自然言語処理 108－14
(1995. 7. 21)

シソーラスを利用した言語データ最適一般化アルゴリズム

田中 英輝

NHK放送技術研究所
先端制作技術研究部

157 東京都世田谷区砧 1-10-11
電話 03-5494-2314
E-mail tanakah@strl.nhk.or.jp

あらまし 自然言語処理に利用するための規則をコーパスから学習する研究が最近盛んになっている．これらの研究では，得られた規則の適用範囲をいかに一般化するかが大きな課題となる．なぜなら，コーパスから直接学習される規則はそのままでは適用範囲が極端に狭いからである．現在はこの問題を解決するためにシソーラスを利用した手法が試みられている．このとき，シソーラス上のどの概念で規則を一般化するかが問題となる．しかしシソーラス上のノードの選び方は，組合せ的に爆発を起こすためその決定は容易ではない．本稿では，この問題を線形時間で解く基本的なアルゴリズムを提案する．本稿の問題は一般的に言うと帰納学習の分野で問題とされていた「構造化属性」の問題に属する．さらに，決定木の最適部分木を求める問題とも等しい．

キーワード 機械学習 構造化属性 一般化 シソーラス コーパス 機械翻訳

A Linear-Time Algorithm for Optimal Generalization of Language Data

Hideki Tanaka
NHK Science and Technical Research Laboratories
1-10-11, Kinuta, Setagaya-ku, Tokyo, 157
phone: 03-5494-2314
E-mail: tanakah@strl.nhk.or.jp

Abstract The proper treatment of structured attributes in inductive learning is getting much attention as this learning technique is now frequently applied to the knowledge extraction in natural language processing. In this context, the problem is finding a set of thesaurus nodes that maximally generalizes words in the learning source, but causes minimum errors. The number of candidate node sets, however, explodes as the thesaurus size increases, and no efficient algorithm has been discovered so far.

In this paper, we propose the algorithm T* which can find the optimal node sets in linear-time. This algorithm first converts the thesaurus into a directed acyclic graph changing this difficult problem into a shortest path problem with a graph where we can use an efficient algorithm. We then show that T* can also be used to find the optimally pruned decision tree.

key words Machine Learning, Structured Attributes, Generalization, Thesaurus, Corpus, Machine Translation

1 はじめに

自然言語処理に利用するための規則をコーパスから学習する研究が最近盛んになっている．これらの研究では，得られた規則の適用範囲をいかに一般化するかが大きな課題となる．なぜなら，コーパスから直接学習される規則は語形を利用したものであり，そのままでは適用範囲が極端に狭いからである．

現在はこの問題を解決するためにシソーラスを利用した手法が試みられている．この場合，シソーラス上のどの概念で規則を一般化するかを求めなくてはならない．しかし，シソーラス上のノードの選び方は，組合せ的に爆発を起こすためその最適な組合せを求めるのは容易ではなく，その最適解を求める手法は従来知られていなかった．

本稿では，この問題を組合せ最適化問題として定式化し，これを線形時間で解く基本的なアルゴリズムを提案する．本稿の問題は一般的に言うと帰納学習の分野で問題とされていた「構造化属性」の問題である．さらに，決定木の最適部分木を求める問題とも等しい．このため本アルゴリズムは自然言語処理の問題に限らず広範囲に応用可能である．

2 問題の定式化

本稿では表 1に示した “take” の翻訳規則をシソーラスを使って一般化する問題を例題とする．表 1は動詞 “take” の目的語の語形が与えられたときに “take” の訳語を決める単純な翻訳規則である[1]．またこの表は，機械学習で使う (属性，属性値，クラス) 形式データの基本単位である．このため，学習に使う最も基本的な言語データとも言える．

