

日本語翻訳タスクへの帰納論理プログラミングの適用

新納浩幸[†] , 阿部修也^{††}

本論文では, SENSEVAL2 の日本語翻訳タスクに対して帰納論理プログラミング (Inductive Logic Programming, ILP) を適用する. 翻訳タスクは分類問題として定式化できるため, 帰納学習の手法を利用して解決できる. しかし翻訳タスクは新たに訓練データを作るのが困難という特異なタスクになっており, 単純に確率統計的な帰納学習手法を適用することはできない. Translation Memory の例文だけ, つまり少ない訓練データのみを用いて, どのように分類規則を学習すれば良いかが, 翻訳タスク解決の 1 つの鍵である. このために本論文では ILP を用いる. ILP は確率統計的な帰納学習手法にはない特徴を有する. それは背景知識を容易に利用可能である点である. 背景知識とは訓練データには明示されない問題固有の知識である. この背景知識によって訓練データが少ない場合の学習が可能となる. ここでは ILP の実装システムとして Progol, 背景知識として分類語彙表を利用することで, 翻訳タスクに対して正解率 54.0% を達成した. この値は, 付加的な訓練データを用いない SENSEVAL2 参加の他システムと比較して優れている.

キーワード: 翻訳タスク, 帰納論理プログラミング, Progol, 分類語彙表, 背景知識

Application of Inductive Logic Programming to Japanese Translation Task

HIROYUKI SHINNOU AND SHUYA ABE

In this paper, we apply Inductive Logic Programming (ILP) to Japanese Translation Task of SENSEVAL2. Translation Task is regarded as a classification problem, and can be solved by inductive learning methods. However, we cannot use general statistical learning methods for this task, because this task has the serious problem that it is hard to create training instances newly. Therefore, the problem is how to learn a classifier from instances in Translation Memory, that is, small training data. To overcome this problem, we use ILP which can handle background knowledge in learning. This is a big advantage over statistical learning methods. Background knowledge means domain specific knowledge which are not described in training data clearly. Using background knowledge, we can learn rules through small training data. In this paper, we used Progol as a ILP system, and 'bunrui-goi-hyou' as background knowledge to achieve the precision 54.0% for Translation Task. This precision is superior to other systems in the contest which did not create new training instances.

KeyWords: *Translation Task, Inductive Logic Programming, Progol, thesaurus, background knowledge*

[†] 茨城大学工学部システム工学科, Department of Systems Engineering, Ibaraki University

^{††} 茨城大学大学院理工学研究科システム工学専攻, Graduate School of Science and Engineering, Ibaraki University (2003 年 4 月より株式会社システム計画研究所)

1 はじめに

本論文では、SENSEVAL2 の日本語翻訳タスクに対して帰納論理プログラミング(Inductive Logic Programing, 以下 ILP と略す) を適用する. 背景知識として分類語彙表を利用することで, 正解率 54.0% を達成した. この値は, 訓練データを新たに作成しない翻訳タスク参加の他システムと比較して優れている.

SENSEVAL2 の日本語翻訳タスクは, Translation Memory(以下 TM と略す) と呼ばれる日英対訳対が与えられ, テスト文中の該当単語を英訳する際に利用できる TM の例文番号を返すタスクである¹(黒橋禎夫, 白井清昭 2001). これは英訳を語義と考えた場合の多義語の曖昧性解消問題となっており, 分類問題の一種である. このため従来から活発に研究されている帰納学習手法を用いて解決可能である. おそらく大規模かつ高品質な訓練データを用いたシステムが, コンテストで優秀な成績を納めるはずである.

