

考察：「文脈空間」 (続「カテゴリーシステム」)

考察「カテゴリーシステム」で、「文脈空間」に触れました。今回は更につっこんで、考察が進展しましたので、それをまとめまして、「カテゴリーシステム」の続編としました。

参考にした文献は、次の通りです。何度も読み返したい本です。

- ・ エージェントアプローチ人工知能 Stuart Russell・Peter Norvig 著 共立出版  
    ・・・主にプランニングについて
- ・ 新編画像解析ハンドブック 高木幹雄・下田陽久 監修 東京大学出版会
- ・ 脳の計算理論 川人光男 著 産業図書  
    ・・・これで、脳の各分野間のパスが双方向であることを知りました。制御というものの面白さも知りました。
- ・ データマイニング 福田剛志・森本康彦・徳山豪 著 共立出版

## 第1章 はじめに

考察「カテゴリーシステム」で、作業品詞という、カテゴリー操作のための入れ物を導入しました。その中で、イメージデータから記号化情報を取り出すシステムを内蔵したものを文脈空間と呼ぶのでした。文脈空間については、途中で考察を打ち切っていましたので、稿を改めまして、以下に記していきます。

人物達が会話するシーンの状況、会話内容を記号化したものはフレームという、データエリア(テーブル)で表現されるのでした(動画システム)。しかし、このフレームによって、全ての状況を表現することは不可能であるということでした。

(例文1.1) レストランに行きました。私はナイフとホークで食べた。

(例文1.2) レストランに行きました。私は予約席で食べた。

(例文1.3) レストランに行きました。私は床で食べた。

(例文1.4) レストランに行きました。カシャンという音がしました。家内は振り返りました。

これらの文では次のようなことが推量できます。

・ 例文1.1の「ナイフとホークで」は「ナイフとホークを使って」で、

・ 例文1.2の「予約席で」は「予約席に座って(足は折り曲げて)」ですね。

・ 例文1.3の「床で」は「床に座って(足はたたんで)」ですね。しかも「私」はイヌではないかという、疑念が湧きさえます。例文1.1は「私」がイヌとは考えられません。チンパンジーならばあるいは。

・ 例文1.4の「カシャンという音」は調理場、あるいは食器が壊れたかな、と想像できます。

これらのことから、ナイフとかホーク、人、イヌ、予約席、床、そういったものが様々な情報を持っていることが分かります。

ナイフを取り上げてみましょう。日常の生活で、ナイフの意味が作られていくことは日常を振り返ってみれば納得のいくことです。

- ・ナイフを手に持ちました。
- ・ナイフに力をいれて、押してみました。そうしたら、まな板に傷がつきました。
- ・ナイフを持って以下のことができました。
  - ・パンを切りました。
  - ・リンゴを切りました。
  - ・野菜を切りました。
  - ・リンゴの皮を切りました。
  - ・鉛筆を削りました。
- .....

毎日毎日こういう体験からナイフの働き、つまり意味が知識の中に構築されていきます。知識は役に立つことです。このまま体験を続けたとして、どれだけ役に立つでしょうか。全く新しいものを切ろうとしたとき、知識が役に立ちません。やはり、今までの知識を有効に使いたい。

既存知識を応用するには知識を抽象化することです。抽象化のレベルで同等となること、比喩が展開できる、既存知識の応用ができるということではないでしょうか。

ということで、ナイフという抽象化物をカテゴリー化することになります。これを筆者は作業品詞と読んでいます。自然言語の品詞みたいな働きするからです。で、このカテゴリーの下に「切る」という、作業品詞を設け、そのさら下に、「パンを切る」「リンゴを切る」「野菜を切る」という具体的な体験を配置させていきます。何か今までにないものを切りたくなったら、作業品詞「切る」を応用して推論します。手にナイフを持って、上から押す・・・というふうに。「削る」はナイフの別の作業品詞ということで、設定していきます。

