統計的部分係り受け解析における係り受け確率の利用法
— コーパス中の構文タグ誤りの検出 —

乾 孝司 [*1] 乾 健太郎 [*1 *2]

[*1] 九州工業大学大学院情報工学研究科
[*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
{t_inui,inui}@pluto.ai.kyutech.ac.jp

品質の高いコーパスを作成するためには，構文解析器によって自動的にタグづけした後，それを人手で修正する作業が不可欠である．コーパス中のタグ誤りを効率的に見つける方法があれば，コーパス修正の人的コストを大幅に削減できると考えられる．本稿では，統計的部分係り受け解析方式によって係り受け確率を推定し，これを係り受けタグの誤り検出に利用する方法について論じる．京大コーパスを用いて実験をおこなった結果，係り受け確率がある程度誤り検出に利用できるという見通しが得られた．

[キーワード] 統計的部分解析, 係り受け確率, 誤り検出, コーパス修正

An application of Probabilistic Partial Parsing
— Detection of Syntactic-Tag Errors in Treebanks —

INUI Takashi[*1] and INUI Kentaro[*1 *2]

[*1] Department of Artificial Intelligence, Kyushu Institute of Technology, JAPAN
[*2] PRESTO, Japan Science and Technology Corporation, JAPAN

We have been exploring a way of enhancing the current state of the art of statistical parsing by reintroducing the notion of partial parsing, which we call probabilistic partial parsing. Among the various advantages of probabilistic partial parsing, in this paper, we discuss the feasibility of applying it to the task of error detection in treebanks. This task is to retrieve erroneous tags from the all tags that are inconsistent with the parser's outputs. The results of our preliminary experiments on the Kyoto Japanese corpus shows that the scheme of our probabilistic partial parsing improves the performance of this retrieval task.

[keyword] probabilistic partial parsing, dependency structure, tag-error detection, corpus refinement

1 はじめに

近年，大規模コーパスの利用によって構文解析の精度を向上させる試みが盛んに行われており，報告されている性能も徐々に上がってきている．たとえば，Wall Street Journal による実験では labelled precision が 86% を越えたと報告されており [1]，日本語でも新聞記事を対象とした係り受け解析の実験で 85% ~ 90% の精度が得られたという報告がある [9, 6, 4, 10]．これらの成果はコーパスの大規模化と言語知識の統計的学習方法の高度化によるところが大きいが，構文解析という作業が実際には意味解釈や省略補完などの作業と不可分であることを考えると，現在のような統計的手法を単独で用いるアプローチには深刻な限界があると予想される．

このような背景から，我々は統計的構文解析に部分解析の考え方を導入することを提案し，その有効性の調査を進めている [2, 3]．本方式では，文節間の係り受け構造のような依存構造の解析を前提とする．確率言語モデルを用いて個々の部分的な依存関係の確信度を計算し，十分に高い確信度をもつ依存関係だけを選択的に決定することにより部分解析を実現する．個々の依存関係の確信度は，確率言語モデルに基づいて順序づけされた文全体の依存構造の上位 n 個の候補 (n-best) による重みつき多数決によって決まる．

この方式には次のような利点があると期待できる．

- 解析器を利用する側から見ると，解析器の出力の精度を用途に応じて任意の高さに設定することができるという利点が期待できる．確信度の高い依存関係だけを決定するようにすれば，解消できる曖昧性の数は減るが，高い精度が得られる．逆に，確信度の低い依存関係も決定するようにすれば，精度は落ちるが，決定できる依存関係の数は増える．このようなトレードオフを利用者が選択できるようになれば，現状の不完全な統計的構文解析器でも用途が広がると考えられる．
- 解析器を洗練・拡張する開発者 (技術者) 側から見ると，解析器の問題点の検出に有用な情報が得られるという利点が期待できる．我々の方式では，従来のように 1 位の解析結果だけを分析する方法とは異なり，上位 n 個の解析結果を横並びで比較する．これによって，1 位の解析結果だけからでは得られない豊富な情報を引き出すことが可能になる．得られた情報は，解析精度の評価，誤りの分析，コーパスのデバッグといった作業に利用できると考えられる．

前者の利点については，文節係り受け解析の場合の実験結果を文献 [2, 3] で報告した．(a) n-best の解の重みつき多数決によって見積った確信度は正解率と強い正の相関関係を示すこと，(b) これを利用すれば係り受け関係の決定について上述のようなトレードオフが得られること，などが確かめられている．これに対し後者の利点については，定性的な議論による示唆に留まり，大規模な実験による仮説の検証が課題として残されていた．

そこで本稿では，コーパスのデバッグに確信度を利用する方法について論じる．京大コーパス [7] のように文節間の係り受け情報（係り受けタグ）が付与されたコーパスは，種々の言語分析や言語知識の獲得に極めて有用である．品質の高いコーパスを作成するためには，構文解析器によって自動的にタグづけした後，それを人手で修正する作業が必要になる．しかしながら実際には，作業者の不注意やゆれなどが原因でタグに誤りが残る場合が少なくない．これに対し，コーパス中のタグ誤りを効率的に見つける方法があれば，コーパス作成の人的コストを大幅に削減できると考えられる．

