自然言語処理　132－12
(1999. 7. 23)

品詞タグつきコーパスにおける品詞体系の変換

乾 健太郎 *1 *2　　脇川 浩和 [3]

*1 九州工業大学情報工学部知能情報工学科
*2 科学技術振興事業団さきがけ研究 21
〒 820-8502 福岡県飯塚市川津 680-4
inui@ai.kyutech.ac.jp
http://www.pluto.ai.kyutech.ac.jp/~inui
*3 INS エンジニアリング

あらまし
近年，信頼性の高い品詞・構文タグつきコーパスに対する需要の増大にともなって，コーパスを共有・再利用することの重要性がますます大きくなっている．しかし，既存のタグつきコーパスでは基礎とする品詞体系が統一されておらず，そのことが共有・再利用の障害となっている．このような背景から本稿では，既存のコーパスの品詞・構文タグを別の品詞体系に基づく品詞・構文タグに変換するアルゴリズムについて論じる．本稿で提案する手法では，ターゲット側品詞体系に基づく文法・辞書でコーパスを形態素・構文解析することによって半自動的にタグ付けを行う．このとき生じる曖昧性は，ソース側タグ情報を最大限に利用することによって高い精度で解消することができる．

キーワード　品詞体系変換，品詞タグ，形態素解析，構文解析

An POS-Tag Conversion Method for Reusing Corpora

INUI Kentaro*1 *2 and WAKIGAWA Hirokazu*3

*1 Department of Artificial Intelligence, Kyushu Institute of Technology, JAPAN
*2 PRESTO, Japan Science and Technology Corporation, JAPAN
*3 INS Engineering Corporation, JAPAN

Abstract

The problems in reusing the POS-tag information of an existing corpus are in the gap between different tag sets; corpora are annotated in terms of different tag sets. While the recent efforts for standardizing tags are important, we still need to explore techniques for the (semi-)automatic conversion between different tag sets in order to maximally reuse the existing tagged corpora. This paper presents an NLP-based method for the conversion between Japanese POS-tag sets, and reports the results of our preliminary experiment.

key words　POS-tag conversion, POS-tag set, corpus, parsing, POS tagging

1 はじめに

自然言語処理の分野では近年，言語コーパスを，言語知識の学習や統計情報の獲得などに利用する研究が盛んである．一般に，コーパスに基づく自然言語処理システムの性能は言語データの量に依存する．そのため，このようなシステムをあつかった研究では，できるだけ大規模なコーパスを利用できることが望ましい．しかし，大規模なコーパスの作成には多くの時間と手間を要するため，十分な量のコーパスを確保するのは容易でない．そこで複数のコーパスを共有したり，再利用したりすることが求められている．しかしながら，コーパスによって品詞体系が異なり，それにともなう単語境界の認定基準も異なる場合が多く，コーパスの共有には何らかの工夫が必要である．

このような背景から本研究では，形態素・構文情報つきコーパスを再利用するために，既存のコーパスの品詞・構文タグ（ソース側タグ）を別の品詞体系に基づく品詞・構文タグ（ターゲット側タグ）に変換するアルゴリズムを提案する．本手法では，ターゲット側品詞体系に基づく文法・辞書でコーパスを形態素・構文解析することによって半自動的にタグ付けを行う．ただし，単純な解析方法では大量の曖昧性が残ってしまう．この問題を解消するために，本手法ではソース側タグ情報を最大限に利用し，ターゲット側の曖昧性を削減する．

2 品詞体系変換アルゴリズム

本手法では，ターゲット側品詞体系に基づく文法・辞書でコーパスを形態素・構文解析することによって半自動的にターゲット側のタグ付けを行う．ターゲット側では，品詞タグの他にこの解析には以下の資源を用いる．

- ターゲット側文法: ターゲット側品詞体系に基づく文節内文法
- ターゲット側辞書: ターゲット側品詞体系に基づく辞書
- ターゲット側連接表: 隣接する品詞間の連接可能性に関する制約
- ターゲット側係り受け表: 文節間の係り受け可能性に関する制約

ただし，これらの資源を使うだけでは，通常の形態素・構文解析と同じ作業になり，大量の曖昧性が残ってしまう．そこで，対象とするコーパスに以下のソース側タグ情報が与えられているものと仮定し，それらを利用することによってターゲット側の曖昧性を削減する．