しかし翻訳タスクでは大規模かつ高品質な訓練データを用意するコストが高い. TM は1つの単語に対して平均して 21.6 例文がある. 今仮にある単語 A の例文として id_1 から id_{20} までの 20 例文が TM に記載されているとする. 新たに訓練データを作成する場合, 単語 A を含む新たな文を持ってきて, id_1 から id_{20} のどれか 1 つのラベルをその事例に与える必要がある. ○か×かの二者択一は比較的容易であるが, 20 個のラベルの中から最も適切な 1 つを選ぶのは非常に負荷のかかる作業である. この理由のために, 実際のコンテストにおいて, 大規模かつ高品質な訓練データを用意する方法をとったシステムは 1 つ(Ibaraki) だけであった. ここでは訓練データを新たに作成せずに, 日本語翻訳タスクを解決することを目標とする. 訓練データを新たに作成しないとしても, TM の例文は訓練データとして扱える. ただし TM の例文を訓練データと見た場合, その量は少量と言わざるをえない. つまり問題は, 少量の訓練データからどのようにして精度の高い分類規則を獲得するかである. そのための戦略として ILP を用いる.

少量の訓練データからどのようにして分類規則を学習したらよいかは, 機械学習における 1 つの重要な課題である. その解決方法として背景知識の利用が提案されている(秋葉泰弘, フセイン・アルモアリム, 金田重郎 1998). 背景知識とは, 訓練データには明示されない問題固有の知識であり, 広く捉えれば, 人間の持つ常識的知識と考えて良い. 一種の知識データベースである. 問題はその背景知識を, どのように学習手法に取り入れてゆくかである. その解決のために提案されているのが ILP である. ILP は訓練データを述語論理の形式で表し, そこから分類規則に相当する規則(述語論理の形式では節に対応) を導出する. 知識データベースは述語論理の形式によって自然に表現できるので, 背景知識の利用の観点からは ILP を用いた学習戦略が優れている(古川康一, 尾崎知伸, 植野研 2001). 更に ILP の背景知識では, 複雑なグラフ構造を持ったものも表現できるので, 近年, CMU の機械学習チームは Web ページの文書分類に ILP を利用している(Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam and Sean Slattery 2000). 更にいくつかの自然言語処理への応用も知

¹ 厳密には, 英訳自体を解答としてもよいが, ここではこの解答形式は考慮しない.

られている (William Cohen 1995)(Mary Elaine Califf, Raymond J. Mooney 1997)(嶋津恵子, 古川康一 2000).

本論文では, ILP の処理系として Muggleton による Progol を利用する (Stephen Muggleton 1995). Progol によって多義語の曖昧性解消を行う. そして背景知識としては分類語彙表 (国立国語研究所 1994) を利用する. 以下 2 章で多義語の曖昧性解消を ILP で行う方法を示す. 3 章では分類語彙表をどのように背景知識として組み込むかを説明し, 4 章で実験, 5 章で考察を述べ, 最後にまとめる.

2 ILP による多義語の曖昧性解消

ILP による分類問題の解決については, 書籍 (古川康一他 2001) に詳しく解説されている. ここでは関連する事柄についてのポイントのみを述べる.

自然言語処理の個々の問題の多くは分類問題として以下のように定式化できる. まず分類先のクラスの集合 $C = \{c_1, c_2, \dots, c_m\}$ を設定し, 次に事例 x を n 個の要素からなる素性ベクトル (f_1, f_1, \dots, f_n) で表す. 各素性は事例を識別するための観点に対応する. k 番目の素性を $attr_k$ と名前をつけておく. 訓練事例は事例 x とそのクラス c_x の対の情報 (x, c_x) であり, これを多数集めたものが訓練データとなる. 確率統計的な帰納学習手法 (決定木, 決定リスト, ME 法など) は訓練データを入力として, 分類器 F を構築する. 分類器 F への入力は事例であり, 出力はクラスである. 分類問題を解決するとは, この分類器 F を作成することである.