このような問題意識に基づき，我々は次のような方法を検討した．前述のように，n-best の解の重みつき多数決によって見積った確信度は正解率と強い正の相関関係がある．したがって，解析器が高い確信度でコーパスの係り受けタグと異なる結果を出力した場合，あるいは解析器がコーパスの係り受けタグに極端に低い確信度を割り当てた場合は，係り受けタグの方が誤りである可能性が高いと予想される．この仮説が正しければ，解析器の出力と係り受けタグが一致しない箇所を確信度の高いものから順に人手でチェックすることによって，効率的にタグ誤りを発見できると考えられる．

2 係り受け確率

前節で述べたように，タグ誤りの検出には，解析器が個々の係り受け関係に割り当てる確信度が重要な役割をはたすと考えられる．文献 [2, 3] で述べたように，我々はこの確信度を係り受け確率によって見積もる．係り受け確率は，ある文節を特定したときに，その文節がどの文節にどの程度の確率で係るかを推定する分布である．入力文を s，その係り受け構造を R とするとき，R に含まれる個々の係り受け確率は，元になる確率言語モデルが $P(R, s)$ の分布を推定するトップダウンなモデルか，$P(R|s)$ の分布を推定するボトムアップなモデルかによって，その推定方法が異なる．以下本節では，まずトップダウンなモデルにおける係り受け確率の推定方法，次にボトムアップなモデルにおける係り受け確率の推定方法を簡単に振り返る．また，3 節以降の議論で用いる用語と記法を 2.3 でまとめて定義する．

2.1 トップダウンな言語モデルにおける係り受け確率の推定

次のような文節切りされた品詞タグつき入力文に対し，その係り受け構造を特定するタスクを考える．

表 1: 例文 (1) の解析結果

係り		b_1	b_2	b_3	b_4	b_5	b_6	b_7	$P(R_i)$
受け	R_1	b_8	b_8	b_4	b_5	b_8	b_7	b_8	5.11e-31
	R_2	b_5	b_3	b_4	b_5	b_8	b_7	b_8	1.32e-31
	R_2	b_5	b_4	b_4	b_5	b_8	b_7	b_8	1.01e-31
	R_4	b_5	b_5	b_4	b_5	b_8	b_7	b_8	7.36e-32
	R_5	b_2	b_8	b_4	b_5	b_8	b_7	b_8	7.35e-32
	R_6	b_8	b_8	b_4	b_5	b_6	b_7	b_8	9.76e-33
	R_7	b_8	b_8	b_4	b_5	b_7	b_7	b_8	8.95e-33
	R_8	b_2	b_3	b_4	b_5	b_8	b_7	b_8	4.71e-33
	R_9	b_2	b_4	b_4	b_5	b_8	b_7	b_8	3.61e-33
	R_{10}	b_2	b_5	b_4	b_5	b_8	b_7	b_8	3.48e-33

(1) [政界にも]$_1$ [二十代、]$_2$ [三十代の]$_3$ [若者が]$_4$ [飛び込み]$_5$ [「戦後政治」の]$_6$ [幕が]$_7$ [上がりました。]$_8$

i 番目の文節を b_i とすると，この文の正解の係り受け構造は，

係り	b_1	b_2	b_3	b_4	b_5	b_6	b_7
受け	b_5	b_3	b_4	b_5	b_8	b_7	b_8

のような係り文節から受け文節への写像関係として表現できる．この入力文に対し，たとえば，トップダウンな言語モデルを用いる白井らの統計的係り受け解析システム [9] は表 1 のような解析結果を出力する．ここで，R_i は入力文に対する係り受け構造の i 番目の候補であり，言語モデル $P(R_i)$ によってランキングされている[1]．

係り受け解析の研究では，一般に 1 位の候補だけが評価の対象となるので，研究者の注意が 1 位の候補に集中する傾向にあった．しかしながら，表 1 のように 2 位以下の候補も一緒に横に並べてみると，より多くの情報がそこから引き出せることがわかる．たとえば，b_1 の係り先の候補を見ると，b_2，b_5，b_8 の 3 つが有力で，いわばシステムがその判断に「迷っている」と言うことができる．これに対し，b_3 の係り先については，上位 10 位までの候補がいずれも b_4 で一致しており，システムがその判断に「自信を持っている」ことがわかる．このように，上位 n 位の候補を横並びで見ると，各文節についての係り先の確率分布（係り受け確率）を見積もることができる．

係り受け確率は，上位 n 位の候補を求めたあと，各係り受け関係を含む候補の確率を出現係り受けごとに足し合わせ，解析結果全体の確率の和で正規化することによって推定できる．

