構文木付きコーパス作成支援統合環境 eBonsai の新しいインターフェース

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

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 [6]．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の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．

本研究では，eBonsai をベースに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．また，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を行い，正解率と作業時間を比較した．その結果，提案手法の方が作業時間が短く，作業効率が良いことが分かった．

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．

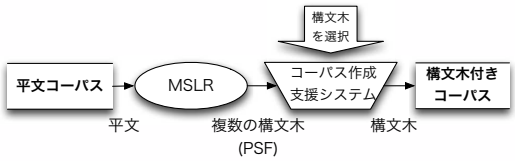

図 1: 構文構造付きコーパス作成の流れ

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．

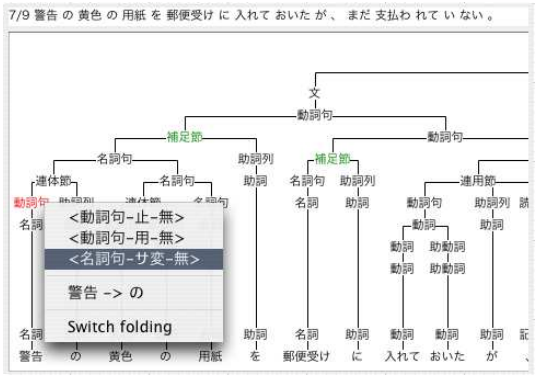

図 2: 従来のインターフェース

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 [2] は，与えられた文法を用いて解析出来る全ての構文木を，圧縮共有統語森 (PSF)[4] の形式で出力する．3. では，出力された PSF の中から正しい構文木を選ぶ作業を人手で行う．

2.1 従来の構文木選択インターフェース

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 新しい構文木選択インターフェース

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，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．

図 3: 提案手法のインターフェース

3.1 作成過程の統制手法

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．

3.2.1 構成素構造の曖昧性解消

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．複数のノードが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．

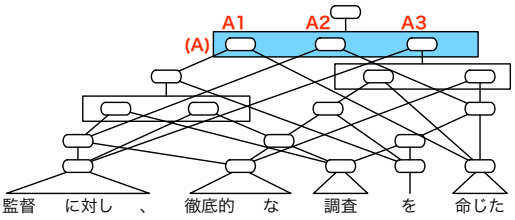

図 4: PSF の状態 (1)

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．

最初に，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 18 個に絞られる．

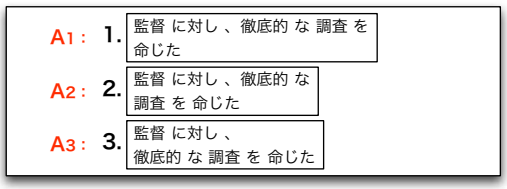

図 5: (A) において作業者に提示する選択肢

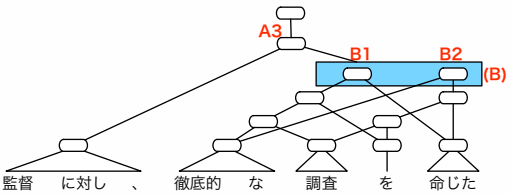

図 6: 選択後の PSF の状態 (1)

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．この例文では，構文木の候補は 9 個に絞り込まれる．

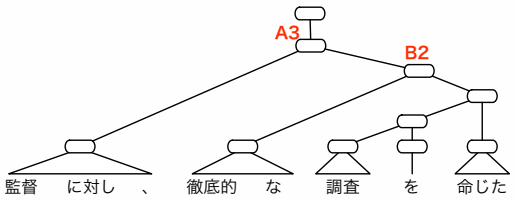

図 7: 構成素構造選択後の PSF の状態 (1)

3.2.2 文法範疇の曖昧性解消

前項に引き続き，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．

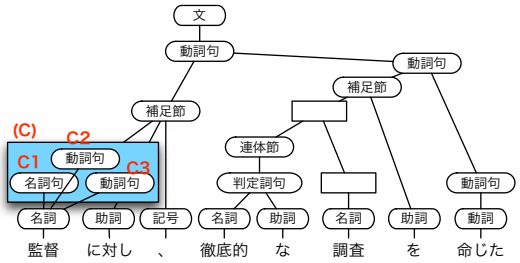

図 8: PSF の状態 (2)

