自然言語処理 113－11
（1996．5．29）

自然言語の並列構造解析へのスキッピングパーザの応用

日下部 俊昭　田中 穂積　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

自然言語処理における困難な問題の一つに，文中に含まれる並列構造の推定がある．比較的長い日本語文には並列構造が含まれていることが普通であり，並列構造の推定は大きな問題になる．並列構造は表層的に類似しているのが普通である．本研究ではこれらを基に並列構造を推定するシステムについて述べる．まず，CFG を使用して文に含まれる句間の広域な接続制約を生かしながら GLR Skip 法を用いて並列構造を推定する一般的な方法を示す．しかし，日本語の場合には文法的な制約が弱くこの一般的枠組はそのまま適用できない．従って言語に固有なヒューリスティックスを導入することで GLR Skip 法の一部を用いて効率良く文節間の意味的類似度を計算し，高精度で高速に統語解析と並列構造推定を行なうことが可能な方法を提案する．日本語文 153 文に対して後者の方法で実験を行なった結果，73.3%の精度で範囲推定を行なえることが分かった．

Application of Skipping-parsers to the Analysis of Coordinate Structures in Natural Languages

KUSAKABE Toshiaki　TANAKA Hozumi　TOKUNAGA Takenobu
Department of Computer Science
Tokyo Institute of Technology

This paper describes the use of a GLR Skipping-parser in the analysis of coordinate structures in natural languages. Coordinate structures typically contain constituents with paralleled surface structures. We pay particular attention to this parallelism, and introduce the notion of 'Similarity of Bunsetsu'(where 'bunsetsu' is defined as the smallest phrase in Japanese).
First, we describe a method of applying a GLR Skipping-parser to find coordinate structures, using a CFG to model global constraints between constituents. Next, we detail a new method of combining a GLR Skipping-parser and Bunsetsu grammar to quickly and efficiently estimate coordinate structure and determine syntactic structure. This method is particularly directed towards languages such as Japanese which contain almost no degree of phrase-level syntactic constraint. We used this latter method to carry out an experiment on the coordinate structures of 153 sentences in the EDR Japanese corpus, and calculated a precision of 73.3%.

1 はじめに

日本語では 80 文字以上の長文の解析は困難であるとされている．その原因は「文が長くなるにつれて 1 つの句の係り先の候補が組合せ的に増えるからである」と考えられる．文を長くする主な原因は，1 文中に多くの内容が並列的に述べられているところにある．従って，この並列構造を正しく認識できれば，長い文を短く区切って解析することができ，文の解析精度の向上が見込まれる．

本研究は文の並列構造の範囲を推定することで解析効率の向上を目標とする．一般に，並列する構造同士は何らかの特徴や性質において類似していると考えられる．例えば次の文では下線を引いた節が並列である．

I played tennis
and drank a glass of water.

上記した文は，次の 2 文に分解される．

- I *played tennis (and)*
- I *drank a glass of water.*

隣接する品詞の接続制約を記述するものに文脈自由文法 (CFG) がある．この文法を扱うための優れた統語解析法として GLR 法がある．しかし，通常の GLR 法では並列構造の有無により上記した 2 文に分解して解析することは不可能であるし，CFG による並列構造の記述にも限界がある．そこで本研究では，GLR 法を拡張した GLR Skip 法というスキッピングパーザを用いて文の構成要素を任意に飛ばし，文全体の解析の他に，文に含まれる並列構造の抽出を行うパーザを提案する．それによれば，スタックトップの状態とそこに割り振られる動作から並列する 2 文を抽出することが可能であることを示す．

日本語の並列構造を抽出する場合には，後置詞句はほとんど全ての句に対して接続可能性を持っている．文法的な制約を用いただけでは制約が緩過ぎるためスキッピングパーザの解析の効果は望めなくなる．従って言語に固有なヒューリスティックスを導入することで GLR Skip 法の一部を用いて効率良く文節間の意味的類似度を計算し，高精度で高速に統語解析と並列構造推定を行なうことが可能な方法を提案する．統語解析では，スキッピングパーザの履歴領域を文節のスタック領域として用いる．次に文節間の類似度を算出し，効率良く並列構造の範囲を決定する．

日本語に対する本手法の有効性を評価するため，EDR コーパス [7] を用いて実験を行ない，本手法の有効性を明らかにするとともに，問題点や今後の課題について検討する．

2 並列構造の推定法

並列構造の推定に関しては様々な手法が提案されているが，そのほとんどが表層レベルの類似性を利用している．これは並列する構造同士は何らかの表層レベルの特徴や性質が類似しているからである．そこで本研究においても表層レベルの類似性を用いる．