形式的には，ある確率言語モデルが生成する係り受け構造の集合を $\mathcal{R}$ とし，その確率分布を $P : \mathcal{R} \mapsto [0,1]$ $\left(\sum_{R\in\mathcal{R}} P(R) = 1\right)$ とする．さらに，ある係り受け構造 $R \in \mathcal{R}$ の文節 b_i が文節 b_j に係ることを式 $R \models r(b_i, b_j)$，R の終端記号列が文 s であることを式 $R \models s$で表すことにする．$R \models s$ を満たす R のうち確率の高い上位 n 個の集合を $\mathcal{R}_H$，$R \models s$ を満たす R のうち残りの確率の低いもの集合 $\mathcal{R}_L$ とすると，このとき，「入力文 s が係り受け関係 $r(b_i, b_j)$ をもつ」という命題に対する係り受け確率，すなわち $P(r(b_i, b_j)|s)$ は次式で推定できる（但し，r は $r(b_i, b_j)$ の略記である）[2]．

$$P(r(b_i,b_j)|s) = P(r|s) \approx \frac{P^{s\wedge r}_{\mathcal{R}_H}}{P^{s}_{\mathcal{R}_H}} \tag{1}$$

$$P^{s}_{\mathcal{R}_H} = \sum_{R\in\mathcal{R}_H : R\models s} P(R) \tag{2}$$

$$P^{s\wedge r}_{\mathcal{R}_H} = \sum_{R\in\mathcal{R}_H : R\models s\wedge r} P(R) \tag{3}$$

2.2 ボトムアップな言語モデルにおける係り受け確率の推定

藤尾らは $P(R|s)$ を推定するボトムアップな言語モデルを提案し，これを部分解析に利用する方法を提案している [10]．藤尾らの言語モデルは，それぞれの係り受け構造が独立であると仮定し，$P(R|s)$ を以下のように展開する．

$$P(R|s) = \prod_{i=0}^{m-1} P(r(b_i,b_j)|s) \tag{4}$$

$$= \prod_{i=0}^{m-1} P(r(b_i,b_j)|f_1,\ldots,f_m) \tag{5}$$

ただし，f は文節属性，m は文節数を表し，s は属性付けされた文節列をあらわす．このとき $P(r(b_i,b_j)|f_1,\ldots,f_m)$ が係り受け確率に相当する．藤尾らはこれをさらに部分分布の積に展開し，パラメータ推定の精度を向上させている．

また，内元らは同様の係り受け確率を最大エントロピー法を用いて推定する手法を提案している [4]．

2.3 解析誤り文節とタグ誤り文節

任意の係り文節 X について，次の 3 タイプの受け文節候補を考える[2]．

- $B(X)$：係り受け確率が最大となる受け文節
- $T(X)$：コーパス係り受けタグが示す受け文節
- $C(X)$：X の真の受け文節

これらのタイプの組合せは図 1のように 5 通りとなる[3]．

[1]実装上また効率性の都合で，すべての候補に共通する周辺分布は $P(R_i)$ の計算に含まれていないので，表 1 の $P(R_i)$ は正確な確率を表しているわけではない．

[2]添字 {B,T,C} はそれぞれ {Best,Tag,Correct} の略．

[3]図 1中の白丸で記した文節の順序には特に意味付けしていない．

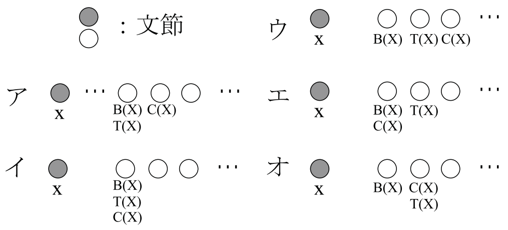

図 1: B(X),T(X),C(X) の組合せ

コーパスを解析結果の正解判定に用いた場合を考えると，解析が正しいと判定された $(B(\mathrm{X}){=}T(\mathrm{X}))$ 文節は { ア，イ } に該当し，解析が誤っていると判定された $(B(\mathrm{X}) \neq T(\mathrm{X}))$ 文節は { ウ，エ，オ } に該当する．また，係り受けタグが誤っている $(T(\mathrm{X}) \neq C(\mathrm{X}))$ 文節は { ア，ウ，エ } に該当する．

以降，本稿では，$B(X) \neq T(X)$ の関係を満たす場合を解析誤り，この時の文節 X を解析誤り文節と呼ぶ．また，$T(X) \neq C(X)$ の関係を満たす場合をタグ誤り，この時の文節 X をタグ誤り文節と呼ぶ．例えば，{ ウ } に該当する文節は，解析誤り文節であり，かつタグ誤り文節である．また，{ オ } に該当する文節は，解析誤り文節ではあるがタグ誤り文節ではなく，{ ア } に該当する文節は，タグ誤り文節ではあるが解析誤り文節ではない．

ここで，X の受け文節 $B(\mathrm{X})$，$T(\mathrm{X})$ の係り受け確率をそれぞれ $P_B(\mathrm{X})$, $P_T(\mathrm{X})$ で表すと，研究の目的は $P_B(\mathrm{X})$, $P_T(\mathrm{X})$ を誤り尤度の推定に利用し，{ ア，イ，ウ，エ，オ } の中から { ウ，エ } だけを優先的に選出することである[4]．

3 予備調査

予備調査として，実際に京大コーパス [7] のテキストを統計的部分係り受け解析することによって，どの程度タグ誤りを発見することができるかを調べた．本節では，この調査の概要と結果を報告する．

