2006－NL－176（11）
2006／11／23

日本語 CCG の語彙項目獲得

小嶋　大起†　戸次　大介‡　宮尾　祐介†　辻井　潤一†§

† 東京大学情報理工学系研究科コンピュータ科学専攻　東京都文京区本郷 7-3-1
‡ 東京大学 21 世紀 COE「心とことば－進化認知科学的展開－」　東京都目黒区駒場 3-8-1
§National Centre for Text Mining, School of Informatics, University of Manchester POBox88, Sackville St, MANCHESTER M60 1QD, UK
† {daiki92,yusuke,tsujii}@is.s.u-tokyo.ac.jp, ‡bekki@ecs.c.u-tokyo.ac.jp

本研究では、コーパス指向の文法開発手法を用いて、日本語 CCG 語彙項目を係り受け情報付きコーパスから獲得する手法を提案する。また、京大コーパスから語彙項目を獲得する実験を行い、その結果を報告するとともに、今後の方針を述べる。

Extracting lexical entries of Japanese CCG

Daiki Kojima†　Daisuke Bekki ‡　Yusuke Miyao†　Jun'ichi Tsujii†§

†Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo　Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan
‡Center for Evolutionary Cognitive Sciences at the University of Tokyo
§National Centre for Text Mining, School of Informatics, University of Manchester POBox88,Sackville St, MANCHESTER M60 1QD, UK
† {daiki92,yusuke,tsujii}@is.s.u-tokyo.ac.jp, ‡bekki@ecs.c.u-tokyo.ac.jp

In this paper, we propose a method for extracting lexical entries of Japanese CCG from a dependency annotated corpus by means of a corpus-oriented method of grammar development. We perform an experiment of extracting lexical entries from the Kyoto Text Corpus, which is annotated with dependency information. We present the result of the experiment and make some remarks on the future work.

1　はじめに

本研究では、コーパス指向の文法開発を用いて京大コーパス [1] から日本語 CCG [2] の語彙項目を獲得する手法を提案する。

英語においては HPSG (Head-Driven Phrase Structure Grammar) [3]、CCG (Combinatory Categorial Grammar) [4]、LTAG (Lexicalized Tree Adjoining Grammar) [5] などに対して、コーパス指向の文法開発手法が用いられ [6] [7] [8]、高被覆な構文解析器が開発されるなど成果を挙げている。それに対し、日本語においては Yoshida [9] による HPSG に基づく構文解析器があるのみである。

CCG は、長距離依存や、等位接続の扱いに関しては HPSG より優れていると言われ、長距離依存などの多い日本語の解析に適した言語理論だと考えられている。よって、CCG の日本語構文解析器を作成するとより高被覆なものができる可能性がある。CCG の日本語構文解析器には、Komagata [10] のものが挙げられるが、この構文解析器は、辞書を手で記述しているため被覆率が低い。そこで本研究では、被覆率の高い日本語 CCG 構文解析器の作成のために、コーパス指向の文法開発の手法を用いて、日本語 CCG の語彙項目を獲得する手法を提案する。

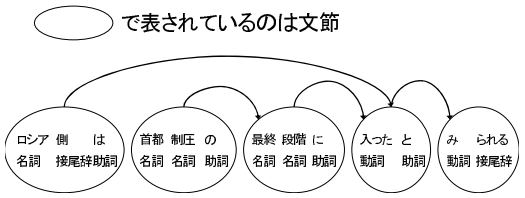

図 1: 京大コーパスの例

2　背景

本章では、京大コーパス、文法理論 CCG、コーパス指向の文法開発について説明する。

$$
\begin{array}{llll}
X/Y & Y & \Longrightarrow & X & (>) \\
Y & X\backslash Y & \Longrightarrow & X & (<) \\
X/Y & Y/Z & \Longrightarrow_B & X/Z & (>B) \\
Y\backslash Z & X\backslash Y & \Longrightarrow_B & X\backslash Z & (<B) \\
X & & \Longrightarrow_T & T/(T\backslash NP) & (>T)
\end{array}
$$

図 2: CCG の文法規則の例

$$
\dfrac{\dfrac{\text{太郎が}}{NP} \quad \dfrac{\dfrac{\text{花子に}}{NP} \quad \dfrac{\text{会う}}{S\backslash NP\backslash NP}}{S\backslash NP}<}{S}<
$$

