人文科学とコンピュータ　40－3
（1998．10．23）

構文解析木を対象とするデータマイニング

雄山真弓　　岡田孝

関西学院大学
情報処理研究センター

この数年来、データマイニングという名の下に、大規模データベースから知識発見を行うことが重要視されている。特に、マーケティングを中心にビジネスへの応用が実社会でも成功しており、理論・開発・応用すべての面で研究が加速化している。コーパスは、まさにこのような大規模データベースである。最近はテキストマイニングの技術も急速な展開を見せているため、現在のデータマイニング技術をそのままコーパスへ適用しても、各種の成果を得ることが期待できる。しかし、構文木情報が付加されたコーパスデータを解析する場合は、当然ながら構文木のもつ構造情報の活用を指向することとなる。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技法毎に、構造情報の取り扱い可能性を評価し、同時に構文木に対する各種のマイニング法を提案する。さらに、構文木からの知識発見法として著者らが開発したデータマイニングのための SYKD システムについて解説をおこなう。

Data-mining Methods for corpuses with syntactic parse trees

Mayumi Oyama　　Takashi Okada

Information Processing Research Center
Kwansei Gakuin University

From these five years, data mining is currently regarded as the key element of a much more elaborate process called knowledge discovery in databases. Data mining deals with the discovery of hidden knowledge, unexpected patterns and new rules from large databases. Especially data mining is closely linked to another important development – data warehousing on business. And the research into theory and newly developed technique is accelerated. Corpuses are large databases andtext 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 its structure. In this paper, we show the inscription of syntactic parse tree first and evaluate techniques which can be used for data mining. And we propose the data mining methods for knowledge discover on corpuses. Lastly, we explain the knowledge discovery system for syntactic parse trees, SYKD.

1. はじめに

情報化の進展によって、さまざまな分野で大規模データベースの蓄積が行われている。また、それらを使った分析や予測がおこなわれ有効な情報を提供している。その中でも、数年前からデータマイニングという名の下に、大規模データベースから、新しい知識発見を行うための手法が注目されている。特に、マーケティングを中心にビジネスへの応用が実社会でも成功しており、理論・開発・応用すべての面で研究が加速化している[1]。コーパスは、まさにこのような大規模データベースである。最近はテキストマイニングの技術も急速な展開を見せているため、現在のデータマイニング技術をそのままコーパスへ適用しても、各種の成果を得ることが期待できる。しかし、構文木情報が付加されたコーパスデータを解析する場合は、当然ながら構文木のもつ構造情報の活用を指向することとなる。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技法毎に、構造情報の取り扱い可能性を評価し、同時に構文木に対する各種のマイニング法を提案する。さらに、構文解析木からの知識発見法として著者らが開発したSYKD[2]について解説をおこなう。

なお、構文木の構造情報としては、木の高さや幅，重心のような通常の属性、属性値の対で表現できる内容もあり、これらは、通常の統計操作を施すことができる。しかし，我々がすでに示したように[3]，このような方法では、例えば作家の識別をできる可能性はあっても，個別の語句の用法に立ち入って新しい知見を得られるわけではないため、本稿で言う構造情報には含めない。