- ソース側単語境界: ソース側品詞体系に基づく単語の境界[1]
- ソース側品詞タグ: 各単語に付与されているソース側品詞タグ
- ソース側文節境界: ソース側の文節認定基準に基づく文節の境界
- ソース側係り受け情報: 文節間の係り受け関係

これらの情報はいずれも，EDR 日本語コーパス [6] や京大コーパス [2]，ATR コーパス [4] といった既存の構文木つきコーパスから抽出できる情報であり，ソース側情報として仮定するのは自然だと考えられる．

本手法は次の手順からなる．

1. 前処理： コーパスから品詞変換表を半自動的に作成する
2. 文節内処理： 文節ごとに次の処理を行う
   - 2.1 品詞変換表を適用し，ターゲット側品詞タグ列の候補を生成する
   - 2.2 得られた品詞タグ列の候補をターゲット側文法で構文解析し，曖昧性を削減する
3. 文節間処理： 文節内処理で曖昧性が残った文節に対し，文節間の制約を適用することによって，曖昧性をさらに削減する

以下，それぞれの処理の概要を述べる．

2.1 前処理

前処理では，二つの品詞体系間の品詞変換表を作成する．品詞変換表は，

（ソース側品詞）$\rightarrow$（ターゲット側品詞の列）

という形の品詞変換規則の集合である．たとえば，「広がりつつ」という文字列は，京大コーパスの品詞体系（益岡文法品詞体系）では，

広がり（子音動詞ラ行基本連用形）
つつ（接続助詞）

と解析されるが，EDR 辞書の品詞体系では，

[1]「単語」の認定は品詞体系が定めるものとし，以下の議論では「単語」と「形態素」という用語を区別せずに用いる．

広が（ラ行五段動詞語幹）
り（ラ行五段活用語尾・五段連用形）
つつ（動詞連用形後接語・接続助詞）

と解析される．この対応からは以下のような品詞変換規則が得られる．

子音動詞ラ行基本連用形 →
　ラ行五段動詞語幹
　ラ行五段活用語尾・五段連用形
接続助詞 → ラ行五段活用語尾・五段連用形

このような変換規則は，ソース側品詞タグとターゲット側品詞タグがともに付与されている訓練用コーパスがあれば，そこから自動的に抽出することができる．そのような訓練コーパスが手に入らない場合は，以下の手順で半自動的に収集する．

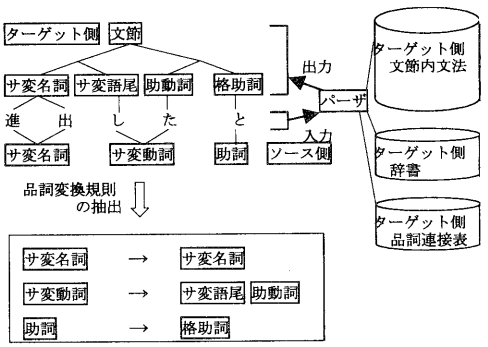

図 1: 品詞変換表の作成

2.1.1 品詞変換規則の候補の収集

ソース側のすべての品詞について品詞変換規則を漏れなく作成する必要がある．そこで，まずソース側の各品詞について，その品詞を含む十分な数の文節を用意する．つぎに，集めた文節をターゲット側の文節内文法・辞書・連接表を用いて解析する．解析で得られるターゲット側品詞タグ候補のうち，ソース側品詞タグと単語境界が矛盾しないものから，図1に示すように品詞変換規則の候補を生成する．ここまでの作業は人手を要しない．

2.1.2 品詞変換規則の洗練

上の作業では，ターゲット側の解析で曖昧性が生じるため，次のような誤った対応関係も品詞変換規則として収集されてしまう．

ナノ形容詞 → 普通名詞 普通名詞 助詞
（ストレートな → スト／レート／な）

これらの不適格な規則は，基本的には人手で取り除くしかない．3節で述べる実験では，対応関係の頻度情報や「名詞」「助詞」といった品詞の大分類の情報を手がかりにして，人手によって規則集合を洗練した．

2.2 文節内処理

2.2.1 品詞変換規則の適用

このようにして得られた品詞変換規則は，ソース側の品詞を非終端記号として左辺に持ち，ターゲット側の品詞（列）を前終端記号として右辺に持つ文法規則と見なすことができる．このことを利用すれば，図2に示したように，品詞変換表をベースとする文法を用いて構文解析することにより，コーパスに品詞変換規則の制約を効率的に適用することができる．