図 3: CCG に基づく構文解析

2.1 京大コーパス

京大コーパスは JUMAN [11] で形態素解析した結果を、KNP [12] で構文解析し、さらに人手で修正した係り受け情報付きコーパスである。図 1 のように京大コーパスには、形態素解析の結果と文節間の係り受け関係を表す情報が付加されている。このほかにも、名詞などに対しては品詞細分類 (“普通名詞”、“人名”など) が、用言に対しては活用型と活用形 (“ラ行子音動詞”、“連用形”など) の情報が与えられている。

2.2 CCG

CCG は語彙化文法の一種であり、多数の語彙項目と少数の文法規則から構成される文法理論である。CCG の語彙項目は以下のように記述される。[1]

太郎は := NP
会う := $S\backslash NP\backslash NP$

これは、“太郎は”という語に対して NP、“会う”という語に対して $S\backslash NP\backslash NP$ という統語範疇 (category) が割り当てられることを表している。CCG では、統語範疇 X/Y を割り当てられた語は、右側にある統語範疇 Y を割り当てられた語と組み合わせられて、また統語範疇 $X\backslash Y$ を割り当てられた語は、左側にある統語範疇 Y を割り当てられた語と組み合わせられて、ともに統語範疇 X となることができる。

CCG の文法規則の例を図 2 に挙げる。上から順に、順関数適用規則 (forward functional application rule) (>)、逆関数適用規則 (backward functional application rule) (<)、順関数合成規則 (forward functional composition rule) (> B)、逆関数合成規則 (backward functional composition rule) (< B)、型繰り上げ規則 (type raising rule) (> T)、と呼ばれる。最右の“()”は、その文法規則の略称を表しており、図 2 の X,Y,T は任意の CCG の統語範疇を表している。

CCG ではこれらの文法規則を語彙項目に適用して文を生成する (図 3)。図 3 ではまず、“花子に”と“会う”が逆関数適用規則を用いて組み合わせられて、その後、“太郎が”と“花子に会う”が逆関数適用規則を用いて組み合わせられる。図 3 を木構造で表したものが図 4 である。CCG の表現としては一般的ではないが、本稿では以降、図 4 の表現を用いる。

本研究では、日本語 CCG 文法理論 [2] に基づく語彙項目を獲得することを目指す。表 1 に、この日本語 CCG 文法理論が、各品詞に対して割り当てる統語範疇を、また、図 5 に、この日本語 CCG 特有の文法規則を挙げる。これは日本語 CCG においては空範疇を用いて実現されている規則であり、それぞれ存在量化規則 (> ∃)、補文繰り上げ規則 (> C) と呼ぶ。この文法理論においては、名詞句と助詞を組み合わせる時には、NP に型繰り上げ規則を適用してから順関数合成規則で助詞と組み合わせる (図 6)。これは、量化された名詞句と、量化されていない名詞句を統一的に扱うためである。このため、名詞句と助詞が組み合わせられた句の統語範疇が $T/(T\backslash NP)$ となり、名詞句と助詞が組み合わせられたものと動詞句を組み合わせる際には、逆関数適用規則ではなく、順関数合成規則を用いることとなる。

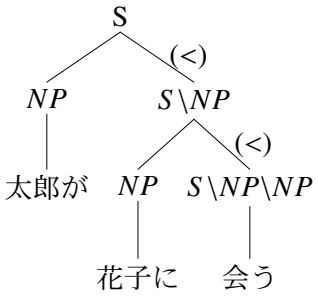

図 4: 図 3 の木構造による表現

普通名詞	N
固有名詞	NP
必須格の格助詞	$((T\backslash NP)/(T\backslash NP))$
非必須格格助詞	$(T\backslash NP)/T$
“の”	$(N\backslash NP)/N$
動詞	$S\backslash NP$,$S\backslash NP\backslash NP$...
補文	CP

表 1: 日本語 CCG の統語範疇

[1] 本稿では統語論に焦点を絞り、意味表示は省略する。

$$N \quad \Longrightarrow_{\exists} \quad T/(T\backslash NP) \quad (>\exists)$$
$$S \quad \Longrightarrow_{C} \quad T/(T\backslash CP) \quad (>C)$$

