人 工 知 能 87－7
（1993. 3. 4）

木パターン学習の自然言語処理のための知識獲得への応用

桜井成一朗† 原口誠† 奥村学‡
†東京工業大学 大学院総合理工学研究科
‡北陸先端大 情報科学研究科

極小汎化アルゴリズムでは，自然な形で選言的な汎化を扱うことができる．本論文では，極小汎化アルゴリズムを自然言語処理で必要となる選択制限を学習する方法について述べる．選択制限の学習においては，選言的な記述が不可欠であるため，極小汎化アルゴリズムが適している．極小汎化アルゴリズムを応用するために，概念階層を用いて訓練例を自ら生成することにより，初期の訓練例が少ない場合でも，汎化を行うことが可能となる．人工的に生成した比較的大規模な概念階層を用いた実験では，極めて効率的に汎化ができることが確認できた．

Seiichiro SAKURAI† Makoto HARAGUCHI† Manabu Okumura‡
†Tokyo Institute of technology
‡JAIST

K-minimal multiple generalization (k-mmg) algorithm, which has proposed by Arimura et al., is an efficient inference algorithm. The k-mmg algorithm can handle disjunctive description naturally. This paper describes an method for learning knowledge for natural language processing. In our method, since examples are generated by using the taxonomic hierarchy, we can obtain general knowledge from a computer dictionary.

1 まえがき

有村ら [1] によって与えられた k-mmg アルゴリズムは木パターン，すなわち項によって表現されたデータの極小汎化を行う汎用のアルゴリズムであり，Plotkin[3] によって与えられた最少汎化の自然な拡張となっている．したがって，概念階層のように木構造によって表現されるデータの汎化アルゴリズムとして用いることができる．k-mmg では，k 個の木パターンによってデータが汎化されるので，自然な形で選言を扱うことができる．本論文では，k-mmg アルゴリズムを自然言語処理で必要となる知識獲得に応用する方法について述べる．

自然言語処理では，選択制限と呼ばれる知識を用いて与えられた文が非文であるかどうかを判定している．選択制限は選言によって与えられる場合があり，概念階層的知識の不備を補うためには，選言的な汎化が必要となる．十分一般的な選択制限を獲得するためには，選言を扱える汎化機構が不可欠である．一般的な汎化機構であるバージョン空間法 [4, 5] に対して選言を扱えるように拡張することも可能であるが，バージョン空間自身が原理的には指数オーダになり得る [9] し，また選言的な記述を得るためには，訓練例の分割が一般に困難である．これに対して，k-mmg アルゴリズムでは，十分な訓練例の下で正しい解に収束し，かつ訓練例のサイズの多項式時間で求めることができる．本論文では，計算機用日本語動詞辞書に記述された動詞を対象として，k-mmg アルゴリズムを応用することによって，効率的に選択制限を獲得する方法について述べる．計算機用日本語動詞辞書には，一つの動詞は複数の格フレームとして記述され，それぞれの格スロットには例題が付与されている．しかしながら，例題の数は限られているので，汎化を行うのに十分ではない．例題は概念階層知識を用いて自動生成し，その正誤をユーザに確認することによって正負の例題を蓄積する．更に，正負の例題に対して k-mmg アルゴリズムを直接適用することは，非効率的になる場合があるので，正例のみに対して k-mmg アルゴリズムを適用し，予め定められた閾値により極小汎化をいくつか求め，含まれる負例の数が最も少ない仮説を選択する．このようにして，仮説生成を行うことによって効率的に選択制限を学習することができる．

2 k-mmg アルゴリズム

有村らの提案した k-mmg アルゴリズムは，Plotkin によって与えられた最少汎化 (lgg) アルゴリズムの自然な拡張であり，k 個の汎化を求めることできる．k 個の汎化は選言的に解釈できるので，選言的な記述の学習を k-mmg アルゴリズムにより容易に実現できる．k-mmg アルゴリズムの概要を図 1に示す．

補助手続き reduced は S を被覆し，かつ一つの木パターンを取り除くと S を被覆しなくなるような k 個の木パターンを返す関数である．素朴なアルゴリズムでは，例題集合の k 分割を求めてから分割された集合それぞれを汎化することによって，k 個の木パターンを求めることができるが，極めて非効率的である．k-mmg アルゴリズムでは，k 個の例題を選択し，その無矛盾で極大な汎化を行うことによって，reduced を k と S の多項式時間推

```
function k-mmg
input k:正整数, S: 正例の集合;
output R: 汎化集合;
begin
      if ( k = 1 ) then
            return lgg(S);
      else if ( reduced(k,S) が存在している ) then
            begin
                  max := reduced(k,S);
                  R := tighten(S,max);
                  return R;
            end
      else return k-mmg(k-1,S);
      endif
end
```

