状況理論に基づく知識獲得における状況の再構成、及び、分節

2J－4

阿部 明典
NTT コミュニケーション科学研究所

1　はじめに

状況依存推論をしながら状況依存知識を獲得する、状況理論に基づく知識獲得システム[1],[2]を提案し、その知識獲得システムに対して、概念クラスタリング[5] 等で行なわれているように背景知識を明示的に与えるのではなく、知識に記録された知識獲得の履歴を背景知識とする状況再構成システム[3]の提案をした。そのシステムでは、状況の再構成に於いて複数の構成の可能性が出ても最初に得られた可能性のみを採用していた。更に、事例が少ない場合は、状況の再構成を誤る可能性がある。本論文で示すシステムは状況の再構成に於いて、複数の可能性を考慮すると共に、再構成された知識ベースを用いて状況依存推論を行う際に、再構成を行なった時点では事例が少な過ぎたため一般化し過ぎた等の理由で推論の結果に誤りが出たものに対し、原因となる状況を分節する。

2　状況の再構成、及び、分節

2.1　知識ベースの構成

基本的に、文献 [3] のものを踏襲しており、普遍知識ベース、状況依存知識ベース、状況データベース、状況管理ベース、状況インデックスより成り立つ。本システムで追加した点は、状況インデックスに last_mark の項を追加し、 [{Situation1}, {situationA}, last_mark] の形式にしたことである。この last_mark は階層構造で最上位にあることを示すマークとして働く。従って、状況を階層構造で表現した場合にこの項を参照するようになる。

2.2　複数の可能性を考慮した状況の再構成

基本動作は文献 [3] に示したものと同じである。文献 [3] に示したものは、状況集合に複数包含関係の可能性が出ても、最初にみつかった候補で状況の構成を行なっていた。本論文で示す方式は状況の再構成に於いて、複数の包含関係を持つ状況集合の可能性が生じた場合は、それらを夫々個別の状況にするか、マージして単独の状況集合にするかユーザに質問を発することにより、複数の状況集合の可能性に対処する。

Step 1: 状況共通集合を集める

状況データベースの状況のデータのうち、複数の知識に関して状況に 2 つ以上の共通集合があれば、それを集める。ない場合は、処理を終了。尚、この集合は、集合の包含関係が複数になることを考慮して、最大集合であるものを全て選ぶ。

Step 2:

複数の包含関係を持つ共通集合がない場合は、goto Step 3。複数の共通集合がある時は、複数の包含関係の可能性をユーザに示す。ユーザがその複数の集合をマージしてよいと示した場合は、マージしたものを一つの状況集合とし、マージしないと示した場合は、状況集合を複数作る。

Step 3: 状況依存知識の更新

出来た共通集合に名前をつける。状況依存知識ベース、状況データベース該当の知識の状況の項を書き換える。状況管理ベースに再構成した状況の状態を記録し、状況インデックスにも変化した結果を追加する。更に、状況インデックスに last_mark を記録するとともに、その下位属性となる状況に関する状況インデックスの last_mark を削除する。

Step 4:

Step 3 までで再構成された状況に対応する状況 (状況が変わると意味を変える知識に対応する普遍知識ベースに於ける知識の状況) を普遍知識ベースから集め、それらと状況の項が一致する知識を更に集める。

Step 5: 普遍知識ベースの更新

出来た共通集合に名前をつける。普遍知識ベース、状況データベース該当の知識の状況の項を書き換える。状況管理ベースに再構成した状況の状態を記録し、状況インデックスにも変化した結果を追加する。更に、last_mark を記録するとともに、その下位属性となる状況に関する状況インデックスの last_mark を削除する。
goto Step 1

2.3　一般化し過ぎた状況の分節

本システムの場合、知識獲得システム自体が殆んどゼロから知識獲得をすることを目的としているので、最初は知識の規模は大きくない。従って、前節に示した状況の再構成も、正しいとはいいかねる場合も生じる。再構成を行なった知識ベースで状況依存推論を行なって誤った解を出した場合、状況の再構成の仕方を誤っている可能性もある。本節で示す手法は、知識の少なさなどの所為で一般化しすぎた状況を分節する手法である。

状況依存推論で、誤った解を出した。

Step 1:

推論の誤りの原因となる知識が単独の状況集合の中にある場合は goto Step 4。それが含まれている状況集合が過去に分節したことがある場合は、goto Step 2。

Step 2:

その分節した集合をユーザに示し、既存の状況集合に含めるか、新たな集合にするか質問する。ユーザがある既存の状況集合に含めてよいと判断した場合は goto Step 5。

Step 3:

