パターンを使った構文解析

乾伸雄，小谷善行
東京農工大学工学部情報コミュニケーション工学科
{nobu, kotani}@cc.tuat.ac.jp

括弧付きコーパスで表現されている構文情報を用いた構文解析手法を提案する．括弧付きコーパスでは句構造文法における非終端記号が表現されていないため，従来の研究では主辞となる単語（または品詞）を用いて，非終端記号を生成する手法が取られていた．これに対して，本研究では句と句の境界に相当する句切り語を導入することで，Ｎグラムによる句の推定を行う．Ｎグラムの尤度はＮグラムの生起確率の積とすることで高精度が得られた．再現率に関しては日本語文に対し 78%，英語文に対し 86%であった．しかし，クロス率に関しては日本語文に対し 19%，英語文に関し 12%と不自然な解析が発生することがわかった．

Using Patterns for Parsing

Nobuo Inui, Yoshiyuki Kotani
Department of Computer, Information and Communication Sciences,
Tokyo University of Agriculture and Technology, 184-8588
{nobu, kotani}@cc.tuat.ac.jp

This paper proposes a parsing method using syntactically-parenthesized expressions in natural language corpora. Since non-terminal symbols of phrase structure grammars are not adopted in such expressions, using the automatically generated symbols estimated from thematic words (or their parts of speech) was necessary in previous researches. Instead of these non-terminal symbols, we estimate boundaries of clauses by N-gram information including segmentation words corresponding to the boundaries. Experimental results indicated higher performance in using the products of N-gram occurrence probabilities as likeliness. Our parser shows 78% and 86% in the recalls for Japanese and English, respectively. But the cross ratios (19% and 12% for Japanese and English, respectively) indicate that our method tends to generate unnatural results.

１．　はじめに

近年，大規模なコーパスが利用されることになったことから，コーパス情報から自動的に様々な自然言語処理システムを作成する研究が進められている．構文解析においては，確率文脈自由文法のような統計情報に基づく方法や事例からの構文解析のようなコーパス情報を直接扱う手法が開発されてきた．これらの手法はコーパスが豊富なほど効果的と考えられるが，ある一つの手法によって作られた構文コーパスはまだ規模が小さいのが現状である．複数のコーパスを混合して使う一つの問題としては句構造文法における非終端記号の種類に様々なバリエーションがあることが挙げられる．非終端記号を用いない構文解析手法ならばこの問題を避けることが可能である．本論文では括弧付きのコーパスから収集された情報を使った構文解析の手法について述べる．

２．　事例を用いた構文解析

言語コーパスから収集された情報を用いた構文解析は，頑健で高精度な手法が開発されている．もっとも良く使われているのは確率文脈自由文法[Chaniak 97, Sekine 95, Pereira 92]に基づく手法である．規則の生成には人手で作成した研究や自動的に規則を生成する研究が行われている．

LTAG[Chiang 00]は文脈自由文法によるあいまい性を解消するために，規則の一部が終端記号に結びついた規則を用いる．現在いくつかの言語において規則の整備が行われている段階である．SuperTAG はおのおのの規則に振られた記号であり，そのNグラムによって規則の適用順序を決定するために用いられる．

DOP[Bod 99]は LTAG と似ているが，LTAG の規則は人間から見て句構造を表すのに適した非終端記号が選ばれるのに対し，一つの規則からひとつひとつの非終端記号全てを含む規則を生成する．コーパスから規則を獲得することを前提に設計された文法である．

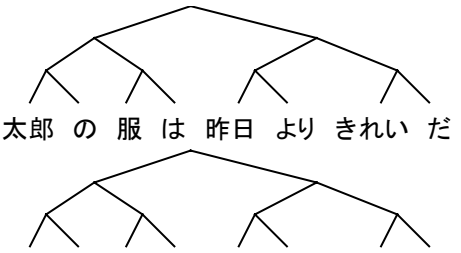
このようにコーパスを使った様々な文法が開発されてきた．基本的にはコーパスに依存する，特に非終端記号に依存するため，大規模なコーパスを利用できないという欠点が指摘できる．複数のコーパスの整合性をとる研究[Chan 99]も行われているが，これに対し，括弧によって句が表現される方法は，文脈自由文法，係り受け文法を通じて共通なものであり，構文情報を持つコーパスの共通的特徴として考えられる．

ただし，文脈自由文法の場合，非終端記号を用いることで，無限の可能性を持つ文を受理可能にしている．逆に言えば，非終端記号を用いない文法では様々な文を受理するために何らかの工夫が必要となる．本稿では第３章に述べる N グラムを使った定式化を行うことでこの問題を解決した．

