社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－NL－172（2）
2006／3／27

システム主導型コーパス作成インターフェースとその評価

野口 正樹　　市川 宙　　橋本 泰一　　徳永 健伸

東京工業大学 大学院情報理工学研究科 計算工学専攻
{mnoguchi, ichikawa, taiichi, take}@cl.cs.titech.ac.jp

近年，構文情報付きコーパスの構築支援に関する研究が盛んに行われている．従来の支援システムは，使いやすいインターフェースの開発などが関心の中心で，作業の順序や作業基準については作業者の解釈にまかされている．一方，協調作業の支援の一つに，作業者間でのノウハウ共有が挙げられる．しかし，ノウハウの共有を行うためには，コーパスの作成過程を正規化する必要がある．本論文では，ノウハウ共有によるコーパスの構築支援の第一歩として，システムが主導となって，どの順に曖昧性を解消すべきかを作業者に提示する手法を提案する．支援システムは，構成素構造の曖昧性をトップダウン・深さ優先で解消し，次に文法範疇の曖昧性をボトムアップ・幅優先で解消する．また，提案手法をシステムに実装し，従来のシステムと提案システムを用い，正解率と作業効率の比較を行った．その結果，初心者の作業者においては，提案システムは作業過程が統制されているにもかかわらず，正解率と作業時間について，従来のシステムとほぼ同等のシステムであることが確認できた．

System Initiate Corpus Making Interface and Evaluation

Masaki Noguchi, Hiroshi Ichikawa, Taiichi Hashimoto, Takenobu Tokunaga
Department of Conputer Science, Toyko Institute of Technology
{mnoguchi, ichikawa, taiichi, take}@cl.cs.titech.ac.jp

Many systems have been developed for creating syntactically annotated corpora. However, they mainly focused on usability in interfaces and decision making in the work process is left to the annotators. To incorporate functionality of knowledge sharing for supporting decision making, we emphasized the importance of noramilizing annotation process. As a first step toward knowledge sharing, this paper proposed a method of system initiative annotation in which the system suggests annotators the order of solving ambiguities. To be more concrete, the system forces annotators to solve ambiguity of constitutent structure in a top-down and depth-first manner, and then to solve ambiguity of grammatical category in a bottom-up and breadth-first manner. We implemented the system and conducted experiments to compare an existing system and the proposed system in terms of annotation accuracy and efficiency. We found that at least for novice annotators, the proposed system provides more efficint annotation means while keeping annotation accuracy comparable with the previous system.

1 はじめに

近年，自然言語処理の分野では，大規模な言語資源に基づく統計的手法が研究の中心となっている．特に構文木付きコーパスは，確率的構文解析モデルの学習データや，構文解析システムの評価用テストセットなどに用いられ，統計的構文解析手法のための重要な言語資源である．しかし，構文木付きコーパスを作成するには，多くの人手と時間を必要とする．そのため，構文木を付与する作業を支援するための研究が行われている．

Penn Treebank コーパスの作成で用いられたツール [3] や Negra コーパスの作成で用いられたツール [5] では，構文木をグラフィカルに表示し，それを操作することで構文木を作成できる．東工大コーパスの作成に用いられたツール [1] では，構文木の集合から一つの構文木を選択することで，構文木の付与を可能にしている．

一方，複数の作業者が一つの作業を行う協調作業や共同作業の支援に関する研究が行われている．その一つに，敷田らの手法が挙げられる [7]．敷田らは，複数の作業者が共通で行う作業過程に着目し，作業結果や判断理由などの情報（ノウハウ）を共有し作業者を支援する手法を提案している．まず，作業過程において，それまでの作業の履歴と，そこで用いられたノウハウを蓄積する．そして，作業者の作業履歴に類似した履歴をもとに，ノウハウを探し出し，作業者に提示する．

これまでの構文木付きコーパス作成支援システムでは，構文木を付与する操作を支援することが研究の中心であり，どのような基準で構文木を付与するかという意志決定に関する支援についてはほとんど関心が払われてこなかった．実際のコーパス作成では，作業者用のマニュアルを用意し，それを作業者が理解することを前提として作業をおこなってきた．作業者の負担を軽減し，コーパスの品質を向上させるためには，作業者間で作業のノウハウを共有し，作業過程の適切な時点で適切なノウハウを参照できるような支援のしくみを導入することが重要である．そのためには作業過程における状況とその状況で有効な情報を対応づけたデータベースをシステムに用意する必要がある．

