タグ付きコーパス管理/検索ツール「茶器」

松本 裕治, 浅原 正幸
奈良先端科学技術大学院大学
{matsu,masayu-a}@is.naist.jp

橋本 喜代太
大阪府立大学
hash@lc.osakafu-u.ac.jp

投野 由紀夫
明海大学
y.tono@meikai.ac.jp

大谷 朗
大阪学院大学
ohtani@utc.osaka-gu.ac.jp

森田 敏生
総和技研
morita@sowa.com

1 はじめに

言語処理および言語学において，コーパスの利用は益々その重要性が認められている．近年の言語処理では，blog などの現実的なデータの処理に対する要求が高まり，処理対象の量や分野などの横の拡がりに伴うコーパスの大規模化と多様化が望まれている．それと同時に，言語表現間の意味関係の特定や意見情報抽出のようなより深い言語解析へと処理内容が進み，このような深い解析を実践した詳細なタグ付与をコーパスに対して行うという深さ方向の要求が高まっている．大規模な言語データに深い言語解析を行ったタグ付きコーパスを構築するためには，精度の高い自動タグ付けシステムを開発することとその結果に残る誤りをいかに効率よく発見し修正するかという方法論を確立する必要がある．

本稿では，科研費の援助を受けて我々が過去３年間にわたって開発してきたタグ付きコーパス管理システム「茶器」について述べる．本システムは，言語処理および言語学と関連分野の研究者を対象に，形態素および統語的依存関係タグ付きコーパスの作成，柔軟な検索，管理を支援することを目的として構築されたものである．これまで計画年度中に部分的な公開を行ってきたシステムと同様，完成したシステムも無償公開を予定している[1]．

日本語の形態素解析と依存構造解析は，我々のグループで開発している茶筌 [3] と南瓜 [2] を用いた解析を想定しているが，それに限る訳ではない．本システムは，形態素情報や依存構造を任意に指定できる柔軟なコーパス検索を実装しており，また，発見されたタグ付け誤りを修正するためのインタフェースを提供している．修正されたコーパスは，茶筌や南瓜を再学習するための訓練データとして用いることが可能であるので，既存のタグ付きコーパスを用いるためだけでなく，むしろ，利用者独自の分野のテキストデータを対象に利用者自らがタグ付きコーパスを構築し利用する環境を提供することを目的としている．

[1] http://chasen.naist.jp/hiki/ChaKi/

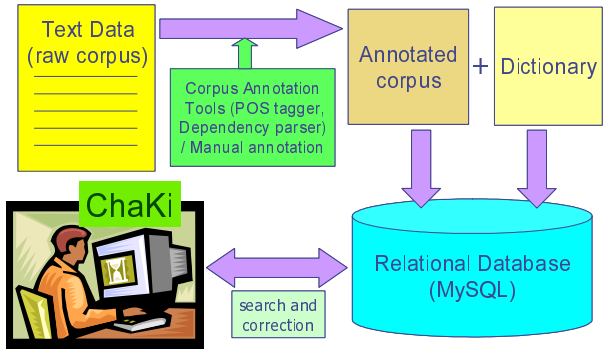

図 1: 茶器の利用状況の概念図

2 茶器の機能の概要

図１が茶器の利用状況を示している．タグ付きコーパスは，既存のもの，自動あるいは手動でタグ付けされたものいずれかの場合があるが，それは辞書と一緒にデータベースに格納される．辞書は利用者が所有しているものでもよいし，特に辞書がない場合は，コーパスに現れた語の集合が辞書とみなされる．茶器は，このデータベースに対して検索・統計処理・誤り修正などを行うインタフェースとして機能する．本節では，茶器の諸機能の概要を示し，次節でそれぞれの詳細について述べる．

1. 辞書との連携：コーパス中の各語が辞書へのポインタとして定義され，コーパスと辞書の不整合が防止される．
2. タグの種類：形態素，文節 (基本句)，依存構造など多段の構造を許容する．また，複合語についてはその構成語の情報を保持できる．また，文ごと

に書誌情報をもたせることができる．

3. 検索機能：形態素や依存構造情報を指定した検索が可能
4. 表示機能：検索結果を KWIC 表示したり，依存構造木の表示などの機能をもつ
5. 誤り修正：分かち書き，品詞情報，依存構造情報などの誤りを修正する機能をもつ
6. 統計機能：検索結果の頻度情報，前後文脈に表れる語の頻度や共起情報の計算機能をもつ
7. 言語非依存：日本語，英語，中国語のコーパスが利用できる