図 5: 日本語 CCG に特有の文法規則

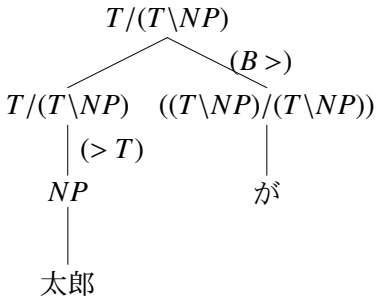

図 6: 名詞句と必須格の格助詞の組合せ

2.3 コーパス指向の文法開発

従来の文法開発では、文法規則と辞書 (語彙項目) を手で記述していた (図 7)。一方、コーパス指向の文法開発においては、文法規則は従来の文法開発手法と同じように手で記述するが、辞書はコーパスから獲得する (図 8)。すなわち、対象とする文法 (本稿では日本語 CCG) の構文情報付きコーパスを入力とし、その文法の語彙項目を獲得する。ここでいう CCG の構文情報付き木構造とは、木の形が CCG における導出木と同型であり、かつ各ノードにおいて適用されている文法規則が明示されている木構造である。例えば、図 4 に示す木構造 1 つのみを含むコーパスからは、木構造の葉ノードから以下の 3 つの語彙項目が獲得できる。

$$\text{太郎は} := NP$$
$$\text{花子に} := NP$$
$$\text{会う} := S\backslash NP\backslash NP$$

しかし現時点では日本語 CCG の構文情報付きコーパスは存在しないので、コーパス変換規則を定義して、既存の日本語構文情報付きコーパスからこれを作る必要がある。

3 京大コーパスから語彙項目を獲得する手法

本章では、京大コーパス [1] から日本語 CCG の語彙項目を獲得する手法を説明する。

3.1 京大コーパスから木構造へ

まず、京大コーパス (図 1) の係り受け情報を元に、これを図 9 のような木構造に変換する。こ

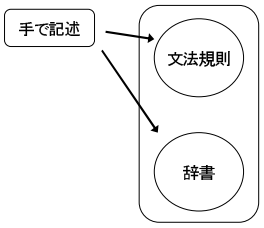

図 7: 従来の文法開発

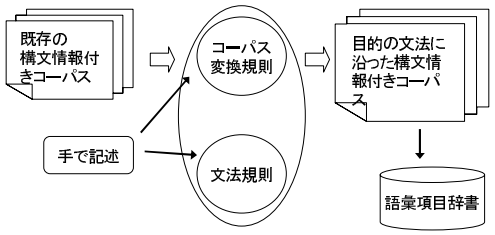

図 8: コーパス指向の文法開発

のアルゴリズムを以下に示す。文末の文節から文頭の文節に向けて以下の手順を繰り返す。

- 文節 A とそれに係る直近の文節 B に対して、
  - 文節 B に係る文節がなければ、親ノードを作り、文節 A と文節 B を兄弟とし、親ノードをこのステップの返り値とする。
  - 文節 B に係る文節 C があれば、文節 B と文節 C に対してこのアルゴリズムを再帰的に適用し、その結果のノードと文節 A に対し、上のアルゴリズムを適用する。

このアルゴリズムを適用すると葉ノードは文節となるが、文節内の形態素は右隣の形態素に係るという仮定のもとに、文節内での部分木を構成する。

3.2 助詞の繰り上げ、接頭辞、接尾辞、複合名詞の処理

京大コーパスを木構造に変形した後 (図 9)、まず助詞の繰り上げを行う。助詞の繰り上げ操作は、京大コーパスの係り受け情報を元に作った木構造と、日本語 CCG が想定している木構造の差を埋めるために行われる。日本語 CCG の場合、ある中間ノードの支配する部分木の右端の葉ノードが助詞である場合、必ず助詞はその中間ノードの直接の子ノードとなると仮定している。例えば、“首都制圧の最終段階に”という句が導出される際の木構造を図 10 に示す。一方京