ILP では訓練事例 (x, c_x) を以下の節で表現する.

```
class (x, c_x).
attr_1 (x, f_1).
attr_2 (x, f_2).
...
attr_n (x, f_n).
```

これらの節が訓練データとなる. ILP ではここからあるクラスにのみ共通して見られ, 他のクラスには見られないある種のパターンの発見を行う. これは本質的に帰納推論の処理である. その結果, 例えば, 以下のような節を ILP は出力する.

```
class(X,c) :- attr_5(X, h).
```

これは事例 X の 5 番目の素性が h であれば, 事例 X のクラスが c であることを示している. また左辺が $class(Y,c)$ である節が他になくしかも, 事例 X の 5 番目の素性が h でなければ, 事例 X のクラスは c でないことも示している.

ここまでは特に確率統計的な手法と大きな違いはない. 確率統計的手法にはない ILP の大きな特徴は, 訓練データ中に任意の述語や節を記述できる点である. この与えられた訓練事例

の集合以外から追加される述語や節を背景知識と呼ぶ。つまり問題固有の知識や、人間の常識といったものを背景知識として訓練データ内に簡単に追加できることが ILP の大きな特徴となっている。特に、述語論理の節の表現力は高く、ILP は表形式(素性ベクトル)では表現できない複雑な構造を持つ訓練事例も表現できる。

以下、ILP によって多義語の曖昧性解消を行う。利用する素性は以下の 4 種類を用いた。

- 対象単語の直前の単語 e1
- 対象単語の直後の単語 e2
- e1 から前方 3 単語 e3
- e2 から後方 3 単語 e4

例を示す。対象とする多義語を「与える」として「彼では力不足という印象を与えるかもしれない。」という文は、以下のように形態素解析される。第 1 列が表記、第 2 列が原型、第 3 列が品詞を表す。

彼	彼	普通名詞
で	で	格助詞
は	は	副助詞
力	力	普通名詞
不足	不足	サ変名詞
と	と	格助詞
いう	いう	動詞
印象	印象	普通名詞
を	を	格助詞
与える	与える	動詞
かも	かも	接続助詞
しれ	しれる	動詞
ない	ない	形容詞性述語接
。	。	句点

ここから以下の素性が得られる。

- e1 = 'を'
- e2 = 'かも'
- e3 = {'と', 'いう', '印象'}
- e4 = {'しれ', 'ない'}

例えば、この例文の ID が sen25 であり、この文の「与える」の語義 ID が ataeru2 だとすれば、この例文に対するデータは、以下の節で表現される。

```
class(sen25, ataeru2).
```

```

e1(sen25, 'を').
e2(sen25, 'かも').
e3(sen25, 'と').
e3(sen25, 'いう').
e3(sen25, '印象').
e4(sen25, 'しれ').
e4(sen25, 'ない').

```

3 分類語彙表の利用

前節の設定で、訓練事例を節に変換すれば、ILPにより分類規則が節の形で得られる。ここで得られる規則を、背景知識を利用することで更に高めることも可能である。

本論文では、背景知識として分類語彙表(国立国語研究所 1994)を利用する。分類語彙表は木構造をもったシソーラスである。ただし木のリーフノードにのみ単語が配置されている。つまり木のあるノード以下に位置するリーフノードの単語群はそのノードの階層において同一の意味クラスに属すると考えて良い。当然階層が上がるほど、同じ意味クラスの単語は増加することになる。

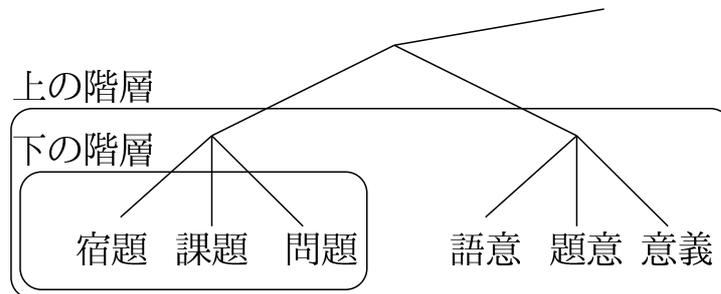


図 1: 分類語彙表

例えば図 1 において一番下の階層で考えると、「課題」「宿題」「問題」は同じ意味をもつグループとなり、「語意」「題意」「意義」は別の意味のグループとなる。1つ上の階層で考えると、これらの単語はすべて同じ意味をもつ単語と見なせる。本論では一番下の階層でみた場合のグループの単語を同じ意味をもつ単語とした。