3.1 調査の概要

係り受け解析には，白井ら [9] が開発した言語モデルを使用し，2.1 で述べた方式で係り受け確率を推定した．係り受け確率の推定には，上位 300 位の解析木を用いた．京大コーパスの 1 月 4 日分 1,115 文の品詞タグつきの単語列を解析対象とし，クローズドテストをおこなった．このうち解析に成功した文は 1,059 文 (8083 文節) で，文節当たりの係り受け正解率は 83.7% であった．

つぎに，解析誤り 1,241 文節のうち，ランダムに選択した 941 文節を手作業で調査し，タグ誤りの可能性のある文節を抽出した．この作業は，京大コーパス作成の作業基準 version1.4[7] に基づいておこない，作業基準に合わない係り受けタグをタグ誤りの候補と判定した．また，この他に，作業者の言語直観に合わない係り受けタグも誤りの可能性があると考え，タグ誤りの候補とした．ただし，並列関係や同格関係に付随する係り受けタグについては，タグづけの基準自身に見直しを要する点が多いことが最初からわかっていたので，今回の調査では対象外とした．この作業の結果，57 文節のタグ誤りの候補が得られた．これらの候補について，コーパス作成者に問い合わせたところ，57 文節のうち 46 文節がタグ誤りと判定された．解析誤り 941 文節のうち，46 文節 $(46/941 = 4.9\%)$ がタグ誤りであったことになる（実際には，この他に並列関係や同格関係に付随するタグ誤りが存在する）．以下，この作業によって得られたタグ誤り文節の集合を [4 日分データ] と記す．

[4]仮説より，今回はアに該当する文節の誤りは考慮しない．

表 2: 抽出したタグ誤り文節 $P_T(\mathrm{X})$ の分布．

$P_T(\mathrm{X})$	0.0	(.0 , .10]	(.10 , .20]	(.20 , .30]	(.30 , .40]	(.40 , .50]
[4 日分データ]	17	10	5	6	4	4
[差分データ]	18	8	8	5	3	1

上の作業と並行して，京大コーパス version1.0 と version2.0 のデータの差分を利用して，タグ誤りを抽出する作業をおこなった．具体的には，2 つの版を比較し，文の文節区切りがすべて等しく，係り受けタグだけが異なる文節をタグ誤り文節と見なした．この作業によって抽出されたタグ誤りは，43 文節であった．この後さらに，抽出した文節を含む文を解析し，係り受け確率を計算した．以下，この作業によって得られた誤り文節の集合を [差分データ] と記す．

3.2 結果

以上の作業によって抽出したタグ誤り文節 $(46+43 = 89$ 文節) を対象として，係り受け確率 $P_T(X)$ の分布を調べた．結果を表 2 に示す．表は，$P_T(X)$ の確率区間範囲ごとのタグ誤りの頻度を表している．この結果から，[4 日分データ]，[差分データ] のタグ誤りの集合はともに $P_T(X)$ の小さい区間に偏って分布していることがわかる．逆に言えば，$P_T(X) = 0.0$ の解析誤りを調べるだけで，今回抽出したタグ誤り集合の約 4 割を検出することができ，$P_T(X) \leq 0.10$ の解析誤りに調査の対象を広げれば，今回抽出したタグ誤りの約 6 割を検出することができたことになる．このことは，係り受け確率を利用することにより，実際のコーパスに含まれるタグ誤りを効率的に検出できる可能性があることを示唆している．ただし，この調査はまだ十分な規模でなく，上の仮説が経験的に裏づけられたとは言いがたい．

4 仮想データによる実験

前節で述べた予備調査は，得られたタグ誤りの数が十分でなかったため，係り受け確率との相関関係に統計的有意性を見出すことができなかった．また，コーパス中のタグ誤りの全体集合が未知であるので，解析誤りから発見できたタグ誤りが全体のどれくらいの割合だったのかを調べることもできない．そこで，タグ誤りを人工的に発生させ，係り受け確率の利用がそれらの誤りの検出にどの程度寄与するかを評価する実験をおこなった．本節では，この実験の概要と結果を報告する．

4.1 タグ誤りの生成

以下の手順でタグ誤りを人工的に発生させた．まず，前節の予備調査で得られたタグ誤り(計89文節)を人手で調べたところ，次のような傾向が見られた．

- タグ誤りの係り受け関係は，正しい係り受け関係と同様，他の係り受け関係と交差しない．また，係り先文節はつねに一つである．京大コーパスの作成に当たっては，係り受けタグの修正のためのツールが使われている．作業者が非交差条件を破る修正を行うと，このツールは交差箇所をハイライトさせ，作業者に注意を促す．これにより，人為的ミスによって係り受け関係が交差してしまう可能性はほとんどない．
- 1文中に複数のタグ誤りが生じるケースは少ない（タグ誤りが見つかった79文のうち，複数のタグ誤りを持つ文は7文しかなかった）．
- 名詞と「ノ」以外の助詞からなる文節が同じく名詞と「ノ」以外の助詞からなる文節に誤って係る場合が目立った（89箇所のタグ誤りのうち24箇所）．たとえば，次の文では「支部は、」の係り先が誤って「日本で」になっているが，正しくは「進めてきた」である．