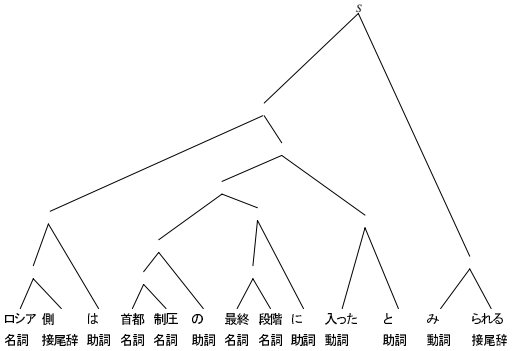

図 9: 京大コーパスを木構造に変換

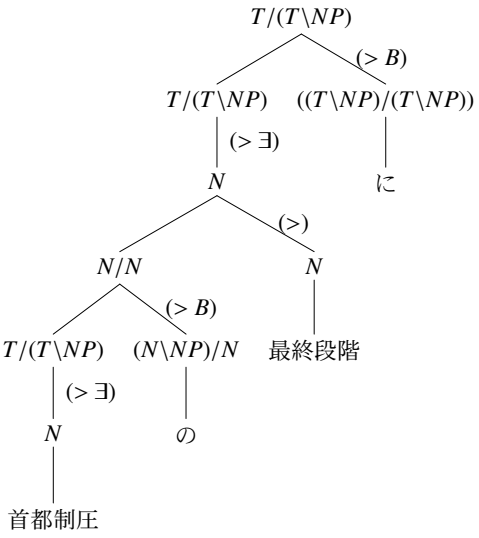

図 10: 日本語 CCG における助詞の位置

大コーパスでは“首都制圧の”は文節“最終段階に”に係ると記述されているため、助詞“に”だけを繰り上げて図 10 に示す木構造を得る。次に京大コーパスで、接頭辞と名詞、名詞と名詞、名詞と接尾辞という品詞の形態素が並んでいる場合、全て複合名詞にする変形を施す。図 11 では、名詞“ロシア”と接尾辞“側”を 1 つの“ロシア側”という複合名詞に変換している。この結果、図 9 から図 11 の木構造を得る。

3.3 単項規則を適用するノードの挿入

例えば、図 6 に示したように、名詞が助詞と組み合わされる場合、型繰り上げ規則を適用する必要がある。このように、日本語 CCG において単項規則を適用する必要がある箇所に、新たにノードを挿入する。図 12 では、“ロシア側”等のノードにこの操作が見られる。他には、主要部が助詞である句と、主要部が動詞である句

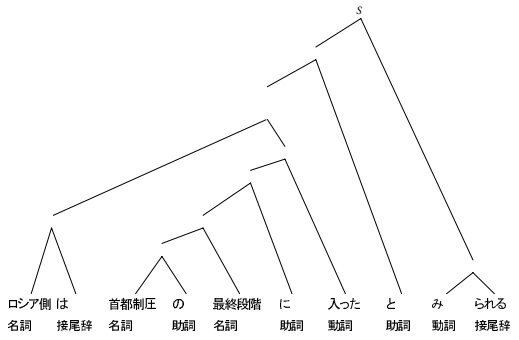

図 11: 助詞の繰上げ、接頭辞、接尾辞、複合名詞の処理

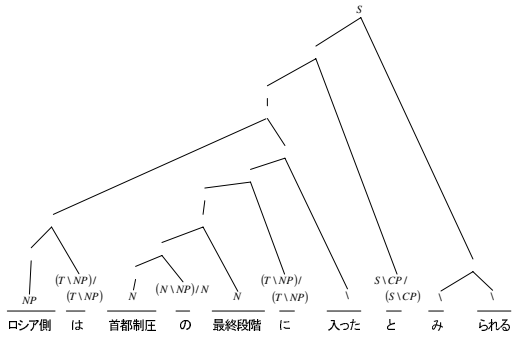

図 12: 単項規則を適用するノードの挿入及び、葉ノードへの語彙項目の割り当て

を組み合わせるときなどに、単項規則が適用される。

3.4 葉ノードへの統語範疇の割り当て

京大コーパスの品詞を CCG の統語範疇へマッピングすることで、活用しない単語 (名詞、指示詞、副詞、助詞など) に当たる葉ノードに、CCG の統語範疇を割り当てる。図 12 では、“入った”、“み”、“られる”以外の単語に統語範疇が割り当てられている。