図 2 に示した規則は、素性(e_1, e_2, e_3, e_4 の単語)と、分類語彙表に含まれる単語と、その単語と同じ意味を持つ単語を結び付ける。つまり、ILP は同じ意味の単語を同じ単語として扱うようになる。述語 b は、分類語彙表の単語とその意味の番号の組を表す。述語 $e_1 \sim e_4$ は、文の素性を表す。述語 $e_{1-w} \sim e_{4-w}$ は、単語の代わりに語義を用いることによって述語

e1 ~ e4 を拡張したものであり, e1 ~ e4 の代りに素性として用いられる.

```
e1_w(ID, Number) :- e1(ID, Word), b(Word, Number).
e2_w(ID, Number) :- e2(ID, Word), b(Word, Number).
e3_w(ID, Number) :- e3(ID, Word), b(Word, Number).
e4_w(ID, Number) :- e4(ID, Word), b(Word, Number).
```

図 2: 単語を語義に一般化する規則

4 実験

本論文では ILP の実装システムとして Muggleton による Progol(Stephen Muggleton 1995) を利用した. Progol への入力形式は, Prolog 形式の述語や節であり, 本論文で説明に用いた形式で行える. ILP の実装システムは他にも存在するが, Progol が最もよく利用される代表的なシステムである.

まず, TM の形態素解析結果から素性(e1, e2, e3, e4) を抽出する. また例文番号を分類先のクラスとする. 例文番号, クラス, 素性の情報を節に変換し, Progol の入力ファイルを作成する. 入力ファイルを Progol に読み込ませて, 規則を生成した. テストは翻訳タスクのコンテストで用いられた全 40 単語(各単語 30 問, 計 1,200 問) が対象である. それらに対して, Progol により得られた規則を使い, 多義語の曖昧性解消のテストを行った(実験 1). 次に, TM の例文の他に背景知識として分類語彙表を用いて, Progol により規則を生成した. 得られた規則を使い, 40 単語に対してテストを行った(実験 2). 実験 1 と実験 2 の結果を表 1 に示す. 表 1 の TM の列は実験 1 の結果を示し, TM+背景知識 の列は実験 2 の結果を示している.

表 1: 背景知識の効果

見出し	TM	TM+背景知識	↓	↓	↓
ataeru	0.167	0.167	kiroku	0.267	0.267
baai	0.033	0.033	koeru	0.967	0.867
chikaku	0.333	0.500	kokunai	1.000	1.000
chushin	0.367	0.367	kotoba	0.700	0.933
deru	0.333	0.333	mae	0.167	0.000
egaku	0.333	0.333	mamoru	0.033	0.033
hakaru	0.600	0.567	matsu	0.867	0.867
hana	0.667	0.700	miseru	0.733	0.933
hantai	0.800	0.933	mitomeru	0.233	0.233
ima	0.067	0.867	mondai	0.533	0.533
imi	0.400	0.567	motomeru	0.867	0.867
ippan	0.333	0.533	motsu	0.333	0.867
ippou	0.633	0.533	mune	0.233	0.267
iu	0.033	0.033	noru	0.300	0.267
jidai	0.700	0.733	shimin	0.867	0.967
jigyou	0.500	0.500	sugata	0.200	0.133
kaku_n	0.800	0.733	tsukau	0.700	0.667
kaku_v	0.967	0.967	tsukuru	0.633	0.367
kau	0.467	0.833	tsutaeru	0.400	0.367
kiku	0.500	0.500	ukeru	0.400	0.433
↓	↓	↓	平均	0.487	0.540

平均の正解率は TM のみは 48.7% であり, TM+背景知識 では 54.0% であった. 分類語彙表を背景知識として用いた効果が確認できる. またこの 54.0% という値は, 付加的な訓練データをを用いない翻訳タスクの他のシステムの正解率と比較しても, 優秀な値と言える.

