

対話型 EBL システム

(説明木を利用した領域知識の獲得支援システム)

打 橋 知 孝* 滝 寛 和** 堀 内 英 一**

*NTT 情報通信処理研究所

** (財) 新世代コンピュータ技術開発機構

1. はじめに

知識ベースの構築においては、知識ベースの運用の過程で事例を用いて知識を洗練していくことが重要である。この知識洗練の手法として説明に基づく学習 (EBL) が行われている。

今回報告する対話型 EBL システムは、EBL における「説明」を利用し、ユーザに刺激となる情報を与え、対話的に知識を獲得・洗練する 1 つの試みである。

2. EBL の基本処理

知識獲得問題に対する 1 つのアプローチとして、説明に基づく学習^[1,2] (EBL) が提案されている。

EBL では、以下の①～④を入力とする。

①領域知識 (domain theory): 学習する分野に関するルールや事実

②目標概念 (goal concept): 学習される概念の記述

③訓練例 (training example): 目標概念の具体例

④操作性規範 (operationality criterion): 学習結果の満たすべき規範。学習結果が利用性・有効性の高い記述であるように定めておく規範

これらの入力をもとに以下の 2 つのプロセスにより処理が進められる。

- 1) 学習システムに目標概念及びその具体例である訓練例を与えて領域知識に基づく目標概念の説明構造 (説明木) を生成する説明プロセス。
- 2) 説明構造中の定数を変数化し、変数の拘束の情報を明らかにする。また与えられる操作性規範を満たすように項目を選択し、説明構造を利用性、有効性の高い記述に変換する一般化プロセス。

つまり、EBL では正しい学習が行えるように、既に存在する豊富な背景知識 (領域知識) を活用して事例 (訓練例) を説明し、それに基づいて的確な一般化を行った結果を獲得している。

EBL の処理に関しては、処理に必要な各情報 (①～④)、特に説明の基本となる領域知識が完全であることがその前提となっている。しかし、実際問題として考えると、領域知識が完全である可能性は低いので、実用上、適用範囲が狭いことが弱点となっている。従来の EBL システムでは、不完全な領域知識下では説明に失敗していた。

この点に注目し、今回、対話型 EBL システムの構築を行った。

3. 対話型 EBL システム

今回構築した対話型 EBL システムは、領域知識の不足によって説明に失敗した際、ユーザによる不足知識の追加を支援して説明成功に導き、最終的に説明木及び獲得できる概念を出力するシステムである。

3.1 対話による説明木の生成

説明失敗時にユーザに失敗時の状況を提示し、ユーザとの対話により不完全な知識を極力完全にしていき、説明成功へと導く。

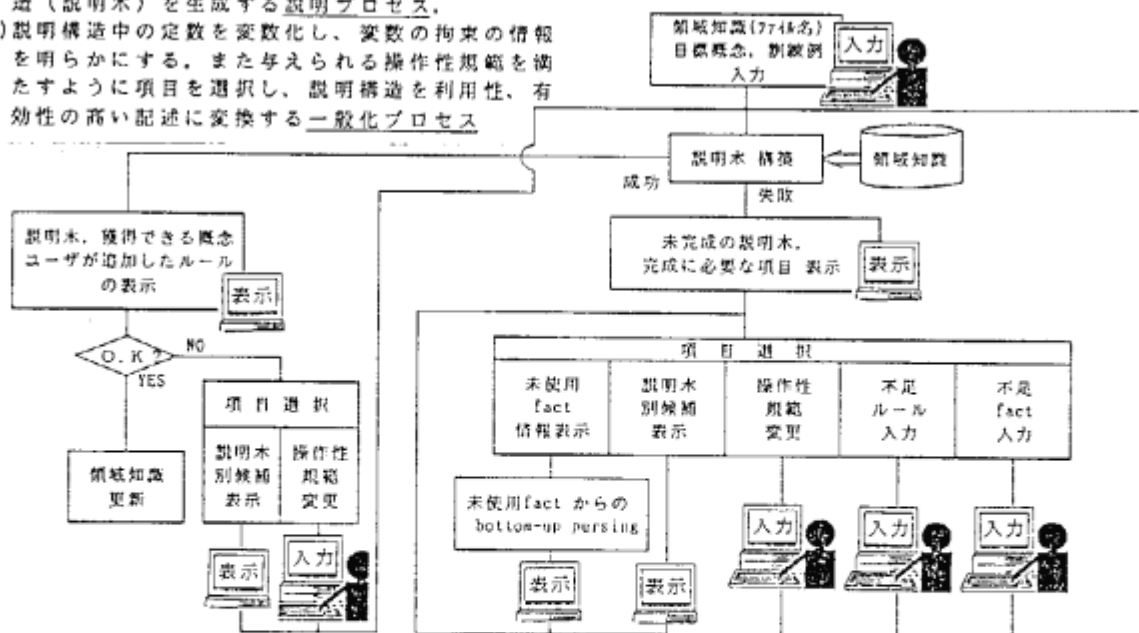


図 1. システムの処理の流れ

Interactive Knowledge Acquisition System for EBL
Tomotaka Uchihashi*, Hirokazu Taki**, Eiichi Horiuchi**

