

第5章 技術詳細

ここでは、今までの章で述べてきたことを実現するための技術を細かに記したい。

5.1 基本技術

データエリアやアルゴリズムを表現するための基本的な技術をここにまとめて紹介する。それは、次のものである。

(1) コネクション

データやアルゴリズムの関連を表記する方法である。

(2) 入れ子(アコーディオン)

データやアルゴリズムを拡大して詳細に見たり、縮小して大域的に見たりするためのデータの表記方法である。

(3) データ、コマンド表記

辞書などの外部ファイルでデータを持つときは、データの種類と値とを明確にわかる形で持っている必要がある。また、プログラム間を流すデータも表記が明確に定まっていることが必要である。

一方で、メモリ内データはプログラムを作りやすく、プログラムが早く動くものでなくてはならない。

(4) ボクセルとキーフレーム

記号では表現しにくいもの、例えば人体の部分の名称などを表現するときには、イメージ表記が必須となる。

(5) 曖昧性管理

自然言語処理は曖昧性の中での処理である。曖昧性をどう表記し、解決していくか、これを検討しておくことは重要である。

5.1.1 コネクション

物には色、形、重さ、名前といった属性がある。色は、色空間の一点と物とを結びつけるものであり、形は、構造を表す3次元空間のイメージと物とを結びつける物である。色、形、重さ、名前とうとう、すべてを空間とか物としてみても、その一つと、何かとを結びつけることとして、頭に浮かぶイメージをとらえようというのが、ここでいうコネクションである。対応とか連想とかによって、計算機内のすべてのデータを分解して考えていけば、柔軟でわかりやすいプログラムができるという経験から生まれた技術である。

「りんご」には、多くの属性がある。「赤い」、「植物」、「果物」、「木」・・・。これらをあらかじめ並べ立てることはできないし、属性の間にも関連がある。「植物」と「果物」とは、「植物」と「木」とは。そこで必要な都度関連を定義できるという柔軟性が要請される。

どういう関連かということを書いて、物と物との間を結ぶ。これだけを、動的なデータ構造として定める。

(物1) ----- (コネクションの種類) ----- (物2)

図5.1 コネクション

5.1.2 入れ子 (アコーディオン)

人が物を観察するとき、物を拡大して詳細にみたり、縮小して大局をみたりする。物を認識するには、適切な大きさで解析する必要がある。

自然言語でも、「赤い花」で一つの名詞として捉える場合と、「赤い」と「花」というふうに、形容詞と名詞に分解して捉える場合がある。それは処理の状況によるのである。認識処理の中では、データを拡大、縮小を自在に行えるようになっていなくてはならないのである。

大局データと詳細データをコネクションで結ぶ構造を取って、この要請に応えよう。このコネクションを入れ子、あるいは伸び縮自在ということで、アコーディオンと名付ける。

```
(大局) -----入れ子----- (詳細)
          ----- (詳細)
          ----- (詳細)
```

図5.2 入れ子 (アコーディオン)

5.1.3 データやコマンドの表記

データは基本的に、データの種類とその属性と値からなるので、XML の表記を採用するのが自然である。

```
<種類 属性=XXXX>値</種類>
```

図5.3 データやコマンドの表記の基本

これだと、種類の入れ子も簡単に表現できる。木構造のデータやコマンドはこれで全て表現できるわけで、ネットワークが必要となったら、コネクションを表現するタグを定義して、それをこの木構造の中に埋め込めば解決する。

一方で、実行時にこの XML 形式でデータを扱うとオーバーヘッドが生じて不利である。そこで、実行データはフレームで表現し、辞書やファイルでデータを持つときとか、プログラム間でデータの授受を行うときには XML 形式を採用することとする。

なお意味処理の中では、あらかじめ決められない、状況によるデータ項目があるものである。この意味では文脈 X を取るべしというような場合である。辞書やファイルの中に変数も表記できるようになっているのが必須である。

変数は\$を先頭につけて表すこととする。\$Xは他の辞書で定義されている変数で、\$\$Xは作業変数で実行時に決まる物と規約する。

5.1.4 ボクセルとキーフレーム

イメージデータを扱うとき、どうしても記号に落とせない場合がある。既存の記号の組

み合わせで表現できないから、プリミティブな記号として定義すると、プリミティブが無数にできてしまう。そんなとき、ボクセルを使うとイメージを荒くするだけだから簡単に表現できる。ボクセルの各ピクセルに情報をコネクションするわけである。ボクセルを介する情報検索にも使える。ボクセルの拡大、縮小も入れ子（アコーディオン）で表現できる。

