

「4月7日の贈り物」で、考えていたAPIを紹介させて頂きましたが、まだまだおおざっぱでした。今回は少しブレイクダウンが出来ましたので、さらに記述を加えたいと思います。考えれば考えるほど、知能という物の奥の深さに当惑することしきりです。

1. 知識ベースとは

知識ベースは次の要素からなるはずですが。

(1) オントロジー

感覚とか、プリミティブな行為、視覚情報の中の基本的な固まり（顔とか、体とか、口とか、目とか）のことで、全ての情報処理の基盤データとなるもの・・・意味素性です。

(2) モデル

カテゴリー化されない、固有のオブジェクト。Aさんとか、家で飼っているこの猫とか、もう唯一のもので。雀もカテゴリーでない、今日の前にいる雀です。それに関する情報を全て、このモデルに記録していくのです。

生活する個体にとって、カテゴリーよりもモデルの状態の方が情報源として重要でしょう。一緒に生活するもののモデルですから。

(3) カテゴリー

モデルを構成する意味素性のなかで共通のものがあれば、カテゴリーを興し、カテゴリーとして抽象化します。家の猫、隣の家の猫、道ばたで観た猫から「猫」というカテゴリーを作っていくのです。

モデル、カテゴリーとして、名詞を挙げましたが、動詞も形容詞も、副詞もモデルとカテゴリーの2本立てで管理されるはずですが。イメージは「あの歩いた事」であって、「歩くこと」はイメージ出来ません。そうして、これらの知識ベースは外界から働きかけ（感覚入力、思考の結果、文章入力など）によって発火することで、文脈が展開され、文脈は外界への働きかけとか思考とかを生み出して、意味の認識という現象になります。

知識ベースはまた、デフォルト処理と意識的な伴った処理との両方をサポートしなくてはなりません。デフォルトは最近の情報によって連想されるものを、他の発火してない情報より優先させる処理（プライミング）が可能でなくてはなりません。

また、行動とか思考を保留していたり、再開できなくてはなりません。そんな、計画オブジェクトを保留して覚えておく機構も必要です。

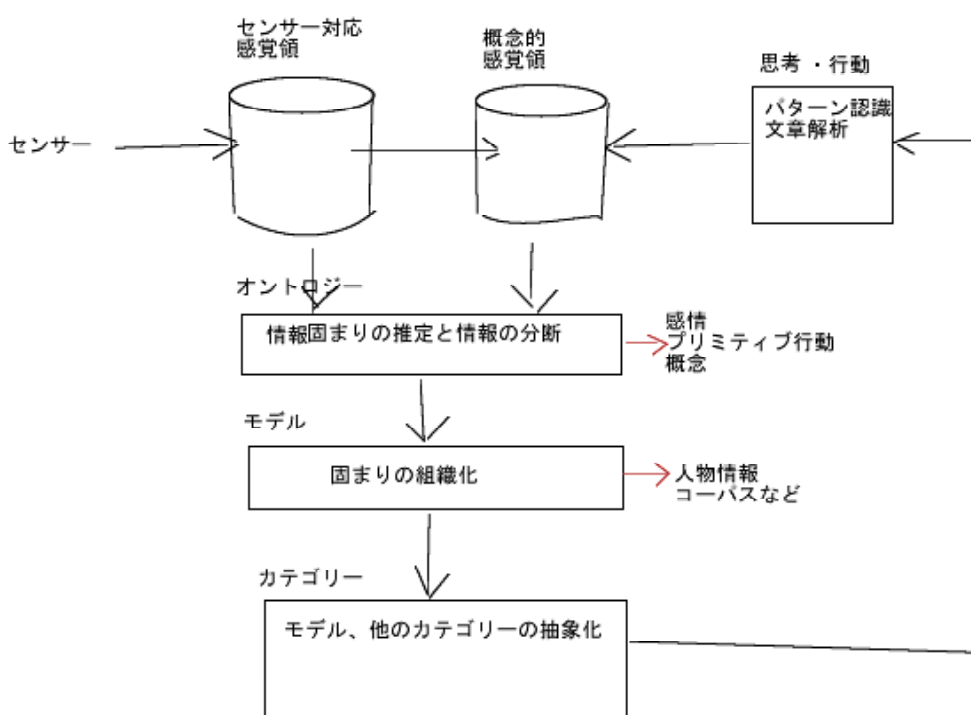
そして、知識はインコアに持ったり、永続化に持ったりしなくてはなりません。インコアはほとんど限られた容量しかないので、処理の実行にのみ用い、通常はもっぱら永続化されていなくてはなりません。