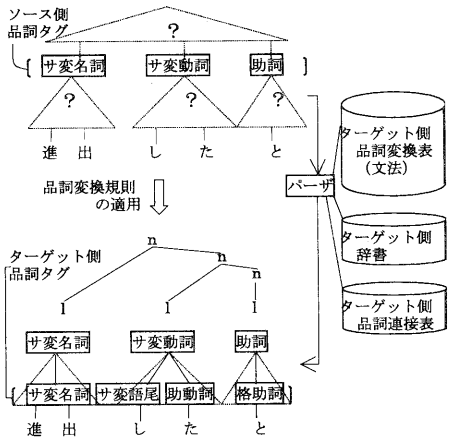

図 2: 品詞変換表の適用

図2のように，この構文解析の入力は，ソース側単語境界とソース側品詞タグの情報を付与した一文節の文字列である．パーザは，これらソース側の制約に無矛盾な構文木だけを出力する．解析に当たっては，ターゲット側辞書の他に，ターゲット側品詞連接表の制約を適用することによって解析結果の曖昧性の抑制する[2]

[2]我々が実験で用いた東工大で開発されたMSLRパーザ[3]は，文脈自由文法と品詞連接表の制約を同時に適用して形態的・構文的曖昧性を抑制することができる．また，同パーザは，入力文に部分的な構文構造が与えられると，それに無矛盾な構文木だけを出力することができる．

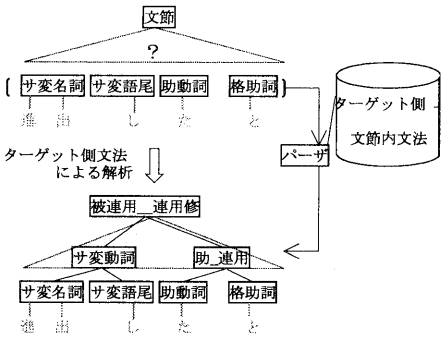

図 3: ターゲット側文法による解析

2.2.2 ターゲット側文法による解析

このフェーズでは，ターゲット側文節内文法の制約を適用することによって，2.2.1の処理で残る曖昧性を削減する．具体的には，2.2.1の処理で得られるターゲット側品詞列の各候補とターゲット側文節境界を入力として，ターゲット側文節内文法による構文解析を行う（図 3）．

この過程で，文節内文法に合わない品詞列は棄却される．一方，文法に合った品詞列は文節にまとめ上げられる．文節内文法をうまく設計しておけば，図 3のように，文節の属性を表すラベルを文節の節点に割り当てることも可能である．このラベルは次の文節間処理で利用する．

入力にターゲット側文節境界の制約を入れるのは，文節境界の制約が曖昧性解消に有効であることが経験的に明らかになっているためである．ただし，ソース側とターゲット側で文節境界の認定基準が異なる場合がある．たとえば，「広がりつつある」という文字列には，全体で 1 文節とする解釈と，「広がりつつ／ある」のように 2 文節とする解釈が考えられる．このような場合，上述の文節内文法による解析は受理されない．そこで，上述の解析で受理されなかった文節列については，ターゲット側文節境界の制約を取り除いて，解析を再度行う．これによって，ソース側文節境界に比べてターゲット側文節境界が細かい場合は，扱うことができる．ただし，逆の場合は現在のアルゴリズムでは扱えないので，何らかの工夫が必要である．

2.3 文節間処理

ターゲット側の曖昧性は，上で述べた文節内の制約だけでは解消できないものも多いと予想される．たとえば，「太郎／と」のような文節では，「と」が格助詞なのか並立助詞なのか，文節内の情報を参照するだけでは決定できない．ところが，助詞「と」の場合について言えば，それを含む文節が用言に係っていれば格助詞，体言に係っていれば並立助詞であることがわかる．このように文節内処理で残る曖昧性の中には，文節間の係り受け情報を利用すれば解消できるものがある．

文節間処理では，文節境界をまたいで隣接する 2 つの品詞の連接可能性のチェック，および文節ラベルを用いた係り受け可能性のチェックを行う．

3 実験

提案したアルゴリズムの有効性を確認するための事例研究として，以下のセッティングのもとに実験を行った．

- ソース側：
  - コーパス： 京大コーパス 10,697 文（105,900 文節）
  - 品詞体系： 益岡文法に基づく品詞体系（品詞数 416 個）
- ターゲット側：
  - 品詞体系： EDR 日本語単語辞書に基づく細品詞体系（品詞数 621 個）[3]
  - 文節内文法： 東工大の植木らが開発した文法 [6] を独自に修正したものを利用（1462 規則）
  - 辞書： EDR 日本語単語辞書を拡張して使用
- 解析器： MSLR パーザ [3]

