社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技法
TECHINICAL REPORT OF IEICE.
KBSE97-25 (1998-01)

文書から概念を獲得するための文法推論

花川賢治　　武田英明　　西田豊明

大阪府立工業高等専門学校
電子情報工学科

奈良先端科学技術大学院大学
情報科学研究科

〒630-02 大阪府 寝屋川市 幸町 26–12
E-mail hanakawa@ecs.osaka-pct.ac.jp

あらまし 本研究は、電子化された大量の文書から知識を獲得することを目的とする。文書には2種類の知識の形態が存在する。一つは、自然言語理解により獲得できるような明示的な表現による知識であり、もう一つは暗黙的な制約として現れる知識である。我々は後者の知識を大量の文書に対する文法推論により概念の構造として獲得方法について述べる。本稿では論理に基づく推論ルールを使った文法推論システムを提案し、このシステムが簡単な記号列を学習できることを確認する。さらに、このシステムを実際の文書から知識を獲得するのに応用することについて論じる。

和文キーワード 文法推論, 知識獲得, 論理プログラミング, 概念学習

Grammatical Inference for Concept Acquisition from Documents

Kenji Hanakawa　　Hideaki Takeda　　Toyoaki Nishida

Osaka Prefectural College of Technology

Nara Institute of Science and Technology

26–12 Saiwai-cho, Neyagawa, Osaka
E-mail hanakawa@ecs.osaka-pct.ac.jp

Abstract The purpose of this study is to acquire knowledge from large scale natural language documents. There are two types of knowledge in the documents. One is explicitly represented knowledge which is acquired using natural language processing. The other is implicit constraint. In this paper, how to acquire implicit constraint using grammatical inference from the documents is described. We propose a grammatical inference system which uses inference rules based on logic, and show that the system can learn easy pattern of character lists. We also discuss its application to knowledge acquisition from real documents.

Key words grammatical inference, knowledge acquisition, login programming, concept learning

1 はじめに

近年、情報の電子化が進み、その有効利用が盛んに模索されている。そのような技術として、大規模データベースから知識を自動獲得するデータマイニングの研究 [4] と並び、大規模文書から知識を自動獲得する研究が重要である。文書には自然言語処理で獲得できるような明示的に表現された知識と、直接解釈できない暗黙的な制約の 2 種類の形態の知識が存在すると考えられる。本稿では、この暗黙的な制約を文法推論を使って獲得することについて述べる。

データベースが一様の構造が与えられているのに対し、文書の個々の自然言語文はまちまちな構造を持つ。そのため、文書からの知識獲得にはデータベースからの知識獲得で応用された手法は必ずしも有効ではないと考えられる。自然言語文書から知識獲得を行うためには、個々の文の構造を解析するための文法が必要になる。つまり文法は知識獲得の際必要となる道具の一つと考えられる。

しかし、本稿では少し視点を変えて、文法そのものが獲得された知識であると考える。本稿での文法は一般的な意味での文法よりも広い範囲を指す。通常使われる狭い意味での文法は、統語カテゴリが基本となる。それに対し本稿での文法は、統語カテゴリではなく、もっと具体性の高いカテゴリも含める。たとえば、「名詞句」だけでなく、「動物」もカテゴリとして扱う。このように文法を拡大解釈すれば、文法に現れる非終端記号は概念に対応し、文法を獲得することは、人間の知識の一部分である概念体系を獲得することに他ならない。

文書から文法を学習するということは、互いに置き換えても文書として成立するような単語列を発見し、それらを同じ非終端記号に含めるという操作を繰り返すことである。文法推論が進むにつれ非終端記号は一般化され、対応する単語列の集合はしだいに大きくなる。つまり、我々はこのような非終端記号を概念であると考える。文法推論の過程で、異なった一般化のレベルが得られるので、概念の上下関係 (IS-A の関係) についても調べることができ、概念体系を得ることができる。

文法推論と概念獲得の関係を図 1 に示す。文のレベルを上の平面、概念のレベルを下の平面とする。例文を L_0 で、例文に含まれるすべての単語からなる集合を W^* であるとする。文法推論は L_0 を包含する L_1、L_2 を生成する過程である。文法推論により、文法が例文 (S_0) と類似した文 (S_1,S_2) を受理するできるようになるのは、文法が単語を扱う規則から、類似した

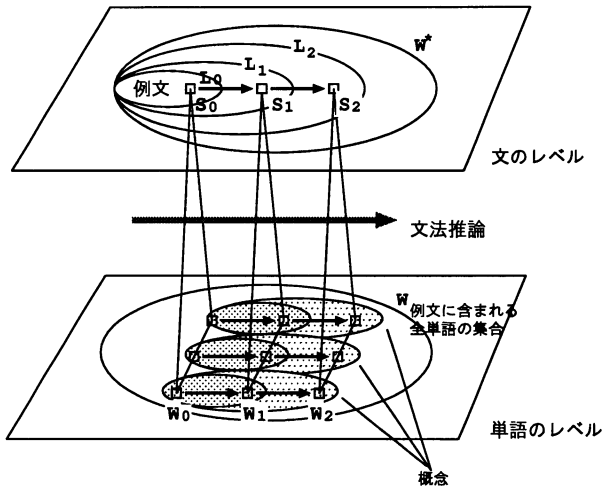


図 1: 文法推論と概念の獲得

単語の集合 ($\{W_0, W_1, W_2\}$) を扱う規則に変わるからである。この類似した単語の集合が概念であると考えることができる。単語の集合の包含関係は概念の IS-A 関係になる。実際には、単語だけでなく単語列のレベルでも同様のことが言える。