N グラムモデルと非終端記号を融合したモデル[Wu 99]も提案されているが，非終端記号の多様性から充分なコーパスが利用できないのが問題である．本稿では非終端記号を用いない解析を試みる．

３．　パターンを使った構文解析

本研究は過去の提案[Inui 01-1, Inui 01-2]の発展型である．まず，パターンを使った構文解析の元となる文法について説明する．非終端記号として形態素を扱うが，実際の実現では品詞も使われている．括弧付き表現の文例として次をあげられる．



(((((太郎)(の))(服))(は))(((昨日)(より))
((きれい)(だ))))

図1　括弧付き表現の構文木

図１において，上の木は形態素によって表現したが，下の木は助詞以外を品詞によって表現している．このように一つの括弧付き表現から得られる木には様々なバリエーションがある．どのような情報を使って木を表現するかはコーパスの量に依存して決定されるが，形態素に近い情報を使うことで，詳細な表現を行うことができる．係り受け解析では形態素情報が主として使われる[Collins 96]が，全ての係り受け情報を収集するのは困難と考えられる．

このような構文木を生成するために，括弧付き表現から規則を収集する．構文解析では非終端記号（品詞や形態素）の列を木構造に変換するが，これは例えば次のような規則で表現することができる．ここで，「→」は書き換え規則であり，左辺から右辺に構造を変換することを表す．

例1　書き換え規則（形態素情報）

（太郎の）→((太郎)(の))

(服は)→((服)(は))
(太郎の服は)→((太郎の)(服は))
・・・

例2　書き換え規則(助詞だけ形態素)
(名詞の)→((名詞)(の))
(名詞は)→((名詞)(は))
(名詞の名詞は)→((名詞の)(名詞は))
・・・

例１，２に示した書き換え規則を用いることによって，図１のような括弧付き表現を生成することが可能である．この規則では文脈自由文法のような開始規則は存在せず適用可能な規則がなくなった時点で解析は終了する．文脈自由文法と異なり，曖昧性の少ない文法であるため,解析精度の向上が期待できるが,非終端記号を用いない表現であるため，受理可能な文には限りがある．例１の規則を用いた場合は，解析したい文そのものが規則に表現してあることが必要となる．例２の場合でも，多数の規則が必要となる．一般に短い形態素列ならば，例えば,「名詞が」のように頻出するものならば規則が適用可能となるが,長い形態素列の場合，困難となる．

一つの問題解消の方法に，複数の木を組み合わせる方法がある．これは TAG で用いられる derivation という同一な非終端記号を合わせることと等しい. 例えば, ((本)(を))と((赤い)(本))を組み合わせることで，(((赤い)(本))(を))という構造が生成できる. 終端記号による規則のモデル化では，品詞を用いた場合などかなり曖昧性を持った解釈が生成されるので，本論文ではこの方法は用いない．

これに対して，N グラムを用いた近似方法を説明する．例１，２に示した書き換え規則では，左辺の終端記号列を右辺の構造表現に変換するが, TAG と異なり２レベルの構造であるため，句切り記号を用いることで終端記号列から終端記号列への変換で表現することができる．

定義　句切り記号@
句切り記号@は句と句の間の境を表す終端記号である．順番を表すために$@_i$(i=1,2,・・・)と表現することができるが，順番が問題とならない場合，単に@と書く．

例3　句切り記号
(太郎の)→((太郎)(の)) ⇔ 太郎の→太郎@の
(服は)→((服)(は)) ⇔ 服は→服@は
(太郎の服は)→((太郎の)(服は))
⇔ 太郎の服は→太郎の@服は

例３は例１に対応したものであるが，例２に対応した句切り記号を使った書き換え規則も構築できる．句切り記号を含んだ規則において，右辺から左辺は一意的に決定することができるので，構文解析の規則は次のような終端記号列の集合で表現することができる．

例4　例3に対応した規則集合
{太郎@の，服@は，太郎の@服は，・・・}

この集合を N グラム集合に変換する. N グラム集合は，様々な長さを持つ終端記号列を固定長の集合に句切る操作である．上記の終端記号列はおのおの一つの句を構成しているので，句頭の記号+および句尾の記号*を考えることができる．例えば，バイグラムに変換すると，例３に対して次のバイグラム集合を得ることができる．

例5　例3に対応したバイグラム集合
{+太郎，@の，の*，+服，服@，@は，は*，+太郎，太郎の，の@，@服，服は，は*，・・・} ={+太郎(2)，@の(1)，の*(1)，+服(1)，服@(1)，@は(1)，は*(2)，太郎の(1)，の@(1)，@服(1)，服は(1)，・・・}