2.1節では，GLR Skip 法について説明し，2.2節では，GLR Skip 法を用いて並列構造の前部と後部を推定する一般的な枠組について述べる．

2.3節では，2.2節で述べた一般的な枠組の一部を利用した，日本語に特化した並列構造の推定方法を提案する．

2.1 GLR Skip 法

GLR Skip 法は，文中に含まれるノイズをスキップし頑健な解析を行なうために GLR 法を拡張したものである [2]．GLR Skip 法の処理の流れを図 1をもとに説明する．GLR Skip 法では，実処理領域と履歴領域を用いる．実処理領域は，現在の先読み語で shift が行なえる領域で，GLR 法のグラフ構造化スタック (GSS) と同様のスタック領域である．履歴領域は，実処理領域で shift を行なった直後のスタックを記録しておく領域である．処理手順は以下の通りである．

1. 履歴領域の全てのスタックを実処理領域にコピーする．
2. 実処理領域で先読み語 (n10) に対して，reduce 可能なものを全て reduce する．(図 1では<n>が<n_>に reduce されているが，n71 に対しては reduce が失敗している．)
3. 先読み語 (n10) を shift する．(図 1では<n_>に n10 が shift されている．shift に失敗した他のスタックは実処理領域から取り除かれる．)
4. shift が終了直後のスタックを全て履歴領域にコピーする．

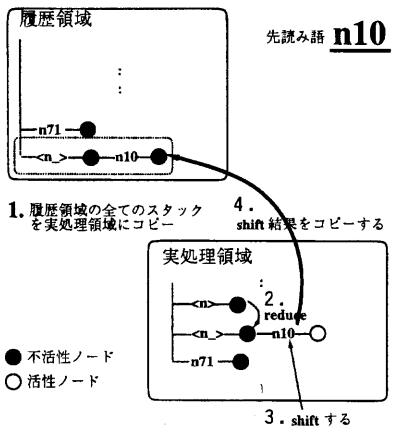

図 1: GLR Skip 法の動作

このようにすることで，実処理領域において現在の先読み語で動作が行なえない場合でも，過去のスタックを履歴領域からコピーし，処理を続けることができる．それにより，結果的に過去のスタックに shift された単語と現在の先読み語との間に存在した単語列がスキップされる．なお，履歴領域の全てのスタックトッ

プに動作が行なえない場合には，先読み語が文末記号でない限りは先読み語を無条件にスキップする．[1]

2.2 並列構造を推定する一般的な手法

GLR Skip パーザをうまく利用することにより，文全体の解析ではなく文法に適合した全てのサブセット文を解析結果として出力できる．並列構造とは文法的，機能的に似た句や節が繰り返し述べられる構造を指すので，部分解析結果には並列構造の組が含まれている．スキップするタイミングは並列構造を示す表層レベルのマーカ (例えば and や or，コンマなど) がスタックトップに現れた時である．並列構造の組を見つけるためにスタック間の類似度を算出し，最も類似しているスタック同士を並列構造とする．以下で具体例を示し，並列構造の推定方法を説明する．表 1の文法を基に，GLR

表 1: 単純な英語の文法

1)	S	→	S_1 and S_1.
2)	S_1	→	NP,VP.
3)	S_1	→	VP.
4)	NP	→	I.
5)	NP	→	tennis.
6)	NP	→	a,glass.
7)	NP	→	NP,PP.
8)	VP	→	played,NP.
9)	VP	→	drank,NP.
10)	PP	→	of,NP.

例文: I played tennis and drank a glass of water.

Skip 法を用いた解析を行なう．

表 2: 表 1 の文法による LR(1) 表の一部

	···	drank	···	NP	VP
0	···	sh2	···	4	9
		:			
3	···	re4	···		
		:			
5	···		···		16
		:			
14	···		···		
		:			
17	···	sh18	···	19	22
		:			

図 2は，GLR Skip 法を用いた並列構造の抽出過程の一部であり，“and” を shift した直後の履歴領域を示している．図中のスタックトップノードの状態と表 2から，次にくる “drank” は図の 3 つのスタックにしか繋がらない．従って履歴領域には “drank”(状態 0 において “drank” を shift) と “NP drank”(状態 3 で先読み語 drank に対して表 2の re4 を行ない NP を push してから drank を shift)，および “S_1 and drank”(状態 17 で先読み語 drank を shift) のスタックができる．“S_1 and drank” のスタックは文全体の解析を行なうスタックであるので，“drank” と “NP drank” のスタックだけが並列構造の候補として残ることになる．そしてこれらを実処理領域に移し同様の解析を最後まで続けると，“NP played tennis”，“NP drank a glass of water”，“drank a glass of water” という 3 つのスタックが生成される．この内，最初のスタックのトップは並列構造を示すマーカの直前の語 (tennis) になっているので，並列構造の一部をなすことが分かる．残りの 2 つのスタックから，より適切な候補を選択するには，2.4節で説明する類似度によるスコアリングの手法を併用する．このように文法を使って状態の違いにより繋がる語を絞り，並列なスタックの候補を効率良く計算することができる．