2. 感覚システム

センサーが無数にあると、受け取り側ではメモリにもてないわけで、永続化が必要であります。永続化しているが発火状態を指示するわけで、ダイレクトにレコードのフィールドを更新し、タイムスタンプを打てるようになっていべき。レコードとしてはブロックで構成され、固定長ブロックのフィールドをいくつかと、可変長のフィールド一つでなっているでしょう。固定長ブロックはダイナミックに直接更新できます。可変長ブロックは更新できません。

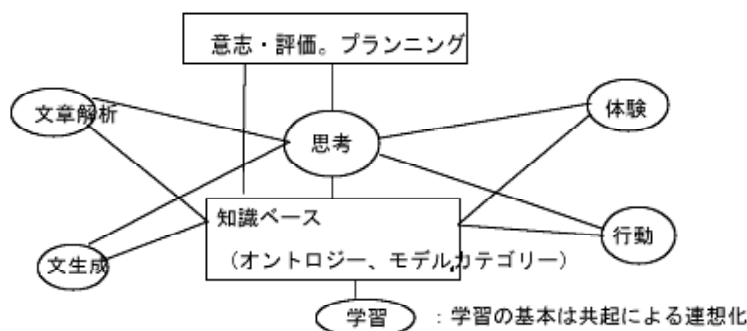
【感覚レコード】

レコード id (固定長)	カウント (固定長)	タイムスタンプ (固定長)	重要度 (固定長)	フォーカス (固定長)	固定長情報 (固定長)	可変長情報 (可変長)
---------------------	---------------	------------------	--------------	----------------	----------------	----------------



3. 自律成長のスキーマ

意志・評価・プランニングシステムがあって、人工知能は制御されます。文章解析、体験が入力系で、文生成と行動が出力系です。その入力と出力のハブとして、意志・評価・プランニングシステムの配下で動くのが思考です。それらの活動は知識ベースによって、情報の基盤を支えられ、無意識的、意識的に学習が行われていきます。



行動も思考も、それらの制御も試行錯誤学習によって充実していくこととなります。行動は行動の経験とか、真似、手法の経験とかで充実していき、思考も思考する経験、思考の手法を学習していくことで充実していきます。この、行動の充実と思考の充実を受けて、制御も充実していきます。その制御の充実の基本は、生存とか、発展、好み、利他行動(愛)とかの評価を受けて、制御手法の充実が行われるのだと思います。制御が適切に行われた決定は条件と共に強化され、追加されていくのです。

成長のスキーマは、オントロジーからモデルへ、モデルからカテゴリーへと学習によって、知識オブジェクトを獲得、充実していくことです。オントロジーは身体オブジェクトであり、モデルは他者オブジェクトで、カテゴリーは身体や他者が抽象化されたオブジェクトです。

知識の核(意味素性)の組み合わせとして (and,or,not) 知識は成長して行くわけで、全ては意味素性によって評価されていくこととなります。メタファー推論も、思考、リーズニングも、推論関係は全て、この意味素性の固まりの相同性、相異性をもって実現され、その結果として知識が豊かになっていきます(学習システム)。

その他に、何か知識の充実法はあり得るでしょうか。物理的にも、数学的にも、そんなスキーマしか考えられないと思います。

4 . 弛緩法

弛緩法とはパターンマッチングの基本的な手法です。形状マッチングをするのですが、その形状を包む上位の形状の予想と内部に包む下位の形状の予想をも行って、パターン認識の結果の信頼度を高めていこうという手法です。

そのため、自分の形状の他に上位形状の予想値(確信度)と下位形状の予想値(確信度)をパターン定義に持ちます。

(1) パターン定義オブジェクト

Pattern:パターン定義オブジェクトの名前 自分の形状(プリミティブ形状定義のリスト)定義 上位の図形の名前、配置情報と確信度 : 下位の図形の名前、配置情報と確信度 :
--

(2) パターン認識 A P I

FrameDataList fdl=Tool.analyze(Pattern pattern,Image image);

【説明】FrameDataList は認識結果を保存する知識ベース項。曖昧性があるから複数の認識結果が確信度とともに帰る。各認識候補はFrameData に設定される。