表 1の翻訳規則は，目的語の部分に語形そのものが記述してあるため適用範囲が非常に狭い．これに対処するには，目的語の語形のかわりにシソーラスの上位概念を使用することが考えられる．例えば，表 1の語形をシソーラスで照合した結果，図 1のような部分的なシソーラス Tが得られたとする．そうすると，この部分シソーラス上の適切な上位ノードで表 1の語形を置換することで翻訳規則の一般性を高めることができる．このときなるべく上位概念で置換したほうが規則の一般性を高めることができる．しかし，過度に一般化すると表 1の訳語との矛盾が生じてしまう．求めたいのは適度な一般化である．

ここで一般化の適切さを定量的に議論するために，

[1] 正確には翻訳規則の一部である．

表 1: “take” の翻訳規則

目的語 (属性)	訳語 (クラス)
boy	連れていく
her	連れていく
dog	連れていく
cat	連れていく
lion	運ぶ
lily	持っていく
rose	持っていく

部分シソーラス Tの各ノード pに次のようなスコアを考える．

$$S(p) = -G(p) + E(p) \tag{1}$$

式 (1) の右辺第 1 項は p のシソーラス上での一般化の度合いを，第 2項は pによるデータの一般化によって生じるエラーの度合いを表す関数とする．各関数の具体的内容はさまざま考えられるが，本稿では一例として次のような関数を採用する．

$L(p)$ を p の支配するリーフ総数を与える関数とする．また $C(p)$ を p の支配するリーフに対応する図 1の最大数の訳語の数を与える関数とする．そして，

$$G(p) = L(p) - 1 \tag{2}$$

$$E(p) = L(p) - C(p) \tag{3}$$

とする．式 (2) は p をリーフにした場合に減るリーフの数であり，上位ノードほど，すなわち一般性の高いノードほど大きな値となる．式 (3) は p の支配下のリーフに対応する訳語すべてを，支配下の最大数の訳語で置き換えた場合に発生する誤り数である．この結果，式 (1) の値が小さなノードほど好ましいノードとなる．図 1の数字はこのスコア値を表す．

次に一般化の手法を規定する．表 1の一般化のためには，目的語の語形を置き換える T上のノードの集合を求めなくてはならない．本稿では最も基本的な一般化と考えられる，リーフの単語を「もれなく」，かつ「重なりなく」支配するようノード集合での置換を考える．これは形式的には次の 2 つの定義の条件を満たすノード集合となる．

r	部分シソーラス Tのルートノード
p	Tの任意のノード
N	ノードの全体集合
$\hat{L}(p)$	p の支配リーフの集合

定義 1
ノードの集合 $P = \{p_i \mid p_i \in N, i = 1, \ldots, k\}$が与えられ，$\forall p_i, p_j \in P, (i \neq j)$ に対して，$\hat{L}(p_i) \cap \hat{L}(p_j) = \phi$であるとき Pを一意被覆という．

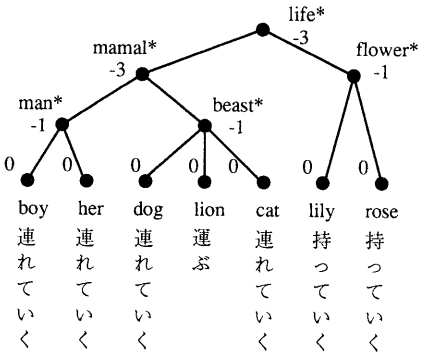

図 1: 部分シソーラス

定義 2

ノードの集合 $P = \{p_i \mid p_i \in N, i = 1, \ldots, k\}$が与えられ，$\cup_{i=1}^{k} \hat{L}(p_i) = \hat{L}(r)$ を満たすとき Pを完全被覆という．

また，この 2 条件を満たすノード集合を一意完全被覆なノード集合と呼ぶ．

以上の準備によって，解くべき問題は次のように記される．

最適一般化問題

コストが各ノードに与えられた木において，一意完全被覆なノード集合の中で，ノードのコストの合計が最小になるようなものを求めよ．

このようなノード集合の数は一般に膨大で，深さ 3 の 10 分木の場合には 1.28×10^{30}程度となる[2]．

3 アルゴリズム T*

前章で定式化した最適一般化問題を高速に解くアルゴリズムを示す．このアルゴリズムの本質は，木を一旦グラフに変形して，問題をグラフの最短経路問題に定式化するところにある．