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。

```
(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。

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。

(I) 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、それを明示するため、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。

(II) 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。

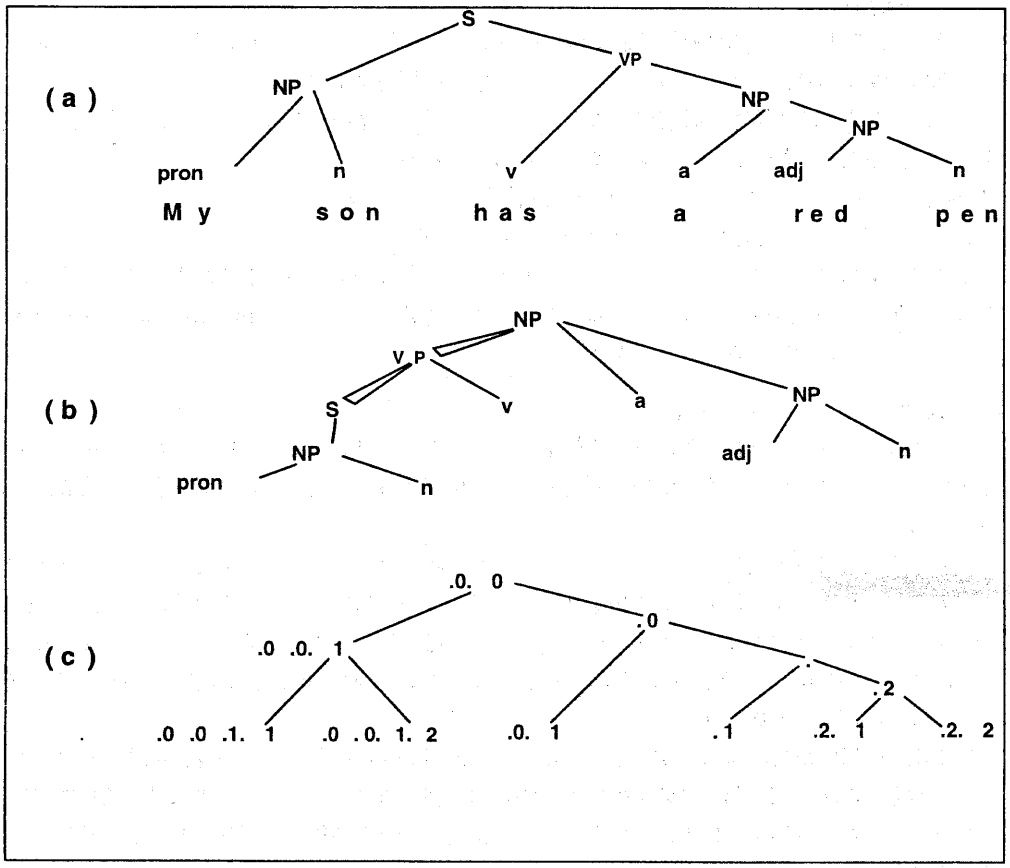

図1．木構造表現法

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。以下各技法毎に、構造情報取り扱いへの拡張可能性を簡単に評価してみよう。

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。

4節で我々の用いた方法論は、逆転木化した上で、field-of-view と呼ばれる探索範囲を順次拡大し、その中での識別力ある節点および変数の探索に決定木を使う方法である。この方法は、機械的に短時間で結果を得たいという場合には不適当であるが、対話的に研究者の興味を反映させながら解析を行う場合は、適当な方法論であるといえる。

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。

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. 構文解析木を対象とした知識発見システムＳＹＫＤ

ここでは、我々が開発した、構文解析木を対象とした知識発見システムＳＹＫＤについて、その方法論とシステムの紹介をおこなう。方法論は、逆転木化した上で、field-of-view と呼ばれる探索範囲を順次拡大し、その中での識別力ある節点および変数の探索にＩＤ３法を適用し、特徴的なパターンの探索を行うものである。本方法は、機械的に短時間で結果を得たいという場合には不適当であるが、対話的に研究者の興味を反映させながら解析を行う場合は、適当な方法論であるといえる。

4.1 ＳＹＫＤシステムの方法論

4.1.1 構文木の構造

次のような文章（A）の構造を、リスト構造で表現したもの(B)と、構文木構造で表したもの（C）を図２に示す。ここで、（B）では[ ]、（C）では文字が黒く囲まれている部分が、分析の中心となる viewpoint である。viewpoint とは、構文木の部分構造のパターンを探す出発点となるもので、（B）のデータを作成する段階で、予め設定しておくものである。また、（C）の各節点には、（B）のデータか

ら作成された構文木のトポロジー的属性と、各節に含まれる文法的属性が表示されている。これらの５つの属性（V1,V2,V3,V4,V5）の意味を以下に示す。

V1：現節点から viewpoint 節点への方向を示す属性値で、０の場合は親節点、ｉの場合はｉ番目の子節点を示す。

V2：親節点の数を示す。

V3：子節点の数を示す。

V4：親節点の下で、現節点が主要な意味を担っているか（ｔ）否か（T）を示す。

V5：品詞や節に含まれる文法的属性で１５種類の品詞と M,S,N,I[２]などの合成関係演算子のいずれかを示す。

(A)

「聞き上手」は、思いやりが身についた人の境地なのだろう。

(B)

