自然言語処理　138－10
（2000．7．19）

# 相対インデックス法による文構造分析法の開発と結果

雄山真弓、岡田孝、黒崎茂樹

関西学院大学情報メディア教育センター

〒662-8501 兵庫県西宮市上ケ原
0798-54-6043 / oyama@kwansei.ac.jp

データマイニングは、大規模データベースから新しい知見の獲得―これまで予期できなかったパターンの導出、新しいルールの導出―を行う方法である。構造を持つデータを対象とするデータマイニングの手法をコーパスデータに適用すれば、これまで発見できなかった文構造上の新たな知見が得られるであろう。本論文は、はじめに構文解析木の表記法について論じ、相対インデックス法を構造のあるデータに適用し、これまで行われてこなかった構造を持つデータを対象とするデータマイニングの手法を開発した報告を行う。次に、ＥＤＲの英語コーパスデータを用いて自動詞の think と他動詞の think を含む文の構文構造上の特徴について、分析を行った結果を報告する。

データマイニング、相対インデックス、構文解析木、知識発見

# Data Mining of Sentence Structures using Relative Indexing of Vertices

Mayumi OYAMA, Takashi OKADA, Shigeki KUROSAKI
Center for Information & Media Studies, Kwansei Gakuin University
Nishinomiya, Hyogo, 〒662-8501
0798-54-6043 / oyama@kwansei.ac.jp

Data mining deals with the discovery of hidden knowledge, unexpected patterns and new rules from large databases. Corpuses are large databases and text-mining techniques are making rapid progress. If we apply the technique of data mining to discover knowledge in corpuses, we will get good results. But, to analyze corpuses with syntactic parse trees we have to treat their structures. In this paper, we show the inscription of syntactic parse tree first and we propose the data mining methods using the relative indexing of vertices for knowledge discover from corpuses. We show the distinctive structural features found between an intransitive verb "think" and a transitive verb "think".

data mining, relative indexing, syntactic parse tree, knowledge discovery.

## 1. はじめに

本研究の目的は、構造をもつデータを対象としたデータマイニングツールの開発とその応用を行うことにある[1][2]。言葉や文は構造を持つデータである。我々は、複雑な表現をしようとすればするほど、文のなかで使われる単語の係り受け構造が複雑になる。構文解析木（以下、構文木）はこの係り受け関係の複雑さを括弧などによって表現したリスト形式のデータである。近年では、日本語や英語の膨大な文例を集めた大規模データベースの蓄積が行われている。特に、日本電子化辞書研究所で作成されたＥＤＲコーパスは、英語や日本語の文例と共に、文を構成する単語の品詞やその共起関係、文の係り受け構造を表現する構文木などを含むコーパスデータである [3]。構文木のような、構造をもつ大量のデータは、テーブル型のデータベースでは蓄積できないため、構造そのものを分析する場合、一般の統計分析の手法では扱うことが難しかった。

ここで述べる構造のあるデータの分析とは、単に構文木全体のパターンを分類する方法ではなく、文のなかに含まれる単語の使い方等で異なるグループに分けられたデータから、それぞれのグループが持つ特徴的な部分パターンを探索することを意味する。例えば、英語の文例から動詞（例えば think）を１つ選択しそれが自動詞として使われる場合と他動詞として使われる場合で、その動詞の置かれた位置を視点として、全ての周辺構造とその属性を調べて、自動詞としての使い方と他動詞としての使い方に違いや共通点等が見いだせるかどうかを探索し、新しい知識を発見するものである。

本論文では、最初に構文木の構造表記法について考察した後、データマイニングでよく使われる各種技法毎に、構造情報の取り扱いの可能性を評価する。さらに、構文木の構造表記法として提案した相対節点インデックス化を英語コーパスに適用し、構造、単語、品詞を組み合わせたデータを作成する。最後に、構文木に含まれる品詞とその構文木中の位置関係で、使われる頻度の多いルールを得た結果を報告する。

