, 2007.

[6] Okanohara Daisuke, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun’ichi Tsujii. Improving the scalability of semi-markov conditional random fields for named entity recognition. In *Proc. of ACL*, 2006.

[7] Yoshimasa Tsuruoka and Jun’ichi Tsujii. Chunk parsing revisited. In *Proc. of IWPT*, pages 133–140, 2005.

[8] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*, 2002.

[9] Manabu Sassano. Linear-time dependency analysis for japanese. In *Proc. of COLING*, 2004.