  例. *支部は、 [日本で] 昨年 五月 条約が 発効したのをきっかけに、 条約の 浸透を 監視する 同委員会の 招へい準備を [進めてきた。] (950104216-004)

次に，これらの傾向をふまえて以下の手順でタグ誤りを発生させた．ただし，名詞と「ノ」以外の助詞からなる文節を便宜的に「補足語文節」と呼ぶ．

1. コーパス中から以下の条件を満たす文を網羅的に集める．
   (a) 複数の補足語文節を含む．
   (b) 文中のある補足語文節 P の係り先を $T(P)$ でない別の補足語 Q としたときに，その係り受け関係が文中の他の係り受けと交差しない．ただし，$T(P)$ は P の元の係り受けタグである．たとえば，次の文では，「国連総会が」の係り先「任命する」を「十一人で」に置き換えても，他の係り受け関係と交差しない．

   例. 世界の 地域 別に 選ばれ、 *国連総会が [任命する] [十一人で] 構成される。(950104020-003)
2. 1で作った文の集合から，n 個の文をランダムに選ぶ．
3. 2の各文について，
   - 1.b. の条件でタグ誤りを発生させる．
   - ただし，1文中にタグ誤りの発生のさせ方が複数ある場合は，ランダムに1つを選ぶ．

今回の実験では，上の手順にそって，コーパス全体にランダムに分布するように949個のタグ誤りを発生させた．

4.2 実験のセッティング

タグ誤りの検出実験では，京大コーパス19,950文（これを文集合 $\mathcal{S}_{19950}$ と呼ぶ）に対し，10分割のcross validationをおこなった．構文解析モデル，部分解析方式，その他のセッティングは予備調査と同様である．トレーニングデータには実際の解析を模擬するためにタグ誤りを挿入したコーパスを用いた．文節当たりの係り受け正解率は82.7%であった．以下では，このセッティングにおける解析結果を[OURS]と記す．

また，藤尾らの部分解析方式を用いた解析も同時に行った．藤尾らの言語モデルと部分解析方式は構文解析システム茶掛[10]上に実装されており，今回の解析実験ではこのシステムを利用させていただいた．茶掛は，平文を入力として受けとり，形態素解析システム茶筅[11]の出力を元に係り受け解析を行う．このシステムを使って $\mathcal{S}_{19950}$ の各文を解析した結果，コーパスの文節情報と同一の文節区切りを出力した文が9,448文あった．これを文集合 $\mathcal{S}_{9448}$ と呼ぶ．$\mathcal{S}_{9448}$ における文節当たりの係り受け正解率は85.7%であった．以下では，このセッティングにおける $\mathcal{S}_{9448}$ の解析結果を[FUJIO]と記す．

4.3 結果の分析

[OURS]では，22,401文節の解析誤りがあり，このうちタグ誤りは840文節あった．今回の実験では，タグ誤りの総数(949文節)うち，9割(= 840/949)近くの誤りが解析器の出力と異なっていた($B(X) \neq T(X)$)ことになる．すなわち，少なくとも今回のセッティングでは，解析誤りの文節を調べるだけで約9割のタグ誤りが検出できることになる．

以下では，解析誤りの集合(22,401文節)からいかに効率的にタグ誤り(840文節)を検出できるか，という観点から実験結果を分析する．

まず，解析誤りとタグ誤りの出現頻度の分布を係り受け確率 $P_B(X)$ と $P_T(X)$ の区間の組 ($P_B(X)$, $P_T(X)$) についてを調べた．頻度の高かった確率区間を表3，表4に示す．ただし，係り受け確率の確率区間には，[0.0,0.01)から[0.99,1.0)の100区間および

表 3: 解析誤り文節 (22401 文節) の頻度

$(P_B(\mathrm{X}),P_T(\mathrm{X}))$	出現頻度	$(P_B(\mathrm{X}),P_T(\mathrm{X}))$	出現頻度
(0.99, 0.0)	1041	(0.96, 0.0)	138
(0.98, 0.0)	277	(0.91, 0.08)	138
(0.98, 0.01)	245	(0.96, 0.03)	135
(0.97, 0.0)	177	(0.81, 0.18)	124
(0.97, 0.02)	149	(1.0 , 0.0)	462

表 4: タグ誤り文節 (840 文節) の頻度

$(P_B(\mathrm{X}),P_T(\mathrm{X}))$	出現頻度	$(P_B(\mathrm{X}),P_T(\mathrm{X}))$	出現頻度
(0.99, 0.0)	189	(0.93, 0.0)	16
(0.98, 0.0)	52	(0.91, 0.08)	14
(0.97, 0.0)	32	(0.94, 0.0)	12
(0.95, 0.0)	28	(0.98, 0.01)	11
(0.96, 0.0)	21	(1.0,0.0)	9

1.0 の 101 区間を用いている．例えば，$P_T(X) = 0.01$ は確率区間 [0.01,0.02) を表している．また，全体の分布を図 2に示す．“×” が解析誤り文節，“◆” がタグ誤り文節を表している．これらの図表から，タグ誤りは $P_T(X)$ が低い (かつ $P_B(X)$ が高い) 区間に偏って分布していることがわかる．したがって，タグ誤りを効率的に検出するには，解析誤りのうち，$P_T(X)$ が低い文節，あるいは $P_T(X)$ が低く $P_B(X)$ が高い文節を優先的に調べればよい．