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，構文木の候補が 3 個に絞られる．

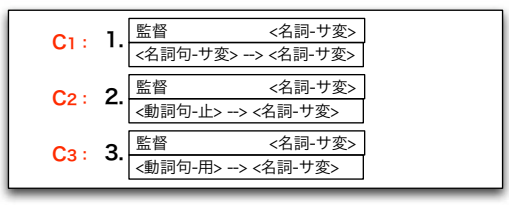

図 9: (C) において作業者に提示する選択肢

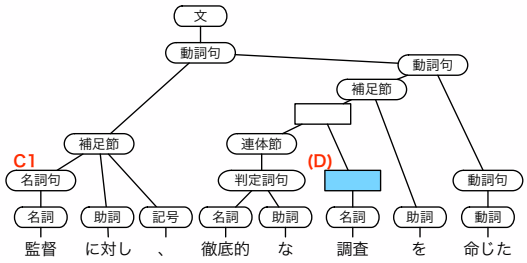

図 10: 選択後の PSF の状態 (2)

次に，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．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．

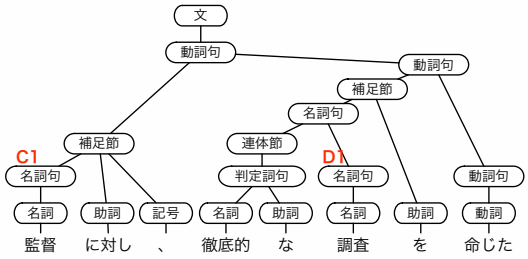

図 11: 選択された構文木

4 評価実験

従来の eBonsai の選択手法と提案手法とで，正解率・作業時間について比較するため，被験者実験を行った．テストデータには，すでに構文木が付与された文 400 文を用いた．被験者には，従来の eBonsai のシステムを使用したことがあるコーパス作成の熟練者 2 名と初めてコーパスを作成する初心者 18 名を用いた．表 1 に正解率，表 2 に 1 文あたりの平均作業時間を示す．

熟練者と初心者両方について，正解率については従来の手法の方が高いが，作業時間については提案手法の方が短いという結果となった．従来のシステムに慣れている熟練者にとっても作業時間の短縮が見られた．

5 まとめと今後の課題

eBonsai をベースに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，本研究では，システム側が作業過程を統制する手法を提案した．また，eBonsai の従来のインターフェースと

表 1: 正解率 (%)

	従来のシステム	提案システム
熟練者	78.5	74.3
初心者	53.5	49.6

表 2: 1 文の平均作業時間 (秒)

	従来のシステム	提案システム
熟練者	42.5	35.4
初心者	49.6	43.2

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を行い，正解率と作業時間を比較した．その結果，正解率は従来のシステムが良かったが，提案手法の方が作業時間が短く，作業効率が良いことが分かった．提案したインターフェースに加え，正解率を上げるための支援が必要である．

今後は，大規模な構文木付きコーパスの作成において，システム側で統制した状況に応じたノウハウの蓄積と，ノウハウの利用方法の決定と実装が必要になる．

参考文献

[1] Hiroshi Ichikawa, Masaki Noguchi, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka. ebonsai: An integrated environment for annotating treebanks., Oct 2005.

[2] 白井清昭, 植木正裕, 橋本泰一, 徳永健伸, 田中穂積. 自然言語解析のための MSLR パーザ・ツールキット. 自然言語処理, Vol. 7, No. 5, pp. 93–112, 2000.

[3] the LINC Laboratory of the Computer and Information Science Department at the University of Pennsylvania. The penn treebank project. http://www.cis.upenn.edu/~treebank/.

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

[5] Hans Uszkoreit and Thorsten Brants et al. @nnotate. http://www.coli.uni−saarland.de/projects/sfb378/negra−corpus/annotate.html.

[6] 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.