*NTT Communications and Information
Processing Laboratories

**Institute for New Generation Computer Technology

ユーザから情報を獲得するには知識推定の刺激が必要である。

ここでは説明失敗までに獲られた不完全な説明木を利用し、以下の情報を提示する。

- ① 情報不足により中断した説明木とその木の途絶えた箇所（この項目が存在すれば説明木の構成は完成するので木完成のために必要な情報と言える）
- ② 訓練例として入力しておきながら説明試行時に利用しなかった事実が leaf となるようなサブ説明木を考え、そのサブ説明木の root（サブ説明木の root と前記の機能で示される項目とを結び付けるルールを追加すれば、その説明は成功することになる）

以上の提示を、ユーザへの刺激とし、ユーザとの対話的な処理を進め、領域知識の獲得を行う。

また、説明成功の場合には、各入力の変更による知識獲得の結果（説明木、変数の拘束関係、獲得できた概念）の変化を確認できる。

3.2. 処理の流れ

対話型 EBL システムの処理の流れを図 1 に示す。この処理により、領域知識として不足しているルールの補充を対話的に行うことができる。

4. 処理例

図 2 に示す領域知識（この例では、
`hate(john, john) :- depressed(john).`
 というルールが不足している）をもとに説明試行を行った例を示す。

```
kill(A,B):-hate(A,B), weapon(C), possess(A,C),
depressed(W):-heavy_fall(W),
depressed(W):-serious_ill(W),
possess(U,V):-buy(U,V),
possess(U,V):-get(U,V,P),
get(U,V,P):-met(U,P), carry(P,V),
weapon(Z):-gun(Z).
```

(a) 領域知識

```
suicide(john), serious_ill(john),
gun(colt), met(john, tom), carry(tom, colt)
```

(b) 訓練例

図 2. 入力情報（一部）

```
suicide(john)
├─ kill(john, john)
│  ├─ hate(john, john) (← defect)
│  │  └─ weapon(colt)
│  │     └─ gun(colt)
│  └─ possess(john, colt)
│     └─ get(john, colt, tom)
│        └─ met(john, tom)
│           └─ carry(tom, colt)
```

図 3. 説明失敗時の表示

訓練例で与えられた fact の内、次のものは未使用です。
`{serious_ill(john)}`

未使用 fact が何から導かれるのか表示します。
`serious_ill(_2555) → depressed(_2555)`

図 4. 未使用 fact についての表示

説明失敗時の表示から `hate(john, john)` に関する情報が不足していることが分かる（図 3）。

未使用の事実 (`serious_ill(X)`) について問い合わせてみると `depressed(X) :- serious_ill(X)` というルールの存在が明らかになった（図 4）。

この情報から `hate(X, X)` と `depressed(Y)` が関係づけられれば説明木が完成することが分かる。そこで `hate(X, X) :- depressed(X)` というルールを追加して再試行すると、説明成功に至り、システムは説明木、変数の拘束関係、獲得できた概念を出力する（図 5）。

5. まとめ

今回、EBL の適用範囲拡大を目的とし、説明失敗時に各種情報を出力してユーザの不足情報追加を支援して不完全な知識を極力完全にするように働くシステム（対話型 EBL システム）の構築を行った。このシステムを用いれば各種情報の変更に伴う説明構造・獲得概念の変化を出力でき、不完全な領域知識の下でも EBL の処理を行うことができる。

今後の課題としては、再処理にあたり入力される情報と既に入力されている情報との矛盾性の管理などが考えられる。

【参考文献】

- [1] DeJong, G. and Mooney, R.: Explanation-Based Learning, Machine Learning, Vol. 1, No. 2, pp. 145-176 (1986).
- [2] Mitchell, T. M., Keller, R. and Kedar-Cabelli, S.: Explanation-based generalization: A unifying view, Machine Learning, Vol. 1, No. 1, pp. 47-80 (1986).
- [3] 打橋, 滝, 堀内: EBL 実験システム, ICOT-TR-49, 1988

```
suicide(john)
├─ kill(john, john)
│  ├─ hate(john, john)
│  │  └─ depressed(john)
│  │     └─ serious_ill(john)
│  └─ weapon(colt)
│     └─ gun(colt)
│  └─ possess(john, colt)
│     └─ get(john, colt, tom)
│        └─ met(john, tom)
│           └─ carry(tom, colt)
```

(a) 完成した説明木

```
suicide(_2601) :- serious_ill(_2601),
gun(_3548),
met(_2601, _3769),
carry(_3769, _3548).
```

(b) 獲得した概念

今までに次のルールが追加されています。
`hate(_3919, _3919) :- depressed(_3919).`

(c) 追加ルールの表示

図 5. 説明成功時の表示（ルールの追加による成功）