でも、ナイフという作業品詞がもっとも強力な応用を持つのは、抽象化が最高の作業品詞を持つときですね。つまり意味プリミティブを持ってくればよいということになります。「separate something」なんて作業品詞の下に、「切る」とか「削る」とかを配置する・・・ということになります。

ここまでは、記号論の世界で対応できる話です。発想の転換をしてみましょう。ナイフを美術品と見たらどうなるでしょう。いろいろな意味がそこに現れてくるでしょう。更に、ナイフは錆びるものや、錆びないものがあります。もう、ナイフには無数に意味が付着していきます。

さらに、世の中にはナイフだけではありません。自動車やなにやらあって、考えていくと、意味記号数が爆発してしまいます。

そこで、この困難を解決したい。物事をそう考えていくと、イメージのまま、物事を知識として保持し、必要に応じて意味記号化しようという考えが浮かびます。パターン認識において、ナイフという単語を利用するのに必要な意味記号群のレベルまで、一気にパターン認識して、ブレークダウンしていくことは不可能でしょう。無数にある意味要素を一度に解析していかなければなりません。パターン認識用に意味解析の元になるデータも持たねばなりません。人間の頭にも、そんな広範囲な知識はないことは内観してみれば分かりま

す。

第一、ナイフは切る道具であるというだけの意味しかないわけではありません。始めは、ナイフを美術品として、捉えることがなくても、また、そんな体験がなくても、ものとしてあれば、他の美術品との相同性はあるわけで、何かのきっかけに誰に教わるでもなく、ナイフを美術品にしてしまう発想は生まれるでしょう。それは、ナイフが「綺麗なもの」と意味記号化されたときです。そして、「美術品」という作業品詞に登録されたときということになります。

本論では、文脈空間の動作概要を先ず説明し、データエリアの構造を記していきます。

文脈空間の応用にも触れて、最後に、文脈空間を利用するのに必要となるであろう関連技術を紹介していきたいと思います。

API 案は Java を想定していて、付録に示しています。

## 第2章 文脈空間の動作概要

文脈空間は意味を保持する機能を担うだけなので、これを使う、上位のシステムが想定されています。意味を扱うということで、直接的には自然言語システムで、あとは推論、思考（プランニング、リーズニング）システムなどが考えられます。

文脈空間は、イメージデータを取り込み、このイメージに意味を付加したり解析したりする。また、その逆に解析したイメージや意味を取り出したりします。意味は記号であり、記号の集まりであるフレームの構造をも形成しています。

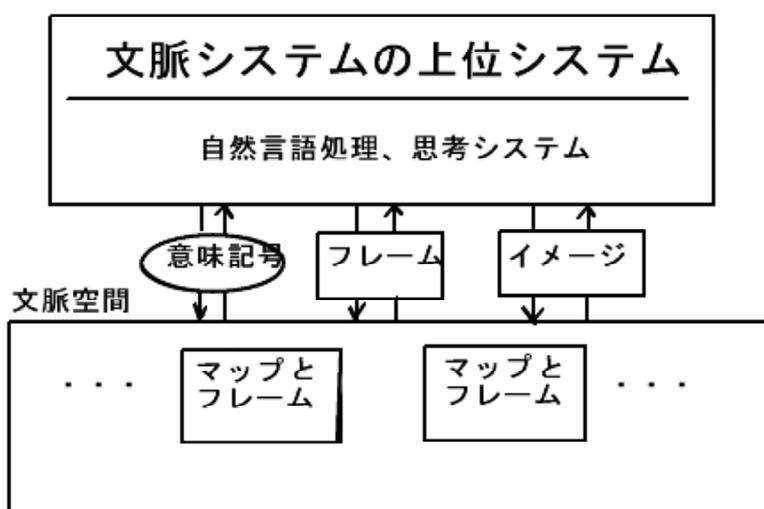


図2.1 文脈空間の入出力

(例文2.1)ピッチャーはボールをキャッチャーに投げた。

例文2.1を文脈空間にするには、次の手順を踏みます。

(1)ピッチャーのイメージを用意し(パターン認識の途中成果でもいいです)文脈空間の一つのマップシステムとして登録します。そのピッチャーのイメージのある部分は頭に、ある部分は腕に、ある部分は手に、足にと分類されて、情報が付加されているでしょう。情報付加はイメージを構成するピクセル毎にたいして行います(コネクション)。