図 1: k-mmg アルゴリズム [1]

論で実現している．したがって，バージョン空間によって学習されるような探索空間の問題に対しても，容易に応用することができる．

3 日本語基本動詞辞書 [7]

本論文では，計算機用日本語基本動詞辞書 [7] の動詞を対象として，選択制限の学習について述べる．同一の読みを持つ動詞は複数個の格フレームを構成し，一つの格フレームは以下の様な構造をしている．

動詞の読み, 識別子, 格スロットの並び

また，一つの格スロットは

格助詞, 上限, 例題の並び

という構造をしている．例として，動詞「あいする」の格フレームを図 2に示す．動詞「あいする」は「ヲ」と「ガ」の二つの格助詞を取り得るので，この格フレームには二つの格スロットがある．

図 2の場合，格助詞『ガ』を伴う例としては「彼」と言う名詞が与えられ，格助詞『ヲ』を伴う例としては，選言によって例題が与えられている．すなわち，格助詞『ヲ』を伴う

読み：”あいする　　”,
識別子：”００１”,”００１”,”００２”,”重要動詞３１１”,
格スロット$_1$：”ガ”,”12*** * **”,”彼”,
格スロット$_2$:”ヲ”,
”12*** * **／156** * **／12*** * **”,
”妻、子、主人／犬／家族、日本、会社”

図 2: 格フレームの例

名詞は,

$$\{\text{妻, 子, 主人}\} \cup \{\text{犬}\} \cup \{\text{家族, 日本, 会社}\}$$

という三つの選言として与えられ，少なくとも３つのカテゴリに分類されることを示している．また，各カテゴリのそれぞれの上界は 12*** * **, 156** * **, 12*** * **として与えられている．“12*** * **” 等の ID は分類語彙表 [8] の ID である．分類語彙表は７階層あり，分岐の枝数は最上位階層と第２階層が５個で第３階層から第５階層までは高々１０個程度の規模である．第６階層と第７階層は１０個以上の分岐がある．分類語彙表には，親を二つ持つような概念はなく，代わりに ID を複数持つ概念が含まれている．

一つの格フレームは図 3に示すような項として表現される．

f(読み, 識別子, [s(格助詞$_1$, 正例$_1$, 負例$_1$, 木パターン$_1$),
s(格助詞$_2$, 正例$_2$, 負例$_2$, 木パターン$_2$),
$\vdots$
s(格助詞$_n$, 正例$_n$, 負例$_n$, 木パターン$_n$)]).

図 3: 項による格フレームの表現

図中の一つの構造体 s によって格スロットを表現し，また [1] と同様に各正例$_i$は概念階層知識をコンパイルした表現として与えておく．例えば，正例 a_nの概念階層中の上位概念が a_{n-1}で，a_nの上界が a_1で，各 a_iの上位概念が a_{i-1}であるとき，$a_1(a_2(\ldots(a_n)\ldots))$ という項で表現する．但し，a_nの下位概念となるような例題は取り除いておく．このような項で例題を表現しておけば，θ-包摂によって一般化を捉えることができる．項 C が項 D をθ-包摂するとは，$C\theta = D$なるθが存在することを言う．x を変数とするとき，$i < n$ であるような $a_1(a_2(\ldots(a_i(x))\ldots))$ を $a_1(a_2(\ldots(a_n)\ldots))$ の一般化と呼ぶ．このように概念階層をコンパイルしておくことによって，概念階層を参照することなく汎化を行うことができる．複数の ID を持つ概念については，ID それぞれについてコンパイルした正例を与える．

図 1中の補助手続き reduced の中では，２つの木パターンに対して，一方を包摂し，他方を包摂しない極大な木パターンを求める [2] が，選択制限の学習においては，同一の訓練

例が一つの格フレーム中に複数回出現していたとしても，独立したデータとして扱われるので，高速化を図ることができる．すなわち，木パターンでも1引数の関数記号だけが現れるので，極大な木パターンの探索が容易になる．具体的には，最悪 $O(n^2)$ 個の候補があるが，すべての関数記号が1引数の関数記号であるので，$O(n)$ 個の候補だけを考慮すればよい．

4 訓練例の生成

選択制限を説明するような背景知識を与えておくことはできないので，訓練例の構文的類似性に基づいた汎化によって選択制限を学習せざるを得ない．しかしながら，辞書に記録されている訓練例は限られており，記録されている訓練例だけでは十分な汎化が期待できない．十分一般的な選択制限を学習するには，バージョン空間法と同様に学習システム自らが訓練例を生成し，適当な数の訓練例を蓄積する必要がある．訓練例の生成に際しては，その正誤は必然的に外部の教師に委ねられるため，訓練例を盲目的に枚挙するのは好ましくなく，概念階層のサイズが大きくなると手に負えなくなる．したがって，適当なサイズの訓練例を生成するように制御しなければならない．訓練例生成の手続きの概要を図4に示す．

```
input: 格フレーム f, 正整数 k;
begin
    for each slot s of f do
        for each disjunct d of s do
            begin
                max := find_maximal(d)
                d := remove_children(d, max)
                for each e ∈ d do d := d ∪ select_sibling(k, d, e)
            end
end
```