そこで次に，[OURS] に含まれる解析誤り文節に対し次の 3 つの方法で優先度を与え，優先度の高い順にタグ誤りかどうかを調べるというタスクを想定し，そのときのタグ誤り検出の効率を再現率と適合率で評価した．

Random： ランダムに選択．

Low： $P_T(X)$ の低い順．

Low +： $P_T(X)$ 値の低い順．ただし，$P_T(X)$ が同一の確率区間内にある場合は，$P_B(X)$ の高い文節を優先する．

適合率と再現率は次式で与えられる．

$$\text{適合率} = \frac{\text{検出したタグ誤り文節数}}{\text{選択した文節数}}$$

$$\text{再現率} = \frac{\text{検出したタグ誤り文節数}}{\text{解析誤り中に含まれるタグ誤り文節数}}$$

結果を図3に示す．Random に比べ Low および Low+ はタグ誤りを効率的に検出できることがわかる．再現率が低い場合に Low+ の適合率が落ち込んでいるのは，$(P_B(\mathrm{X}), P_T(\mathrm{X}))$=(1.0，0.0) における適合率が極端に低いことを反映している．しかし，$(P_B(\mathrm{X}), P_T(\mathrm{X}))$=(1.0，0.0) のときに適合率がさがる原因については，現在のところ不明であり，今後さらに詳しく調査する必要がある．

また，図 4は選択する文節数を増加させた時のそれぞれの順序付けにおける再現率の変化を表している．図 4は，例えば，5000 文節を選択した時点で，Random で選択した場合に比べ，Low，Low+ のそれぞれの順序で選択した場合では，約 3 倍のタグ誤り文節を検出できることを示している．

なお、[FUJIO] の解析誤り文節を対象とした評価でも，[OURS] と同様の結果が得られた (図 5，図 6)

4.4 複数モデルの利用

次に，[OURS] と [FUJIO] に含まれる文節の係り受け確率を混合して順序付けに利用することを試みた．

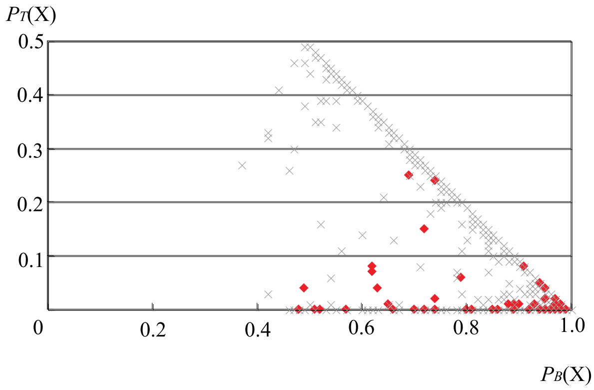

図 2: 解析誤りの分布

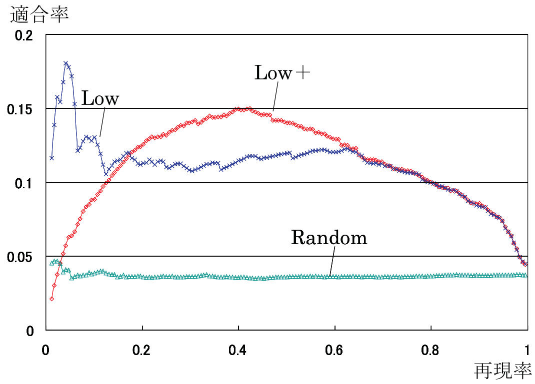

図 3: 再現率と適合率の関係 [OURS]

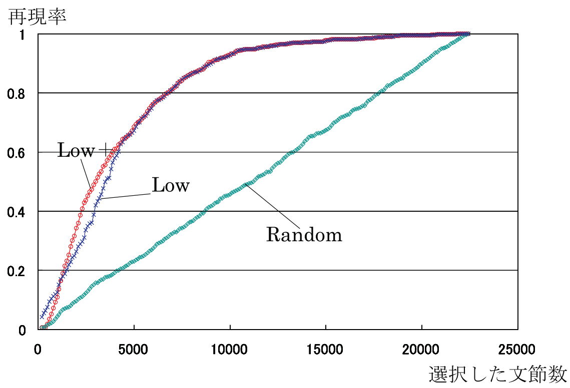

図 4: 選択文節数と再現率の関係 [OURS]

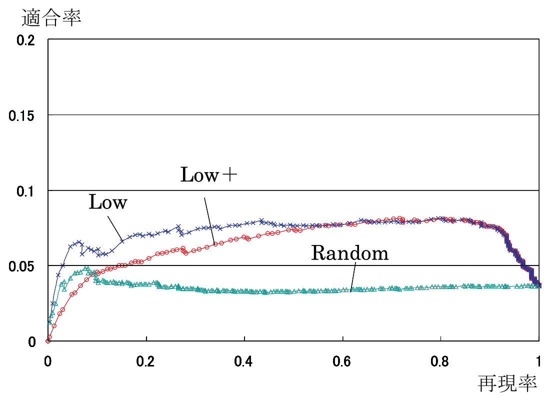

図 5: 再現率と適合率の関係 [FUJIO]