評価に当たっては，次の 2 つのレベルのタスクを想定した．

構文木決定タスク： （ターゲット側）構文木を品詞列とともに決定することを目標とするタスク

品詞列決定タスク： 文節内構文木の決定を目標とせず，品詞列さえ決定できればよいとするレベル

[3]ここで用いる細品詞体系は，EDR 辞書の品詞と左右連接属性の組み合わせを一意に特定するように定義された品詞ラベルの集合である．

3.1 品詞対応規則の作成と辞書の拡張

今回の実験では，ソース側品詞タグ，ターゲット側品詞タグともに付与された訓練コーパスが存在しないので，2.1項で述べた手順に従って品詞対応規則を半自動的に抽出した．

まず，ソース側の各品詞について，それを含む文節を 15 個ずつ用意した．これらの文節に対し2.1.1で述べた処理を行ったところ，次のような品詞変換規則の候補が約 4,000 パターン得られた（表1 参照）．

ナノ形容詞ダ列基本形 → **am2 em21**
（ストレートだ → ストレート／だ）

イ形容詞イ段基本形 → **aj1 ea11**
（女らしい → 女らし／い）

2.1.1で述べたように，規則の左辺はソース側品詞，右辺はターゲット側品詞列である．上の例のターゲット側品詞にはそれぞれ以下のような定義が与えられている．

am2: 形容動詞 （左属性）形容動詞 （右属性）ダ活用形容動詞語幹・連体「の」あり

em21: 形容動詞語尾 （左属性）ダ活用形容動詞活用語尾連体「の」あり （右属性）形容動詞終止形

aj1: 形容詞 （左属性）形容詞 （右属性）語幹がイ段で終わる・形容詞語幹

ea11: 形容詞語尾 （左属性）イ段終止形容詞語幹後接語尾（右属性）形容詞終止・連体形・助動詞終止・連体形た・ぬ

今回の実験では，ターゲット側資源による形態素解析の際に，ヒューリスティクスや統計的手法を用いて解析結果を絞り込むといったことは特に行わなかった．したがって，解析結果の中には正しくない候補が大量に混じっており，そこから次のような不適格な候補が数多く生成された．

ナノ形容詞ダ列基本形 → **n10 n10 jo41**
（ストレートだ → スト／レート／だ）

イ形容詞イ段基本形 → **n10 jd36 ea6**
（女らしい → 女／らし／い）

n10: 普通名詞 （左属性）普通名詞 （右属性）普通名詞

jo41: 助詞 （左属性）準体言助詞 （右属性）準体助詞の

jd36: 助動詞 （左属性）助動詞 （右属性）形容詞語幹

表 1: ソース側品詞数と品詞対応規則数

自動獲得した規則候補数	3935 パターン
手作業による洗練後の規則数	631 パターン
助詞・副詞などの追加後の規則数	704 パターン

ea6: 形容詞語尾 （左属性）形容詞活用語尾 （右属性）形容詞終止・連体形・助動詞終止・連体形た・ぬ

しかしながら幸いなことに，不適格な候補のほとんどは，上の例のように不適格であることが容易に判断できるものであった．たとえば，形容詞から普通名詞（n10）への変換は明らかに不適格だと判断できる．このようにして規則候補を人手で取捨選択したところ，631 パターンの候補が適格な規則として残った．

ただし，今回ターゲット側品詞体系として用いた EDR 細品詞体系では，助詞や副詞など，一部の品詞について非常に細かい分類がなされているので，ソース側各品詞について 15 文節を用意するだけでは，得られる変換規則に漏れが生じることが作業の過程で明らかになった．そこで，EDR 辞書マニュアル [1] の記述を手がかりにして，規則の追加を行なった．追加した規則は 73 パターンであった．

辞書は，EDR 日本語単語辞書をもとに作成した．ただし，固有名詞をはじめとする名詞類の未登録単語については，ターゲット側における形態素解析結果とソース側とで単語境界が一致している語をコーパスから抽出し，ターゲット側辞書に登録した．その他の未登録単語については，手作業で登録した．とくに表記のゆれによるものが多かった．