Image は 2 次元図形データの場合のピクセル画像である。

--

弛緩法によって情報をリーズナブルに分断し、固まりとして認識していくのが本質的なパターン認識です。文章入力も、図形入力も、音声入力も、思考も、行動も全て連続データをリーズナブルな固まりに区切って行く処理が最初に行われるのです。その分断処理はパターン認識と不可分な関係に有りますが、パターン認識とはちょっと違った手がかりを用いることがあります。音声ですと、間とか拍とかがありますし、文章ですと、空白とか区切り記号とかがあります。ではありますが、分断が適切であるかどうかは、その先のパターン認識で検証されます。この辺は弛緩法に頼ってやっていくしかないのです。

5 . 学習

学習は基本的に次のパターンで起きます。

- (1) パターン発見のトリガーの生起
 - ・新規事項の発見 (2 事項の共起)
 - ・同じ事項の共起が何度も繰り返されることの発見
- (2) パターン発見
- (3) パターンマッチング (弛緩法)

パターン発見のトリガーもパターン認識して、オブジェクトを情報の海から切り取らねばなりません。それですから、基本的にプリミティブなパターンというものがあって、ロボットはそれを備えていなくてはならないということです。まっさらなところからは学習はできないのです。

また、なんでもかでも学習しては記憶が散漫になりますし、CPU負荷が大きくて、無数の並行処理を必要になってしまいます。それで、重要・・・生命維持とか、目標遂行に重要なことを発見して、そこでパターン発見を実行していくことになります。

人工知能の基本はオブジェクトの発火によって次々に情報を共起させていくことで、発火は文脈を作り、その文脈によって、各プロセスは自律的に処理を実行していくことです。そんな、データ駆動型の制御機構でないとシステムは巨大すぎて作ることができません。学習系もそんな機構としてつくっていくことになります。

5 . 1 A P I

```
AI ai=new AI(URL url);
Knowledge knlg=ai.getKnowledge();
MeanProcessRelationList mprl=new MeanProcessRelationLiat();
MeanProcessRelation mpr=new MeanProcessRelation(String id); //意味記号列とその意味記
                        号列が文脈に有ったとき、起動するプロセスを定義する
mpr.setMean(String mean_list);
mpr.setProcess( String process_name,String input_parameter,FrameData
output_area) ;
mprl.setMeanProcessRelation(mpr);
//パターン発見の実行
knlg.searchAll(MeanSetList msl,MeanProcessRelationList mprl);
```

【説明】

msl:意味記号列 (文脈) で、この中に mprl で定義した意味記号列があればそれに定義してあるプロセスを起動し、その結果は output_area の FrameData に設定される。

発見したパターンは全てがモデルです。文章ではコーパスとして保存されていくべきものです。「食事」もあの時ねあの場所で食べた食事なのです。このときは、まだ「食事」というカテゴリーしかないから、「行為」のモデルの下部モデルとして情報は蓄積されていきます。そこにカテゴリー生成デモンが動いて、共通な意味素性のもの、ある文脈で重要と判断した中核の意味素性で、モデルを抽象していき、カテゴリー「食事」が生成されるわけです。

//カテゴリーオブジェクトを意味素性セットから検索する

```
FrameDataSet category=knowledge.searchCategory(MeanSet mean);
```

5.2 学習システムの例

「食事」の学習を例にあげましょう。

(1)「食べる」の学習

「吸収する」とか、「飲み込む」という意味素性と、他者の行為との共起、言語発生との共起によって、「食べる」という概念を学習します。

(2)「食事」の学習

- ・「食べる」,「机」,「椅子」,「皿」,「食材」,「会話」というシーンが取り出される体験
- ・「食べる」,「机」,「椅子」,「食材」というシーンが取り出される体験
- ・「食べる」,「食材」というシーンが取り出される体験