例５において括弧内の数字は頻度を表している. 一つの N グラムは一般に複数の句から発生している．例えば,「+太郎」は句の最初の形態素が「太郎」である句から得られる．このような N グラムを使うことで, ある句の生起確率を推定することができる[北 96].

$$(1)\quad P(w_1\cdots w_n)=P(w_1)P(w_2\mid w_1)\cdots P(w_n\mid w_1\cdots w_{n-1})$$
$$\approx P(w_1)P(w_2\mid w_1)\cdots P(w_n\mid w_{n-k+1}\cdots w_{n-1})$$

式(1)は長さ n の終端記号列の生起確率を長さ k の N グラムを用いて推定した式である．さらに，句頭に k-1 個の句頭を表す記号+，句尾に k-1 個の句尾を表す記号*を挿入する．

+および*だけからなる確率は共通であるから(1)式は式(2)のように書き直すことができる．

$$(2)\ P(w_1\cdots w_n) = P(+\cdots+w_1\cdots w_n *\cdots*)$$
$$\approx P(w_1 \mid +\cdots+)P(w_2 \mid +\cdots+w_1)\cdots$$
$$P(w_n \mid w_{n-k+1}\cdots w_{n-1})\cdots P(* \mid w_n *\cdots*)$$

ここで，条件付き確率は式(3)によって計算することができるので，式(2)から例５のような N グラム集合（および N-1 グラム集合）から推定することができる．

$$(3)\ P(w_j \mid w_{j-k+1}\cdots w_{j-1}) \approx \frac{freq(w_{j-k+1}\cdots w_j)}{freq(w_{j-k+1}\cdots w_{j-1})}$$

ただし，実験的には条件付き確率の積による推定よりも生起確率の積による推定の方が好成績が得られた．よって，式(2)，(3)の代わりに尤度 L を用いた式(4)，(5)を用いる．

$$(4)\ L(w_1\cdots w_n) = P(+\cdots+w_1)P(+\cdots+w_1w_2)\cdots$$
$$P(w_{n-k+1}\cdots w_n)\cdots P(w_n *\cdots*)$$

$$(5)\ P(w_{j-k+1}\cdots w_j) = \frac{freq(w_{j-k+1}\cdots w_j)}{\text{総頻度}}$$

本論文では，式(6)を用いて句切り位置の推定を行う．これは，次のように生起確率を最大にする句切り記号@の挿入位置を求めることに等しい．

$$(6)\ w = \underset{w=insert(w_1\cdots w_n,i)}{\arg\max} L(w)\quad i=1,\cdots,n-1$$

式(6)において，$insert(w_1\cdots w_n,i)$ は句切り記号を含まない終端記号列に i 個の句切り記号@を挿入する関数である．例えば，次のような例が挙げられる．ただし，連続した句切り記号は意味がないので考慮しない．

例6　挿入関数 insert

{insert(**太郎の服は**,2)} = {**太郎@の@服は**,
太郎@の服@は,**太郎の@服@は**}

確率文脈自由文法の場合は，確率を最大にするような構文木を算出する．このため，非決定的な解析を行うことになる．非終端記号を用いた場合，一つの終端記号列に対して多数の構文木が可能となるため，計算時間がかかる．これに対して，本論文で用いた終端記号列による書き換え規則は重複する木が，品詞だけを使って記述した場合でも，全体の5%程度に過ぎず, 曖昧性の低い文法が得られている．これは，句ごとに最適な分割を行っても解析精度に大きな影響がないこと意味している．このため，本論文では，式(6)を用いて，決定的に構文木を生成する方法を採用した．また，決定的構文解析で充分な精度が得られることも論じられている[Wong 99, Tugwell 00].

実際に尤度を計算する場合，ロバスト性を保つことが必要である．つまり，尤度が０になるのを防ぐ．これは，線形補間式，つまり k 長の N グラムを求めるのにユニグラムから k グラムまでの尤度の重み付き和で計算する．実際には，k グラムから順に 0.999 の重みをかけることでこのディスカウントを行った．