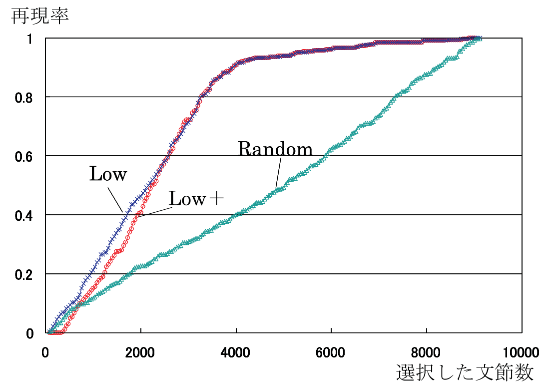

図 6: 選択文節数と再現率の関係 [FUJIO]

両者のセットでともに $P_T(\mathrm{X})\simeq 0$ となる係り受けタグは，2 つの言語モデルからともに強く否定されていることにより，これらの文節を優先的に選択することで，タグ誤り文節の検出率が向上すると予想される．そこで，次の手続きに従って文節を選択した．

1. [OURS] に含まれる解析誤り文節と [FUJIO] に含まれる解析誤り文節の積集合をとり，両者に共通する文節を抽出する．
2. 1. で抽出した文節に付与された各 $P_T(\mathrm{X})$ 値の相乗平均をとる．どちらかの文節 (あるいは両者) が $P_T(\mathrm{X})$=0.0 の場合は値がすべて等しくなってしまうため，$P_T(\mathrm{X})$=0.0 となる文節の $P_T(\mathrm{X})$ 値に 0.01 を加え，その値を用いて相乗平均をとる．
3. 2. で求めた値の低い順に選択する．
4. 1. で抽出した文節をすべて選択した後，抽出されなかった残りの文節に対して，Low の順序で選択する．

図 7の LOW[Committee] が上記の順序で選択した場合の再現率と適合率の関係を表している．また，Low[OURS] が [OURS] による結果，Low[FOJIO] [FUJIO] による結果である．

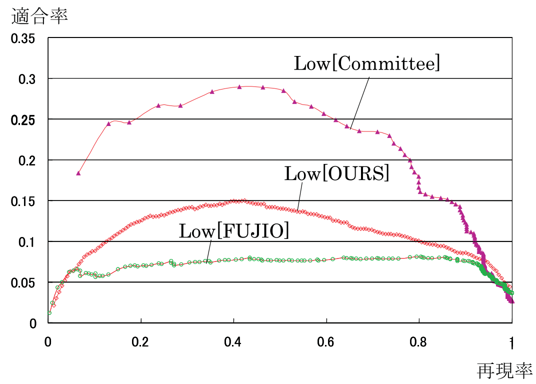

図 7: 複数モデルによる再現率と適合率の関係

図 7より，2 つのモデルの係り受け確率を組み合わせることで，タグ誤り文節の検出率が向上していることがわかる．今回は，2 つのモデルの組み合わせであったが，3 つ以上のモデルを組み合わせて利用することにより，検出率がさらに向上する可能性もある．

4.5 解析精度が高い場合

係り受け解析精度が 90% 以上の解析器を利用した場合のタグ誤り検出率を推測する試みとして，[OURS] の中で比較的精度が高いヲ格の係り受け関係に注目し，ヲ格のタグ誤り文節の検出率を調査した．ヲ格に限定した時の係り受け正解率は約 92% である．

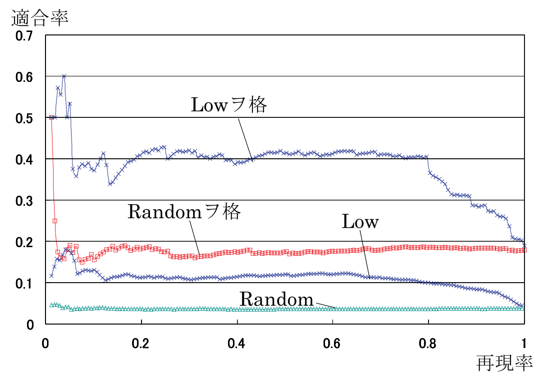

図 8: ヲ格に限定した場合の再現率と適合率の関係

図 8中の Low ヲ格は，Low の順序で解析誤り文節を選択した場合のヲ格に限定した再現率 - 適合率曲線の関係であり，Random ヲ格は，Random で選択した場合のヲ格に限定した再現率 - 適合率曲線である．比較のためにヲ格に限定しない場合の再現率 - 適合率曲線もプロットしている．

図 8によると，Low の順序で選択した場合，ヲ格に限定しない場合の適合率が 20% に満たないのに対し，ヲ格に限定した場合の適合率は 40% 近くに達していることがわかる．今回は誤りタグをランダムに付与しているので，相対的にヲ格のタグ誤りの密度がヲ格に限定しない場合のタグ