## 2. 構造の表記法

構文木は順序づけられた有向木であり，順序付きでなくまた閉サイクルを含む化学グラフなどと比較すると，多様な表現法が可能である。通常のコーパスでは、括弧で表現したリスト構造で表されており、これが一般的な表記法であろう。図１(ａ)に示した “My son has a red pen”の構文木のリスト表記例を次に示す。

```
    My son has a red pen

(S  (NP  (pron  "My")  (n  "son"))
    (VP  (v  "has")
         (NP  (a  "a")
              (NP  (adj  "red")
                   (n  "pen")))))
```

なお、各節点の先頭に斜体で付した記号は通常の文法での概念である。

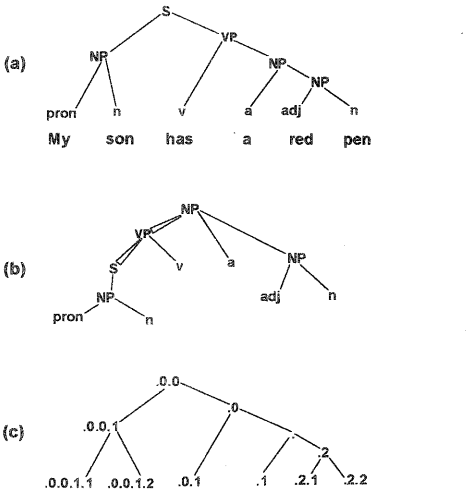


図１．木構造表現法

ここでは、構文木中に特定の視点となる節点を定めた場合に、可能となる２種の表記法について考察する。上記で”a red pen”の名詞句を視点として定めた場合を例とする。

（Ａ）視点を摘んで持ち上げ振り回すと考えれば、視点が根節点となるように変換され、図１(ｂ)のようになる。なお、元来の親節点は左端になるように配置し、それを明示すため、くさび形の記号で示している。実際にこの木をどの様に符号化するかは各種の方法が考えられ、考え方だけを取り入れて、変換せずに元の木のままで扱うこともできる。この木は、各節点の深さが視点からの距離に対応しており、視点を含む部分木を探索する際に便利である。この技法を逆転木化と名付けることとする。

（Ｂ）図１(c)は、図１(a)に示す木のすべての節点に、視点を原点としてユニークな番号付けを行った例である。すべての節点は，原点からその節点に至るユニークな経路を有する。そこで、経路中で親節点への移動には .0 を、

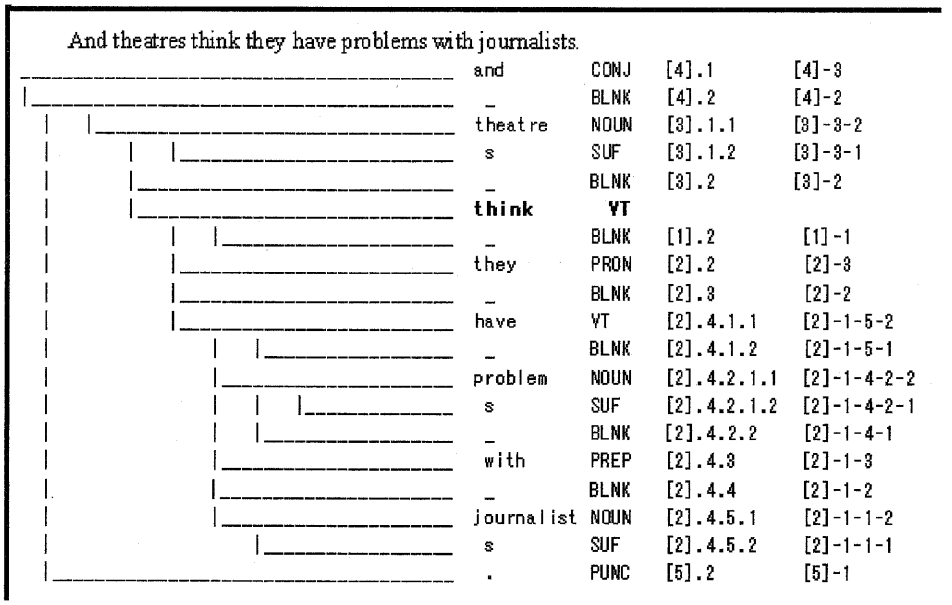


図２．構文木構造と単語の品詞および相対節点インデックス表現ＡとＢ（視点：think）

付し、子節点への移動には、左側から数えた子節点番号を付すと図に示した番号付けができる。この技法を、相対節点インデックス化と名付けることにする。

この様にして得られた番号は、視点から各節点への相対的な位置情報を表しており、節点付随の情報と組み合わせることにより、構造と節点内情報の相関を探索する上で重要な手がかりを与えてくれる。なお，子節点の番号付けは右側から数える方法もあり，最適な番号付けは課題に応じて選択すべきものであろう。