その状況をもとの状況集合から除き、新たな状況集合とする。つまり、該当の状況のみを分節し、新たな状況とする。 goto Step 4

Step 4:

新たな状況集合に名前をつける

Step 5:

Merging and Dividing Situation-sets for The Knowledge Acquisition System Based on Situation Theory
Akinori Abe
NTT Communication Science Laboratories
1-2356, Take, Yokosuka-shi, Kanagawa 238-03 Japan

状況データベースにこの知識と状況のことを追加するとともに、状況管理ベースの第3項のリストから該当の状況を除き、状況インデックスも変化した結果を修正し、新しく作成した状況に関してlast_markを記録、削除する。

終了

3 例

前節に処理アルゴリズムを示した。本節では、簡単な例を用いて、本手法を施した後の状況の様子を示す。

3.1 状況の構成

これまでの推論で、図1に示す知識が得られていたとする。状況データベースを参照すると、≪ pay, money, before, 0 ≫に関しては $\{S_{A0}, Mr\_D\}$ の状況が共通集合になり、≪ serve, dish, yourself, 0 ≫, ≪ pay, service, 1 ≫ に関しては $\{S_{A0}, J\&B\}$ の状況が共通集合になる。そこで、システムはユーザにこれらの状況をマージするかどうかきく。マージする場合は、普遍知識は、

$[S_{B1}, \{\ll pay, money, before, 0 \gg\}, 1]$ などとなり、状況データベースの知識は

$[\ll pay, money, before, 0 \gg, \{S_{A1}, Moss\}]$

などとなり、状況管理ベースの知識は

$[S_{A1}, \ll pay, money, before, 1 \gg, \ll pay, service, 0 \gg, \{S_{A0}, Mr\_D, J\&B\}]$,

$[S_{B1}, \ll pay, money, before, 0 \gg, \ll serve, dish, yourself, 0 \gg, \ll put, lap, in, trash, 0 \gg, \ll pay, service, 1 \gg, \{S_{B0}\}]$

などとなり、状況インデックス[†1] に $[Mr\_D, \{S_{A1}\}, Y]$、$[S_{A0}, \{S_{A1}\}, Y]$、$[J\&B, \{S_{A1}\}, Y]$、$[Mac, \{S_{A0}\}, N]$、$[S_{B0}, \{S_{B1}\}]$ などと記録する。マージしない場合は、共通集合が S_{A1}、S_{A2} などと複数になる。

Universal Knowledge-base:
$[S_{B0}, \{\ll eat, food, ontable, 1 \gg\}, 0]$
$[S_{B0}, \{\ll pay, money, before, 0 \gg\}, 1]$
$[S_{B0}, \{\ll serve, dish, yourself, 0 \gg\}, 1]$
$[S_{B0}, \{\ll pay, service, 1 \gg\}, 1]$
$[S_{B0}, \{\ll put, lap, in, trash, 0 \gg\}, 1]$

Situation Dependent Knowledge-base:
$[S_{A0}, \{\ll pay, money, before, 1 \gg\}]$
$[S_{A0}, \{\ll serve, dish, yourself, 1 \gg\}]$
$[S_{A0}, \{\ll pay, service, 0 \gg\}]$
$[Mr\_D, \{\ll put, lap, in, trash, 1 \gg\}]$

Situation Database:
$[\ll pay, money, before, 0 \gg, \{S_{A0}, Mr\_D, Moss\}]$
$[\ll serve, dish, yourself, 0 \gg, \{S_{A0}, J\&B\}]$
$[\ll pay, service, 1 \gg, \{S_{A0}, Mr\_D, J\&B\}]$
$[\ll put, lap, in, trash, 0 \gg, \{S_{A0}, Hokka\}]$

Situation Maintenance Base:
$[S_{A0}, \ll pay, money, before, 1 \gg, \ll serve, dish, yourself, 1 \gg, \ll put, lap, in, trash, 1 \gg, \ll pay, service, 0 \gg, \{Mac, Lott, KFC\}]$
$[S_{B0}, \ll pay, money, before, 0 \gg, \ll serve, dish, yourself,) \gg, \ll put, lap, in, trash, 0 \gg, \ll pay, service, 1 \gg, \{Heichinro, Chez\_Ino, Sabattini\}]$

Situation Index:
$[Mac, \{S_{A0}\}, Y]$ $[KFC, \{S_{A0}\}, Y]$ $[Lott, \{S_{A0}\}, Y]$
$[Heichinro, \{S_{B0}\}, Y]$ $[Chez\_Ino, \{S_{B0}\}, Y]$ $[Sabattini, \{S_{B0}\}, Y]$