次に確率統計的な手法の 1 つである決定リストと比較してみる. 論文(新納浩幸 2001)では翻訳タスクの正解から語義(例文番号)をグループ化して, TM の例文番号をグループ化することで, 正解率が向上することを述べている. そのため翻訳タスクに対する学習手法を比較する場合, TM の例文のグループ化を揃える必要がある. そこで, ここでは論文(新納浩幸 2001)と同じ手法を用いて, 正解データから例文をクラスタリングし, 同一の訓練データを用いることにした. 確率統計的な学習手法としては, 決定リストを用いた(実験 3). 実験の結果を表 2 に示す.

表 2: 決定リストとの比較

見出し	決定リスト	ILP	↓	↓	↓
ataeru	0.333	0.867	kiroku	0.833	0.933
baai	0.367	0.933	koeru	0.633	0.600
chikaku	0.367	0.467	kokunai	1.000	0.633
chushin	0.200	0.700	kotoba	0.967	0.967
deru	0.367	0.233	mae	0.267	0.200
egaku	0.567	0.267	mamoru	0.367	0.833
hakaru	0.733	0.900	matsu	0.867	0.767
hana	0.800	0.933	miseru	0.933	0.733
hantai	0.900	0.967	mitomeru	0.200	0.300
ima	0.367	0.933	mondai	0.533	0.467
imi	0.767	1.000	motomeru	0.633	0.000
ippan	0.333	0.567	motsu	0.900	0.833
ippou	0.500	0.767	mune	0.267	0.300
iu	0.733	0.900	noru	0.300	0.433
jidai	0.567	0.233	shimin	0.567	0.433
jigyuu	0.667	0.567	sugata	0.367	0.233
kaku_n	0.300	0.800	tsukau	0.533	0.967
kaku_v	0.967	0.967	tsukuru	0.033	0.233
kau	0.733	0.533	tsutaeru	0.367	0.200
kiku	0.467	0.500	ukeru	0.333	0.100
↓	↓	↓	平均	0.548	0.605

平均の正解率は決定リストでは 54.8% であったが、本手法では 60.5% の結果を得た。つまり TM だけを用いた学習システムとしては、決定リストよりも ILP の方が優れていた。

ただし、いくつかの単語については、実験 3 での ILP の正解率が、実験 1 での ILP の正解率や、実験 3 での決定リストの正解率よりも、極端に低くなっている。例えば、mokuteki, jidai, ukeru, baai, egaku, ima などである。これらの正解率が極端に下がっている理由は、学習結果として生成された節の順序の問題である。これは、default 規則にあたるものを適切に設定できなかったことを意味している。これについては次節の考察に記述する。

5 考察

背景知識を用いても必ずしも正解率が高くなるとは限らないことは容易に予想がつく。実際に、実験 1, 2 でも精度が下がる単語がいくつか存在する。また、実験 3 と同様の課題についてさらに分類語彙表を背景知識として用いた実験も行ったが、この場合の平均の正解率も 60.5% から 58.9% に低下した。分類語彙表を用いることで、単語を語義に一般化すれば、ある部分では効果があるが、別の部分では過度の一般化になるので、その影響が現れると精度は下がる。過度の一般化への対処は今後の課題である。

また節を規則として見た場合、節の適用順序が重要になる。これは入力事例と訓練事例に矛盾がないことを仮定する論理を基盤とする推論では問題にならない。しかし現実の問題では訓練データに矛盾する入力も有り得る。例えば、以下のケースを考えてみる。

```
class(X,c1) :- attr_1(X, a).    %% 節 A
class(X,c2) :- attr_2(X, b).    %% 節 B
```