誤りの密度に比べて高いわけではない．それにもかかわらず Random ヲ格が Random よりも高い適合率となっているのは，ヲ格の真の解析誤り (図 1のオ) の文節数が少ないことを反映している．この結果は，広範囲の文節種類において高い係り受け精度をもつ解析器を利用すれば，精度に比例してタグ誤り検出率も向上することを示唆している．

5 関連研究および議論

コーパス誤りの検出に関する研究として，内山 [5] は形態素解析の誤りを発見する統計的尺度を提案しており，この尺度が人手で修正されたコーパスに残る過分割を検出するのに有効であると報告している．しかし，構文タグ誤りの検出に関する研究は，今のところ報告されていない．

コーパス中のタグ誤りを減らす方法として，たとえば黒橋らはコーパスと構文解析器をカップリングさせ，それらを交互に洗練する方法を提案し，成果を上げている [7]．黒橋らの方法では，まずコーパス中の誤りから解析器の誤り傾向を調べ，解析器を洗練する．つぎに，洗練した解析器でコーパスを再度解析し，解析結果が変化した部分だけについて人手で正誤を判定する．このサイクルを繰り返すことによって，人的コストを抑えながらタグ誤りの削減をはかる．本稿で述べた手法は，黒橋らの枠組でも利用可能であり，黒橋らのサイクルで見過ごされてしまうタグ誤りも検出することができる可能性がある．今後は，両者を統合する試みを進める必要がある．

今回の実験では，係り受け確率の利用がタグ誤りの検出にある程度寄与することが確かめられたが，本稿で報告した適合率ではまだ不十分であり，現状では修正作業のコストを劇的に軽減するには至らないと予想される．この点からも，本稿で述べた方法を単独で用いるのでなく，黒橋らが提案しているような枠組と組み合わせる方法を検討すべきであろう．また，実際に人手でタグ誤りをチェックする場合には，文単位で作業するのが自然だと考えられるので，今回の実験ように解析誤りを文節単位で順序づけしたときの適合率・再現率を議論するのは必ずしも適当でない．例えば，タグ誤りの候補を一つしかを含まない文を調べるより，複数のタグ誤りが含まれていそうな文を優先的に調べる方が誤り検出の効率は上がるだろう．本稿で述べた方法では，このような要因は考慮していない．これについても今後の課題である．

6 おわりに

本稿では，統計的係り受け解析における係り受け確率の一利用法として，コーパスの係り受けタグの誤り検出への有効性を実験を通して検証した．その結果，係り受け確率がある程度誤り検出に利用できるという見通しが得られた．今後は，構文解析モデルの性能向上をはかるとともに，文脈を考慮した誤り検出に対する検討を行っていく予定である．

謝辞

実験に当たっては，東京工業大学で開発された統計的構文解析器を利用させていただきました．同大学の田中穂積氏，白井清昭氏，植木正裕氏，橋本泰一氏に感謝いたします．また，奈良先端科学技術大学院大学の藤尾正和氏には，係り受け解析システム茶掛の使用を許可していただきました．心よく実験に協力してくださいました同氏に深く感謝いたします．京大コーパスの誤り判定の際には，京都大学の黒橋禎夫氏に多くの助言を頂きました．深く感謝いたします．

参考文献

[1] E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 15th National Conference on Artificial Intelligence*, 1997.

[2] 乾健太郎, 白井清昭, 田中穂積, 徳永健伸. 統計に基づく部分係り受け解析. 言語処理学会, 第 4 会年次大会予稿集, pp. 386–389, 1998.

[3] 乾孝司, 木村啓, 乾健太郎. 統計的部分構文解析器のふるまいについて. 言語処理学会, 第 5 会年次大会「構文解析–現状の分析と今後の展望–」ワークショップ論文集, pp33–40, 1999.

[4] 内元清貴、関根聡、井佐原均. 最大エントロピー法に基づくモデルを用いた日本語係り受け解析情報処理学会論文誌, Val.40, No.9, pp.3397–3407, 1999.

[5] 内山将夫. 形態素解析結果の誤りを発見する統計的尺度. 情報処理学会, 自然言語処理研究会, NL129-11, 1999.

[6] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 自然言語処理, Vol. 1, No. 1, pp. 35–57, 1994.

[7] 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.

[8] 黒橋禎夫. 日本語構文解析システム KNP 使用説明書 version 2.0b6. 京都大学大学院情報学研究科, 1998.

[9] 白井清昭, 乾健太郎, 徳永健伸, 田中穂積. 統計的構文解析における構文的統計情報と語彙的統計情報の統合について自然言語処理, Vol.5, No.3, pp.85–106, 1998

[10] 藤尾正和, 松本裕治. 語の共起確率に基づく統計的部分解析と冗長解析. 言語処理学会, 第 5 会年次大会「構文解析–現状の分析と今後の展望–」ワークショップ論文集, pp71–78, 1999.

[11] 松本裕治, 北内啓, 山下達雄, 平野義隆. 日本語形態素解析システム茶筅 ver2.0 使用説明書. 奈良先端科学技術大学院大学, 1999.