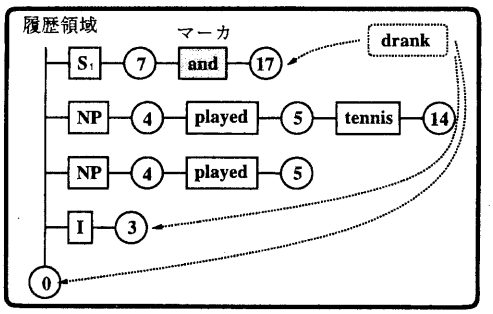

図 2: GLR Skip パーザを用いた例文の解析

2.3 日本語の並列構造解析

2.2節で説明した並列構造推定の一般的な枠組みでは，文法の曖昧性が増えたときや文法の制約が緩い日本語などの言語を扱う場合に計算量が指数的に増大するという問題がある．そのために日本語固有のヒューリスティックスを導入する必要がある．この節では，文節文法を用いて文法の曖昧性を出来るだけ排除し，また GLR Skip 法の履歴領域を利用して並列構造を推定する手法について述べる．

形態素・統語解析に用いる文法は，我々の研究室で研究されている EDR 日本語単語辞書を用いた大規模日本語文節文法 [4] を用いる．本研究では，単語単位に分かち書きをしたものを入力とするが，単語の品詞を一意に決定せず，曖昧性を持たせておく．現在の実装では，形態素・統語解析システム mslr [3, 4] が正解とした分かち書きを入力にしている．解析に用いる LR 表としては，細品詞[2]間の局所的な接続関係を織り込んだ修正 LR 表 [1] を用いる．GLR Skip パーザは，実処理領域で文全体の解析の他に 1 文節だけの解析も行なうことにする．そして履歴領域には 1 文節からなる解析が完了したスタックのみを記録し，そこからのいかなる動作も許さないことにする (図 3)．これにより，2.2で述べたような文法による制約は得られないものの，履

[1] Lavie らの GLR* 法と GLR Skip との違いは 2 つある．第 1 は履歴領域に相当する全ての部分に対して動作が行なえない場合には GLR* では以後の解析が不能になる．第 2 に GLR* では，過去の reduce の結果が履歴として生きており，その履歴に対して現在の先読み語でさらに動作を行なってしまう．これは reduce 直後の動作では先読み語は同一でなくてはならないという原則に違反する．我々の手法ではこのようなことがない．

[2] 具体的な品詞分類と，語と語の繋がりも有しているものを指す．例えば ev16 という細品詞は，1 段活用語尾で，1 段動詞語幹と繋がるという情報を有す細品詞である．

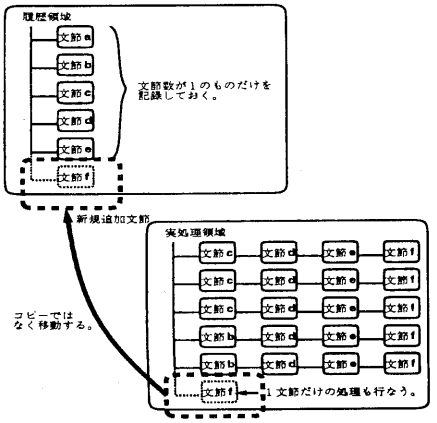

図 3: 文節文法の修正 GLR Skip 法による解析

歴領域の解析結果は文全体の解析結果よりも単純なものになり，解析コストが大幅に改善される．また，次節で説明するように，統語解析の途中で並列構造に関係するキー文節 (またはキーと呼ぶこともある) を検出することにより，その前後で文節 (つまりスタック) 間の類似度も計算することができる．ここで辞書引きや文法により生じる文節の品詞の曖昧性はそのままにしておく．これは，早期に曖昧性解消を行なうことで並列構造解析に有効な情報まで失われることを防ぐためである．