これまでの研究で正規言語および制限された文脈自由言語を対象とした文法推論については、理論的な背景を持った推論テクニックが提案されている [1]。しかし、自然言語を対象とした実用的な文法推論に使用できるような理論的に十分に研究されたテクニックは存在しない。実用的な効率で大量の文書から学習させるシステムを構築するには、部分的にヒューリスティクスに頼らざるを得ない。

そこで、本稿では、ヒューリスティクスの発見を促すような論理に基づく推論ルールの表現を使った文法推論システムについて検討を行う。

以下、2 節で単純な概念体系の利用方法について述べ、概念体系を自動獲得することの意義を明確にする。3 節で文法推論ルールと文法推論の実行例について述べる。4 節で実際に文書から概念構造を獲得するために解決すべき課題について述べる。

2 概念体系

ここでは、概念体系の利用方法について述べ、獲得すべき概念体系がどのようなものかを明確にする。

概念を体系化したものとして様々なものが存在する。古くから、辞書、シソーラスなどの形態で語彙を整理することが行われてきた。最近では、知識システムを構築する上での基本としてオントロジー [6] が研

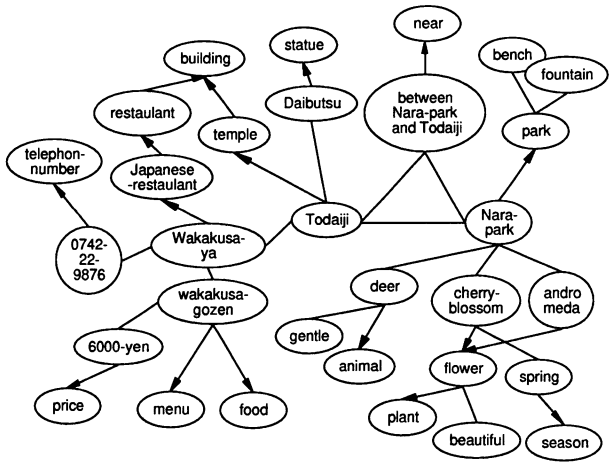


図 2: 連想グラフ

究されている。また、大規模知識ベースの研究 [7][2] では大量の知識を一階述語論理のようなコンピュータの推論に利用できる形式で収集することが試みられている。このような概念体系は人間が持つ知識をコンピュータに利用できる形態で表現したもっとも基本的なものであると考えられる。概念体系は分野、個人により微妙に異なり、時代とともに変化する。従来、概念体系の構築は人手でなされていたが、人手に頼る限り多様な概念体系のニーズに対応することは不可能であり、コンピュータによる自動構築が強く求められる。

自動構築を行うには概念体系の表現形式は単純であるほうが望ましい。我々は非常に単純化した概念体系として、弱構造化情報ベースについて研究を行ってきた [3]。弱構造化情報ベースでは、概念をノード、概念間の関係をリンクとする連想グラフという知識表現形式を使用した。連想グラフのリンクは概念間の上下関係を表す IS-A リンクと上下関係以外の概念間の関係を表す水平リンクの 2 種類を使用する。一般的な意味ネットワークとは異なり、水平リンクにはラベルを付けない。

連想グラフによる情報を構築は極めて容易である。情報の構築者は概念間に関連性があるかどうか調べ、上下関係があれば IS-A リンク、それ以外の関係があれば水平リンクで概念間を結ぶ。図 2は筆者が奈良の観光ガイドブックの一部分を見て、その内容を連想グラフ化した例である。

一般に、人間とコンピュータの間のコミュニケーションには何らかの言語が使用される。言語を厳格にすると、コミュニケーションに要する労力が増大する。弱構造化情報ベースは、厳格でない言語として

表 1: 弱構造化情報ベースの質問と回答

質問文	回答
聖徳太子はいつ寺を建てたか	法隆寺 607 年 法隆寺の建立
東大寺近辺のバスの停留所は	バス停大仏殿前 バス停
東大寺と興福寺の距離は	914 m 東大寺と興福寺の距離
興福寺から 1000m 以内の駅	309 m 近鉄奈良駅
塔が 30m 以上ある寺は	興福寺の五重塔 51 m 興福寺
4 月上旬に咲く花	4 月上旬 馬酔木
池のあるハイキングコース	くろんど池 私市府民の森コース
5000 円以上のメニュー	8000 円 塔の茶屋の茶懐石