3.2 品詞体系変換： 構文木の決定

前述のセッティングのもとで文節内処理および文節間処理を行ったところ，表??のような結果が得られた．表中の数字 (u)，(u^0)，(u^1)，(u^2) は，それぞれ解析結果の候補が一意に決定できた文節の数を表す．ただし，(u^0) は品詞列が一意に決まった文節の数を示し，残りの (u^1)，(u^2)，(u) はそれぞれ構文木が一意に決まった文節の数を示す．同様に，(a^*) は複数の構文木（品詞列）の候補が残った文節の数，(r^*) は解析に失敗した文節の数を表

す．

文節内処理（2.2項）ではまず，図2のように品詞変換規則を用いてターゲット側品詞列の候補を生成する（表??の「変換規則のみ」の列を参照）．候補の生成は文節総数の 89% の文節で成功し，そのうちの 78%（73,460 文節）について候補を一意に絞ることができた (u^0)．

次に，上で得られた品詞列の各候補について，ターゲット側文節内文法を用いて解析し，文節内統語構造を生成する（図3）．ただし，品詞変換規則による解析が失敗した 11,783 文節 (r^0) については，文節の文字列を入力とし，文節内文法，品詞連接表，および辞書による形態素・構文解析を行った．さらに，この過程で解析に失敗した文節については，文節境界の制約を緩和して，再度解析を試みた．この処理の結果，文節総数の 63%（67,603 文節）の文節に一意の構文木を与えることができた (u^1)．

最後の文節間処理では，残った構文的曖昧性の解消を目的として文節間制約を適用する．文節内処理で用いた文節内文法は，品詞列を文節単位にまとめ上げる働きを持ち，このとき各文節に係り属性（被連用／被連体）と受け属性（連用／連体／無）の組を表す非終端記号を割り当てる．たとえば，「国連改革と」という文節には次のような構文木を与える．

```
[bun,[b_体_用[noun,[nn,[n_,[n,[n10, 委員長]]]]]
                [j_体,[joshi_体,[jo_体,[jo1, と]]]]]
[bun,[b_体_体[noun,[nn,[n_,[n,[n10, 委員長]]]]]
                [j_体,[joshi_体,[jo_体,[jo3, と]]]]]
```

ここで非終端記号 b_体_用 は，連体修飾を受ける文節であり，かつ自らは後方の文節に連用修飾する文節であることを表している．いま我々はソース側の文節係り受け情報を利用することができるので，係り文節と受け文節双方の係り受け属性を調べれば，たとえば上の例では「（委員長）と」の品詞が jo1（普通の格助詞）か jo3（並立助詞）かを決定できる可能性がある．このような処理をほどこした結果，構文的曖昧性を持つ文節の 41%（14,692 文節）について，曖昧性を解消することができた（表??中の「"+ 文節間制約" · "from (a^1)"」参照）．これに，文節内文法による解析ですでに構文木が一意に決まっている 67,301 文節 (u^1) を加えると，結局文節総数の 77%（81,993 文節）について構文木が一意に決定できたことになる．このことから，少なくとも今回の実験設定では，文節間制約の利用（ソース側係り受け情報の利用）がパーフォーマンスの向上に寄与したことがわかる．

ただし，構文木が一意に決定できた文節1 (u^1) のうち，7.2% に当たる 5,864 文節は文節間制約の適用で棄却されている．これについては，無作為に抽出した事例を調べたところ，「名詞句＋判定詞」あるいは「名詞句＋読点」といった文節の文節内文法での扱いが不適当で，係り受け関係の制約を満たさなかったものが約 70% を占めていた．たとえば，京大コーパスによると，「国連改革を／前提に／考えていく」という文では「国連改革を」が「前提に」に係る．しかしながら現在の文節内文法では，これら2つの文節をそれぞれ次のように解釈するので，この係り受け関係を受理することができない．

```
[bun,[b_体_用[n_sahen[nn[n_[n[n10, 国連]]]]
                [n_sa[n12, 改革]]]
        [j_用 [joshi_用 [jo_用 [jo1, を]]]]]]
[bun,[b_体_用[noun[nn[n_[n[n10, 前提]]]]]
        [j_用 [joshi_用 [jo_用 [jo1, に]]]]]]
```

この例について言えば，たとえば「前提に」の「に」を判定詞と解釈し，「前提に」に対して "連用後置詞句を補語にとる文節" という解釈も与えられるように文節内文法を拡張するといった対処が考えられる．いずれにせよ，このように文節間制約を満足しない構文木については，たとえ一意に決定できたとしても，それを適格な構文木と考えるべきかどうかは議論の余地がある．