2.4 類似度の計算法と範囲の推定法

この節では，前節の文節の統語解析により得られた履歴領域のスタックの間の類似度を計算し，並列構造の範囲を推定する手法について述べる．

キー文節の検出・文節間の類似度の計算 この部分では，キー文節となる文節を推定し，その前後で類似度計算を行なう．キー文節とは，並列構造の存在を示す重要な文節である．文全体の解析が終ってからキー文節の推定や類似度の計算を行なう手法も考えられるが，履歴領域の文節を利用すれば以下のことが行なえので，統語解析と並行して行なうことにする．

- GLR Skip 法を用いているため，先読み語が1つでも，「…も…も」などの呼応による並列以外のキー文節の推定ができる (図 4)．
- キー文節が分かれば，類似度を同時に計算できる．

キー文節の検出 本研究で用いた日本語文法の自立語の品詞分類は，「名詞，動詞，形容詞，形容動詞，副詞，感動詞，接続詞，その他の修飾語」である．ここでは付属語を残りの語尾や助詞，助動詞と定義する．

日本語の並列構造を含む文には，それを示す印 (表層レベルのマーカ) が存在する．本研究では並列構造を以下の2つとし，並列構造の印はその分類別に並列のキー文節として，以下のように決める．

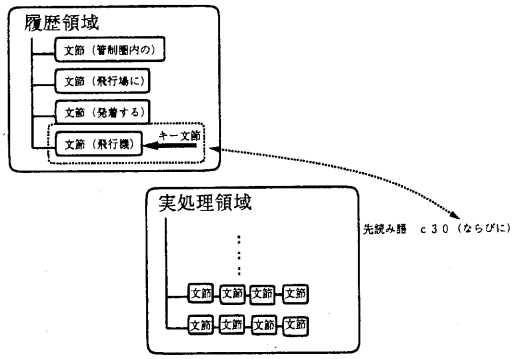

図 4: 統語解析中の履歴領域でのキー推定

1. 名詞並列
   その文節の自立語が名詞で以下が成立つとき：
   - 付属語に次の語を持つ場合；「と，～も，や，かつ，か，とか，なり，やら，だの，だけで (は) なく，／，・」但し，助動詞を付属語に持つとき (e.g. 「僕だと」) はキーとしない．ここで「～」は 0 個以上の付属語の連続を表す (以下同様)．
   - 文節の先読み語に読点または以下に示す接続詞がある場合；「および，または，ならびに，あるいは，もしくは，ないし，そして，そうして」

   但し，確実に名詞並列でないと分かるものは排除する．「時相名詞 (e.g. 今後，通常)+読点」などや「…と」の文節の先読み語が「いう」「思う」「する」などの引用助詞を伴う動詞語幹である場合は，キーとしない．ここで「+」は左右の語が連続することを表す．
2. 述語並列
   自立語が用言であるか，名詞+助動詞の文節であり，以下が成立つとき：
   - 文節の活用形 (文節に含まれる活用する語の活用形) が連用形であり，先読み語が読点である場合；但し，文節の活用形が連用形でも助詞を持つ場合は，「て～」または「で」でない限りキーとしない．
   - 付属語に以下の語を持つ場合；「の+に対して，とか，か，し，ところで，が，ず+に，だけで (は) なく，けれど (も)，なり，やら，だの，たり，～も」．但し，助詞「の」を助詞「が」の前に持つ場合，格を示すと考えられるので，並列のキーとしない．
   - 文節の先読み語に，名詞並列のところと同様の接続詞がある場合．

   読点を伴わない連用中止は，全て次の述語に係ると推定できるので，並列のキー文節とはしない．

文節間の類似度の計算 統語解析の途中でキー文節が発見できれば，それ以後の文節とキー文節以前の文節

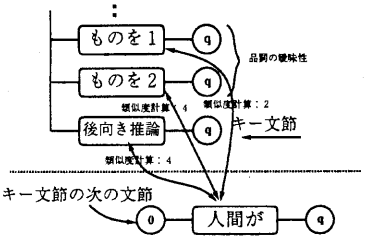

図 5: キーの前後での類似度計算

との間で類似度を計算する．図5は，GLR Skip 法を用いた統語解析途中の履歴領域である．「後向き推論」という文節は直後に読点が来たために名詞並列のキー文節であると推定されている．従って，図のようにその次の文節「人間が」とキー文節の前の文節群との間の類似度だけを計算し，キー文節以前の文節間の類似度は計算しない．

