

入試問題を解く人工知能の想定できる知識構造とプロセス A P I を考えてみました。これは信濃プロジェクトにも流用できる技術です。

## 1 . 知識構造

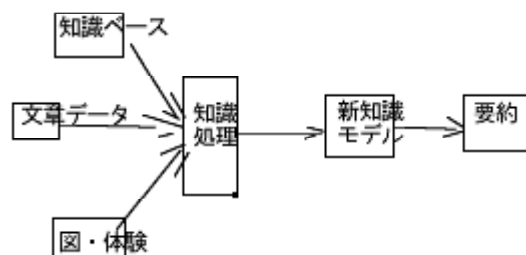


図 処理の概略

問題の知識表現を作ったり、知識をアセンブリしたり、回答を推論したりする知識処理には、既存の知識の固まりである知識ベースと問題の文章を知識表現したデータと図や体験の新規事項が入力します。知識処理では新規の回答の含まれる知識を創成してモデルとして出力します。この結果を要約して回答にします。

知識処理の基盤は連想（共起関係処理）とパターンマッチングであるとししました。知識のアセンブルはこの2つの機能によって実現します。ただ、コンピュータ学のパターンマッチングのように単純ではありません。知識表現には揺らぎがあるからです。ある知識表現と知識ベースのデータをマッチングさせるとき、視点を変えた知識を得なくてはマッチングできないとか、要約して知識表現にどれだけ近いかという評価をしていかねばならないでしょう。視点変換と要約機能は基盤プロセス要素となります。

知識は文ではなく文章となって始めて働きますから、知識はレコードの固まり、・・・R D B だとテーブルが基本的な単位となります。私はこれをモデルと呼んでいます。知識ベースはモデルの固まりと捉えられます。連想もパターンマッチングも基本的にはこのモデル単位に行います。

## 2 . プロセス A P I

今考えているプロセスは以下の通りです。

### ( 1 ) オントロジー構成管理

オントロジーはモデル毎に設定されねばなりません。また、他のプロセス群から利用できるデータとなっていなければなりません。また、シノニムのように相互に関係があります。それらを管理するのです。

### ( 2 ) 解析パターン管理

意味構造パターンを辞書にしたり、オンメモリにしたりして管理します。パターンは意味を理解するための解析用データです。

### ( 3 ) 推論 ( プロダクションシステム )

### ( 4 ) 推論 ( Prolog 型推論システム )

### ( 5 ) 文章要約

必要な意味を指定して、モデルから文を選択します。また、視点変換して表現を変えます。

### ( 6 ) 知識データをアセンブルすること

パターンマッチングや連想を利用して、必要な知識を収集して1つのモデルにします。

### ( 7 ) 個別解析要素群

知識処理用のライブラリです。

### ( 8 ) 黒板システム

### ( 9 ) フォーカス管理

### ( 1 0 ) 問いを知識表現すること

### ( 1 1 ) 5 W 1 H 1 F を解析すること

### ( 1 2 ) cause/result 関係を解析すること

おわり