一方，得られた構文木が文節内制約と文節間制約をともに満足する場合 (u^2) は，他の候補が存在しない限り，その構文木の信頼性は十分に高いと期待できる．そこで実際に，本手法で一意に決定できたターゲット側タグ情報の信頼性を推定する調査を行った．文節内制約と文節間制約をともに満足する構文木を一意に決定できた文節（(u^2) 76,129 文節，総文節数の 72%）の中から無作為に 70 文節を抽出し，その適格性を人手で調べた結果，70 文節とも適格な構文木が与えられていることがわかった．小規模な調査ではあるが，この結果は，一意に決定されたターゲット側のタグが十分に信頼できる

表 2: 実験結果

変換規則を用いない場合		変換規則を用いた場合							
文節内・文節間制約を適用		変換規則のみ		+ 文節内制約			+ 文節間制約		
(u)nique	18,598	(u^0)	73,460	from (u^0)	65,664	(u^1)	from (u^1)	61,437	(u^2) 76,129
				from (a^0)	1,032	67,301	from (a^1)	14,692	(72%)
				from (r^0)	605				
(a)mbiguous	78,597	(a^0)	20,657	from (u^0)	7,387	(a^1)	from (a^1)	18,863	(a^2) 18,863
				from (a^0)	19,478	35,750			(18%)
				from (r^0)	8,885				
(r)ejected	8,705	(r^0)	11,783	from (u^0)	409	(r^1)	from (u^1)	5,864	(r^2) 10,908
				from (a^0)	147	2,849	from (a^1)	2,195	(10%)
				from (r^0)	2,293		from (r^1)	2,849	

ことを示唆している．

このように今回の実験設定では，最長一致法のようなヒューリスティクスや統計的手法による曖昧性解消を行わなくても，統語的な制約だけで 70% 以上の文節にユニークな構文木を与えることができた．このことは我々がソース側品詞タグ情報を利用していることに因るところが大きいことは明らかであろう．そこで試みに，ソース側品詞タグ情報を利用しない方法で，すなわち変換規則を用いずにソース側文節境界情報とターゲット側資源だけを用いて同様の実験を行ってみた．結果は，表??の「変換規則を用いない場合」の列に示した通りである．容易に想像できるように，品詞体系の変換ではソース側品詞タグ情報をいかに利用するかが重要であることがこの結果から示唆される．この意味で，今回行った品詞変換規則の抽出方法については，さらに改良すべく検討する必要があると思われる．

3.3 品詞体系変換： 品詞列の決定

文節内構文木の決定を目標とせず，品詞列さえ決定できればよいという立場に立つと，同じ実験から表3のような結果が得られる．前述のように，品詞変換規則を適用するだけで品詞列が一意に決まったのは 73,460 文節あった，これに文節内制約を加えると，新たに 3,672 文節の品詞列が一意に決まった．さらに文節間制約を加えると，新たに 12,395 文節の品詞列が一意に決まった．合わせて 89,527 文節（総文節数の 85%）の品詞列が特定できたことになる．この結果からも，文節間制約の利用が曖昧性の解消に大きく貢献することがわかる．

表 3: 実験結果： 品詞列の決定

品詞列が一意に決まった文節の数		
変換規則のみ	73,460	(69%)
+ 文節内制約	+ 3,672	(3.4%)
+ 文節間制約	+ 12,395	(12%)
合計	89,527	(85%)

前述のように，このうちの 81,993 文節については構文木も一意に決まっており，さらにそのなかの 76,129 文節 (u^2) では構文木が文節間制約も満たしている．品詞列が一意に決まっていて，構文構造に曖昧性がある場合，その約 90% が複合名詞句内の構造的曖昧性であった．複合名詞句の構造的曖昧性は構文的制約ではほとんど解消できないことがすでにわかっているので，今回のようなタスクにこの種の曖昧性を持ちこむことは建設的でない．これについても，何らかの工夫が必要である．

このように今回の設定では，構文木の決定を目標とするタスクと品詞列だけを決定すればよいというタスクの間には，とくに大きな違いが見られなかった．このことは主として日本語の文節の性質に起因するものと推測できる．