```
(T S(t M(T S(t S(t S(t S(T 記号 "「")(t 名詞 "聞き上手"))
(T 記号 "」"))[T 助詞 "は"])(T 記号 "、"))(t S(t M(T S(t
M(T M(T S(t 名詞 "思いやり")(T 助詞 "が"))(t S(t 動詞
"身につ")(T 語尾 "い")(T 助動詞 "た")))(t 名詞 "人"))
(T 助詞 "の"))(t 名詞 "境地"))(T 助動詞 "な")(T 助詞
"の")(T 助動詞 "だろ")(T 語尾 "う")))(T 記号 "。"))
```

(C)

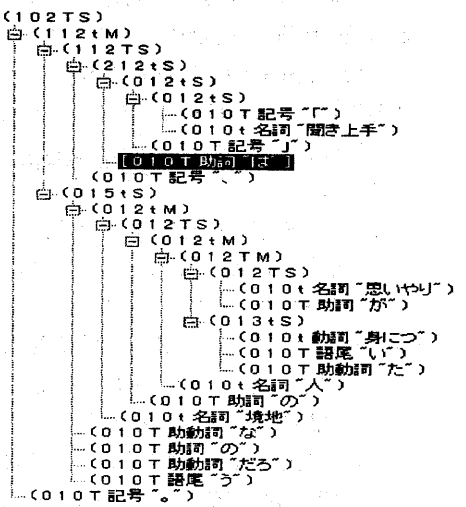

図２　データ構成 (A), (B), (C)

4.1.2　構造を探索するための viewpoint の設定と field-of-view

構文木中で構造を調べる核となる節点を viewpoint とする。具体的には、文中で良く使われる単語や助詞、文の区切りなどに使う読点でもよい。これらは、分析をおこなう前に決定しておく。

viewpoint に連結した複数の節点の集合を field-of-view と呼ぶ。図３は viewpoint と field-of-view の概念図である。field-of-view 内の各節点は、(V1,V2,V3,V4,V5) の５つの属性が節点毎にあり、これらの全ての属性を対象として、ID3 法によるデータ分類を行う。このように、SYKD システムでは、クラスラベルの選択と viewpoint の設定が必要である。次に分析の経過で必要になるのは、field-of-view の展開方法である。、field-of-view の展開は、インタラクティブにおこなえるシステム、つまり field-of-view の展開はすべて利用者の指定により行うこと、また利用者が複数の方向へ field-of-view を展開し、そのそれぞれにおいて解析を遂行できるように設計されている。

4.1.3　ID3 法について

field-of-view が決定した段階で、次に field-of-view 内に含まれる節の属性全てについて ID3 法で分類をおこなう。viewpoint から出発して、どの節のどの属性がクラス分けに効いているかを全属性について ID3 法で調べる。その結果、分類に関係しない属性については？のマークをつけてパターンの表示をおこなう。

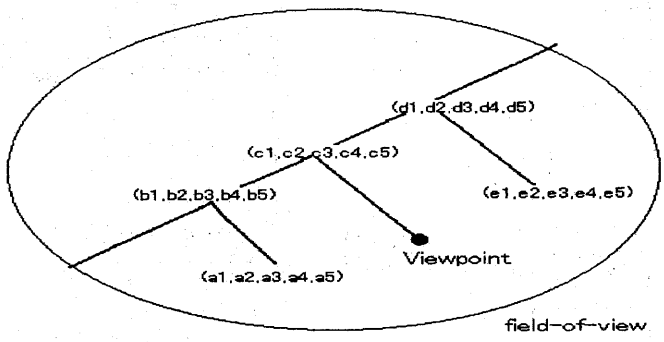

図3　viewpoint と field-of-view の概念図

ここで、ID3 とは、分類したいクラスラベルを含んだデータ群を対象として、それぞれに与えられた複数の属性値の中から、どの属性値を使ったらもっとも効率的にクラス分けができるかを調べる方法である。本研究の構文木のデータでは、viewpoint を始点として展開された、field-of-view の各節点につけられた全ての属性について最も識別力のある属性はどれかを順次計算し、初期に設定した分類条件に合ったパターンを見つけていく方法である。なお識別力の強さは、各段階での次式の情報量で比較を行う。

-P+log2P+　－　P-log2P-

ここで P+ ,P-は、確率で、2つのクラスの識別子を＋、－の記号で表している。分類前の情報量と分類後の情報量の重み付き和を比較することにより、分類による情報量の獲得値が計算できる。各属性について全て情報量を調べ、情報の獲得量が大きいものを順次選んで属性値の優先順序を決定する。また、分類計算を行う上で、以下の3つの値を満たす条件にしたがってパターンの検出を行う。

分類条件

1．分類を行うための最少サポートインスタンス数

2．有効なパターンであるための最少のサポートインスタンス数

3．クラス判別に必要な最低の確信度