Figure 1: 学習前の知識ベースの例

3.2 状況の分節

図2に示す知識ベース(一部のみ記載する)で状況依存推論を行なったとする。Hubの状況で推論していて、実は、≪ have, an, alcohol, 0 ≫ では推論の解が正しくなかったとする。状況インデックスを調べると、Hubの状況は S_{A1} にだけ含まれる。従って、システムはHubのみ別の状況にし、S_{A2} とする。(Hubが単独の状況集合に含まれない場合は、質問を発する。)状況管理ベースは、

$[S_{A1}, \ll pay, money, before, 1 \gg, \ll pay, service, 0 \gg, \ll have, an, alcohol, 0 \gg, \{S_{A0}, Mr\_D, J\&B\}]$

$[S_{A2}, \ll pay, money, before, 1 \gg, \ll pay, service, 0 \gg, \{S_{A0}, Hub\}]$

のよう複数の状況集合に分節される。状況インデックスは

$[S_{A0}, \{S_{A1}, S_{A2}\}, Y]$ $[Hub, \{S_{A2}\}, Y]$

などのようにHubに関するところが変更され、Hubが別の状況集合に分節されたことを示す。

Situation Dependent Knowledge-base:
$[S_{A1}, \{\ll pay, money, before, 1 \gg\}]$
$[S_{A1}, \{\ll pay, service, 0 \gg\}]$
$[S_{A1}, \{\ll have, an, alcohol, 0 \gg\}]$

Situation Database:
$[\ll pay, money, before, 0 \gg, \{S_{A1}, Moss\}]$
$[\ll pay, service, 1 \gg, \{S_{A1}\}]$
$[\ll have, an, alcohol, 0 \gg, \{S_{A1}\}]$

Situation Maintenance Base:
$[S_{A1}, \ll pay, money, before, 1 \gg, \ll pay, service, 0 \gg, \ll \mathit{have}, an, \mathit{alcohol}, 0 \gg, \{S_{A0}, \mathit{Mr\_D}, \mathit{J\&B}, \mathit{Hub}\}]$
$[S_{B1}, \ll pay, money, before, 0 \gg, \ll eat, food, on, table, 1 \gg, \ll serve, dish, yourself, 0 \gg, \ll put, lap, in, trash, 0 \gg, \ll pay, service, 1 \gg, \ll have, an, alcohol, 1 \gg, \{S_{B0}\}]$

Situation Index:
$[S_{A0}, \{S_{A1}\}, Y]$ $[Mr\_D, \{S_{A1}\}, Y]$ $[J\&B, \{S_{A1}\}, Y]$
$[Hub, \{S_{A1}\}, Y]$ $[S_{B0}, \{S_{B1}\}, Y]$

Figure 2: 学習前の知識ベース (一部)

4 まとめ

本システムにおいては、概念クラスタリング[5] 等で行なわれているような背景知識を明示的に与えることなく、状況を複数の状況集合の可能性を考慮して再構成することが可能になるとともに、事例が少ないなどの理由で状況の再構成を誤った場合でも、状況依存推論の解が誤りであった場合、その結果を考慮することにより、誤りの原因となる状況を分節可能である。この状況の再構成、分節により、文献[4]で言及されているような状況の階層構造を状況依存推論の結果により、マルチプルインヘリタンスを許して構成可能になり、事例が少ないなどの理由から過汎化のような誤った再構成をしてしまった状況集合を正しい状況集合とすることが可能となる。

今後の課題としては、現在のシステムは教師つきで知識を獲得、学習しているが、ある程度の規模の知識が集まると、推論の正しさの判断を自動的に行なうことが可能になる可能性がある。その判断機能をシステムに持たせ、正しい、状況依存推論を行なうと共に、自動的に正しい状況の構成、分節を行なえるシステムを作成するというものがある。

References

[1] 阿部 明典 : 状況理論に基づく学習, 信学会研究技報, *AI92-82 (1992.11)*

[2] 阿部 明典 : 状況理論に基づく知識の獲得、及び状況の個別化, 信学会論文誌投稿中

[3] 阿部 明典 : 状況理論に基づく知識獲得における状況の再構成, 情処全大, *2P-4 (1993.10)*

[4] 中島 秀之 : 状況に依存した推論, 人工知能学会誌, *pp.392-398 (1992)*

[5] Stepp R. E. , Michalski R. S.: *Conceptual Clustering : Inventing Goal-Oriented Classifications of Structured Objects, MACHINE LEARNING vol. II, pp.471-498, Morgan Kaufman (1986)*

[†1] $[Mr\_D, \{S_{A1}\}, Y]$ の最後の項で、Yと書いてあると、それより上の階層構造がないことを意味する。