ノウハウのデータベースの検索キーとなるのは作業過程における状況である．したがって，作業過程における状況を作業者によらずできるだけ正規化する必要がある．しかし，従来のシステムでは，作業者が操作順序を自由に決めており，最終的に同じ構文木が付与された場合でも，作業履歴が作業者によって異なる場合がある．作業過程の状況を正規化するためには，作業過程を統制する必要がある．

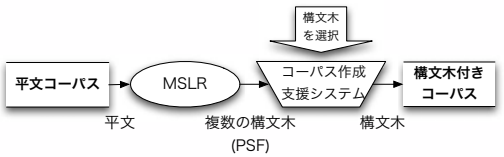

図 1: 構文構造付きコーパス作成の流れ

本研究では，ノウハウを共有する支援を加えた作成支援環境の構築を目指し，その一歩として，システム側が作業過程を統制する手法を提案する．我々が東工大コーパスを作成するために開発した eBonsai[1] に提案手法を組み込み，eBonsai の従来のインターフェースと提案手法を実装したインターフェースを用いた被験者実験によって，正解率と作業時間の点で比較した．その結果，提案手法の方が作業時間が短いことが分かった．

2 構文木付きコーパス作成支援統合環境 eBonsai

eBonsai を使った構文木付きコーパス作成の概要を図 1 に示す．

コーパス作成の流れは以下のとおりである．

1. 平文コーパスから文を取り出す．
2. 文を MSLR パーザで構文解析する．
3. 得られた構文木の集合の中から，正しい構文木を選ぶ．
4. 選んだ構文木を構文木付きコーパスに加える．

2. の MSLR パーザ [4] は，与えられた文法を用いて解析した結果得られる全ての構文木を，圧縮共有統語森（PSF）[6] の形式で出力する．3. では，出

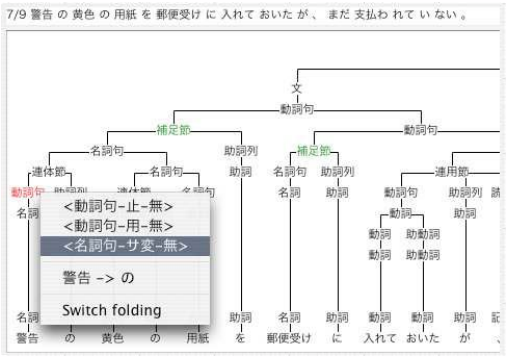

図 2: 従来のインターフェース

図 3: 提案手法のインターフェース

力された PSF の中から正しい構文木を選ぶ作業を人手で行う．

eBonsai の従来のアノテーションインターフェースでは，構文木候補の 1 つを視覚的に表示する．図 2 に従来のインターフェースのスクリーンショットを示す．作業者は，表示された構文木の曖昧性を解消することを繰り返し，候補を絞り込む．作業者が解消する曖昧性には,「構造の曖昧性」と「ラベルの曖昧性」の 2 つがあり，曖昧性を含むノードには色が付けられている．作業者はこれら色のついたノードを自由に選択し，そのノードにおける選択肢から一つを選択することで曖昧性を解消することが出来る．作業者は，この操作を候補が 1 つの構文木になるまで繰り返す．

3　新しい構文木選択手法

協調作業や共同作業といった，複数人で行う作業を支援する研究が行われている．その一つに，敷田らの手法がある．これは，共通の作業過程において，それまでの作業履歴とそこで用いられたノウハウを蓄積し，現在作業者の作業履歴に類似した履歴をもとに，データベースからノウハウを探し出し，作業者に提示する．

この手法による支援を構文木付きコーパスの作成に適応することを考えると，作業過程を構文木を選択することに，作業履歴を作業者の操作履歴に対応づけることができる．しかし，従来の構文木選択のインターフェースでは，作業者が自由に操作できるため，作業過程に多様性が生じ，敷田らの手法を適用するのが困難になる．そこで，作業過程が発散することを防ぐために，システムが作業過程を統制し，作業者の作業過程の多様性を押さえ込む手法を提案する．