特に手には、ボールを持つ機能があるというような情報が添付されている事でしょう。

(2)ボールのイメージを用意し、文脈空間の一つのマップシステムとして登録します。ボールには柔らかいとか、軽いかいろいろな情報が付加されています。

(3)キャッチャーのイメージを用意し、文脈空間の一つのマップシステムとして登録します。そのキャッチャーのイメージのある部分は頭に、ある部分は腕に、ある部分は手に、足にと分類されて、情報が付加されているでしょう。特に、手にはボールを受けるといような情報がついていることでしょう。

(4)シーンを表すマップシステムにピッチャーとボールとキャッチャーを配置します。ボールはピッチャーとキャッチャーの間に配置します。

(5)例文2.1を表現するには、ボールがピッチャーからキャッチャーに移動することが実演されなくてはなりません。アニメーションに成っていればリアルで、この移動運動過程で、沢山の情報が得られることでしょう。例えば、「ピッチャーからキャッチャーが見えた」とか、「ピッチャーはボールを握った。そして、投げた。」とか。「ピッチャーの腕が振られた。」とか、「キャッチャーの腕が一瞬前後に揺れた。」とか。

(1)から(5)までの作業によって、今までイメージで持っていた事柄が、記号化されて、自然言語で利用できるデータとなります。ピッチャーの手からボールが離れるアニメーションとキャッチャーがボールを受けるアニメーションがあればいいのです。もっとも、それらの動きを解析するプロセスが必要になりますが、それは汎用な機構であって、キャッチボール専門の機構となっている必要はないわけです。

意味プリミティブを設計し、それを抽出する解析プロセスを作っていく、そんなインプリメントになります。意味プリミティブはそんなに多くないはずですから、努力すれば、実現可能な作業でしょう。全ての意味を用意するという、フレーム爆発問題にはならないはずです。

文脈空間は、幾何学的問題の解決には必須となります。

(例文2.2)冷蔵庫の中にプリンがあります。冷蔵庫の扉の前に段ボール箱を置いた人があります。プリンを取り出せるでしょうか。



## 第4章 文脈空間の応用

文脈空間はこの上に作業品詞を設定して、利用します。課題遂行の単位である作業品詞には、意味以外に、様々なデータを保持すべきだからです。

ここでは、作業品詞の一部として組み込む立場で、次の2つの課題に挑戦してみます。

- (1) プランニング
- (2) リーズニング

### 4.1 プランニング

行動計画の策定です。どんな風にロボットは行動すべきなのかを考察しながら、作業品詞、文脈空間を考えていきます。

(課題4.1) 冷蔵庫の中に爆弾があって、それを撤去したい。冷蔵庫は隣の部屋にある。

ロボットはまず、目標の爆弾に至る大まかなルートを計画します。それは、マップシステムを使って行うのですが、次のような地図データは与えられるとします。

- ・爆弾は冷蔵庫の中にある。
- ・冷蔵庫は隣の部屋の中にある。

これで、プランは、

- ・現在地点 隣の部屋 冷蔵庫 爆弾
- となります。この動きは全て作業品詞で覆います。
- ・作業品詞1 = 現時点 隣の部屋
  - ・作業品詞2 = 隣の部屋 冷蔵庫
  - ・作業品詞3 = 冷蔵庫 爆弾