3.1 アルゴリズム本体

まず，グラフ作成部分の説明を行う．先に使った記号以外では以下を使う．

[2]深さ n の k分木の場合，ノード集合の総数は $a_1 = 2, a_n = (a_{n-1})^k + 1$ の漸化式で表される．

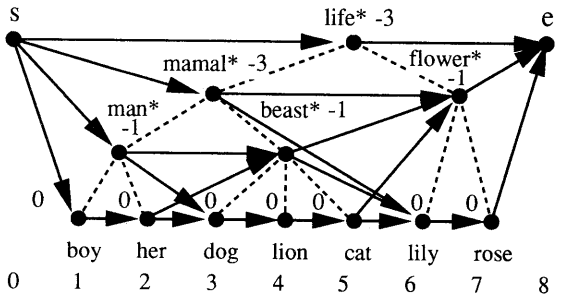

図 2: 横断路グラフ

n	Nの要素
$L_p(n)$	n が支配する最左リーフノードの番号
$R_p(n)$	n が支配する最右リーフノードの番号
s	開始ノード
e	終了ノード

尚，木のリーフノードには 1 から m までの番号を付与し，番号でリーフを参照する．

ステップ 1

始め

開始ノード s と終了ノード e を生成する．

開始ノードには番号 0，終了ノードには番号 $m+1$ を付与する．

終わり

ステップ 2

始め

すべてのノード $n \in N \cup \{s\}$から式 (4) の接続条件を満足するノードの集合 Hに向けて有向辺を張る．

$$H = \{x \mid x \in N \cup \{e\}, L_p(x) - 1 = R_p(n)\} \quad (4)$$

終わり

以上のアルゴリズムによって図 1に示した木から図 2の有向グラフが生成される．

本稿では図 2のグラフを横断路グラフ，また横断路グラフの s から e に向かう各経路を横断路と呼ぶ．さらに，各横断路上にあるノードの集合 (ただし s, e は除く) を横断路ノード集合と呼ぶ．横断路グラフについて次の 2 つの命題が成立する．

命題 1 横断路グラフは非循環有向グラフである

命題 2 一意完全被覆なノード集合の全体集合と横断路ノード集合の全体集合は同一である

命題 2 が成り立つので，最適な一意完全被覆のノード集合を求める問題は，横断路グラフ上の s から e

表 2: 最適に一般化された "take" の翻訳規則

目的語	訳語
mamal*	連れていく
flower*	持っていく

への最小スコアの横断路を求めてその横断路ノード集合を求める問題となる[3]. このための最短経路アルゴリズムには既存のものが利用できる. 尚, 以上の手続きによって最適な一意完全被覆のノード集合を求めるアルゴリズムを T*と呼ぶ.

先に示した図 2の場合は

$$s \rightarrow mamal* \rightarrow flower* \rightarrow e$$

が最小コスト経路となり, 表 1の語形を置換するノード集合として, {mamal*, flower* } が求められる. 表 2に最適に一般化された翻訳規則を示す.

3.2 アルゴリズムの説明

紙面の都合上, 詳しい証明は省略するが, アルゴリズムおよび命題 1,2 が成り立つ理由を直感的に説明する.

一意完全被覆なノード集合を作るには, ノードを勝手に組合せることはできない. 例えば図 2のノード "man*" をノード集合の要素とした場合には, ノード "man*" の支配下のリーフを支配するようなノード, 例えば "mamal*" を要素にはできない (一意被覆に反する). またリーフ $R_p(man*)+1$, すなわち "dog" を支配するノードを要素に選ぶ必要がある (完全被覆のため). 以上のことより, ノード "man*" を一意完全被覆なノード集合の要素とした場合, 接続条件 (4) を満たす "dog","beast*" ノードのどちらかを要素にする必要があることが理解されよう. 横断路グラフは, このような関係にあるノード間に有向辺を張ったグラフであり, 一意完全被覆のノード集合のすべてを尽くしている.