3 茶器の諸機能の詳細

3.1 コーパスと辞書の連携

タグ付きコーパスがデータベースに読み込まれる際に，コーパス中の単語が辞書項目へのポインタとして表現される．辞書にない単語がコーパス中に現れた場合は，辞書に仮登録され，その語へのポインタとなる．したがって，コーパスは，基本的には辞書にない語を含むことが許されず，例外的な語は明示的に管理される．利用者が特に辞書をもたない場合は，コーパスに現れたすべての語が例外的な語として登録されるだけなので，特に支障はない．現状のタグ付きコーパスでは，実際にあり得ない分かち書きや語が含まれることがあるが，そのような不整合を自動的に検出することが可能となる．現存するタグ付きコーパスでもこのような不整合は散見される．例えば，非常に多くの研究で用いられている Penn Treebank においても，“have” に動詞の過去形である “VBD” というタグが付与されている箇所があるなど実際にはあり得ない品詞タグが付与されていることがある．このようなエラーは，コーパス読み込み時に自動的に検出することができる．

3.2 文に付与されるタグの種類

形態素解析，文節 (または基本句)，依存構造の解析を施したコーパスを関係データベースに格納する．各文がそれを構成する形態素へのポインタの列として表現される，現在の辞書では，各単語は，表層形，原形，品詞 (階層的定義を許す)，活用型，活用形，読みなどの情報をもつが，コーパスごとにどの情報を検索対象の項目として用いるかを指定することができる．文節間の依存構造が付与されたデータの場合は，各文は文節の列としても表現され，文節間の依存関係が記録される．辞書内の語は自身が複合語の場合は，

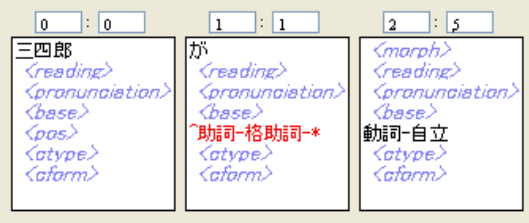

図 2: 単語検索質問の例

構成語へのポインタを定義することができる．例えば,「お好み焼き」を「お好み」という名詞と「焼き」という接尾辞が複合したものとして定義され，さらに「お好み」が「お」という接頭辞と「好み」という名詞の複合語として定義されているならば，辞書の定義によって，これらすべての構成語が辞書の定義によって「お好み焼き」に関連付けられることになる．英語の場合の multiword expression も同様に扱われる．

利用者によっては，巨大な一つのファイルによってコーパスを管理するのではなく，階層をもったディレクトリ構造 (フォルダ) を書誌情報として用い，その下にコーパスを分類して管理している場合がある．また，コーパス内の各文に何らかの文脈情報を付与する場合がある．茶器には，指定したコーパスに対して茶筌や南瓜などの言語解析ツールを適用して解析済みコーパスを作成してデータベースに格納する機能を提供するが，あるフォルダ内のコーパス群を一括して解析してデータベース化する時は，トップのフォルダから各コーパスが存在する位置までのフォルダ名の列が，各文がもつ情報 (そのようなものがあれば) と連結されて，書誌情報として文ごとに添付される．

3.3 コーパスに対する検索機能

文字列，単語，依存構造を対象とした検索機能を提供する．いずれの場合も，検索対象は文であり，質問にマッチした文が KWIC(KeyWord In Context) 形式で表示される．文字列検索では，正規表現を用いた任意の文字列が検索可能である．単語検索では，各語がもつ任意の情報を指定した連続あるいは非連続の単語列を検索することができる．例えば，図 2 は，単語検索質問の例であり,「三四郎」,「が」という格助詞，自立動詞を含む文を検索しようとしている．単語の相対的な位置が各単語ボックスの上の数字対によって指定される．$< 0,0 >$ が中心位置を表している．自立動詞の上の $< 2,5 >$ は，この動詞が「三四郎」から見て，２単語目から５単語目の位置にあることを指定している．依存構造検索では，文節間の任意の依存構造を指定した検索が可能である．例えば，