単語列を採用し、人間にとって容易に利用できるインターフェースを提供した。

厳格さを除きすぎると曖昧性が増し正確に情報が伝わらないという問題が生じる。弱構造化情報ベースでは、構造的な曖昧性を連想グラフを使用して解消する方法を使って、実用レベルの正確さを実現している。表 1は弱構造情報ベース上で行った質問と回答の例である。表の左の列の簡単な自然言語の質問に対しコンピュータが人間の質問意図に合う回答を返している。

すなわち、弱構造化情報ベースは、表現の厳格さを取り除くことで、情報の組織化と利用の両面での簡便性を高めた。弱構造化情報ベースの研究により、概念と概念の関係を単純に記述するというような低度な概念の体系化でも有用性があることがわかった。低度な概念の体系化は低コストで大規模に行うことが可能であるので、早期に大きな利益を得ることが期待できる。そのためにまず重要になるのは、概念間の単純な関係を文書から自動的に抽出する技術である。

3 文法推論による概念体系の獲得

文法推論とは、言語の学習とも呼ばれることもあるが、帰納推論の一種で、与えられた例文からそれらの文の文法規則を推論することをいう。文法推論は、正例（文法的に正しい例文）だけからの学習と、正例と負例（文法的に誤った例文）の両方を使った学習に大別される。大量の文書から学習を行わせるばあい、現実問題として負例を収集するのは難しいので、ここでは正例だけからの学習を用いることにする。

過去の研究で、正例のみからの学習は唯一の言語を決定できないということが明らかにされ、文法の学習における負例の必要性が認識されるようになった。しかし、我々は学習されるべき言語が一意であることを要求しない。応用面を考えると、要求される文法の一般化のレベルは一定ではない。たとえば、ある文が構文的に正しいか否かを判定させるばあいには一般

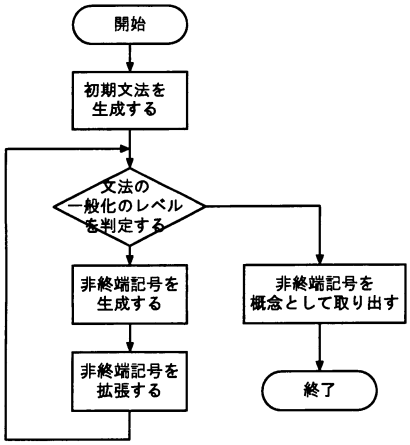


図 3: 文法推論のサイクル

化が高い文法が必要になるが、意味のある文か否かを判定させるばあいにはそれよりも一般化の低い文法が必要になる。したがって、目的に応じて一般化の程度を任意にコントロールできればよいと考える。

本稿で提案する文法推論方式は図 3のような手続きである。最初に、例文だけを受理する初期文法を生成する。次に、文法を一般化する非終端記号の生成と拡張を繰り返す処理を行う。このサイクルが回るたびに、非終端記号の一般化が進み、それに対応する単語列の集合が大きくなり、文法が受理する言語も大きくなる。必要なレベルまで一般化がなされたと判断したら、このサイクルを停止する。その時点でそれぞれの非終端記号はある概念を表し、非終端記号に書き換えられる記号列はその概念の下位概念を表すと予想される。

文法推論の結果から連想グラフを構築する方法を簡単な例を使って説明する。「ポチ 吠える」という文が与えられたときに、まず「犬 吠える」、次に「動物 動作」と文を一般化したとする (図 4の左側)。このばあい、ポチ→犬→動物 という一般化と、吠える→動作 という一般化が行われたので、それぞれを IS-A リンクで表す。一方、ポチ–吠える、犬–吠える、動物–動作 は概念間の関連を表すので水平リンクで表す (図 4の右側)。多数の文について同様のことを繰り返すことで連想グラフが構築される。

4 論理に基づく文法推論

文法推論は、書き換え規則の書き換え規則を使って議論することが多いが、本稿では、より明解な表現形式として論理に基づく表現 [5] を用いる。Perira と Warren は、Prolog プログラムの基本要素である

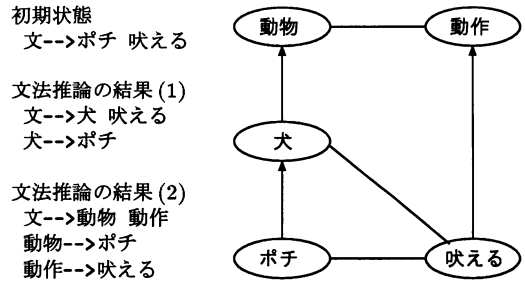


図 4: 文法推論と連想グラフ

表 2: 生成規則の表記

書き換え規則	Prolog	引数を省略した Prolog
$S \rightarrow NP\ VP$	$s(X,Z) :- np(X,Y), vp(Y,Z).$	$s :- np, vp.$
$NP \rightarrow DT\ N$	$np(X,Z) :- dt(X,Y), n(Y,Z).$	$np :- dt, n.$
$VP \rightarrow V\ NP$	$vp(X,Z) :- v(X,Y), np(Y,Z).$	$vp :- v, np.$
$N \rightarrow boy$	$n(X,Y) :- boy(X,Y).$	$n :- boy.$
	$boy([boy\|X],X).$	boy.
$N \rightarrow dog$	$n(X,Y) :- dog(X,Y).$	$n :- dog.$
	$dog([dog\|X],X).$	dog.
$DT \rightarrow the$	$dt(X,Y) :- the(X,Y).$	$dt :- the.$
	$the([the\|X],X).$	the.
$DT \rightarrow a$	$dt(X,Y) :- a(X,Y).$	$dt :- a.$
	$a([a\|X],X).$	a.
$V \rightarrow beats$	$v(X,Y) :- beats(X,Y).$	$v :- beats.$
	$beats([beats\|X],X).$	beats.