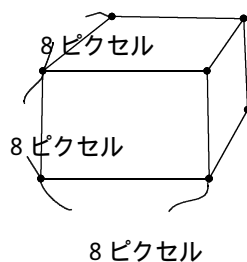


図5.4 ボクセル

人体の部分定義するボクセルであれば、間接とか曲げが利く部分とか、固定部分とかをピクセル情報として定義し、さらに定義部分のピクセル全部に同じ識別子をコネクションすればよい。

キーフレームは、ボクセル表現と記号表現の間である。ボクセルとの関連をコネクションで持って、処理の高速化に利用する。

5.1.5 曖昧性管理

自然言語処理の根本問題として、曖昧性の克服がある。

(例文1) 「私はウナギだ。」

は、「私はウナギという生き物である」のか「私はウナギを食べたいのか」、「私はウナギを捕まいたい」のか、よくわからない。前後の文との関係から解釈していくべきものなのだろう。

だが、「私はウナギだ」という文はそれはそれで、曖昧を含みつつ味わいのある文である。曖昧は敵ではないのである。必要に応じて、厳密化するのが正しい姿勢である。

では、曖昧性をどう表現していくか。可能な厳密解釈を複数コネクションさせるという方法がある。厳密解の意味をフレームで表す。これをカットと呼べば、「私はウナギだ」の発言はエピソードということになる。エピソードに複数のカットをコネクションするのである。コネクションにもその曖昧性を消す条件、このカットを選ぶ条件というものをコネクションする。後の曖昧性処理のためである。

コネクションとして、ファジー表現を持つことも、計算機パワーが大きければ可能であろう。

曖昧性をどう克服していくか。それは弛緩法である。隣接する文とか、文章が作り出す文脈情報との関連で、曖昧性を順次除去していく。

手順としては、曖昧性がコネクションされているエピソードがあれば、そこから近隣のエピソードの解析をして、確定カットの条件が満足されないかを調べる、というふうになる。すなわち、処理の中では、この確定カットを選んで、情報収集するという手順ということである。これは仮定法である。弛緩法は仮定法を多用する手法である。

もうひとつは、「曖昧性フラグを必要に応じてたてていき、決めうちで処理を続行する。問題があれば、バックトラッキングして、別のルートを取る。」というものである。この場合も、仮定法を多用することになる。

5.1.6 基本意味辞書

記号に意味を定義しなければならない。記号は、プリミティブかプリミティブで構成された文またはボクセルで最終的に表現されなくてはならないものである。実戦では、それらを辞書の形式で持つことになる。

辞書の形式は次のようになる。

意味識別子（単語など）	基本意味（プリミティブ）
	付加情報
	修飾文（もしくは修飾文の識別子）
	説明文（もしくは説明文の識別子）
	文法（もしくは文法の識別子）
	文法付加情報
	同義語群
	異義語群
	ボクセルの識別子

図5.5 基本意味所書の構造

識別子はデータのキーとなる項目である。

5.1.7 基本知識辞書

人間は様々な知識を動員して、文を解析して理解している。知識の各項目は様々なコネクションを持ち、他の項目と関連を持っている。その中で体系だったものを基本知識辞書として持つことが想定される。学術は知識を体系立てることもあるから、このような辞書の必要性が予想できるのだ。

知識識別子	直接上位語
	直接下位語
	間接上位語
	間接下位語
	メトリック（属性、値、ファジー性）
	ボクセル識別子
	その他のコネクション群

図 5.6 基本知識辞書の構造

識別子はデータのキーとなる項目である。

5.2 応用技術

当面の問題の文解析と意味解析、文生成にこれらの基本技術を応用することを考察していく。

5.2.1 文解析

文の解析は英語文も日本語文も左から右に向かって行う。

(例文2)「俊は、昨日買った赤いバンドをしめて会社に行った。」

解析点と未解析部分の開始点を管理しながら解析を行うことになる。

<ND>俊</ND><AX>は</AX> 昨日買った赤いバンドをしめて会社に行った

解析点

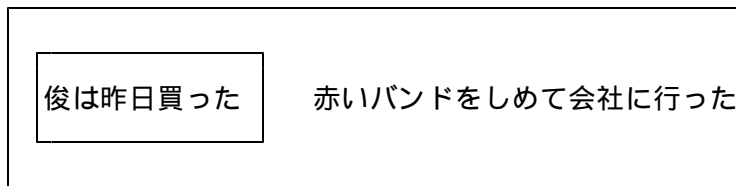
未解析部分の開始点

解析点では、助詞「は」を検出したので、「<ND>俊</ND><AX>は</AX>」がまとまって後続の動詞に掛かっていく（修飾する）と判断する。そうして、その旨を作業メモリに持