類似度の計算方法は以下の通りである．

1. 文節の品詞 (自立語もしくは語尾による品詞) が一致した場合に類似度 2 を加える．
2. 文節の品詞が名詞のとき自立語の細品詞が一致した場合に類似度 2 を加える．
   但し，自立語が複合語である場合は，接尾辞，単位，そしてその 1 つ前の名詞のいずれかが一致すれば類似度 2 を持つとした．さらに，
   (a) 自立語が同じ単語である場合には，類似度 10 を加える．
   但し，自立語が複合語の場合は，その主辞が一番最後にあると仮定し，その単語が一致した場合とする．
   (b) 自立語が同じ単語でなかった場合には，
      i. 形態素の部分一致 (先頭から，もしくは最後方からの一致のみを見る) があれば一文字ごとに類似度 2 を加える (最大 10 で打ち切る)．但し，複合語は一つの単語として見る．
      ii. 自立語の意味的な近さを調べるために，自立語の単語について分類語彙表のコードを調べる．それぞれの上位 3 桁目からの一致を調べ，一致する桁ごとに類似度 2 を加える (最大 10 で打ち切る)．両者の語義が複数ある場合には，一番近い語義で比較する．但し，自立語が複合語の場合には，その主辞が一番最後にあると仮定し，その単語で分類語彙表を引く．その単語が同一なら分類語彙表による類似度は考えない．
3. 同一の助詞がある場合には，助詞の一致 (辞書引きの規則が一致しているもの) 一組について類似度 3 を加える．助詞の性質上，文節の最後からの一致を見る．
4. 文節の品詞が動詞で自立語がサ変名詞でサ変語尾があるとき類似度 3 を加える．

また，文節の品詞が一致しなくても，述語並列のキーである場合は

- 「…が強く，…が可能な機械を…」

という場合があり得る．

従って，述語並列のキーと，述語となり得る文節 (自立語が用言，もしくは名詞+ 助動詞) の間に類似度 2 を認める．

キー文節に対応する後部の文節の検出 統語解析を終えた段階で，全てのキー検出と並列構造の範囲推定に関係する文節間の類似度計算も終了する．

ここでまず最初に行なう処理は，呼応する助詞による並列のキーの認定処理である．呼応する助詞は，「もなりだのやらとか」などで，これらが 2 つ以上現れた場合には，最後の文節だけキーとならない．これは統語解析時に，それらの助詞の出現回数をカウントしておくことで容易に検出可能である．

次に行なう処理は，見つかったキー文節に対応する並列構造の後部の文節の検出である．キー文節に並列すると思われるこの文節を後部の終点と呼ぶ．もちろん一つのキーについて複数の後部が見つかることがあるが，これは 2.4節の処理で適切なものを選択することにする．

後部の終点 (単に終点と呼ぶ) を検出するには，次のようにキー文節毎に検出方法を変える必要がある．

名詞並列の検出 名詞並列の後部を検出するための，ヒューリスティックス：

1. 名詞を自立語に持つ文節が並列することができるのは，次の “名詞並列以外の読点” までであると仮定し，この制約の中で，助詞の一致によるポイントを除く類似度で，最も類似度の高い文節を後部とする．
   付属語の一致による類似度は，次節で行なう処理で有効となる部分であり，キー文節の助詞が (格ではなく) 並列を表すものである場合が多いと考え，後部を見つけるときには無視することにした[3]．
   以上の制約で後部が見つからない場合には，2 を行なう．
2. 自立語の細品詞が形式名詞以外の名詞並列は，最初に格要素と読点の両方が後方に出現した時，それより後ろの名詞とは並列しない．

但し，次のものは特別に優先して検出する．

- 自立語の細品詞が形式名詞ならば，同じ形態素の形式名詞にしか並列しない (e.g. …ことや…こと)．またそれ以外の名詞並列のキー文節が，形式名詞と並列することもない．
- 「名詞+読点」のキー文節は，その直後の文節が名詞並列のキーであるならば，それと並列する．
- 「…も…も」のような助詞の呼応による並列があれば，それに従って終点を決める．