図２は、英文 [And theatres think they have problems with journalists.] について構文木の係り受け構造と記号を含む単語とその品詞、think を視点とした相対節点インデックスの値を記号化して表現している。節点にぶら下がる子供の数を左側から数えた表現をＡで、右側から数えた表現をＢに示している。[4] は視点(think)から数えて４番目の親の位置 0.0.0.0.を表している。Ａの表現の[3].1.1 は、視点 think から３つ目上の親に付いた子供中で、左側から数えて１番目の子供のそのまた左側から１番目の子供を意味する。Ｂの表現では[3]-3-2 は think から３つ上の親についた子供中で、右側から数えて３番目の子供に移りそのまた右から数えて２番目の子供を意味する。Ａ、Ｂの表現は、共に相対節点インデックス法を用いている。分析では、両方のデータを使うことによって詳しい情報が得られる。本研究で使用した相対節点インデックス化データは、変換プログラム EDRitem を LISP で作成し、ＥＤＲ英語コーパスデータを入力データとして作成した。

## ３．データマイニング技法と構造情報の取り扱い

現在データマイニングに利用されている主要な技法は以下のようなものであろう。

１．多変量解析
２．決定木およびルールの導出 [4]
３．帰納論理プログラミング [5]
４．Graph based induction [6]
５．ニューラルネットワーク
６．クラスタリング
７．相関ルール探索 [7] [8]

これら技法は、その内容として教師付き学習と教師なし学習の何れかに限定されているものと、その双方を含むものがある。また、構文木のトポロジカルな形状だけに興味のあるケースは少なく、実際に使われている語まで掘り下げて解析する場合には、属性値の種類が非常に多くなることを考慮する必要がある。属性値数を制限して解析する場合には、意味論にまで踏み込んで概念階層化等の方法により、何らかの符号化を行う必要があると考えられる。以下、各技法について、構造情報取り扱いへの拡張可能性を簡単に評価しておく。

### 3.1 多変量解析

多変量解析の方法は、本来連続数値情報の解析から発展してきた。これまでの多くの研究成果もあり、信頼性のある方法論である。従って、内容についての解説は不要であろう。

この方法では、構造情報を直接取り扱うことは、原理的に無理がある。しかし、特定の語句に視点を定めた上で解析する場合、以下のような方法が考えられる。

a. 相対節点インデックス化により得られた節点番号を属性とする。
b. 個々の文における当該位置の語句の内容を属性値とする。

こうすれば、通常の表形式のデータがそろうこととなり、各種の解析が可能である。この考え方は、多変量解析に限らず他の技法でも適用できる。なお、属性値数が多いことと missing value が頻発することに留意し、それに対応できる技法を選択すべきである。

### 3.2 決定木およびルールの導出

決定木は教師付き帰納学習で多用される簡単な方法であり、計算時間も早い。しかし、前項で述べた相対節点インデックス化による方法は、原理的には適用できるが、属性値の種類が多く、現実的には適用困難であろう。

この方法の適用対象を逆転木に拡張し、浅い節点から順に識別力の高い変数を選択して、決定木を構成する方法は、我々がすでに発表した[9]。しかしながら、決定木は元来、識別に用いる変数を greedy に探索するため、識別力のない節点を飛び越して、その節点に繋がる深い節点に存在する良い知識を得ることは難しい。機械的に判別すればよいと言う立場ではなく、興味ある知識を得るという立場からは、この方法は制御が困難であろう。

### 3.3 帰納論理プログラミング

発見概念を記述する述語を自動的に生成するもっとも柔軟な方法であるが、大規模コーパスへの適用は計算時間の点から当面困難と予想される。

### 3.4 Graph based induction

この方法によると、連結された特徴的な部分グラフを高速に探索することができる。また、教師付き学習の場合にも拡張されている。属性値数を制限できれば、今後有効に活用できると期待される。