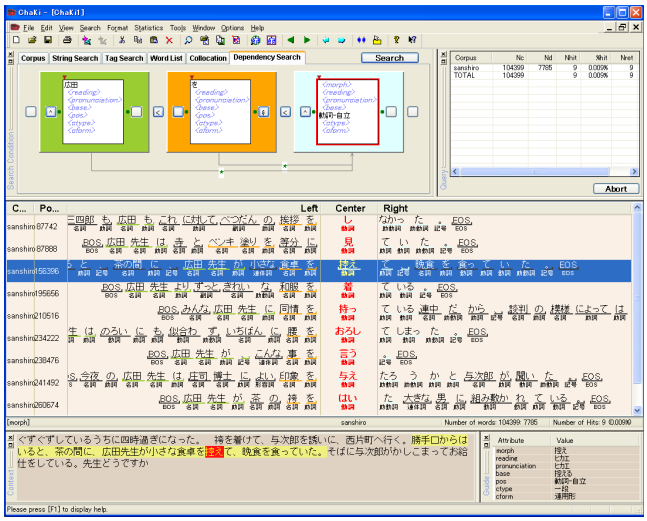

図 3: 係り受け構造検索の例

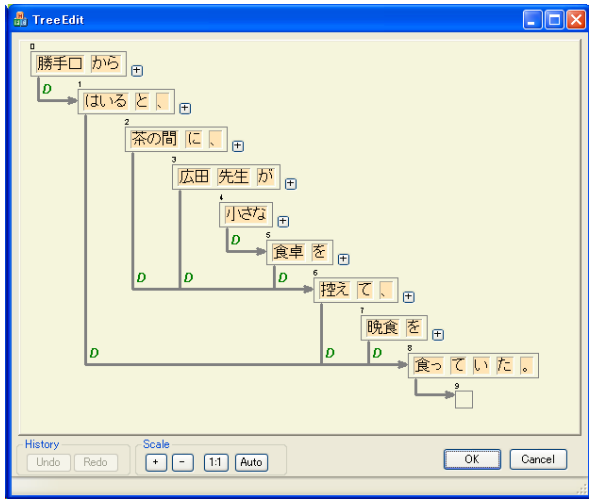

図 4: 係り受け木の表示

図 3 では，３つの文節の間の依存関係が指定されている．色づけされた３つのボックスが文節に対応し，それぞれ「広田」,「を」,「自立動詞」を含むことが指定されている．このように，文節内の単語（あるいは単語列）は，単語検索と同様に指定し，文節間の依存構造がそれぞれのボックスを繋ぐ矢印によって指定される．文節間および単語間の小さいボックスは，それぞれの相対位置を示す記号が選択可能である．図 3 では文節間の相対位置が不等号によって示されている．それ以外に，２つの文節あるいは単語が直接前後関係にある，文内または文節内の先頭位置にある，末尾位置にあるなどの指定を行うことができる．

3.4 検索結果の表示機能

文字列検索，単語 (列) 検索，依存構造検索のいずれの結果も基本的には文単位で行われ，KWIC 形式で表示される．図 3 は，依存構造検索結果の表示の例 (中央が KWIC 表示) である．単語検索，依存構造検索の場合は，各文は単語ごとに区切られ，それぞれの語の下に任意の情報 (例えば品詞名) を１つだけ表示できる．単語にカーソルを合わせると，別の (図では右下の) ウィンドウにその単語の全情報が表示される．また，一つの文の前後文脈を別の (図では下部の) ウィンドウに表示することができる．

文を選択して，その依存構造を別ウィンドウで表示することが可能である．図 4 は，図 3 中の選択された文の依存構造木を表示させた例である．各列が単語列よりなる文節を表し，それらの間の矢印が依存関係を表している．依存関係には任意個の関係名が定義可能である．図では，“D” という関係名だけが表示されている．

前節で述べた通り，複合語にはその構成語の定義が辞書内部で行われている．単語検索においては，どの構成語によってもその単語が検索されるが，KWIC 表示においては，辞書に登録された複合語のまま表示するか，最も細かい構成語のレベルで表示するかを選ぶことができる．前者の複合語表示の場合には，辞書内で構成要素の定義をもっている複合語は斜体によって表示されるので，どの語が複合語定義をもつかを知ることができる．１つの複合語を選択して，その内部構造を表示させる機能も提供している．

3.5 誤り修正機能

検索と並んで最も重要と考えている機能が誤り修正機能である．想定している誤りは，分かち書き誤り，品詞等の形態素情報誤り，文節区切り誤り，依存構造誤りである．表層文字列には誤りはないと仮定しており，本システムで表層文字列を編集することはできない．単語レベルの誤り (分かち書き誤りと形態素情報誤り) を修正するためのモジュール (TagEdit) と，依存構造レベルの誤り (文節区切り誤りと依存構造誤り) を修正するためのモジュール (TreeEdit) が用意されている．図 4 は，実は TreeEdit の画面であり，連続した文節の連結，文節内の任意の位置での分割，および，依存関係を表す矢印の修正と依存関係名の変更などの機能をサポートしている．TagEdit は，単語の連結と分割および品詞等の形態素情報の修正をサポートしている．TagEdit の特徴は，同じ誤りを含む複数の文を選択した一括修正が可能であること，および，形態素情報の修正が必ず既存の単語の選択という形で行われること (前に述べたようにコーパス