definite clause(確定節) が構造的に文脈自由文法の文法規則とまったく同一であることに注目し、DCG を提案した。DCG は正規言語でのグラフ表現 (決定性オートマトン) と同様に、書き換え規則と同一の表現能力を持つ直感的にわかりやすい形式である。

DCG では生成規則は Prolog の規則で記述される。非終端記号に相当するものは規則の左辺の述語に相当する。終端記号は事実で記述される。DCG では s という開始記号に相当する述語を証明する過程で文の解析を実行する。Prolog の表記に従うならば、生成規則を構成する述語は 2 個の変数を持たねばならない。また、終端記号を表す事実の第一引数は単語の文字列をリストに含む。しかし、規則の述語の変数は固定であるし、事実は述語名を単語と一致させることにすると、引数は表記する必要はないので、本稿では引数を省略することにする。表 2に書き換え規則、Prolog、引数を省略した表記例を示す。

論理に基づく文法推論は、まず例文だけを証明する初期文法に相当する節集合を生成した後、節集合の要素を書き換える操作を繰り返す処理である。

初期文法の生成段階では、文法が与えられていない状態なので文の構造が決定できない。そこで、可能性のあるすべての構造を受理するような初期文法を

$p1:-boy,p3.$	$p1:-p8,p5.$	$p1:-p9,dog.$
$p2:-the,boy.$	$p3:-beats,p5.$	$p3:-p7,dog.$
$p4:-p2,beats.$	$p4:-the,p8.$	$p5:-a,dog.$
$p6:-p2,p7.$	$p6:-p4,a.$	$p6:-the,p9.$
$p7:-beats,a.$	$p8:-boy,beats.$	$p9:-boy,p7.$
$p9:-p8,a.$	$s:-p2,p3.$	$s:-p4,p5.$
$s:-p6,dog.$	$s:-the,p1.$	$the.$
$boy.$	$beats.$	$a.$
$dog.$		

図 5: 初期文法

生成する。また、一般に生成規則は右辺は任意の数だけあってよいが、ここでは二つに限定する。

初期文法の生成処理は以下のように行う。

1. すべての単語に対応する事実を生成する。本来述語名は任意であるが、先に述べたように単語と事実の述語名を一致させる。

2. 例文に含まれる全ての部分文字列に対応する述語を定義する。ただし、文全体に相当する述語はsとする。

3. 二つの部分文字列を連結すると一つの部分文字列になるすべての組み合わせについて規則を生成する。

この手順で生成された初期文法は、例文だけを受理する。図5に “the boy beats a bog” という文から生成した初期文法を示す。

文法推論のサイクルは、節集合に対し節の書き換えルールを繰り返し適用することにより行う。節書き換えルールは演繹を反転させたものに相当し、論理的に常に正しい事実を導くとは限らないが、現在わかっている事実を一般化する。文法推論においては、節書き換えルールを適用することにより、文法を一般化し、受理する言語を大きくすることができる。

以下に基本的な 3 種類の節書き換えルールを対称形を省略し簡略化して示す。実際のルールはもう少し複雑で、同様の条件が複数重なったとき起動するようになっている。ルールに現れる A,B,P,Q,X,Y は述語が代入される変数を表す。ルールの上辺はルールが起動される条件を示し、ルールの下辺は変更すべき状態を示す。つまり、ルールが適用されると、上辺にないが下辺にある節は追加され、上辺にあるが下辺にない述語は削除される。

概念の発見

$$\frac{P:-A,X \quad P:-A,Y}{P:-A,Q \quad Q:-X \quad Q:-Y}$$

表 3: 文法推論で段階的に生成される言語（2 回目の推論サイクル後長さ 9 の文は 42 個、長さ 11 の文は 132 個生成された。表にはその一部を載せた。）

長さ	例文	最初の推論サイクル後	2 度目の推論サイクル後
1	a	a	a
3	a+a	a+a	a+a (a)
5	a+a+a	a+a+a	(a)+a a+a+a a+(a) (a+a) ((a))
7	(a+a+a) a+(a+a) a+a+a+a	(a+a)+a a+a+a+a a+(a+a) (a+a+a)	((a))+a (a+a)+a (a)+a+a (a)+(a) a+(a)+a a+a+a+a a+a+(a) a+(a+a) a+((a)) (a+a+a) ((a)+a) (a+(a)) ((a+a)) (((a)))
9		a+(a+a)+a (a+a)+a+a a+a+(a+a) a+a+a+a+a	(((a)))+a ((a+a))+a (a+(a))+a ((a)+a)+a (a+a+a)+a (((a+a))) ((((a))))
11	(a+a)+(a+a)	a+a+(a+a)+a a+(a+a)+a+a (a+a)+a+a+a a+a+a+a+a+a a+a+a+(a+a) (a+a)+(a+a)	((((a))))+a (((a+a)))+a ((a+(a)))+a (((a)+a))+a ((a+a+a))+a (a+((a)))+a ((((a+a)))) (((((a)))))