つぎに、作業品詞1の課題を遂行するのですが、その課題が次のように展開されます。それは知識による作業品詞の下位作業品詞の展開です。

- ・作業品詞1.1 = 現時点 隣の部屋への通路またはドア
- ・作業品詞1.2 = 通路またはドア 隣の部屋

作業品詞1.1に曖昧性があるので、それを解くパターン認識プロセスを起動します。部屋の中に人が通れるくらいの長方形があるかどうかで判断することになるでしょう。これは、文脈空間の情報によります。文脈空間の要求している情報にあう部分が見つければ、そこへ移動します。

ドアであれば、ノブがあるか探すでしょう。これも文脈空間の保持する知識の要求によって行動します。

作業品詞1.1と作業品詞1.2がクリアできたら、作業品詞2の課題に移ります。

作業品詞2では、知識により下位の作業品詞に展開します。

- ・作業品詞2.1 = 冷蔵庫を探す
- ・作業品詞2.2 = 隣の部屋 冷蔵庫

冷蔵庫の位置が曖昧です。それを解くパターン認識プロセスを起動します。白い直方体

の箱を探すことになるでしょう。これも文脈空間の情報によります。見つければ、作業品詞 2 . 1 の課題はとけて、作業品詞 2 . 2 の課題を遂行します。この部分も、障害物があったりしたら、それを解決すべく、更に作業品詞は下方に向けて分解されていくでしょう。

作業品詞 3 では、知識により下位の作業品詞に展開します。

- ・作業品詞 3 . 1 = 冷蔵庫を開ける
- ・作業品詞 3 . 2 = 爆弾を取り出す

これらも、文脈空間の保持する知識によって遂行していきます。必要に応じて、下位の作業品詞群への展開を行っていくのです。

とまあ、簡単に書いてしまいましたが、難しい問題はいくつもあります。

- ( 1 ) 一連の作業を網羅する知識をどうやって獲得するのか。
- ( 2 ) 作業品詞間の意味的整合をどう評価して、これはいけると判断するのか。
- ( 3 ) パターン認識はどうやるのか。

というようなことが、先ず浮かびますね。

( 1 ) の作業はロボットが生活しながら徐々に学習していくしかないことでしょう。学習の効率化のために、新しい事態には既存の知識を利用して、知識を組み合わせるつまり、推論できなくてはなりません。そんな機構が文脈空間の外側に必要になってきます。

( 2 ) の作業は、課題をなにかマップで表現して、解決できるピースをジグゾウパズルのようにマップに配置していき、現状から目標までにパスができるまで、試行錯誤していくことになるでしょう。ピース間の連結は意味処理でやります。文脈空間の出番です。意味は ( 1 ) と同じく、学習によって獲得していくしかないでしょう。

( 3 ) のパターン認識は、自然言語処理と平行して研究していく課題で、一部、文脈空間のマップシステムでも研究すべきことがらです。

#### 4 . 1 リーズニング

リーズニングは常時人の頭の中で動いている思考です。何か言われたとき、「なぜ、彼女はあんなことを言ったのだろう」と考える。ご飯がうまい。「なぜだろう」と考える。本を読んで、頭の中の知識を整理する。そんなのもありますね。あれはこうゆう事だったのかと、本を読んで合点がいく。

私は、夕方になって夜の来るのを、雲が太陽光を遮るからだ、信じたことがあります。子どもなりのリーズニングですね。

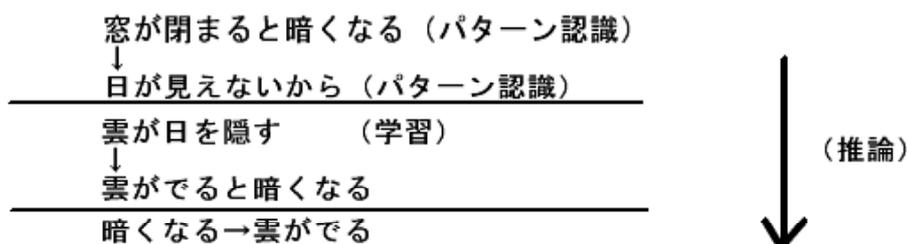


図 4 . 1 「夜になるのは雲がでるから」

図4.1では、パターン認識と意味処理が複雑に絡んでいますから、文脈空間が必要な場面です。窓とか、日とか、雲とか、意味記号で表現したり、イメージで表現したりしなくては、この問題は学習の根本からして、解けない問題となってしまいます。