は辞書項目へのポインタとして表現されているため) である．なお，辞書にない単語を作成することも可能であるが，その場合は，例外的な語として新たに辞書に登録される．

3.6 統計解析機能

単語検索や依存構造検索の結果得られる KWIC 表示に対して，中心語と前後に出現する単語の簡単な頻度統計や共起情報の計算などの機能を提供する．頻度統計については，任意の形態素情報 (例えば，原形，品詞名，活用形など) を指定した頻度を取ることができる．共起情報については，中心語と前後に出現する単語の間の相互情報量などの共起尺度の計算を提供している．この計算も任意の形態素情報を指定した計算が可能である．

3.7 その他の機能

多言語対応 ：特定の言語に限定せず，様々な言語のコーパスにも対応可能であることを目指している．現在は，日本語，英語，中国語のコーパスを扱っている．

配布に対する制限 ：フリーソフトである関係データベースシステム MySQL[2]をコーパスの格納と検索に利用しており，その他の部分はプロジェクト内で構築しているので，これまでも開発中のシステムをフリーソフトウェアとして配布してきた．最終システムも無償公開の予定である．なお，インタフェース部はウィンドウ上でのみ利用可能である．

言語処理ツールとのインタフェース ：現実の利用としては，利用者が個別にもつタグ付きコーパスを対象としているが，タグ付け支援のために，現時点では，茶筌 [3] と南瓜 [2] を日本語の分かち書き＋品詞タグ付けと依存構造解析に用いるため，茶器からこれらのツールを呼び出す機能を設けている．１文が１行になったテキストデータがあれば，これを茶筌と南瓜で解析して，データベースに格納することができる．また，中国語のタグ付けの自動化のために，茶筌の中国語化と中国語辞書の作成を行っている [1].

本システムの実装は，本体データベースには MySQL(version 4.1) を用いてコーパスと辞書の格納を行い，ユーザインタフェース部には VisualC++を，MySQL への検索要求の生成と結果の表示処理には Ruby を用いている．

[2] www.mysql.com/

4 おわりに

科研費の援助を得て，過去３年にわたって開発してきたタグ付きコーパス管理・検索システム「茶器」を紹介した．本システムが目指したのは，コーパス利用者がタグ付きコーパスを作成し，様々な形で検索や表示を行い，また，コーパス中の誤りを修正することによって，質の高いコーパスの利用を可能にする環境である．また，どの言語にも複合語や熟語のように複数の語がまとまって一つの単語として機能することがあり，利用者によって検索対象が異なるため形態素解析を一つの出力に決めてしまうことができないという問題があった．これに対する完全な解決とは言えないが，複合語の定義を辞書中に記述し，どのレベルでも検索できる機能を提供した．統語構造については，単語あるいは文節間の依存構造に限定しているが，英語のように句構造解析が主流の言語では，句構造データの取り扱いが望まれるかも知れない．句構造については，各句構造規則に対してどの要素が主辞となるかがわかれば，機械的に依存構造に変換することができ，現在，Penn Treebank などはこの方法により依存構造に変換している．句構造に比べると依存構造では一部情報が失われるが，句構造規則を理解する必要がなく，直観的な検索が行えることと，タグ付けが容易であるという利点があり，特定の言語理論に依存しないデータの取り扱いが可能ではないかと考えている．

謝辞: 本システムの構築に様々な形で協力いただいた研究室の現メンバーおよび過去の在籍者に感謝する．なお，本研究は，文部科学省科学技術研究補助金 基盤研究 B「言語研究のためのコーパスの作成と利用に関する研究」(研究期間:平成 15 年度〜17 年度, 課題番号: 15300046) の支援を受けて行ったものである．

参考文献

[1] Chooi Ling Goh, et al., “A Practical Morphological Analyzer Based on Penn Chinese Treebank Standard,” 本論文集, 2006.

[2] 工藤拓, 松本裕治, 「チャンキングの段階適用による日本語係り受け解析」, 情報処理学会論文誌 Vol.43,No.6, pp.1834-1842, 2002.

[3] 松本裕治, 「形態素解析システム『茶筌』」, 情報処理, Vol.41, No.11, pp.1208-1214, 2000.

[4] Matsumoto, Y., et al., “ChaKi: An Annotated Corpora Management and Search System,” International Conference on Corpus Linguistics, 2005.