概念の拡張

$$\frac{P:-A,X \quad Q:-X}{P:-A,Q \quad Q:-X}$$

概念のマージ

$$\frac{P:-A,B \quad Q:-A,B}{P:-A,B \quad Q:-A,B \quad P=Q}$$

{a,(,),+}の 4 種類の文字を使って作られる 7 種類の文を正例として上記の節書き換えルールを使って文法推論させる実験を行った。実験の結果表 4に示す文を生成する文法を得た。一度目の推論サイクル後は、比較的例に似た文が生成されるが、二度目の推論サイクル後は、例と類似性が低い文も生成されるようになり、生成される文の数が多くなっている。なお、偶数の長さの文は生成されなかった。

表 4: 文法推論で獲得した節

最初の推論サイクル後	2回目の推論サイクル後
q2 :- '(a+a)'.	s :- s , '+a'.
q2 :- 'a'.	s :- s , q1.
q2 :- 'a+a'.	q1 :- '+' , s.
q2 :- 'a+a+a'.	s :- q6 , ')'.
q2 :- q2 , '+a'.	q6 :- '(' , s.
q1 :- '+' , q2.	
s :- q2 , q1.	

推論の過程で作られた節の一部を表 4に示す。q1,q2,q6 は文法推論で獲得した非終端記号である。最初の推論サイクルで、同じ前後関係を持つ記号列のグループとして q2 が生まれ、q2 を使用した一般化した規則が追加される。2 回目の推論サイクルで、s と q2 が同一化され、すべての q2 が s に書き換えられる。また、'(' と')' に関係した規則も追加される。

5 考察

前節では、帰納推論に基づく簡単な節書き換えルールにより、段階的な文法推論が実現できることを確認した。前節での文法推論により獲得された非終端記号は意味のある単語列のグループに対応している。したがって、文法推論によりある種の概念を獲得したと考えることができる。

しかしながら、この手法を実際の自然言語で書かれた文書に適用するのには、以下に示す課題がある。

意味のある節の抽出　文の長さが長くなると、指数関数的に節の数が増える。節の数が多いために、推論処理の効率が低下する。また、一つの文を導く節の組み合わせが多数存在するために、文の解析および生成の効率も低下する。そのため、実際の技術文書に現れるような長い文を処理することは不可能である。この問題を解決するために、冗長な節を取り除く処理と推論結果に基づき節の重要度を判別する処理が必要である。

節書き換えルールの最適化　節書き換えルールとして、3 種類の非常に単純なものを提案したが、これらは最適なものではない。節書き換えルールの開発が今後の課題である。

節書き換えルールは論理の形式をしているので、コンピュータによる候補の自動生成および人間の直感的な選択が比較的容易になると考えられる。本研究では、次のステップとして、文法推論のための節書き換えルールの獲得支援システムを計画している。

右辺が 3 述語の節の導入　単純さのために、節は右辺の述語の個数を 2 個以下にするという制限を加えたが、そのために文法推論の能力がどのような制限を受けているか調べる必要がある。さらに右辺が 3 述語の節を導入した場合について検討する必要もある。

既存知識との融合　既に得られている自然言語の文法および概念体系を与えることにより、より精度の高い推論を効率よく行う。基本的には、既存の情報を節の形式で記述して初期文法に加える方法で可能であると考えている。

6 おわりに

文書から概念体系を獲得する手段として、段階的に文法推論を実行する方法を提案した。ヒューリスティクスの獲得が容易になるように論理に基づく推論ルールを使用した文法推論処理を提案し、簡単な記号列の学習できることを示した。今後は実際の文書を使った概念獲得を行っていきたい。

参考文献

[1] K. Fu and L. Booth. Grammatical inference: Introduction and survey – part 1. In *IEEE Transactions on Systems, Man and Cybernetics SMC-5*, pp. 95–111, 1975.

[2] R. V. Guha. Representation of defaults in Cyc. In *Proc. AAAI-90*, pp. 608–614, 1990.

[3] 花川賢治, 武田英明, 西田豊明. 柔軟な問い合わせのための弱構造化表現. システム制御情報学会論文誌, No. 10-7, pp. 341–350, 1997.

[4] 河野浩之. データベースからの知識発見の現状と動向. 人工知能学会誌, No. 12-4, pp. 3–10, 1997.

[5] 松本祐治. 論理型言語による自然言語へのアプローチ. 淵一博（編）, 自然言語の基礎理論, pp. 51–86. 共立出版, 1986.

[6] 溝口理一郎. 知識の共有と再利用研究の現状と動向. 人工知能学会誌, No. 9-1, pp. 3–9, 1994.

[7] 西田豊明. 大規模知識ベース. 情報処理, No. 35-2, pp. 130–139, 1994.