図 4: 訓練例生成の手続きの概要

訓練例集合は，必ずしも極大ではないので，*find_maximal*によって極大な訓練例を求める．*find_maximal*では，2分探索により各訓練例に対して極大な概念を探索する．2分探索を用いれば，サイズ n の訓練例に対して $O(logn)$ 回の質問をすることになる．極大な訓練例が求まれば，それらの下位概念となる訓練例を *remove_children* を呼び出して取り除いておく．極大な訓練例での置き換えは効率化のために行う．次に，*find_sibling*により概念階層中で同一の深さでかつ深さが一つ上の概念が共通しているk個の概念を任意に選択し新しい例として付け加える．したがって，質問回数は，例題の数をm，訓練例の最大のサイズをnとすれば，$O(mlog(n)+km)$ となる．n は高々10であるので，$k>2$ ならば質

問回数は$O(km)$となる．生成された訓練例の正誤はユーザに質問することによって，正負を区別して記憶する．このパラメータkは，必要に応じて調整する．

5 k-mmgによる探索

k-mmgアルゴリズムは，与えられたkと訓練例のサイズの多項式時間で訓練例の極小汎化を計算できるが，負例がある場合には多項式時間推論は保証されなくなる．したがって，負例をどのように扱うかが問題となるが，選択制限の学習においては，訓練例の大部分が正例であり，多くの負例が存在しないので，正例のみを汎化する．選択制限の記述空間が正例のみから学習可能なクラスであるならば，極小汎化アルゴリズムにより極限における同定が期待できる．正例から学習不可能であったとしても，近似解として十分な精度を期待できるであろう．したがって，基本的には正例のみの汎化を行い，負例は汎化することなく直接記憶しておく．選択制限がいくつの選言によって記述可能であるかは予めわからないので，ビーム探索の手法を用いて，固定のビーム幅で仮説を生成していく．すなわち，ある固定のパラメータによって，選言の数を限定しておく．学習が完了する前に，選択制限を用いる必要がある場合には，生成された仮説の中で尤もらしい仮説を用いて，非文の判定に利用する．ここでは，仮説の中で説明してしまう負例の数が最小で，選言の数が少ない仮説を用いる．但し，与えられた例が負例かどうかは汎化せずに記憶した負例との一致によって検査し，基本的に過剰汎化を許すことにする．また，辞書を基にして汎化を行う場合，辞書に記述された選言を尊重するかどうかも問題となるが，ここでは辞書中の選言による記述に対しては，再分割だけを行うことにする．

6 実験結果

計算機の主記憶不足のために，分類語彙表を直接用いた実験ができないので，概念階層を人工的に作成し，例題の生成とk-mmgにより汎化に要する時間を測定した．実際に作成した概念階層の規模を表1に示すが，各レベルのノードから出ている枝数を固定して概念階層を作成した．尚，実験はSparcstation/IPX上のSICStus prolog上で行い，native code compilerでコンパイルしたもので測定した．

レベル	1	2	3	4	5	6	7	総ノード数
枝数	5	2	3	3	3	10	10	27000

表1: 作成した概念階層

人工的に作成した概念階層ではあるが，概念階層の深さは分類語彙表と同一にした．また，格フレームについても，概念階層なしでは評価が困難であるので，乱数を用いて人工的に生成した．但し，生成される格フレームは，辞書に記述された格フレームと同一構造

とした．したがって，この実験は学習の速度のみを評価するもので，選択制限の学習の評価としては別の実験が必要である．

学習の速度を計るために，乱数により生成した５つの格フレームについての学習時間を測定した．実験結果を表 2に示す．各格フレームの最初の実行時間が例題生成と汎化に要した時間であり，各スロットの部分に記述した実行時間は前者が例題生成に要した時間であり，後者が汎化に要した時間である．単位はすべてミリ秒であり，実行時間には GC に要した時間も含まれている．生成された例題の正誤については，乱数により 95%は正しい例題として扱った．また，例題生成のパラメータとして，同一レベルの概念に関する例題は 3 つ乱数を用いて選択し，一つの選言を更に小さな選言に分割するパラメータとしては $k = 2$ を用いた．人工的に作成した概念階層ではあるものの，27000 の概念を持つ概念階層上での汎化に十分な速度であると言えるであろう．k-mmg による汎化に要した時間と比較して，例題生成にかなりの時間を要しているが，素朴な方法により例題生成を実現しているので，概念階層の記憶方法及び参照方法も含めて例題生成に関してはまだ改善の余地があると考えられる．