[3]例えば，「僕や彼とは考えが違う者やそうでない者は…」の場合に「僕や」と「者や」が並列するとは考えにくいので助詞の一致によるポイントは考えない．

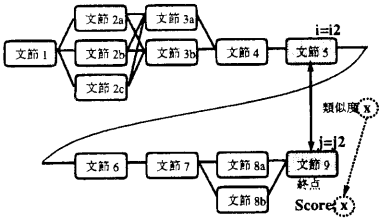

図 6: 前部推定の step1

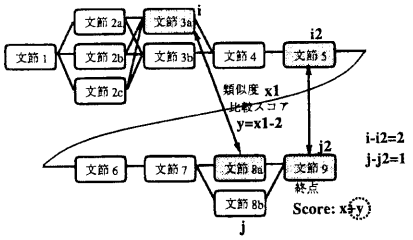

図 7: 前部推定の step3

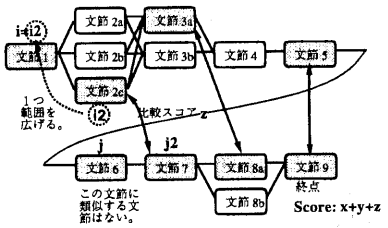

図 8: 前部推定の step6

述語並列の検出 述語並列の検出は次の制約 (ヒューリスティクス) を用いる．但し，ここでも助詞の呼応による並列は優先して検出する．

- 名詞述語や用言の連用形は，直後の名詞述語もしくは用言の連用形には並列しない．但し，直後の連用形が読点を伴っている，つまり，述語並列のキー文節である場合には，その限りでない．

この制約を考慮した上で，キー文節と最も類似度の高い文節を終点とする．この場合も，付属語の一致による類似度は無視する．

並列構造の前部推定 並列のキーと，それと並列するであろう終点が決まったなら，並列構造の後部から，類似度が大きく，バランスのとれた並列構造の前部を推定する．

前部推定のアルゴリズム 終点となる文節毎に以下の処理を行なう．(step1～step7)

始めに，現在類似度比較を行なっているそれぞれ並列構造の前部の文節と後部の文節を示すために，次のインデックスを定義する．

- i…キー文節から現在比較の対象となっている文節までの (前方への) 距離．
- i_2…前回の i の値．初期値は 0.
- j…終点から現在比較の対象となっている文節までの (前方への) 距離．
- j_2…前回の j の値．初期値は 0.

図 6,7,8では，品詞の曖昧性のある文節を縦に並べ，横のつながりは純粋に文のつながりとした (添字のアルファベットが曖昧性)．文節 5 はキー文節，文節 9 は終点である．

step1 まず，終点とキー文節の類似度を現在のスコアとする (図 6)．

step2 jを 1 つ前の文節に動かす．そこがキー文節なら step7 へ．

step3 jに位置する文節と比較スコアが最も高い文節を前部の文節 $i(\geq i_2)$ から見つける (図 7)．文節 i と jの類似度がない ($=0$) の場合は，その文節 i を比較対象からはずし，i を 1 つ前に進める．

比較スコアは，文節数が前部と後部で同じ方が良いという観点から，

- $i = i_2$ならば，$|(i - i_2) - (j - j_2)| \times (-2)$
- $i \neq i_2$ならば，(その文節間の類似度) $+ |(i - i_2) - (j - j_2)| \times (-2)$

とする．このスコアは，1 つの文節数のずれにつき 2 点のペナルティを課しているとみなせる．

但し，jの文節が，どの i の文節とも 0 より大きい類似度を持たないならば step2 に戻る．

step4 比較スコアが最大のものを現在のスコアに加える．j_2に jを代入する．

step5 step3 で見つかったポインタを i_2とする．そこが文の先頭なら step6 へ．さもなくば step2 に戻る．

step6 jがキー文節に達していないならば，jからキー文節までの間にある文節数だけ前部を伸ばす (図 8)．

step7 前部範囲推定を終了する．

ここで，step1～step7 に割り込み条件を導入する．つまり，

- i の値は，名詞並列と引用助詞の前の読点以外の読点よりも前方の文節の位置以下である．
- i_2の値が，名詞並列と引用助詞の前の読点以外の読点の次の文節に指す場合，もしくは j_2の値がキー文節を示す接続詞の文節の位置を指す場合は，そこで推定処理を終了する．

という条件を上の step の任意の地点で有効とし，step6 へ進む．これは，名詞並列を示す読点や引用助詞の前にある読点以外は文を区切る意味合いが強く，並列構造もそれ以前には存在しないという経験則に基づいている[4]．

[4]例えば,「僕と君，そして彼だ.」の名詞並列の読点や「無い物ねだり，ということだし，見苦しいことだと思う.」の下線部の引用助詞の直前の読点は区切りの意味合いを持たないと考える．

以上の手続きにより，終点1つにつき1つの並列構造が決定できる．次に問題になるのは，「どの終点が正しいか」という点であろう．これは，step3で計算した「スコア」の大きさで決定する．計算の過程から分かるように，前後で似た構造をバランス良く持つペアが最も高いスコアを持つので，最大スコアを持つペアが正しい並列構造であるとみなす．