### 3.5 ニューラルネットワーク

逆伝搬型、あるいは Kohonen 型の何れのネットワークも、基本的に構造入力は困難である。しかも、結果として得られるネットワークの解析が困難であるため、相対節点インデックス化により、何らかの自動識別を行う場合以外は適用範囲が限られるであろう。

### 3.6 クラスタリング

この方法のベースには、ユークリッド空間であれ、記号空間であれ、基本的には事例や生成されたクラスター間の距離概念が必要である。相対節点インデックス化を行い、また属性値の一致数により距離を定義する可能性が考えられる。適当な概念階層化により合理的な距離定義が得られるならば、興味ある結果の得られる可能性がある。

### 3.7 相関ルール探索

元来マーケティング調査を行うため、顧客が購買するアイテム間の相関を調べる目的で提案された方法である。最小サポート数の概念を設定することにより、大容量のデータベースの解析を可能としている。全く異なった分野からのものではあるが、対象が表ではなくアイテムを要素とするリスト構造であったため、時系列的な購買行動の解析、要因・結果分析などに拡張されている。さらに、扱うアイテム群の概念階層化も実用化されている。

また、元来教師なしの学習を対象としていたが、出力されるルールの中からクラス記述を帰結部に有するルールのみを解析することで、教師付き学習への適用も試みられている。さらに筆者らは最近、この方法のシステム的な枠組みを一新することにより、カスケードモデルによる教師付きの学習法を提案している[10]。4.ではカスケードモデルを使った結果も報告する。

これらの発展経過を見ると、相関ルール探索は構文木からのデータマイニングに今後大きな役割を果たす可能性がある。以下、いくつかの具体的な方法論を提

案してみよう。

1. 相対節点インデックス化により得られた節点番号と節点属性をアイテムと考え、通常の相関ルール探索を行う。元来多種多様な商品を対象としたものであるため、属性値の概念階層化などを考慮する必要がなく実行可能である。その結果から、特定の位置関係にある複数の語間の特徴的な相関関係を得ることが期待できる。
2. 上記と同様の処理を、クラス属性を加えてカスケードモデルにより解析すれば、識別に有効な語間の相関関係を得ることが期待できる。
3. アイテムのリスト構造を部分木に拡張することにより、個別節点を抽出するだけでなく、特徴的な部分木全体を獲得することが期待できる。かなりのシステム開発が必要となろうが、大きな成果が期待できる方法論である。

## 4 相対節点インデックス化を使った構造データの分析

### 4.1 入力データについて

ＥＤＲ英語コーパス(約１６万文例)から、動詞 think の自動詞(182 文例)と他動詞(300 文例)を含む文を全て検索し利用した。検索・変換プログラム EDRitem で、データの形式はデータ番号、単語、相対節点インデックス番号、品詞から構成される。相対節点インデックス記号は枝の数を右側から数える場合(A)と左から数える場合(B)がある。以下にデータ構成を示す。

| 番号 | 単語 | 相対節点インデックス記号 | 品詞 |
|---|---|---|---|
| EC00002661 | _ | [3]-2 | BLNK |
| EC00002661 | a | [4]-7-5-3 | ART |
| EC00002661 | _ | [4]-7-5-2 | BLNK |
| EC00002661 | spokesman | [4]-7-5-1-2 | NOUN |
| EC00002661 | _ | [4]-7-5-1-1 | BLNK |
| EC00002661 | for | [4]-7-4 | PREP |

ＥＤＲコーパスは文中のスペースについても構造を与えている。_はスペースを意味する。品詞の種類は記号も含めて

ADJ, ADV, ART, AUX, BE, BLNK, CONJ, DEMO, INDEF, ITJ, NOUN, NUM, PREP, PRON, PTCL, PUNC, SUF, SYM, VI, VT, WH, UNIT, VERB