3.1　作成過程の統制

与えられた文法を使って文を解析した結果，複数の構文木が得られた場合，構文の曖昧性があると言う．句構造文法を採用した場合，構文の曖昧性は，構成素構造の曖昧性と文法範疇の曖昧性から成る．構成素構造の曖昧性とは，構文木の各ノードのラベルを無視し，その構造が複数あるために生じる曖昧性である．文法範疇の曖昧性とは，ノードのラベルの異なりにより生じる曖昧性である．

本手法では，まず構成素構造の曖昧性を解消し，次に文法範疇の曖昧性を解消する．

1. 構成素構造の曖昧性解消：曖昧性には依存関係があり，局所的な曖昧性を先に解消すると最終的に正しい構文木に辿り着けない場合があるため，トップダウン・前順序で解消する
2. 文法範疇の曖昧性解消：単語の語彙情報が有力な手がかりになるため，構成素構造の曖昧性が解消された後に，ボトムアップ・幅優先で解消する

3.2　例

次の例文を用いて，各曖昧性の解消手順を説明する．曖昧性の解消時には，図 3 のように選択肢を作

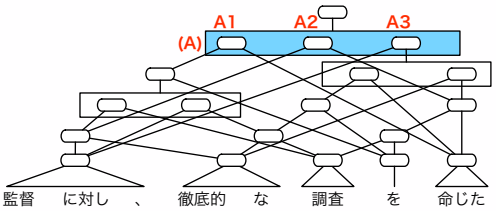

図 4: PSF の状態 (1)

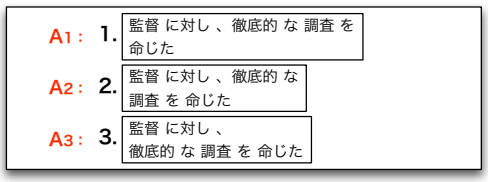

図 5: (A) において作業者に提示する選択肢

業者に提示する．

『監督に対し、徹底的な調査を命じた』

この文を解析した結果，39 個の構文木が得られる．

はじめに，複数の構成素構造からラベル無しの圧縮共有統語森（PSF）を構築する．図 4 に PSF の状態を示す．四角で囲まれた複数のノードはパックされているノードを表す．

PSF のパックされている部分の曖昧性を解消することで，構成素構造の曖昧性を解消する．パックされているかどうかは，ルートノードからトップダウン・前順序で辿りながら探索する．

まず，図 4 のノード (A) がパックされているノードとして検出される．A1,A2,A3 の異なりは構造の異なりであるが，木構造を提示すると煩雑になるので，図 5 のように，文の区切りのみを作業者に提示する．作業者は提示された選択肢から，正しい文の区切りを選択する．図 5 の選択肢では，作業者は正しい選択肢 3 を選択し，PSF の状態が図 6 のように変化する．その結果，構文木の候補が絞られる．

次に，パックされているノードを A3 から引き続き探索し，ノード (B) を検出する．ノード (B) についても同様に，作業者に選択肢を提示する．図 7 に示すように，パックされたノードが無くなり，構成素構造が 1 つ選択されるまで繰り返す．

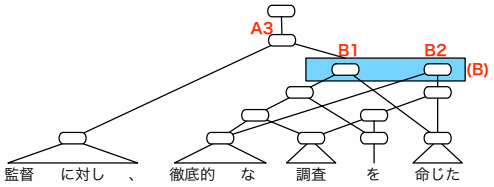

図 6: 選択後の PSF の状態 (1)

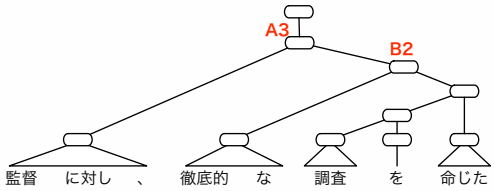

図 7: 構成素構造選択後の PSF の状態 (1)

構成素構造の曖昧性に続き，文法範疇の曖昧性を解消する．残った複数の構文木からラベル付きの圧縮共有統語森（PSF）を構築する．図 8 に PSF の状態を示す．四角で囲まれたノードがパックされているノードを表す．