文書から概念を獲得するための文法推論

Grammatical Inference for Concept Acquisition from Documents

花川賢治* 武田英明** 西田豊明**

* 大阪府立工業高等専門学校 電子情報工学科

** 奈良先端科学技術大学院大学 情報科学研究科

文書から概念を獲得するための文法推論

1. 研究の目的
2. 概念体系
3. 概念体系としての文法
4. 文法推論の関連研究
5. 文法推論のアルゴリズム
6. 実験
7. 評価
8. 課題
9. まとめ

研究の目的

大量データからの低度な知識の自動獲得

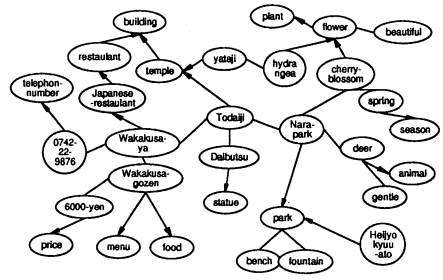
電子化された文書からの概念体系の自動構築

なぜ低度な知識なのか?

構築の自動化の可能性
それなりの有用性
質は劣るが量が豊富
共有と再利用が容易

概念体系

- 概念間の関係を簡単に表現したもの



- 概念体系の利用

  仮定:質問を具体化したものが回答となる。

質問文	回答
聖徳太子はいつ寺を建てたか	法隆寺 607 年 法隆寺の建立
東大寺近辺のバスの停留所は	バス停大仏殿前 バス停
東大寺と興福寺の距離は	914 m 東大寺と興福寺の距離
興福寺から 1000m 以内の駅	309 m 近鉄奈良駅
塔が 30m 以上ある寺は	興福寺の五重塔 51 m 興福寺
4 月上旬に咲く花	4 月上旬 馬酔木
池のあるハイキングコース	くろんど池 私市府民の森コース
5000 円以上のメニュー	8000 円 塔の茶屋の茶懐石

概念体系としての文法 (1)

1. 言語表現

花子　が　ネコ　を 飼う。
太郎　が　イヌ　を 飼う。

2. 言語表現に内在する制約 ⇒ 概念体系

人　　が 動物 を 飼う。
花子は人間である。 太郎は人間である。
ネコは動物である。 イヌは動物である。

3. 狭義の文法

```
S --> NP P NP P V
NP --> 花子        NP --> 太郎
NP --> イヌ        NP --> ネコ
```

文を弱く一般化すると 2 が得られ、
より強く一般化すると 3 が得られる。

概念体系としての文法 (2)

1. 例文

花子がネコを飼う。
太郎がイヌを飼う。

2. 言語表現に内在する制約を満足する文

花子がネコを飼う。
太郎がイヌを飼う。
花子がイヌを飼う。
太郎がネコを飼う。

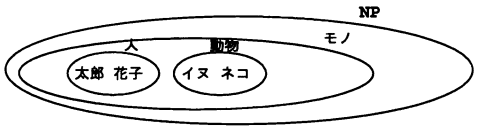
3. 狭義の文法を満足する文

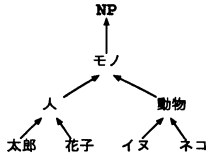
花子がネコを飼う。ネコが花子を飼う。
太郎がイヌを飼う。イヌが太郎を飼う。
花子がイヌを飼う。イヌが花子を飼う。
太郎がネコを飼う。ネコが太郎を飼う。

概念体系としての文法 (3)

文法を獲得する処理により概念体系が得られる。

異なったレベルの一般化を行うことにより、IS-A の関係が獲得できる。





文法推論の理論的研究

正例からの自然言語文法の推論は不可能?

Gold の定理
正の例からだけでは正しい文脈自由文法は学習できない。

Angluin のパターン言語
パターン言語は学習に負の例を必要としないが、自然言語の文法をモデル化できない。

正例からの文脈自由文法の推論

ヒューリスティクスの利用 (Fu,Biermann,Feldman,Klein, Kuppin,Evans,Knobe)

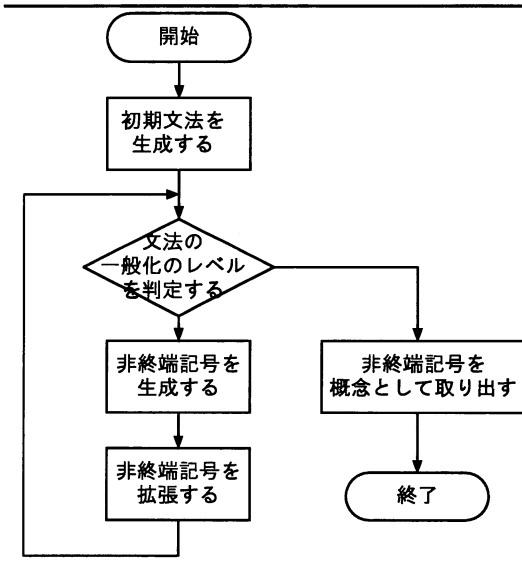
形式的導出語
形式的導出語 (s から始まる文で s に続く文字列の集合) に基づき正規文法を構成することができる。