の23種類がＥＤＲコーパスデータで定義されている。

### 4.2 分析の方法と結果

データマイニング手法を用いて分析を行うには、対象とするデータについて、データクリーニングを行っておくことは分析のために重要である。データ分析用ソフトウェアＪＭＰを使ってデータクリーニングの手作業と分布及びクロス集計をおこなった。データクリーニングの方針として、相対節点インデックス記号や品詞の数が対象としているデータに対して極度に少ないものは省いた。データマイニングを行う対象データは以下の通りである。

think を含む文から取り出した、文番号、構造・品詞・単語の組み合わせのアイテム数

自動詞の場合：3606 件（文例：182）

他動詞の場合：6607 件（文例：300）

分析はインテリジェント・マイナーの相関関係探索手法を用いた。分析に用いたパラメータ値と導出されたルール総数、最小確信度を以下に示す。

| | 自動詞 | 他動詞 |
|---|---|---|
| ・最小サポート値 | 9.5% | 21.5% |
| ・最小確信度 | 100% | 100% |
| ・ルール総数 | 10,693 | 9,331 |
| ・品詞が関係するルール | 51 | 65 |
| ・think 周辺の特徴的構造パターンの数 | 9 | 5 |

最小サポート値は、導出されたルールのトランザクションに対する割合をしめす。アイテム X,Y の関係 X→Y というルールがある場合、ルールの確信度は次のように計算できる。

確信度（X→Y）＝
サポート係数（XUY）／サポート係数（X）

品詞と構造が関係するルールは、自動詞 51 件、他動詞 65 件であるが、そのルールから構造上同じ内容のものをまとめると、自動詞 think の周辺構造の特徴的なパターン９件、他動詞 think の周辺構造の特徴的なパターン５件となった。それぞれのパターンを図３、図４に示す。また、各ルールのサポート係数、文例数と、文例、出典を表１、表２に示す。