ボトムアップ・幅優先に辿って行くと，図 8 中のノード (C) がパックされているノードとして検出される．C1,C2,C3 の異なりは，ラベルの異なりなので，図 9 のように文の区切りとラベルの異なりを提示する．図 9 の選択肢では，C1 が正しいので，作業者は正しい選択肢 1 を選択し，PSF の状態が図 10 のように変化する．その結果，構成素構造の場合と同様に，構文木の候補が絞りこまれる．

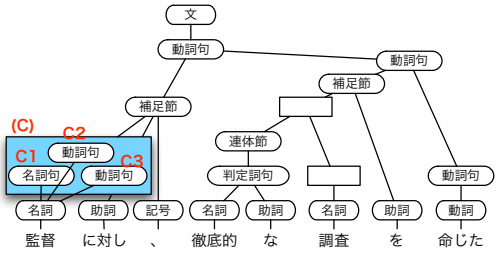

図 8: PSF の状態 (2)

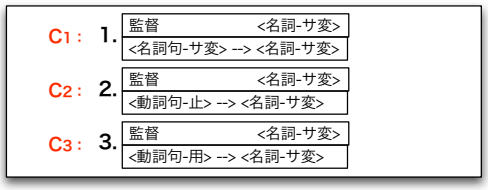

図 9: (C) において作業者に提示する選択肢

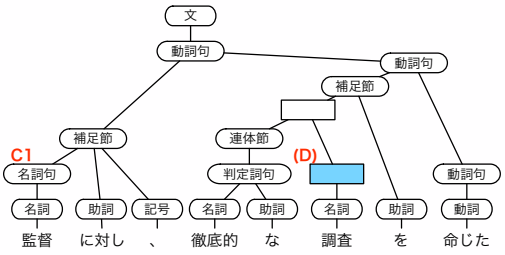

図 10: 選択後の PSF の状態 (2)

さらに，パックされているノードをボトムアップ・幅優先に探索し，ノード (D) を検出する．ノード (D) についても同様に，作業者に選択肢を提示する．

最終的に，パックされているノードが無くなるまでこれを繰り返す．この例文では，図 11 に示す構文木が正しい構文木として選択される．

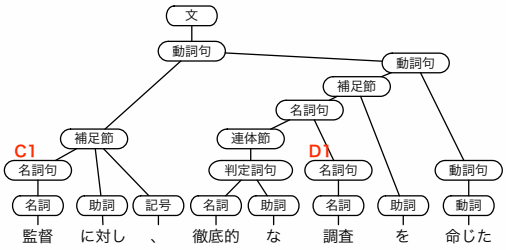

図 11: 選択された構文木

4 評価実験

4.1 実験環境

eBonsai の従来手法と提案手法で，コーパス作成の作業効率を比較するために被験者により評価実験を行った．評価実験には，毎日新聞の記事 17,664 文に構文木を付与したコーパス (東工大コーパス) を用いた．東工大コーパスから抽出した句構造文法 (東工大文法) と MSLR パーザを用いて，東工大コーパス中の文を解析し，PSF を獲得した．そして，東工大コーパスを 200 文づつ 88 のセットに分割し，構文的曖昧性の平均と分散が近い 2 セット (各 200 文) を被験者実験に使用した．

東工大文法の特徴を熟知し，従来手法に慣れている熟練者 2 名と初めて構文木付きコーパス作成に携わった初心者 18 名を被験者として実験を行った．各被験者は，従来手法と提案手法の 2 つのシステムを用いて，PSF から構文木を選択する作業を各 1 セット (200 文) づつ行った．

2 つの手法の作業効率を比較するために，被験者の作業時間と選択した構文木の正解率に注目した．作業時間とは，被験者が与えられた PSF から一つの構文木を選択するまでの時間である．正解率は，次の式で計算する．

$$\text{正解率} = \frac{\text{構文木が一致した文数}}{\text{総文数}}$$

構文木が一致した文数とは，被験者が選択した構文木がコーパスの構文木と一致した文の数である．

4.2 熟練者

東工大文法の特徴を熟知し，従来手法に慣れている熟練者 2 名を被験者として実験を行った．各熟練者は，最初の 200 文を従来手法のシステムを使用し，次の 200 文を提案手法のシステムを使用して，構文木の選択を行った．選択した構文木の正解率と 1 文あたりの平均作業時間を表 1 に示す．