つことになる。

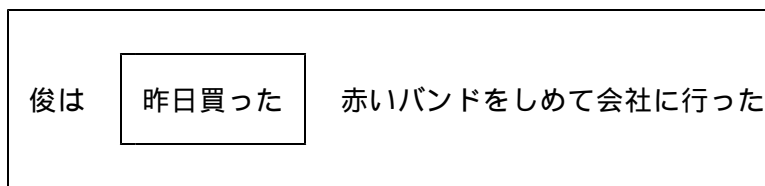
最後まで解析したとき、この文は、

(解釈 a) :



と、

(解釈 b)



が考えられる。

解釈 a のエピソードの形を次にしめす。

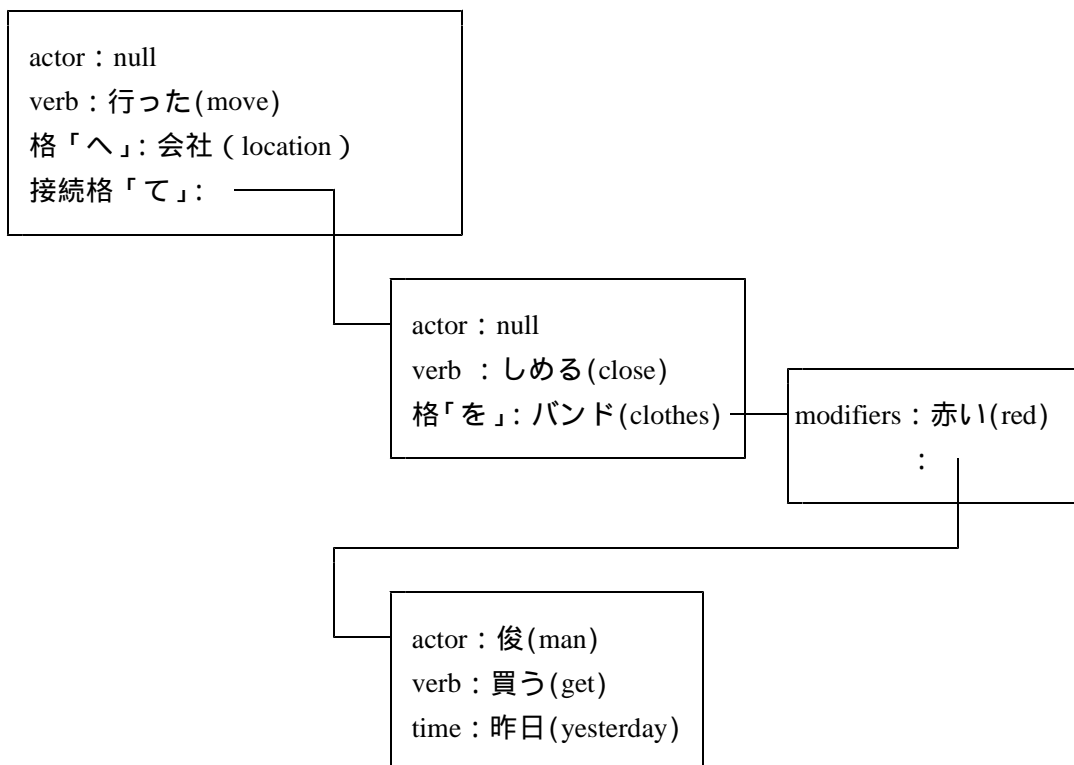


図 5 . 7 解釈 a のエピソード表現

解釈 b のエピソードは解釈 a のエピソードの actor が null のところが、「俊 (man)」にな

る点が異なる。

この曖昧性は、次のようなことを考慮して、消去できる。

- (1) 格助詞「は」は修飾文の主語にならない。修飾文の主語は格助詞「が」である。
- (2) 主文には主語が必要である。

このような事を全てプログラムロジックで作り込むのは、柔軟性がなく現実的でない。プログラムとは独立した、文法データとして、コマンド列で表現すべきである。

それは、次のような形式をしているであろう。

マッチングパターン	処理コマンド列
-----------	---------

図 5 . 8 文法の形式

例えば、「名詞と格助詞が続いていたら、それらを名詞句として動詞を修飾させる」という文法は、次のように書けるであろう。

```
<nd/><ax/>|<ax/>=<np mod=vb><nd>#nd#</nd><ax>#ax#</ax></np>;<nd/>=null;;
```