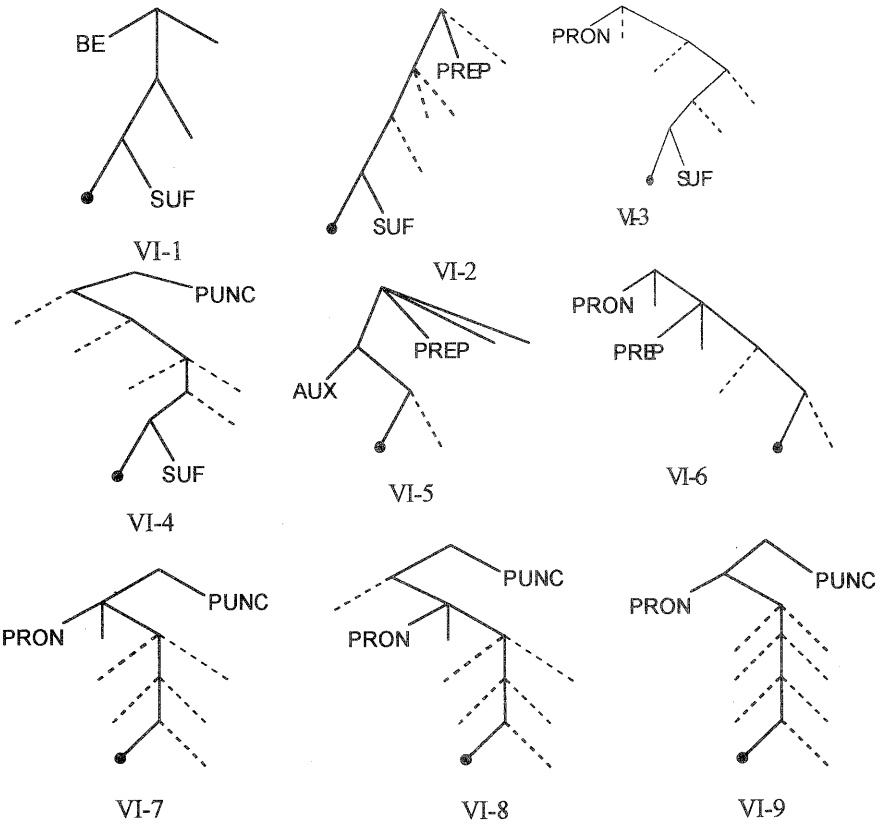


図3．　自動詞 think の周辺構造パターン（黒丸は think の位置）

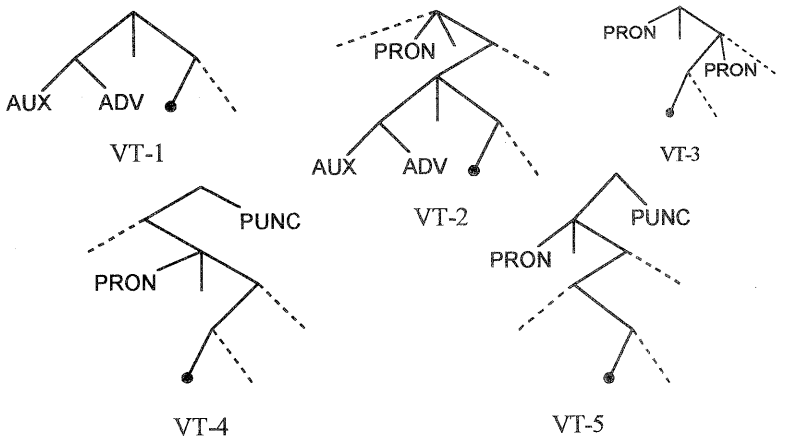


図4．　他動詞 think の周辺構造パターン（黒丸は think の位置）

表1. 自動詞の場合のルールとサポート係数(S),文例（確信度=100%）[出典]

VI-1 (SUF-BE) (S=9.890%, 文例数=18) [The Japan Economic Journal]
Clearly, Nakasone was thinking of a similar role to Saionji's, observers say.

VI-2 (SUF-PREP)(S=9.890%, 文例数=18) [The Japan Times ]
But the postal part has held firm, and this is what everybody thinks of as the ministry's real work.

VI-3 (SUF-PRON)(S=9.890%, 文例数=18) [旺文社表現辞典]
Are you also thinking about a link with the CAM system?

VI-4 (SUF-PUNC) (S=9.890%, 文例数=18) [旺文社表現辞典]
Do you always relate to your customers thinking only about profits?

VI-5 (AUX-PREP)(S=10.989%, 文例数=20) [The Japan Times ]
But before we can think too much about spring, we have Christmas to enjoy.

VI-6 (PERP-PRON)(S=10.440%, 文例数=19) [The Independent]
But how long was it before I was doing so without even thinking?

VI-7 (PRON-PUNC)(S=12.637%, 文例数＝23) [旺文社表現辞典]
Can't you think of a more efficient way to get it done?

VI-8 (PRON-PUNC)(S=9.890%, 文例数=18) [The Japan Times ]
Although such an argument seems plausible on the surface, I don't think it gets to the point.

VI-9 (PRON-PUNC)(S=10.440%, 文例数=19) [旺文社表現辞典]
He obviously thinks he's a great comedian.

表2. 他動詞の場合のルールとサポート係数文例（確信度=100%）（出典）

VT-1 (AUX-ADV)(S=23.667%, 文例数=71) [The Japan Economic Journal]
And in the end, I don't think we should have any doubt.

VT-2 (AUX-ADV-PRON)(S=22.000%,,文例数=66) [The Japan Economic Journal]
But I don't think funds will move so widely.

VT-3 (PRON-PRON)(S=22.000%, 文例数=66) [The Independent]
All those chaps have African forebears, I think you'll find.

VT-4 (PRON-PUNC)(S=24.667%, 文例数＝74) [The Japan Times]
A spokesman for the city said it cannot disclose at this time what price it thinks is acceptable.

VT-5 (PRON-PUNC)(S=30.000 %, 文例数＝90) [The Japan Times]
Besides, I think, Americans like the novel because there is a lot of action in the story.