実験結果から，提案手法は，従来手法より平均作業時間が短いが，正解率が低いことが分かる．従来手法より作業時間が短くなっている要因は，作業者が解消すべき曖昧性を選択する手間が提案手法により改善されているためである．しかし，正解率が低い原因は，システムのインターフェースに原因がある．従来手法が構文木を表示しているため，構文木全体の構造を把握しやすい．一方，提案手法は文字列のみを表示しているため，構文木全体の構造を把握しにくい．特に，熟練者は，文法を熟知してい

表 1: 熟練者による実験結果

	従来のシステム	提案システム
正解率 (%)	78.5	74.3
平均作業時間 (秒)	42.5	35.4

表 2: 初心者による実験結果

	従来のシステム	提案システム
正解率 (%)	53.0	49.6
平均作業時間 (秒)	49.6	42.9

るため，構文木全体が把握できる従来手法の方が正しい構文木を選択しやすい．そのため，提案手法においても，文字列による選択肢だけでなく，木構造を把握しやすいインターフェースを構築する必要がある．

また，選択した構文木の正解率が約 70%から 80%であり，熟練者が作業しているのにもかかわらず高くない．この原因は，構文木付きコーパス作成が作業者による揺れが生じやすいタスクであるためである．このような作業者による揺れをなるべく少なくするために，作業ノウハウの提示や作業者間の知識共有などのシステムの拡張，作業者が分かりやすいインターフェースの作成，東工大文法の改善などが考えられる．

4.3 初心者

初めて構文木付きコーパス作成に携わった初心者 18 名を被験者として評価実験を行った．被験者は，最初の 200 文を従来手法のシステムを使用し，次の 200 文を提案手法のシステムを使用する被験者 (10 名) と最初の 200 文を提案手法のシステムを使用し，次の 200 文を従来手法のシステムを使用する被験者 (8 名) とに分けた．

被験者には，システムの使用方法を学んでもらうために，各システムを使用する前にチュートリアルとして，実験に用いていない 30 文について作業をしてもらった．チュートリアル時には，インストラクターに自由に質問をしてよいが，実験のときには，質問をせずに作業をを行った．初心者が選択した構文木の正解率と 1 文あたりの平均作業時間を表 2 に示す．

実験結果から，提案手法は，従来手法より平均作業時間が短いが，正解率が低いことが分かる．従来手法より作業時間が短くなっている要因は，熟練者と同様に，作業者が解消すべき曖昧性を選択する手間が提案手法により改善されているためであると考えられる．

システムの違いによる正解率，平均作業時間の差の有意性を確かめるために t 検定を行った．その結果，正解率については有意差はなかったが，平均作業時間については有意差があった ($t_{(70)} = 2.64$, $P = 0.01$)．しかし，実際に短縮された作業時間は，1 文あたり約 5 秒とわずかであるため，提案手法と従来手法での作業効率の差はほとんどない．

熟練者の実験結果 (表 1) と初心者の実験結果 (表 2) を比較すると，選択した構文木の正解率が約 25%の大きな差がある．初心者はコーパス作成をしたことがないため，チュートリアル (30 文) を用いて東工大文法やツールの使い方についてのインストラクションを行った．しかし，チュートリアルの 30 文では，文法を十分な理解ができなかったことが原因であると考えられる．

4.4 確率モデル

東工大コーパスから被験者実験に用いた文 (400 文) を除いた文 (17,264 文) を PCFG モデル，PGLR モデル [2] の各確率モデルの学習に用いた．被験者実験に用いた文に対して，各確率モデルの適用したときの生成確率第 1 位の構文木の正解率を表 3 に示す．

実験結果より，確率モデルの正解率が初心者の正解率に近い．さらに，確率モデルの正解率と初心者の被験者の正解率を比較したところ，18 名中 12 名

表 3: 確率モデルの正解率

	PCFG	PGLR
正解文数	168	196
正解率 (%)	42	49

表 4: 各分類の構文木の正解率 (%)