【説明】

- ・ <nd/>は名詞を表す。「花」という特別な名詞なら、<nd>花</nd>とする。
- ・ <ax/>は格助詞である。「は」という特別な格助詞なら、<ax>は</ax>とする。
- ・ |はマッチングパターンの終了を表す。
- ・ =の左辺に、マッチングパターンの中の操作する品詞を指定する。
- ・ =は左辺を右辺のデータで置き換えることを表す。
- ・ <np mod=vb>は名詞句で、動詞を修飾する成分ということを表す。
- ・ #nd#はマッチングパターンの<nd/>の内容をコピーしてできる値であることを示す。
- ・ =null は=の左辺のデータを削除することを示す。

このように、XML を用いると、柔軟にコマンドを記述できる。

5 . 2 . 2 意味解析

エピソードからカットを作ること考える。それは修飾成分を解釈し、格として捉え直すことである。

(例文 2) の意味表現を以下に示す。

actor : 俊 (man)
verb : 行く (move)
target : 会社 (place)

actor : バンド (clothes)
verb : 巻き付く (close)
direct-object : 俊 (man)

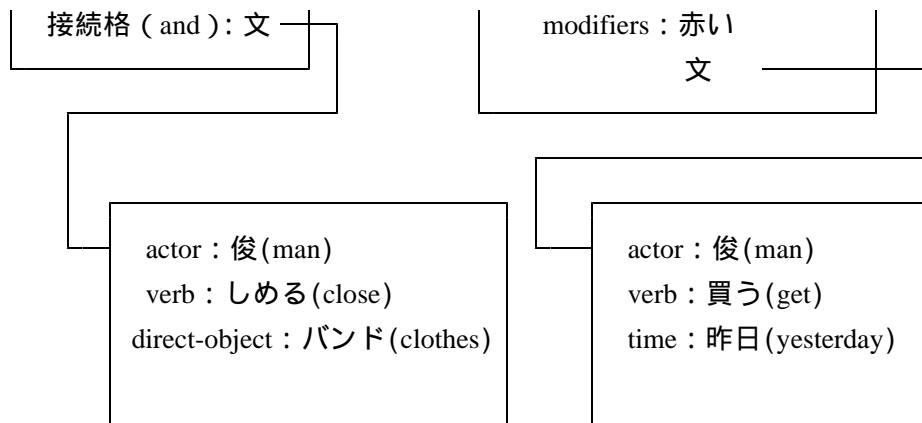


図 5 . 9 意味表現

また格助詞の解析も行うことになる。

(例文 3) 「象は鼻がながい」

「鼻は象がながい」は同じ意味に取れる。

(例文 4) 「私はリンゴが好きだ」

「リンゴは私が好きだ」は意味が異なるようだ。

(例文 5) 「リンゴは私があげた」

「私はリンゴがあげた」は非文のようだ。

「好きだ」という動詞と「あげた」という動詞、また、「鼻が長い」という表現 (be 動詞) では格の取り方が違ってることがわかる。そのため、文法上の付加情報を加えて、細かな文法検査をしていかねばならない。また、この 3 文だけみても、非文とすべきか、正常文とすべきかは、常識による検査を受けていることがわかる。常識辞書が必要になってくるゆえんである。常識はデータコネクションの塊である。

このような事を全てプログラムロジックで作り込むのは、柔軟性がなく現実的でない。プログラムとは独立した、文法データとして、コマンド列で表現すべきである。

このコマンドも XML 形式を転用すれば、柔軟に表現できる。

5 . 2 . 3 文生成

意味表現から、文を生成することを考える。意味表現は時系列データをただ並べたものであるから、何を、どう表現していくかを明確に指定されなくては、処理を実現できない。

(1) 動画システムのどの部分を表現するかを外部から指定すること。

(2) 重要な単語はなにかを明示すること。これにより、表現する内容の粗さを判断する。

(3) 文体を指定する。「です」、「である」とかの語尾だけでなく、一文を電文体にするか、一文に多くの修飾語を組み込んで情報量を多くするかを指定できなくてはならない。

(4) その他

このような事を全てプログラムロジックで作り込むのは、柔軟性がなく現実的でない。
プログラムとは独立した、文法データとして、コマンド列で表現すべきである。
このコマンドも XML 形式を転用すれば、柔軟に表現できる。