節 A は事例の 1 番目の素性の値が a なら分類クラスが c1 であることを示し、節 B は事例の 2 番目の素性の値が b なら分類クラスが c2 であることを示している。この 2 つの規則が学習されたということは、訓練データ内には、1 番目の素性の値が a でしかも 2 番目の素性の値が b の事例が存在しなかったことを意味する。また同時にそのような事例が存在しないことも仮定したことになる。ところが、現実にはそのような事例が入力されることもある。この場合、上記の規則では、クラスは c1 と識別される。一方、上記の規則の順序を変更し、以下の形にすれば、クラスは c2 と識別される。

```
class(X,c2) :- attr_2(X, b).    %% 節 B
class(X,c1) :- attr_1(X, a).    %% 節 A
```

節の出現順序は Progol では考慮されていないようである。訓練データに対応する述語や節の与える順序が、生成される節の順序に影響する。本実験ではこの点は何も対策をたてずに、学習された節をそのまま適用した。しかしこのような節の順序は default 規則が何に対応するかという問題にもなっており、正解率に大きく影響する。実験 3 で正解率が大きく下がる単語は、みなこの問題がからんでいた。この対策も今後の課題である。

また上記の問題とも関連するが ILP では頻度の情報がなくなってしまう。例えば、上記の節 B に合致した訓練事例の数が 10,000 で、節 A に合致した訓練事例の数が 1 のとき、普通に考えれば合致する事例数の多い節 B を優先させるのが正しいであろう。しかし ILP では、頻度の情報が節 B と節 A に反映されず、結果として同じ重みを与えている形になる。この問題は、確率と論理を結び付ける研究と関連しており、いまだ決定版は出ていない状況である (古川康一他 2001)。

少量の訓練データしかない場合、識別精度を高めるには、訓練データ以外の情報、つまり背景知識をいかに取り込むかが重要である。今回の実験では背景知識として分類語彙表を用い

たが、単語を語義で一般化することは通常確率統計的な手法でも実現可能であり (Hussein Aluallim, Yasuhiro Akiba, Takefumi Yamazaki and Shigeo Kaneda 1994), この点では ILP を用いた利点は少ない。ただし翻訳タスクでは背景知識の利用という観点以外からも、ILP を用いた方が適切であると思われる。なぜならここで扱っている少量のデータは統計学というサンプルではないからである。例えば、仮に

語義 c1 の例文 1 から e1=a と e2=b という素性
 語義 c2 の例文 2 から e1=c と e2=d という素性
 語義 c1 の例文 3 から e1=a と e2=e という素性

が得られたとする。確率統計的な手法では、以下の 5 つの確率が高くなる。

確率	対応する例文
$P(c1 e1=a)$	例文 1, 例文 3
$P(c1 e2=b)$	例文 1
$P(c1 e2=e)$	例文 3
$P(c2 e1=c)$	例文 2
$P(c2 e2=d)$	例文 2

そして特に $P(c1|e1=a)$ の確率が高くなる。同じクラス c1 の例文 1 と 3 に素性 e1=a が発生しているからである。ただし、このような確率の算出が妥当なのは、例文 1, 2, 3 がサンプル、つまり等確率で現れる事例という仮定がある。TM の例文はサンプルではありえない。例えば、ある単語は 90% 以上の割合で、語義 c1 の意味用法で利用されとしても、TM のその単語の例文の 90% 以上が語義 c1 の例文であることはない。つまり、TM の例文数から素性に重みをつけるのは意味がない。そのため TM から得られる素性は、同じ重みで評価するのが妥当であろう。今回 ILP が決定リストよりも優れた結果を出せた要因がそこにあると思われる。