４．　解析アルゴリズム

３章で述べた終端記号を用いた文法を用いた解析アルゴリズムについて述べる．アルゴリズムは図２に示す．

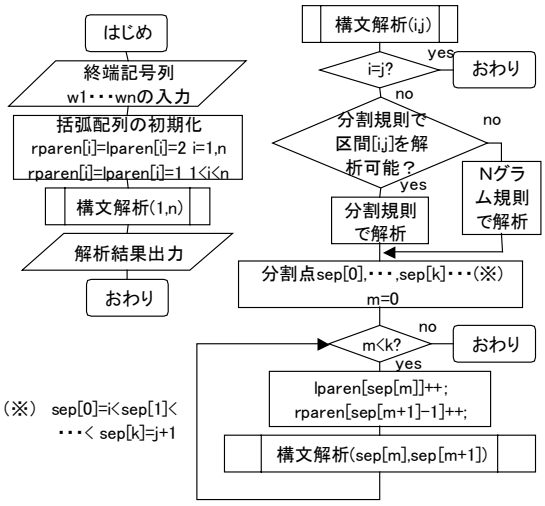


図2　構文解析のアルゴリズム

構文解析の出力となるのは括弧付き表現の構造である．アルゴリズム中で，rparen[i], lparen[i]はそれぞれ i 番目の終端記号（形態素または品詞）のそれぞれ右および左につく括弧数を示す. sep[j]は句として抽出された終端記号の位置を示す．３章で述べたように構文解析はトップダウンに決定的に実行される

ため，高速である．

実際の実現では終端記号列に挿入する句切り記号の数は最大で２個，つまり最大でも一つの句は三つまでにしか分割されないようにした．これは，全ての分割方法を試すのには時間がかかることと，尤度 L の性質から分割数を増やしたものは選ばれにくくなるからである．そのため，生成された構文木は結果として２分部分木（および少数の３分部分木）から構成されることになる．

５．　実験

実験は　EDR コーパス[EDR 96]の日本語コーパスおよび英語コーパスを用いて行った．コーパスのデータを表 1，2 に示す．実際に終端記号として用いたのは次の３種類である．

表1 EDRコーパスのデータ

	学習用(文)	評価用(文)	平均単語数(単語／文)
日本語コーパス	197411	10391	24.6
英語コーパス	119523	6291	26.4

表2 EDRコーパスで使われている品詞

	品詞の種類
日本語コーパス	名詞 助詞 動詞 語尾 記号 数字 助動詞 接続詞 形容詞 感動詞 副詞 連体詞 形容動詞 接尾語 接頭語
英語コーパス	s NUM BLNK NOUN VT SUF ART ADJ PUNC ITJ DEMO VI CONJ ADV AUX PRON INDEF PREP BE UNIT WH SYM PTCL PF

（１） 品詞

（２） 助詞(PREP)だけ形態素

（３） 助詞(PREP)，記号(SYM)だけ形態素

評価は次の二つの基準で行った[Black 91]．クロス率は解析の自然さ，再現率は解析の正しさを表す指標となる．図４～７に解析精度をグラフで表す．

$$\text{クロス率} = \frac{|D|}{|C|} \qquad \text{再現率} = \frac{|C \cap T|}{|C|}$$

C: コーパス中の句の集合

T: 解析木中の句の集合

D: コーパス中の解析木と交差する句の集合

英語，日本語ともに形態素に近い情報を用いた方がクロス率，再現率ともに良い結果が得られる．日本語の場合，助詞に形態素を導入することによって得られる精度の向上は顕著である．本システムは終端記号の取り方がどの程度構文に影響を与えるかを示すことも可能である．英語については，どれもほとんど変わらないため，細かい曖昧性の解消以外には形態素情報は効果がないと推察される．

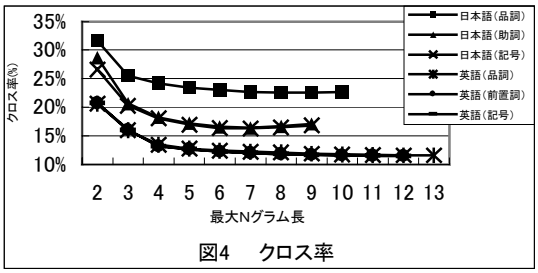


図4　クロス率

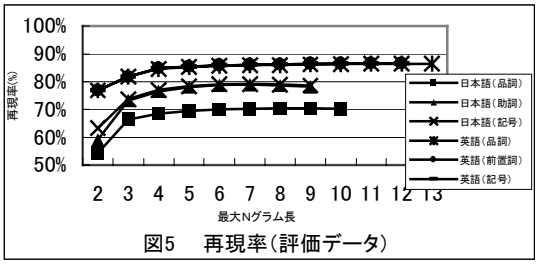


図5　再現率(評価データ)

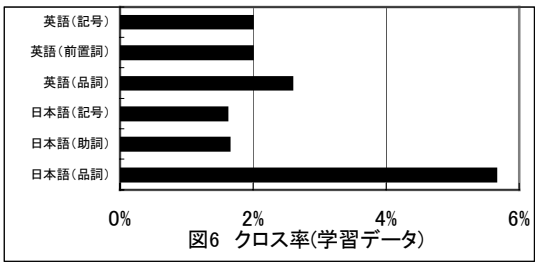


図6　クロス率(学習データ)