論理和による一般化
類似した非終端記号のグループをマージする。類似を判定するヒューリスティクスとして、包含関係、k-末尾同値 などを使用する。

繰返しの仮説
X→a, X→aa のような生成規則があるばあいは、X→Xa という再帰的な生成規則を導入する。

短縮代入
s という文字列が生成規則の右辺に出現する回数が多いときには、新しい非終端記号 A をつくって、s をすべて A に置き換え、A→s という生成規則を付け加える。

文法推論のアルゴリズム–文法の一般化サイクル



文法推論のアルゴリズム–初期文法の生成

1. 例文に含まれる全ての部分文字列に対応する記号を生成する。ただし、文全体に相当する記号は s とする。
2. 二つの部分文字列を連結すると一つの部分文字列になるすべての組み合わせについて生成規則を生成する。

例文が “(a+a)” のときの初期文法

```
'(a' :- '(' , 'a'.        'a+' :- 'a' , '+'.
'(a+' :- '(' , 'a+'.      'a+a' :- 'a' , '+a'.
'(a+' :- '(a' , '+'.      'a+a' :- 'a+' , 'a'.
'(a+a' :- '(' , 'a+a'.    'a+a)' :- 'a' , '+a)'.
'(a+a' :- '(a' , '+a'.    'a+a)' :- 'a+' , 'a)'.
'(a+a' :- '(a+' , 'a'.    'a+a)' :- 'a+a' , ')'.
'+a' :- '+' , 'a'.        s :- '(' , 'a+a)'.
'+a)' :- '+' , 'a)'.      s :- '(a' , '+a)'.
'+a)' :- '+a' , ')'.      s :- '(a+' , 'a)'.
'a)' :- 'a' , ')'.        s :- '(a+a' , ')'.
```

文法推論のアルゴリズム–グループの提案

$$\frac{P:-A,X \quad P:-A,Y}{P:-A,Q \quad Q:-X \quad Q:-Y}$$

同様の生成規則に現れる記号 X,Y が存在するとき、新しい記号 Q をつくり、生成規則 $Q:-X$,$Q:-Y$ を加える。

```
s :- 'a+' , a .
s :- 'a+' , 'a+a' .
s :- 'a+' , '(a+a)' .
s :- 'a+' , 'a+a+a' .
        ⇓
s :- 'a+' , q1 .
q1 :- a .
q1 :- 'a+a' .
q1 :- '(a+a)' .
q1 :- 'a+a+a' .
```

文法推論のアルゴリズム–書き換え規則の一般化

$$\frac{P:-X,A \quad Q:-X}{P:-Q,A \quad Q:-X}$$

$Q:-Y$があったとすると、**P:-Y,Q** が追加されたのと同じ効果が生じる。すなわち文法の一般化が行われる。

$\frac{P:-Q,A \text{ にマッチする生成規則の数}}{Q:-X\text{にマッチする生成規則の数}}$ が閾値を越えた時に適用する。

```
s :- a , '+a' .
s :- 'a+a' , '+a' .
s :- 'a+a+a' , '+a' .
```

$\frac{s:-q1,+a \text{ にマッチする生成規則の数}}{q1:-X\text{にマッチする生成規則の数}} = 3/4$

$\Downarrow$

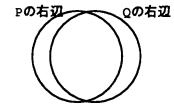
```
s :- q1 , '+a'
```

文法推論のアルゴリズム
– 非終端記号のマージと包含関係の獲得

- マージ

$$\frac{P:-A,B \quad Q:-A,B}{P:-A,B \quad Q:-A,B \quad P=Q}$$

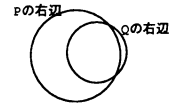
$\frac{P\text{の右辺}\cap Q\text{ の右辺}}{P\text{の右辺}}, \frac{P\text{の右辺}\cap Q\text{ の右辺}}{Q\text{ の右辺}}$ の両方が閾値を越えたとき適用する。



- 包含関係の獲得

$$\frac{P:-A,B \quad Q:-A,B}{P:-A,B \quad Q:-A,B \quad P:-Q}$$

$\frac{P\text{の右辺}\cap Q\text{ の右辺}}{P\text{の右辺}}, \frac{P\text{の右辺}\cap Q\text{ の右辺}}{Q\text{ の右辺}}$ の片方が閾値を越えたとき適用する。



文法推論のアルゴリズム
– 構造データの付与と既知文法との融合

- 構造データの付与

  文の構造を示すことにより、初期文法の生成に制限を加える

  [if,(,a,),a,+,a,;] ⇒ [[if,(,a,)],[[a,+,a],[c]]]

```
s :- 'i(a)' , 'a+ac' .
'i(a)' :- 'i(a' , ')' .
'i(a' :- 'i(' , a .
'i(' :- i , '(' .
'i(a' :- i , '(a' .
'(a' :- '(' , a .
'i(a)' :- 'i(' , 'a)' .
'a)' :- a , ')' .
'i(a)' :- i , '(a)' .
'(a)' :- '(a' , ')' .
'(a)' :- '(' , 'a)' .
'a+ac' :- 'a+a' , c .
'a+a' :- 'a+' , a .
'a+' :- 'a , '+' .
'a+a' :- a , '+a' .
'+a' :- '+' , a .
```