また, この接続条件に従うと, 有向辺の方向はノードの支配リーフの番号が増加する方向のみに制限されているため, 非循環グラフになることも理解されよう.

[3]通常の最短経路問題では辺にコストが付与されている. 現在の問題の場合ではノードにのみコストがある最短経路問題を解くことになる.

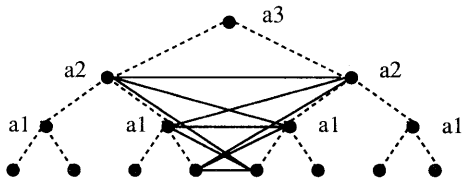

図 3: 2 分木の横断路数

4 計算量

木構造上の最適ノードを求める問題に T*を適用した場合の計算量を評価する. この計算の本質部分は横断路グラフを作成する部分, および最短経路問題を解くことにある.

前者の計算量は, 横断路グラフの辺の数のオーダーの計算量となる. また, 非循環グラフの最短経路問題については, 辺の数のオーダーで求められるアルゴリズムが知られている (Gondran 84). このため T*の計算量を評価するには横断路グラフの辺の数を評価すればよい. 本章ではシソーラスとして k分木を考え, この横断路グラフの辺の数を計算して T*の計算量を評価する.

この問題を考察するために, 2 分木を例に考え方を説明する. 尚, 以下の議論では s, e ノードは含めない.

段数 $n(n=0,1,2,\ldots)$ の横断路グラフの辺の総数は左右の対称な $n-1$ 段の横断路グラフの辺と, 中央の部分の n^2本の辺の合計で求められる. 図 3に a_3 の例を示す. つまり, 段数をサフィックスとして, 第 n 段の 2 分木の横断路グラフの辺の総数は,

$$a_1 = 1,\ a_n = 2a_{n-1} + n^2 \qquad (5)$$

で表される. 同様な考え方で k分木の n 段の横断路グラフの辺の総数は,

$$a_1 = k-1,\ a_n = k \cdot a_{n-1} + (k-1)n^2 \qquad (6)$$

となり, これを解いて一般式

$$a_n = \frac{k(k+1)}{(k-1)^2}k^n - n^2 - \frac{2k}{k-1}n - \frac{k(k+1)}{(k-1)^2} \qquad (7)$$

を得る.

ここで, 一般化したいデータの数であるリーフの総数 $s(s=k^n)$ で辺の数を評価する. そうすると式 (7) の最大オーダーである第 1 項は $\frac{k(k+1)}{(k-1)^2}s$ となり, 横断路グラフの辺の数はリーフの数の線形オーダーで押

さえられる．すなわちT*の計算量はリーフの数の線形オーダーとなることが示された．

5 自然言語処理との関連

本章では，自然言語処理の分野での規則の一般化を取り扱った最近の研究と本研究の比較を行う．次に自然言語に応用する場合のT*の問題点について述べる．

5.1 従来研究との比較

自然言語処理での一般化は，語形をあるデータ構造に従ってまとめる操作と見なすことができる．本稿ではこの構造のことを汎化構造と呼ぶ．

本研究と同じく木を汎化構造に使う研究としては(野見山 93)や，(李 94)，(Almuallim 94)などの動詞の格フレームの一般化の研究がある．(野見山 93)，(李 94)では，(属性，属性値，クラス)形式の学習データ中の属性値(語形)をシソーラスで照合して部分シソーラスをまず作成する．そしてその後，このシソーラス上の適切な上位ノードで単語を置換する方針を取っている．すなわち基本的には本研究と同じアプローチである．