	選言の数	例題生成前 例題数	例題生成後 例題数	実行時間 (msec)
格フレーム 1				21308
スロット 1	2	3+4	正例:11+13 負例:4	15090+399
スロット 2	1	3	正例:10 負例:2	5520+299
格フレーム 2				11430
スロット 1	1	4	正例:16 負例:0	10500+930
格フレーム 3				31449
スロット 1	1	1	正例:1 負例:3	1590+0
スロット 2	3	2+4+7	正例:7+13+25 負例:7	26500+3359
格フレーム 4				18728
スロット 1	3	2+2+4	正例:7+5+16 負例:4	14330+840
スロット 2	2	1+1	正例:4+4 負例:0	3509+49
格フレーム 5				28938
スロット 1	3	1+6+1	正例:4+19+4 負例:5	14879+1250
スロット 2	3	1+4+1	正例:4+16+4 負例:0	12710+99

表 2: 実験結果

次に，例題生成のパラメータを変更することなく，汎化のパラメータのみを変更して実験を行った結果を表 3に示す．この表からわかるように，許す選言の数を増やすと，急激に汎化に要する時間が増加する．特に，$k = 4$ のときの格フレーム 3 が極端に遅くなっている．これは他の格フレームと比較して，訓練例の数が多いことによるものと考えられる．また，速度の低下も $k = 4$ 程度であれば，許容できる範囲であると考えられる．

	k=2 実行時間 (msec)	k=3 実行時間 (msec)	k=4 実行時間 (msec)
格フレーム 1	21308	44188	190899
スロット 1	15090+399	15059+17150	15090+95810
スロット 2	5520+299	5909+6070	5599+74400
格フレーム 2	11430	35820	470710
スロット 1	10500+930	10490+25330	10520+460190
格フレーム 3	31449	164989	4711828
スロット 1	1590+0	1580+0	1580+9
スロット 2	26500+3359	26659+136750	26719+4683520
格フレーム 4	18728	21159	20906
スロット 1	14330+840	14290+2870	14289+15599
スロット 2	3509+49	3699+300	3510+369
格フレーム 5	28938	52838	554679
スロット 1	14879+1250	14889+1720	14890+10070
スロット 2	12710+99	12199+24030	12090+517629

表 3: 実験結果 (k-mmg のパラメータ変更)

7 あとがき

本論文では，有村らの極小汎化アルゴリズムを応用して，選択制限を獲得する方法について述べた．本方法では，辞書に記述された情報と概念階層知識を基にして，訓練例が生成される．生成された訓練例の正誤については，ユーザと対話することによって確認し，効率的に選択制限を獲得することができる．人工的に生成された比較的大規模な概念階層を用いた実験により，その学習の高速性が確かめられた．実際の分類語彙表を用いた実験を行っていないので，自然言語処理のためにどの程度有効な知識が獲得できるかという問題，すなわち選択制限の学習の質的な評価は今後の課題である．大規模な分類語彙表を高速な処理も重要な課題である．

選択制限を選言によって近似する方法では限界があるので，構成的帰納の方法を導入することによって，選択制限のより柔軟な獲得を実現する必要があると考えられる．

謝辞

k-mmg について詳しく御教示頂いた九州工業大学の有村博紀氏に感謝致します．

参考文献

[1] H. Arimura, H. Ishizaka, T. Shinohara, S. Otsuki. A Generalization of the Least General Generalization. RIFIS Technical Report, RIFIS-TR-CS-63, 1992.

[2] H. Arimura, T. Shinohara, S. Otsuki. Polynomial time inference of unions of tree pattern languages. In ALT 91, 105–114, 1991.

[3] G. D. Plotkin. A Note on Inductive Generalization. Machine Intelligence 5, 153–163, 1970.

[4] T. M. Mitchell. Version Spaces: A Candidate Elimination Approach to Rule Learning. In *IJCAI-77*, pp. 305–310, 1977.

[5] T. M. Mitchell. Generalization as Search. Readings in Artificial Intelligence, Tioga, 1981.

[6] H. Hirsh. Incremental Version-Space Merging: A General Framework for Concept Learning. Kluwer Academic pub, 1990.

[7] 計算機用日本語基本動詞辞書 IPAL(Basic Verbs)，情報処理振興事業協会技術センター，1987.

[8] 分類語彙表，国立国語研究所，秀英出版, 1964.

[9] D.Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. Aritificial Intelligence, 26(2):177–121,1988.