また ILP の背景知識として、今回は分類語彙表を用いたが、任意の節が組み込めることは大きな魅力である。特に、Web ページは解析の観点によっては、複雑な構造をもつことになり、そのような複雑なデータ構造からの学習には ILP が利用できるため、今後応用範囲が広がる研究分野だと思われる。

6 おわりに

本論文では、SENSEVAL2 の日本語翻訳タスクに対して ILP を適用した。ILP は背景知識を容易に学習に組み込めるという確率統計的な手法にはない長所がある。翻訳タスクは少量の訓練データしか利用できない分類問題と見なせるため、翻訳タスクは ILP の格好の応用となっている。ここでは ILP の実装システムとして Progol、背景知識として分類語彙表を利用することで、正解率 54.0% を達成した。この値は、訓練データを新たに作成しない翻訳タスクの他システムと比較して優れている。また語義のクラスを同一にした訓練データを用いて、確率統計的手法の 1 つである決定リストと比較したところ、決定リストの正解率 54.8% に対して、ILP では 60.5% となり、決定リストよりも良い結果が得られた。

分類語彙表を利用した場合の過度の一般化をどう押さえるか、出力される規則の優先順序をどのように制御するかが今後の課題である。

参考文献

- Hussein Aluallim, Yasuhiro Akiba, Takefumi Yamazaki and Shigeo Kaneda (1994). “Induction of Japanese-English Translation Rules from Ambiguous Examples and a Large Semantic Hierarchy.” 人工知能学会誌, **9** (5), 730–740.
- Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam and Sean Slattery (2000). “Learning to Construct Knowledge Bases from the World Wide Web.” *Artificial Intelligence*, **118** (1-2), 69–113.
- Mary Elaine Califf, Raymond J. Mooney (1997). “Relational Learning of Pattern-Match Rules for Information Extraction.” In *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pp. 6–11.
- Stephen Muggleton (1995). “Inverse Entailment and Progol.” *New Generation Computing, Special issue on Inductive Logic Programming*, **13** (3-4), 245–286.
- William Cohen (1995). “Learning to Classify English Text with ILP method.” In *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pp. 3–24.
- 古川康一, 尾崎知伸, 植野研 (2001). 帰納論理プログラミング. 共立出版.
- 国立国語研究所 (1994). 分類語彙表. 秀英出版.
- 黒橋禎夫, 白井清昭 (2001). “SENSEVAL-2 日本語タスク.” 電子情報通信学会言語とコミュニケーション研究会, NLC-36~48, pp. 1–8.
- 秋葉泰弘, フセイン・アルモアリム, 金田重郎 (1998). “例からの学習技術の応用にむけて, 2. 応用の課題に対する解決法.” 情報処理, **39** (3), 245–251.
- 新納浩幸 (2001). “SENSEVAL2 日本語翻訳タスクに向けて作成した語義判別学習システム Ibaraki.” 電子情報通信学会言語とコミュニケーション研究会, NLC-36~48, pp. 25–30.

嶋津恵子, 古川康一 (2000). “データベースからの知識発見システム DM-Amp – 設計と実装とエキスパートシステム開発への応用.” 人工知能学会誌, **15** (4), 629–637.

略歴

新納 浩幸: 1985年東京工業大学理学部情報科学科卒業. 1987年同大学大学院理工学研究科情報科学専攻修士課程修了. 同年富士ゼロックス, 翌年松下電器を経て, 1993年茨城大学工学部システム工学科助手. 1997年同学科講師, 2001年同学科助教授. 情報処理学会, 人工知能学会, 言語処理学会, ACL各会員. 博士(工学).

阿部 修也: 2001年茨城大学工学部システム工学科卒業. 2003年茨城大学大学院理工学研究科システム工学専攻博士前期課程修了. 同年4月より株式会社システム計画研究所.

(2002年4月22日 受付)

(2002年7月17日 再受付)

(2002年9月12日 採録)