しかし野見山は，ノードに統計的な評価値を付与しているが，ノードの選択問題を組合せ最適化問題としては捉えずに解いているために解の最適性は不明である．ただし，後で述べるようにイディオムの問題を重視しており，最適化すべきでないデータを認識する能力を持たせている点で優れている．

一方，(李 94)では，ノードの選択の問題を本研究と同じく最適な横断路[4]を求める問題として定式化している．本研究との違いはスコアに相当する量にMDL原理に基づいた量を採用している点と，横断路を求めるアルゴリズムとして，2つの準最適アルゴリズムを提案している点である．李が提案するスコアがノードのスコアの合計値として求められるものであれば，問題設定は本研究と全く同じになるため，直ちにT*によって最適解を求めることができる．もし現在のスコアで不可能であるにしても，MDLの評価関数の設計には自由度があるため，多少の変更で最適解が求められるようにできるであろう．

(Almuallim 94)も上記2研究と本研究と同じ形式の入力と木を汎化構造として使う．この研究では，まず(属性，属性値，クラス)形式の学習データの属性値(単語)をシソーラスを使って展開し，ビットベクトルに変換する．この後，決定木アルゴリズム(ID3)を使って局所最適なビット群を選択させている．このため上記の2つの研究や本研究で実現している，一つの属性(格要素)におけるシソーラスの階層の混在を自由に許す一般化にはなっていないと思われ，一般化の柔軟性に欠ける．むしろ後に述べるシソーラスの錯綜に対する問題意識の方が強いと思われる．

以上，関連研究との比較を行ったが，いずれの研究とも基本的には問題設定は似ている．本研究はこの問題を組合せ最適化問題として定式化し，その最適解を求めることに成功している．このため，情報処理の問題としては最も分かりやすい手法であると思われる．ただし，T*を実際の問題に使うには以下の問題を考える必要がある．

5.2 問題

錯綜

これまでの説明ではシソーラスと木を等価のものと見なしてきたが，シソーラスとして錯綜階層を考えることができる．錯綜階層とは，親ノードへのリンクが複数許された階層で，シソーラスに複数の視点を混在させたい場合などに使われる．この錯綜階層を言語データの汎化構造として積極的に利用した研究はこれまでのところ見当たらない．これは，錯綜階層は複雑で取り扱いにくいことと通常の大きなシソーラスが木構造を基本としているためである．しかし，通常のシソーラスは単語のレベルとその直接の上位概念で錯綜しており，正確には錯綜階層となっている．従来この錯綜は意味のある錯綜というより，単語の曖昧性の問題として捉えられており，解消すべく努力がなされている．例えば先の(Almuallim 94)のビットへの展開はこの現れである．T*は錯綜階層に対しては直接的には使えないため，なんらかの対処が必要になろう．これは今後の課題である．

イディオム

言語の大きな特徴は，一般化できない特殊なデータの存在である．T*は一意被覆であるため，あるノードの下のリーフを特別扱いする操作はできない．このため，なんらかの特殊処理が必要となる．これについては野見山が行ったような事前に一般化できないデータを統計量に基づいて認識して除外する手法が有効であろう．

複数属性

本研究で一般化に使った翻訳規則の例は，格要素を一つしか含んでいない．現実の問題に適用する場

[4](李 94)ではシソーラスのカットと呼んでいる．本稿ではグラフ理論のカットとの混同を恐れて横断路とした．

表 3: 計算量比較

	求める解	計算量
CART	準最適部分木系列	$O(s\log s) \sim O(s^2)$
OPT	最適部分木系列	$O(s^2)$
T*	最適部分木	$O(s)$