	PCFG 以下	PCFG 以上	PGLR 以下	PGLR 以上
従来システム	41.2	60.4	47.5	63.7
提案システム	35.1	58.8	41.8	65.2

表 5: 各分類の平均作業時間 (秒)

	PCFG 以下	PCFG 以上	PGLR 以下	PGLR 以上
従来システム	43.5	53.4	49.7	49.3
提案システム	35.9	47.4	42.3	44.3

被験者の正解率が PGLR モデルの正解率より低かった．さらに，12 名中 7 名については，PCFG モデルよりも正解率が低い結果であった．

4.5 初心者の分類

4.4 節より，PCFG モデル，PGLR モデルそれぞれの構文木の正解率よりも低い被験者と高い被験者に分類し，その違いについて検討した．分類方法と被験者の数を下記に示す．

- **PCFG** 以下:
  正解率が PCFG モデルよりも低い初心者 7 名
- **PCFG** 以上:
  正解率が PCFG モデルよりも高い初心者 11 名
- **PGLR** 以下:
  正解率が PGLR モデルよりも低い初心者 12 名
- **PGLR** 以上:
  正解率が PGLR モデルよりも高い初心者 6 名

そして，各分類の被験者の平均作業時間と選択した構文木の正解率を表 5，表 4 に示す．

従来手法と提案手法にかかわらず，選択した構文木の正解率が確率モデルよりも低い初心者は，平均作業時間が短い傾向にある．熟練者の平均作業時間と比較すると，選択した構文木の正解率が確率モデルよりも高い初心者は，作業時間が長い．一方，正解率が確率モデルよりも低い初心者は，作業時間が短い．正しい構文木を選択することができない初心者は，チュートリアルの不足などの原因による学習不足が考えられる．作業時間を考慮することによって，ノウハウを獲得してもよいほど初心者が上達したかどうか，正しい構文木を選択できるようになったかどうかの判断材料として利用することができると考えられる．

5 まとめと今後の課題

本研究では，構文木付きコーパス作成支援システム eBonsai をベースに，ノウハウ共有機能を導入するための準備として，システム主導で構文木を選択する手法を提案した．提案手法では，構成素構造の曖昧性をトップダウン・深さ優先で解消し，次に文法範疇の曖昧性をボトムアップ・幅優先で解消する．また，提案手法をシステムに実装し，従来のシステムと提案システムを用い，正解率と作業効率の比較を行った．その結果，初心者の作業者においては，提案システムは作業過程が統制されているにもかかわらず，正解率と作業時間について，従来のシステムとほぼ同等のシステムであることが確認できた．

今後の課題として，正解率の向上のために，作業ノウハウの提示や作業者間の知識共有などのシステムの拡張，作業者が分かりやすいインターフェースの作成などが考えられる．構文木の選択作業は作業者による揺れが生じやすいタスクであるため，東工大文法の改善や作業時間を基にした作業ノウハウの獲得方法を考えることも必要になる．

参考文献

[1] Hiroshi Ichikawa, Masaki Noguchi, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. ebonsai: An integrated environment for annotating treebanks. In *The 2nd International Joint Conference on Natural Language Processing.*, Oct 2005.

[2] Kentaro Inui, Virach Sornlertlamvanich, Hozumi Tanaka, and Takenobu Tokunaga. Probabilistic GLR parsing: A new formalization and its impact on parsing performance. 自然言語処理, Vol. 5, No. 3, pp. 33–52, 1998.

[3] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1994.

[4] K. Shirai, M. Ueki, T. Hashimoto, T. Tokunaga, and H. Tanaka. Mslr parser tool kit — tools for natural language analysis. *Special Interest Group of Natural Language Processing (IPSJ-SIGNL)*, Vol. 7, No. 5, pp. 93–112, 2000.

[5] Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*, Washington, DC, 1997.

[6] M. Tomita. Efficient parsing for natural language, 1986.

[7] 敷田幹文, 門脇千恵, 國藤進. フローに連携した組織内インフォーマル情報共有手法の提案. 情報処理学会論文誌, Vol. 41, No. 10, pp. 2731–2741, Oct. 2000.