計算において、サンプル数が1の条件を満たさない属性については、処理を行わない。また2，3の条件を満たすパターンを見つけた場合は、システムに記録しておく。以上構文木から知識発見を行うための方法について述べてきた。次にこれらの方法をつかって作成した SYKD システムの概要を述べる。

4．2　　ＳＹＫＤシステムの構成

SYKD システムは図4に示すような構成をとっている。また、メイン画面を図5に示す。

コーパスデータベースから ProDef という検索プログラムによって学習用構文木ファイル、およびテスト用構文木ファイルを準備する。テスト用の構文木ファイルは、学習されたパターンの正確さを検証するために用い、そのフォーマットは学習用のものと全く同じである。構文木ファイルのデータ構成は、クラスラベルと図2の（B）に示す木構造をリストで表現したもので、節点毎に埋め込まれた各種の属性値、[] で示される viewpoint 位置などの情報を持つ。

SYKD プログラムは、入力ファイルとして、学習用構文木ファイル、テスト用構文木ファイル及び学習用のパラメータを指定する。さらに、タスクと呼ばれる単位を基本として利用者とのインターフェースを取る。

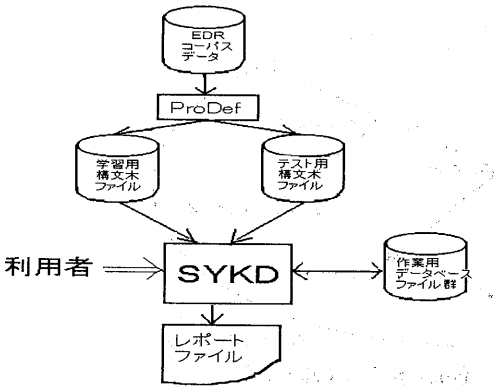

図4　SYKD のシステム構成図

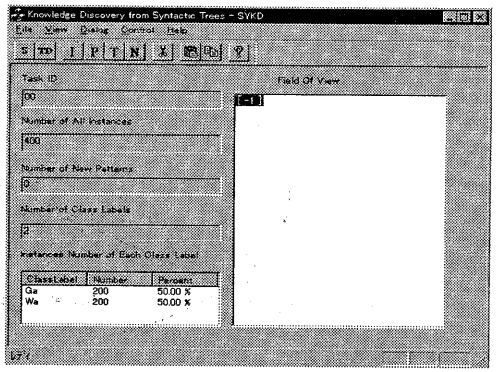

図5　SYKD のメイン window

SYKD システムは、EDR 日本語コーパスデータベースを入力データとして、構文木上で共通する記号(例えば読点)や品詞の周辺構造の属性を調べ、よく使われているパターンや数は少ないが特徴的と見られるパターンを発見し、構文木からの新たな知識を発見するデータマイニングのシステムである。

5．　まとめ

会話的な操作による発見を目的とした全節の方法論から離れて、高速かつ自動的なデータマイニングを目的とする場合、上記の評価を元に今後の方向性を考えると以下のようになろう。なお、以下の記述は筆者らの主観的なものである点に留意されたい。

まず、相対節点インデックス化により、与えられた視点からの属性：属性値対の表を作成することが第1段階であろう。第2段階としては研究の効率性を考え、既存の商用ソフトウェアを利用できる統計解析、相関ルール解析をこの表に対して実行し、それぞれの方法を評価する。さらに高度の結果を求める場合には、第3段階として、直接木構造を取り扱うための相関ルール探索システムあるいはカスケードモデルによるシステムを構築して、解析を実施する。いずれにしろ、今後、データマイニングの諸技術が構文木のような構造化オブジェクトの解析に大きな役割を果たしていくことは間違いないと考えられる。

参考文献

[1] 沼尾編:"特集 大規模データベースからの知識獲得", 人工知能学会誌、Vol.12, No.4 (1997)
[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（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」pp71-78(1996)
[3] 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,pp69-77(1998)
[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). (1994)
[8] Agrawal, A. et. al.: Fast Alogorithms for Mining Association Rules, Proc. VLDB, pp.487-499
[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).