3.5 親子間で適用される文法規則の決定

主要部後置言語 (head-final language) である日本語の特性を利用して、注目したノードが支配している部分木の右端の葉ノードの品詞を見ながら、親子間に適用する文法規則を決定する。文法規則を決定するための規則の一部を表 2 に示す。例えば、図 13 での“首都制圧の最終段階に”+“入った”に適用される文法規則は、順関数適用規則であるが、これは、“首都制圧の最終段階に”の句の主要部分が助詞の“に”であり、“入った”の句の主要部分が動詞だからである。

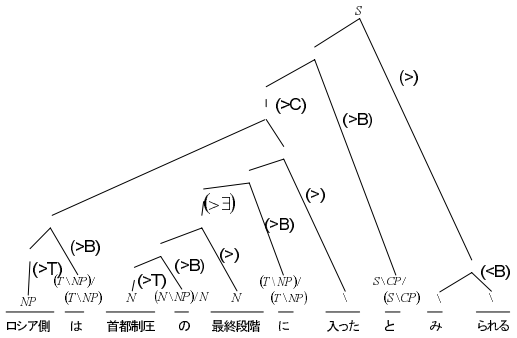

図 13: 文法規則の決定

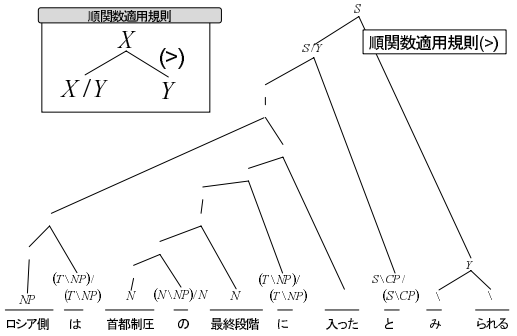

図 14: 文法規則をトップダウンに適用:順関数適用規則

助詞句＋動詞句	順関数適用規則
普通名詞＋形容詞性接尾辞	逆関数適用規則
名詞句+助詞	順関数合成規則
用言＋用言	逆関数合成規則
固有名詞＋用言	単項規則
固有名詞+助詞	

表 2: 文法規則を決定する規則の例

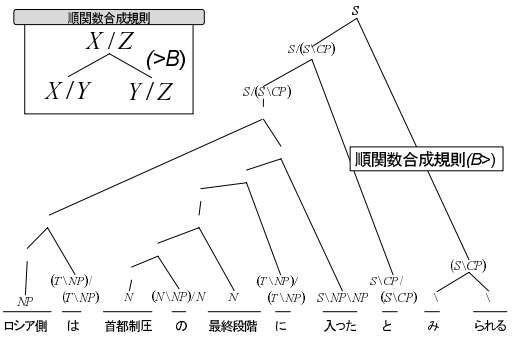

図 15: 文法規則をトップダウンに適用:順関数合成規則

3.6 文法規則をトップダウンに適用

これまでの変形で得られた木構造に、文法規則をトップダウンに適用する。図 14 は、根ノードに順関数適用規則を適用したところを示す。その結果、左の子の統語範疇が S/Y、右の子の統語範疇が Y であると計算される。図 15 では根ノードの左の子に順関数合成規則を適用したところを示す。その結果、根ノードの左の子、孫ともに $S/(S\backslash CP)$ の統語範疇を持つことが分かる。このように木の根からトップダウンに文法規則を適用すると、日本語 CCG に基づく完全な構文木が完成する (図 16)。ここで、“ み ”、“ られる ” の統語範疇に X が現れるが、動詞はほかの統語範疇と組み合わせられて、全体として文 S になるという制約を設けているため、“ み ”、“ られる ” ともに完全に統語範疇を決定できる。

3.7 語彙項目の獲得

最後に、CCG 構文木の葉ノードから語彙項目を獲得する。図 16 の構文木からは、“入った”、“み”、“られる”の統語範疇がそれぞれ $S\backslash NP\backslash NP$、$S\backslash CP$、$S\backslash S$ であることが得られる。

4 評価

本章では京大コーパスから語彙項目の獲得を試みた結果と、エラー解析について述べる。

4.1 実験結果

京大コーパスの最初の 30000 文から語彙項目の獲得を試みた結果 (表 3)、15352 文から文中の単語全ての語彙項目の獲得に成功し、274284 単語に対して、64789 種類の語彙項目が得られた。この結果には、語彙項目が現れた文の文脈情報や、単語固有の情報が全て入っている。その中の無視できるものを削除することにより、獲得できる語彙項目の種類数は減るものと考えられる。