4 関連研究

品詞体系の自動変換に関してはすでにいくつかの先行研究が見られる．

植木らは，品詞体系の違いは文節内構造にしか影響しないと考え，文法を品詞体系非依存部分（文節間間構造）と依存部分（文節内構造）の2階層に分けて記述するアプローチを提案している．品詞体系により依存部分の文法のみを入れ替え，非依存部分に共通の文法を用いることで品詞体系の違いを吸収しようとする試みである [6]．品詞体系の違いを文節内構造で吸収するという考え方は，本稿で述べたアルゴリズムでも重要な前提であり，ソース側文節境界や係り受け情報をターゲット側での形態素・構文解析における制約として用いることの根拠になっている．このように植木らの考察は示唆に富むが，具体的な変換アルゴリズムや実験結果はまだ報告されていない．

一方，田代らは，ソース側品詞タグとターゲット側品詞タグがともに付与された訓練コーパスから，ソース側の単語・品詞対とターゲット側の単語・品詞対の対応規則を抽出し，これと通常の形態素解析技術を併用するアルゴリズムを提案している [5]．本稿で述べたアルゴリズムでは品詞対品詞（列）の対応規則（品詞変換規則）しか用いないのに対し，対応規則の中に単語表記情報を組み込んでいる点が特徴的である．田代らはのアルゴリズムでは，訓練コーパスに現れないパターンに対応するために，必要に応じて上の対応規則を品詞対品詞の対応規則に一般化して用いる．したがって，田代らの対応規則は，我々の品詞変換規則を一般化したものと見なすことができ，品詞変換における弁別能力も高いと考えられる．また，田代らの対応規則では複数のソース側単語が複数のターゲット側単語に対応するという一般的な対応関係を扱うことができるのに対し，我々の品詞変換規則はソース側単語が一つの対応関係しか扱えない．一方，我々のアルゴリズムでは，品詞変換規則の他に，ソース側の文節境界や係り受け関係といった情報や，ターゲット側の文節内制約・文節間制約といった資源を用いる．これらの情報や資源は田代らの algorithm では利用されていない．定量的な評価については，田代らは品詞数 15 の体系から品詞数 32 の体系への変換実験の結果を報告しているが，本稿で報告した実験では品詞数 416 から 621 への変換を行ったので，両者の結果を単純に比較することは難しい．むしろ，両手法はたがいに対立するものではないので，今後はこれらの統合を検討すべきだと考えられる．

5 おわりに

本稿では，形態素・構文解析器を用いて既存のコーパスのタグを異なる品詞体系に変換するアルゴリズムを提案した．品詞タグや係り受け情報など，ソース側の情報を最大限に利用することにより，単純に形態素・構文解析する場合に比べ，タグづけの曖昧性を大幅に削減できることを，一つのケーススタディを通して示した．ただし，本稿で提案したアルゴリズムのままでは，ソース側の単語・文節境界よりもターゲット側の単語・文節境界が荒い場合を扱うことができない．この点の拡張が今後の課題である．また，既存の形態素・構文解析技術と併用する実験についても進めていく必要がある．

謝辞

実験に当たっては，東京工業大学で開発された MSLR 構文解析器を利用させていただきました．開発者の田中穂積氏，白井清昭氏，植木正裕氏，橋本泰一氏（同大学）に感謝いたします．また，植木正裕氏からは，同氏が開発した文法の他，本研究に対する有益なコメントをいただきました．深く感謝いたします．

参考文献

[1] EDR. 電子化辞書仕様説明書 第2版. Technical report, 日本電子化辞書研究所,3 1995.

[2] 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト． 人工知能学会全国大会予稿集, pp. 58–61, 1997.

[3] Li, Tanaka A method for integrating the connection constraints into an LR table. In *Proceedings of Natural Language Processing Pacific Rim Symposium '95*pp703–708,1995.

[4] 田中穂積 (東工大), 竹澤寿幸 (ATR), 衛藤純司 (ランゲージウェア). MSLR 法を考慮した音声認識用日本語文法. 情報処理学会研究報告 (音声言語処理研究会)15-25, pp. 145–150, 1997.

[5] 田代敏久, 森元． 形態素情報付きコーパスの再構築手法． 情報処理学会論文誌, Vol.37, No.1, pp.13–22,1 1996.

[6] 植木正裕, 白井清昭, 徳永健伸, 田中穂積. 構造つきコーパスの共有化に関する一考察. 情報処理学会研究報告 (98-NL-128)128-9, pp. 61–66, 1998.