合には複数の格要素を取り扱う必要がある．これについてはいろいろな拡張方法があろう．次章で述べる決定木アルゴリズムとの組合せもその一つの例である．

6 その他の関連研究

本研究は，決定木アルゴリズムを利用して機械翻訳用の規則を学習しようとする著者の研究 (田中 95) との関連で行われている．決定木アルゴリズムでは表 1のような (属性，属性値，クラス) 形式のデータ (複数の属性を使う) を利用するため，構造を持った属性の場合は本稿と同じ問題が発生する．すなわち「構造化属性」の問題である (Quinlan93)．T*はこの基本部分を解決しており，決定木アルゴリズムと組み合わせることで，この問題を解決することができると思われる．

さらに 2 章で定式化した問題は，CART の決定木の最適部分木を求める問題 (Breiman84) と同一である[5]．この問題は最適解を求めることが困難な問題として知られており，CART では準最適解を求めるアルゴリズムを提案している．

従来，この問題を解くには，さまざまな大きさの部分木 (部分木系列) に対して，それぞれの最適解を探す方針がとられていた．CART ではすべての部分木を尽くせないため，最適部分木が求められない問題があった．これに対して，すべての大きさの木を尽くすアルゴリズム OPT が (Bohanek 94) で提案されている．このため OPT は T*の解を含めすべての大きさの部分木の最適解を求める能力がある．ただし計算量は $O(s^2)$ である (s は k分木のリーフ数)．決定木の枝刈りの問題の場合でも最終的には最適な部分木を求めることが目的であるため，T*を使った方が効率が高く有利である．表 3に枝刈りプログラムと T*の比較を示す (Bohanek 94)．

最後に T*の適用可能な問題について記す．本稿では木構造を汎化構造に利用したが，この他の汎化構造を利用することもできる．例えば意味カテゴリ (単語集合の直和分割) を木の代わりに汎化構造に利用できる．また，森を使うことも問題ない．

さらに，ある特殊なグラフを利用すると離散数値データの最適区間分割を $O(N^3)$(Nはデータの個数) で求めることが可能なことが分かっている．この応用については本稿の主旨からはずれるため，いずれ稿を改めて報告したい．

[5]本稿の部分シソーラスを決定木とみなし，コストにテストサンプルでの決定木の各ノードの分類エラー使ったとする．

7 おわりに

本稿では，翻訳規則をシソーラスで一般化する問題を例に取り，これを最適に一般化するシソーラスのノードを求める手法を提案した．またその計算量はシソーラスのリーフノードの数の線形オーダーであることを示した．

さらに，本アルゴリズムと他の研究との比較を行った．この中で本アルゴリズムは最適な決定木の枝刈りを求めるアルゴリズムであることも示した．

今後は，実際の言語データを使った一般化実験，決定木アルゴリズムとの結合などを実施していく予定である．

参考文献

(田中 95) 田中 英輝, 動詞訳語選択のための「格フレーム木」の統計的な学習, 自然言語処理, vol.2. No.3, 掲載予定, (1995)

(野見山 93) 野見山 浩, 事例の一般化による機械翻訳, 情報処理学会論文誌, Vol. 34, No. 5. pp.905–912, (1993)

(李 94) 李 航・安倍 直樹, シソーラスと MDL 原理を用いた格フレームの一般化, 自然言語処理における学習シンポジウム予稿 (鎌倉), pp.1–8, (1994)

(Almuallim 94) H. Alumallim et.al., Two Methods for Learning ALT–J/E Translation Rules from Examples and a Semantic Hierarchy, proc. of Coling94, pp.57–63, (1994)

(Bohanek 94) M. Bohanek and I. Bratko, Trading Accuracy for simplicity in Decision Trees, Machine Learning, 15, pp. 223–250, Kluwer Academic Press, (1994)

(Breiman 84) L. Breiman et.al, Classification and Regression Trees, Chapman & Hall, (1984)

(Gondran 84) M. Gondran and M. Minoux, Graphs and Algorithms, John Wiley & Sons, (1984)

(Quinlan 93) J. R. Quinlan, C4.5 Programs for Machine Learning, Morgan Kaufmann, (1993)