4.2 エラー解析

最初の 100 文から語彙項目の獲得を試みた結果、60 文から語彙項目を獲得することに成功した。語彙項目獲得に失敗した 40 文について、その失敗の原因を分析した。その結果、失敗した原因は以下の様に分類された。

1. 実装上の問題のために語彙項目の獲得に失敗しているもの
   - 助詞が二つ重なる場合：14 文
     例 :“ 報道では ”

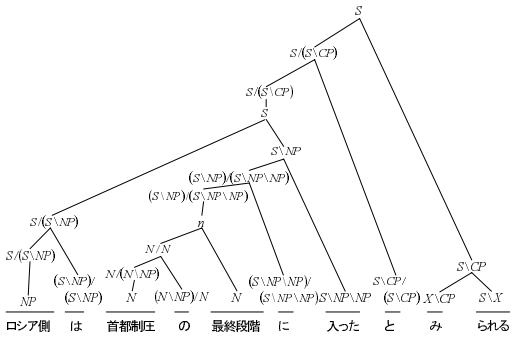

図 16: 文法規則をトップダウンに適用した結果

実験で使用した総文数	30000
CCG 構文木に変換した文	29999
語彙項目を獲得した文数	15352
語彙項目を獲得した単語数	274284
獲得した語彙項目の数	64789

表 3: 実験結果

- 連用中止形を含むもの：6 文

  例 :“ 微妙な時期なので、解散するべきではない ”
- その他：12 文

2. 文法理論で扱えないために語彙項目の獲得に失敗しているもの
   - 格助詞句が普通名詞に係る：8 種類

     例 :“ 戦後五十年を契機に ”

前者の原因による失敗は、今後実装を改善することによって解決することができると考えられる。したがって、将来的には、本手法により 90%以上の文から語彙項目を獲得できることが期待される。

5 まとめと今後の課題

本稿では、係り受け情報付きコーパスからの日本語 CCG の語彙項目を獲得する手法を提案した。京大コーパスを用いた実験の結果、30000 文中 15352 文から語彙項目を獲得することに成功した。今後の課題としては、失敗の原因を観察し、コーパス変換規則の実装を改良することで、実際に獲得できる語彙項目の数を増やすことが挙げられる。また、現在の日本語 CCG の理論では説明できない構文については、理論言語学の研究へのフィードバックを行い、理論のさらなる改良が必要である。

参考文献

[1] 黒橋禎夫, 長尾真. 京都大学テキストコーパス・プロジェクト. 言語処理学会第 3 回年次大会発表論文集, 1997.

[2] 戸次大介. 組合せ範疇文法 (CCG) による日本語の活用体系　東京大学, 2006.

[3] C. Pollard and I. Sag. *Head-Driven Phrase Structure Grammar*. The University Chicago Press, 1994.

[4] M. Steedman. *Surface Structure and Interpretation*. The MIT Press, 1996.

[5] Yves Schabes and Aravind Joshi. An earley-type parsing algorithm for Tree Adjoining Grammars. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics (ACL 1988)*, pp. 258–269, Buffal NY USA, 1988.

[6] Fei Xia. Extracting Tree Adjoining Grammars From Bracketed Corpora. In *5th NLPRS*, 1999.

[7] Julia Hockenmaier and Mark Steedman. Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. In *3rd LREC*, 2002.

[8] Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. Corpus-Oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the first International Joint Conference on Natural Language Processing(IJC-NLP 2004)*, Hainan Island, China, 2004.

[9] Kazuhiro Yoshida. Corpus-Oriented Development of Japanese HPSG Parsers. In *The 43rd ACL Student Research Workshop*, 2005.

[10] N. Komagata. A Computational Analysis of Information Structure Using Parallel Expository Texts in English and Japanese, 1999.

[11] 黒橋禎夫, 長尾真. 日本語形態素解析システム JUMAN version 3.61 京都大学, 1998.

[12] 黒橋禎夫, 長尾真. 日本語構文解析システム KNP version 2.0 京都大学, 1998.