### 4.3 カスケードモデルによる分析

カスケードモデルのプログラム DISCAS を使って分析した結果を以下に示す。分析に用いたデータはインテリジェント・マイナーと同じである。パラメータは以下の通りである。

| | |
|---|---|
| Min-support: | 5% |
| Threshold: | 0.04 |
| 自動詞のルール | 16件 |
| 他動詞のルール | 14件 |

カスケードモデルのルールの特徴は、{[4]-3PRON: no} のように [4]-3 の位置に PRON が存在しないことを示す構造パターンを導出する。ルール例を以下に示す。

相対インッデクスの右側からの表示と左側からの表示を全て含むデータを用いた結果である。

() 内はルールがあてはまる文例数。

```
R1:  IF [[3].2 PREP: y ]  THEN  Verb = VI
(50)
R2:  IF [[2]-3-1ADV: y] on [[3].1 PRON:
       n] THEN Verb = VT (76)
R3:  IF [[2].2 PREP: y] THEN Verb=VI (47)
R4:  IF [[2].2 PRON: y] on [[4]-3PRON:n]
       THEN Verb = VT (73)
R5:  IF [[4].2 PREP: y] THEN Verb = VI
(39)
R6:  IF [[3]-3PRON: y] on [[4].1 PRON:
       n]
       THEN Verb = VT (143)
R7:  IF [[3]-3PRON: n] on [[4]-3PRON: n]
       THEN Verb = VI (171)
R8:  IF [[3]-3PREP: y] THEN Verb = VI (56)
R9:  IF [[3]-3PRON: y] THEN Verb = VT (175)
R10: IF [[2].1.1AUX: y] on [[3]-3PRON:
       n] THEN Verb = VT (63)
R11: IF [[2].1.2ADV: y] on [[3]-3PRON:
       n]
       THEN Verb = VT (57)
R12: IF [[2].2PRON: y] THEN Verb = VT (74)
R13: IF [[3]-3PRN: n] on [[4].1PRON: n]
       THEN Verb = VI (172)
```

## 5. まとめ

構文木データに、相対節点インデックス化を使って構造を含むデータの分析を行った。データマイニングの基本であるデータクリーニングをＪＭＰのソート機能やエディター機能、グラフ機能を有効に使って比較的簡単に行うことができた。言語データの分析には、人間とコンピュータとのインタラクティブな作業も大切であることを実感した。今後の研究としては、think 以外の動詞について分析をおこなう予定である。特に、あらゆる表現を難しい単語を使わなくても表現できるといわれている基礎動詞（be, do, have, see, take, get, put, keep, say, go, come, make, let）の 13 種について自動詞と他動詞の関係を調査する。さらにこれまで、行ってきた日本語のコーパスについても同様な方法で研究をおこなう。また、視点とした枝位置を基準とした番号付けによるインデックス化も試みる予定である。

## 参考文献


[1] 雄山、岡田、李：”構文解析木を対象とするデータ解析法の研究（1）－方法論についての一考察－”,重点領域研究、シンポジウム「人文科学における数量的分析」 pp.71-78 (1996)

[2] 雄山、岡田、李：”構文解析木を対象とするデータ解析法の研究－構文木からの知識発見システム SYKD の開発と応用－”, 情報処理学会研究報告 98-CH-38, pp.69-77 (1998)

[3] EDR：EDR 電子化辞書 1.5 判仕様説明書,日本電子化辞書研究所(1996)

[4] Quinlan J.R.: “C4.5: Programs for Machine Learning”, Morgan Kaufmann, (1993); 古川訳: “AI によるデータ解析”、トッパン (1995).

[5] 国藤編: “小特集 帰納論理プログラミング”、人工知能学会誌、Vol.12, No.5 (1997)

[6] 吉田、元田: “逐次ペア拡張に基づく帰納推論”, 人工知能学会誌, Vol.12, pp.58-67 (1997).

[7] Agrawal, A. et. al.: “Database Mining: A Performance Perspective”, IEEE Trans. on Knowledge and Data Engineering, Vol. 5, No.6, pp.914-925 (1993).

[8] Agrawal, A. et. al.: “Fast Alogorithms for Mining Association Rules”, Proc. VLDB, pp.487-499 (1994)

[9] Li, G. et.al.: “Knowledge Discovery from Syntactic Trees”, 1996 年度人工知能学会全国大会（第 10 回), 08-01, 東京 (1996).

[10] Okada, T.: “Rule Induction in Cascade Model Based on Sum of Squares Decomposition” Principles of Data Mining and Knowledge Discovery, pp.468-475, Lecture Notes in Artificial Intelligence 1704, Springer (1999).