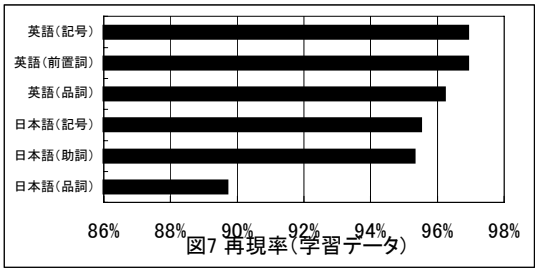


図7 再現率(学習データ)

Ｎグラム長を長くすることでも精度の向上が期待される．図４，５の結果からは，長さ５程度でほとんど収束傾向にあることが読める．これはコーパスのサイズとの関係が強い．

図６，７に示された学習データとの比較では評価データの精度は低い．これは，Ｎグラムを用いたことによる低下である．より高い精度を保つために，Ｎグラムによる推定とともに木の接合による推定も取り入れることが

必要と思われる．

６．　考察

従来，コーパスを用いた構文解析では日本語において 70%以上[Shirai 97]，英語では 90%前後[Chiang 00]が報告されていることが多い．これに対して，本手法は全く人為的なヒューリスティックを取り入れてないにも関わらず，最大で再現率 78%(日本語)，86%(英語)を得，日本語では同等，英語でも若干の精度の低下が見られているに過ぎない．本手法の特徴は高ロバスト性，高速な計算にあるため，従来手法と何らかの形で併用することにより，自然言語アプリケーションの開発に役立つと考えられる．ただし，再現率の高さに比べ，再現率が最大の場合，19%(日本語)，12%(英語)というクロス率の高さの改良が必要である．

７．　おわりに

本論文では，非終端記号を用いない句のパターンによる構文解析について述べた．

謝辞 本研究の一部は学術振興会科学研究費補助金(12780266)の支援を受けて行われた．

参考文献

[Chaniak 97] E. Charniak: Statistical Parsing with a Context-free Grammar and Word Statistics, Proc. AAAI 97, pp.598-603, 1997

[Chiang 00] D.Chiang: Statistical parsing with an automatically-extracted tree adjoining grammar, ACL 2000, pages 456-463, 2000

[Bod 99] R. Bod and R. Kaplan: A Probabilistic Corpus-Driven Model for Lexical Functional Analysis. COLING-ACL-98, 1998

[Sekine 95] S. Sekine and R. Grishman: A Corpus-based Probabilistic Grammar with Only Two Non-terminals, Fourth International Workshop on Parsing Technology, 1995

[Tugwell 00] D. Tugwell: Towards a Dynamic Syntax for Language Modeling, 3rd Int. Workshop on Test, Speech, Dialogue, pp.33-38, 2000

[Wu 99] J. Wu and S. Khudanpur: Combining Nonlocal, Syntactic and N-Gram Dependencies in Language Modeling, Eurospeech'99, vol 5, pp2179-2182, 1999

[Chan 99] D.K. Chan and D.Wu: Predicting Unlikely Part-of-Speech Categories, 5th NLPRS, pp. 38-43, 1999

[EDR 96]EDR: EDR Electric Dictionary Manual Ver. 1.5, 1996

[Collins 96] M. Collins.: A New Statistical Parser Based on Bigram Lexical Dependency, 34th ACL, pp.184-191, 1996

[Inui 01-01] N. Inui,, T. Kotani: Robust N-gram Based Syntactic Analysis Using Segmentation Words, 15th PACLIC, pp.333-343, 2001

[Pereira 92] F. Pereira, Y. Schabes: Inside-Outside Reestimation from Partially Bracketed Corpora, 30th ACL, pp.128-135, 1992

[Wong 99] A. Wong and D. Wu: Are Phrase Structured Grammars Useful in Statisticl Parsing?, 5th NLPRS, pp. 120-125, 1999

[Black 91] E. Black, etc.:.A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars, Fourth DARPA Speech and Natural Language Workshop. 1991

[北 96] 北研二，中村哲，永田昌明:音声言語処理，森北出版, 1996

[Shirai 97] Shirai, K., Tokunaga, T. and Tanaka H.: Automatic Extraction of Japanese Probabilistic Context Free Grammar From a Bracketed Corpus, Journal of Natural Language Processing, 4(1): pp.125-146, 1997 (In Japanese)

[Inui 01-02] 乾伸雄，小谷善行：品詞列に基づく構文解析，情報処理学会自然言語処理研究会，01-NL-144, 2001