イメージを意味記号に展開できれば、後はカテゴリーシステムと同じです。作業品詞の体系を「～ならば、～」という情報を中心にした形で作り込んでいくという問題に帰着できます。また、リーズニングでは、比喩の処理が重要になります。内容検索と意味のマッピング処理が重要な技術になります。

## 第5章 関連技術

文脈処理は曖昧性処理を含む難しい技術です。曖昧性を意識した技術として、黒板システムがありますが、もう一つの脳の構造をヒントにした技術を紹介します。脳は無数の領野を持ち、その分散処理によってパターン認識などの作業をしているのですが、全ての領野の結合は双方向なのだそうです。つまり、領野Aから領野Bにデータが流れるときには、領野Bから領野Aへのデータの流れるそうです。双方向の神経パスがある。

視覚の認識ですと、V1とか、V2、V3といった、領野があり、データはだんだん高度な認識を担当していくようになっていくともいいます。機能の階層構造がある。プログラム技術でいいますと、レイヤーパターンということです。レイヤーパターンといえば、インターネットのTCP/IPプロトコルなどの応用技術がありますが、これのデータの流れるは、一方向です。戻しというか、バックトラッキングというか、そういう機能は持ちません。

そこで、脳にならって、レイヤーを双方向にデータをパスする新しいアーキテクチャが考えられます。それを、梯子型レイヤーパターンと呼びましょう。これは、曖昧性処理に強力な機能を提供してくれます。つまり、上位で曖昧性が減少したとか、複数の両立する解釈があり、それを分離したとき、その情報があれば、下位では曖昧性を更に減少できるかも知れません。情報をつけて、下位に処理を差し戻すわけですね。そんなバックトラッキングができる。良い感じですね。

### 5.1 自然言語処理での例

次の図のような階層が考えられます。

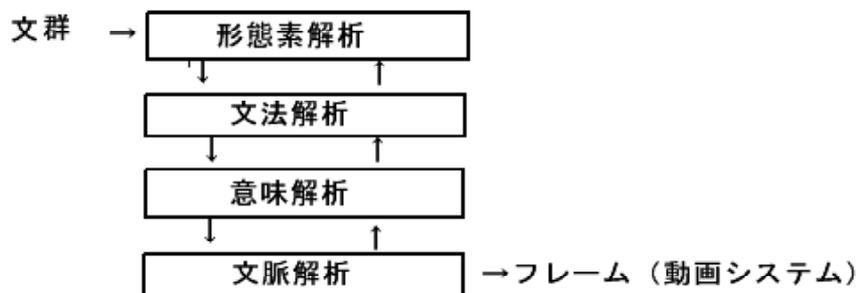


図5.1 自然言語処理

(例文 5 . 1) 信子はリンゴをナイフで切った。学校でみんなに分けてやるのだ。

(例文 5 . 2) 蟻がナイフで遊んでいた。

形態素解析で間違えていたら、文法とか、意味処理で非文となるので、そのむね情報をつけて、一段下に差し戻してやる。そこでもだめなら、さらに下に差し戻す。というような処理ができます。

「ナイフで」というのは、「ナイフを使って」か「ナイフの上で」かは、曖昧のあるところです。意味解析で引っかかったら、文法に戻してやり、文脈解析で引っかかったら意味処理に戻してやります。