3 実験

3.1 日本語の並列構造解析の評価実験

括弧付きEDRコーパス約20万文からランダムに抜き出した23単語以上の文(平均30.4単語，平均文字数45.6文字)の内，156文を用いて2.3節による評価実験を行なった．この156文に対して単語にタグ付けされた細品詞は2313個で，文法は我々の研究室で開発した日本語文法を多少変更し利用した．この文法のサイズは規則数904，連接属性を導入したSLR(1)表の状態数は1042である．

コーパスには，予めタグや木構造などが付されているが，EDR単語辞書の登録語や区切りとの対応が皆無に等しく，そのままでは使用できない．そこで選択した文には前処理として以下のような操作を行なうことにする．

1. mslrを用いて，EDR辞書(単語数25万語)に基づく形態素解析を行なう．mslrには辞書に基づく分かち書きだけを行なわせた．
2. mslrで解析できた文の内，正しく分かち書きできなかった文を人手で修正する．
3. 解析結果を基に，細品詞[4]によるタグ付けを行なう．正しく分かち書きを行なった文について，その単語それぞれに細品詞を割り振る．

並列構造推定範囲を評価するために，以下の観点を取り入れた．

- 提題・取り立て助詞「は」を最後にもつ文節は主題を表すので，もし始点にあるなら範囲から外す．
- 述語並列では，前部の始点の前に連体修飾文節(名詞+の，連体詞，用言の連体形など)があった場合には，そこまでを前部の範囲とする．
- 述語並列の前部の始点の前にガ格以外の格がある場合は，ほとんどが前部中の用言と並列するはずなので，そこまで範囲を広げる．
- 述語並列の後部の終点が連用形でありその次の文節が用言である場合には，終点の次の用言までを後部とする．

なお，本手法で計算した並列構造が正しいかどうかの評価は人手で行なった．

156文の内パーザが受理できなかった文は3文あり，それらの文は評価対象から外した．文中に含まれるキー文節は全て検出されたが，並列でない部分からの過剰検出もあった(キー文節の品詞曖昧性を含む)．

目標となる範囲推定は「形態素で見たときの範囲」であるから，品詞の曖昧性によって生じた範囲の数は以下のrecall/precisionの計算には考慮していない．つまり「形態素で見たときの範囲」だけをカウントしている．

結果を表3に示す．recallとprecisionの定義は次の通りである．

$$recall = \frac{(\text{範囲推定の正解数})}{(\text{正しい並列構造の範囲の数})}$$

$$precision = \frac{(\text{範囲推定の正解数})}{(\text{スコア1位の推定範囲数})}$$

範囲推定のprecisionが73.3%と期待した値よりやや低めの結果が得られた．EDRコーパスは科学技術論文などと違い，新聞記事や雑誌など，前部と後部の類似性がきちんと見られない場合が多く見られる．また文脈を見なければ，人が見ても範囲が曖昧なものも多かった．そのために並列構造の範囲推定精度が悪くなる傾向があると考えられる．文単位では1文内の並列構造を完全に推定したか，並列構造がないと推定したものは100文であった．従って文単位の範囲推定精度は65.4%となる．

また，本手法の有効性を調べるために黒橋らの手法[5]と本手法との比較を行なった．黒橋らの手法では本手法と異なる辞書や形態素解析システム(JUMAN [8])を用いており，JUMAN+knp[5]による解析結果の単純比較はできない．特に問題となるのは区切り誤りであり，本手法では，区切り誤りはないものと仮定しているのに対し，JUMAN+knpでは区切り誤りを含んでいる．JUMANにおける区切り誤りは実験で扱った153文のうち1割程度の文に見受けられた．knpの推定した範囲については，係り受け解析などを利用して正解を得たものは除いた[6]．

本手法によればrecall/precisionが90.6%/73.3%であるのに対し，黒橋法によれば59.7%/44.6%と著しく差が現れている．これはJUMANのタグ付けが1種類で，しかも間違いを多く含むことに起因している．JUMANは，名詞と用言を間違えることが多く，黒橋法では類似度による評価が大きく変わるためである．一方，我々の手法では，品詞の曖昧性を扱うことが考慮されており，それにより確実に範囲推定の精度が向上することが示された．さらに述語並列に関しては，本手法では完全に後部を検出できたもののうち，黒橋法では後部推定が完全に行なえなかったものが多くあり，後部推定の手法も有効であったと考えられる．

3.2 統語的曖昧性の解消実験

統語的曖昧性とは，次の2つを指す．

- 品詞の曖昧性

[5]黒橋法を計算機上に実装したもの．

表 3: 解析結果の評価 (並列のキー文節単位の正解率)