- 既知の文法規則との融合

  初期文法に既知の文法規則を追加する。

アルゴリズムの実験的検証

ある文法により生成された文のいくつかを正例として与え、もとの文法と同等の文法が推論されることを検証する。

- 手順
  1. 制限長以下のすべての文を生成し L とする。
  2. L からランダムに取りだした要素で Sを生成する。
  3. Sを正例として推論を実行する。
  4. 推論で得た文法から制限長以下の文を生成し L'とする。
- 閾値、文法の一般化サイクルの回数は固定とする。
- 評価基準

  $\frac{S\text{の要素数}}{L\text{ の要素数}}$ を小さく設定したとき、

  $\frac{L'\text{の個数}}{L\text{ の個数}}$ が1に近いと正しい文法推論をしたとみなす。

実験 1

- 終端記号 {a,+,(,)}
- 非終端記号 {s}
- 開始記号 s
- 生成規則

$s \rightarrow (s)$
$s \rightarrow a$
$s \rightarrow s+s$

```
(a)+((a))
((a))
a+(a)+a
((((a))))
a+a+a+a
a+((a)+a)
((a)+a)
```

$S = 20$ $S/L = 31\%$

長さ	L	L'	$L'/L(\%)$
1	1	1.0	100
3	2	1.9	95.0
5	5	4.9	98.0
7	14	13.6	97.1
9	42	39.3	93.5
1〜9	64	60.7	94.8
11	132	113.1	85.7

実験 2

- 終端記号 {a,b,+,-}
- 非終端記号 {s}
- 開始記号 s
- 生成規則

$s \rightarrow sos$
$s \rightarrow a$
$s \rightarrow b$
$o \rightarrow +$
$o \rightarrow -$

```
a-a-b+b
b-a+a+b
b-b+b+b
a+b-b
a-a-b
a-a+a
```

$S = 20$ $S/L = 11.8\%$

長さ	L	L'	$L'/L(\%)$
1	2	2	100
3	8	6.6	82.5
5	32	22.3	69.7
7	128	63.7	49.8
1〜7	170	94.6	55.6
9	512	143.3	28.0

実験 3

- 終端記号 {a,+,(,),if,;}
- 非終端記号 {s,ex}
- 開始記号 s
- 生成規則

$s \rightarrow ex;$
$s \rightarrow if(ex)s$
$ex \rightarrow a$
$ex \rightarrow ex+ex$

```
i(a+a+a)a;
i(a+a+a+a)a;
a;
i(a+a)a+a;
a+a+a+a;
i(a)a;
```

誤った文法が推論された。

```
a+if(a)if(a)a;
a+if(a)a+a+a;
a+a+a+if(a)a;
a+a+if(a)a+a;
```

実験 4

例文に部分的に構造を与えて実験 3 と同じ文法を学習させた。

- 終端記号 {a,+,(,),if,;}
- 非終端記号 {s,ex}
- 開始記号 s
- 生成規則

$s \rightarrow ex;$
$s \rightarrow if(ex)s$
$ex \rightarrow a$
$ex \rightarrow ex+ex$

```
i(a+a+a)a;
i(a)[[a+a+a][;]]
a;
i(a+a)[[a+a][;]]
a+a+a+a;
i(a)a;
```

$S = 20$ $S/L = 37\%$

長さ	L	L'	$L'/L(\%)$
2	1	1	50
4	1	1	25
6	2	2	100
8	3	2	67
10	5	4	80
12	8	5	63
14	13	7	23
16	21	6	35
2〜16	54	28	51.8

評価

- 非常に簡単な言語に対しては、例文を生成した文法と類似度の高い文法を獲得することができた。
- 例文を生成した文法と完全に一致する文法を獲得することはまれであった。(文法規則を少なくさせる制約が必要?)
- 本アルゴリズムは言語の特徴を捕らえることには成功していると考えられる。
- 実験 3 は C のような人工言語の学習の可能性を探るために行ったが、構造的な情報を付けないと正しい推論が行われなかった。
- 実験 4 では実験 3 の例文に構造的な情報を加えることにより、誤った推論は行われなくなった。
- 実験 4 では文法の推論は不十分にしか行われなかった。
- 本アルゴリズムは文法が少し複雑になると効果的でなくなる?
- 推論後の生成規則から「概念」をみつけることは容易ではない。

まとめ

- 文法推論を使って概念体系を獲得する方法を提案した。
- 正例から文脈自由文法を推論するアルゴリズムを提案した。
- 本アルゴリズムが非常に簡単な文法の推論に有効であることを確認した。

課題

文法推論能力の向上

- 文法規則書き換えルールの最適化
- 右辺が 3 述語の文法規則の導入
- C プログラム例から C の構文を導き出すことを目標とする

獲得した文法規則から概念を得る方法の開発

- 意味のある文法規則の抽出

自然言語を対象とした実験