## 5 . 2 パターン認識での例

パターン認識も曖昧性の多い処理です。例えば、次のような文字を認識してみましょう。



文字例A



文字例B

図 5 . 2 飾り、崩しがある文字

文字例 A では黒丸がじゃまで、普通パターン認識できません。ですが、黒丸パターンを認識して、それを取り去って、再度文字認識すれば、パターン認識に合格できそうです。そう、ここでも差し戻しのできる、梯子型レイヤーパターンが生きてきそうです。

文字例 B では、文字が崩れています。そこで、閉曲線を取り去ったり、曲線を直線にしたり、傾きのあるものは横、縦だけにしてみたりして、上位の変形レイヤーと下位のテンプレートマッチングレイヤーの間で、行きつ戻りつさせます。そうこうしているうちに、「Little」と単語を構成している、部品がそろってくるはずですが。効率を考えれば、崩しを想定した、パターン当てはめレイヤーを設けたりすることになると思いますが。

崩し字の認識は、昔の手書きの本の電子化には欠かせませんので、ここは頑張るべき所ですね。

## 5 . 3 パターン認識での文脈空間の位置づけ

ロボットの外に広がる世界から切り出したイメージデータを、一気に意味記号のセットには落とせないと述べました。まずは、「家」とか「人」とかの大枠の物体認識とそれらの配置を認識して、文脈空間に落とし込むことになるでしょう。そのなかで、雑音を取り除かれ、また、決まり切った表層な意味は解析されるでしょう。これを外部パターン認識と呼ぶことにします。

そうして、文脈空間のイメージデータを必要に応じて詳細に解析して、必要な意味を記号化していく。記号化して、自然言語処理で使えるようにします。これを、内部パターン認識と呼びましょう。

こんな、外部パターン認識と内部パターン認識の2段階に分けると、パターン認識も見通しがよくなるのではないのでしょうか。

#### 5.4 梯子型レイヤーアーキテクチャの留意事項

層間の依存関係を最小にしなければ、層構造にする意义がありません。となれば、エージェントアプローチで、並行処理を狙うしかないでしょう。

並行処理を管理するには、やはり適者生存戦略を基本にするのがよさそうです。生きているプロセスが有るか無いかは、次の2つのカウンタで管理できます。

- ・共有データの変更カウンタ
- ・動作プロセスのカウンタ

そして、共有データは多層にわたるコラム構造にするのが良いでしょう。どの層でも、上下も含めて、データを参照できるからです。

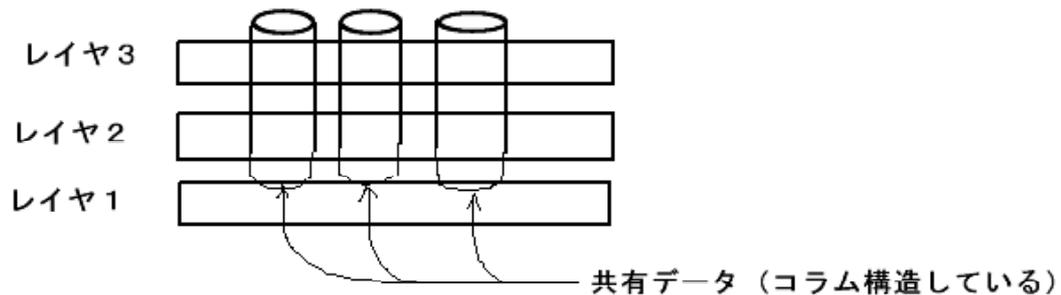


図5.3 梯子型レイヤーアーキテクチャの共有データの構造

## 第6章 まとめ

考察「カテゴリーシステム」で、紹介だけになっていた、「文脈空間」を掘り下げてみました。まだまだ、考えて技術を精密化していかなければならない状況です。「文脈空間」の考え方は自然言語処理や思考システム、パターン認識にも利用できるはずですが、というか、キー技術になると感じています。更に前進していきたい。

以上