	名詞並列	述語並列	計
並列構造の数	55	84	139
本手法	名詞並列	述語並列	計
推定した並列構造の範囲数	69	103	172
並列構造の範囲を正しく推定した数	47	77	124
recall/precision	88.9%/68.6%	91.7%/76.2%	90.6%/73.3%
黒橋法	名詞並列	述語並列	計
推定した並列構造の範囲数	71	115	186
並列構造の範囲を正しく推定した数	34	49	83
recall/precision	63.0%/48.5%	58.3%/42.6%	59.7%/44.6%

- 品詞曖昧性のある文節同士の組合せによる構造木の曖昧性 (図 9の枝分かれ)

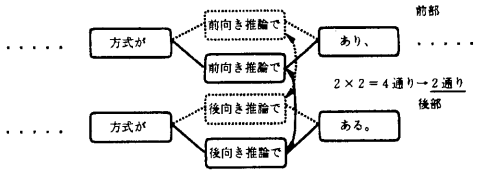

図 9: 組合せによる曖昧性の解消

類似度を見て最大スコアを求める際，類似度の最も高いものを選択することにより統語的な曖昧性を解消できる可能性がある．このことを確認するために，並列構造推定に完全に成功した文から 30 文を取り出し，品詞の曖昧性解消について調べてみた (表 4).

表 4: 統語的曖昧性の解消の実験結果

	正しい解消	誤った解消	未解消
名詞並列	1	1	9
述語並列	8	5	18
計	9	6	27

表 4において，正しい解消とは並列構造内の全ての曖昧性解消に成功したこと，誤った解消とは 1 つでも曖昧性解消に失敗したことを指す．正しい解消を行なった場合の 30 文全体の統語木の統語的曖昧性は 31.3%減少した．

表から分かるように，未解消の数が大きく，曖昧性を解消できなかった場合がほとんどであった (64.3%). 特に名詞並列は，並列構造の範囲が狭い場合が多いのでほとんど曖昧性解消の手立てにはならなかった．曖昧性解消に失敗したのは，同一カテゴリの文節 (形態素は一致するが品詞に曖昧性を持つ文節群) から類似度が低いものを削除した場合だけであり，対応する文節同士の組合せを解消することについては誤ることがなかった．

以上から，表層的な類似度という尺度だけで品詞の曖昧性解消を行なうことはできない．よって並列構造の範囲を推定する段階では，

- 早期に品詞の曖昧性を解消してはならない．
- 統語的曖昧性解消は，組合せの数を減らすだけにとどめる (図 4).

ということが言える．

4 おわりに

本稿では，GLR Skip 法という統語解析法を示し，それを自然言語の並列構造解析に用いる新しい手法を提案した．これは自然言語一般に適用できる手法である．しかし，言語によってはそれぞれに適したヒューリスティックスを用いねばならない．日本語の場合には文法的な制約が弱くこの一般的枠組はそのまま適用できないため，GLR Skip 法の利点を十分に生かすことが不可能である．本稿で述べた日本語固有のヒューリスティックスを導入した並列構造の抽出法の有効性を明らかにするため，日本語 153 文を解析した結果，約 73.3%の精度で並列構造解析を行なうことが可能であった．類似度を用いた統語的曖昧性解消実験では，統語木の組合せの数を減らせる可能性があることを確認した．

今後の課題として，日本語以外の他言語での実験，類似度の算出方法の洗練，日本語では文節の区切りの曖昧性の扱いなどが挙げられる．

参考文献

[1] H. Tanaka, T. Tokunaga, and M. Aizawa. Integration of morphological and syntactic analysis based on lr parsing algorithm. *Journal of Natural Language Processing*, Vol. 2, No. 2, pp. 59–74, 1995.

[2] 日下部俊昭. 自然言語の構造解析へのスキッピングパーザの応用. 修士論文, Tokyo Institute of Technology, February 1996.

[3] 伴光昇. 形態素解析と統語解析の統合処理システムに関する研究. 修士論文, Tokyo Institute of Technology, 1994.

[4] 植木正裕. EDR 辞書を対象とした日本語文法の試作と形態素・構文解析用ツールに関する研究, 1995.

[5] 黒橋禎夫, 長尾眞. 長い日本語文における並列構造の推定. 情報処理学会論文誌, Vol. 33, No. 8, pp. 1022–1031, 1992.

[6] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 言語処理学会学会誌, Vol. 1, No. 1, pp. 35–56, 1994.

[7] 日本電子化辞書研究所. EDR 電子化辞書仕様説明書, 第 2 版, Mar. 1995.

[8] 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木裕, 長尾真. 日本語形態素解析システム JUMAN 使用説明書 Version 2.0, 1994. 2.