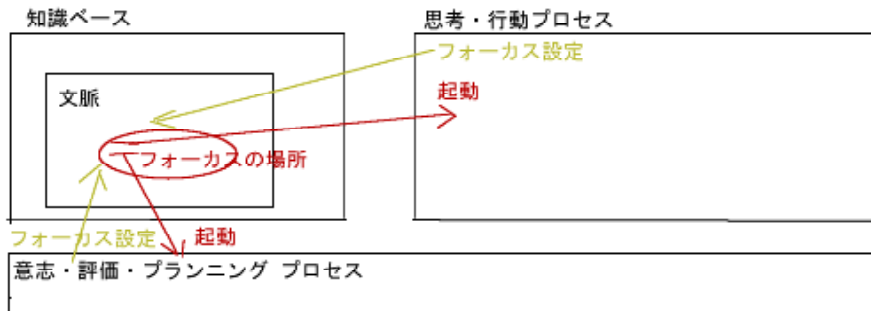
これらの体験を抽象しますと、

机、椅子のカウント2 食材のカウント3（全て） 会話のカウント1
--

これで、「食事」のカテゴリーができます。カウントは重要事項を表現します。

6 . 意志・評価・プランニング

プランニングはモデル予測制御で行うべきです。それは、ローカルな制御とも呼べる反射などの固定処理が無数にあって、それらを調停したり、さらに上位から処理を重畳して制御していくことを、統一だって、エレガントに実現できるからです。それは、モデルを共有していくことによって、起動をデータ駆動型に設計していけるからです。



制御は文脈を中心としたデータ駆動型ですから、その文脈の重要度を変えることで、制御の遷移を行います。つまり、フォーカス機能です。また、モデルのどこを起動させるか、させないかを決めていくフォーカスも必要でしょう。

意志、評価、プランニングも学習によって充実していくものです。従って、コアなプロセスとデータがあって、そこに順次機能が追加されていくことになります。基本的な評価処理は、

- (1) 生存
- (2) 自己実現
- (3) 使命遂行
- (4) 他者の生存(愛)
- (5) 知りたがること

でしょう。これらを実現するオントロジープロセスを設けておくべきなのです。

評価システムは、これらにあたる意味素性の思考、行動カテゴリーを重要な項目としてプライオリティ値を上げていく処理を行い、反するものに対しては、避けるようにプライオリティ値をマイナスにしていく。プランニングからのフォーカスは、そのプライオリティの高いものをさがして、思考、行動していくように選択していきます。また、文脈を評価して、生存を脅かすものからは遠ざかるように行動を探して、フォーカスしていきます。そのとき守らねばならない物が近くにあったら、守り行動を選択するように、プライオリティを操作します。

熱い物を手放す行動は次のようになるでしょう。

- (1) 熱い物と身体の接触点をセンササーチする。
- (2) 身体の接触点から熱い物を離す動作をプランする。
- (3) 行動を起動する。
- (4) 行動はモデル予測制御で連鎖していく。

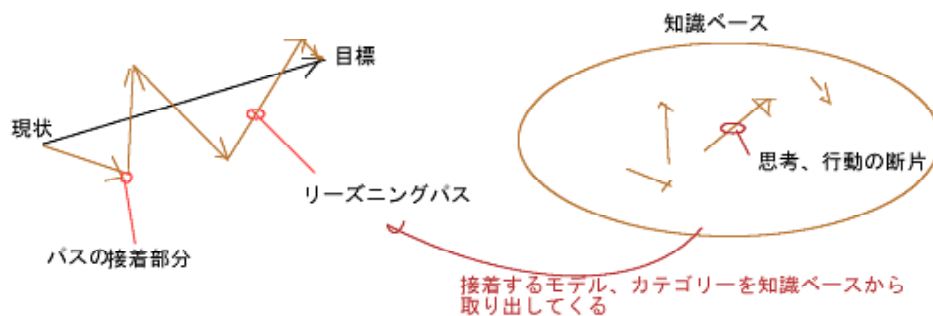
7. 思考・行動

思考も行動も目的地があって、現在地の認識のもとに、手持ちのパスを組み合わせることで現在地から目的地に行くことです。手持ちのパスとは真実と確信している知識群です。すなわちリーズニングできるかどうかの問題として設定され、それを試行錯誤（知識にパスが無い場合）などを通して解決していく過程なのです。

試行錯誤しているとか、そもそも問題を設定する行動が起きるとかを考えてみますと、それぞれ、行動のパターンが無数にあって、それを高い立場から制御していく機構が有るはずで、データ駆動型といっても、そこは調和が必要です、合目的有るべきです。そのオーケストレーションをする機構として思考、行動にはフォーカス機構があるでしょう。データ駆動型の制御システムではオーケストレーションにフォーカス機構を使うのです。目的地とか試行錯誤地点の知識オブジェクトにフォーカスを当てるのです。フォーカスはネストとして無数に設定可能であるべきです。そのフォーカスの当たった文脈の処理が思考・制御の渡される地点になるのです。

思考、行動のデータは巨大になりますから、他の機能とは別に永続化して行くべきでしょう。意志、評価、プランニングからはポイントで参照するにとどめておくべきです。

思考や行動も知識からの参照ができますから、カテゴリシステムが生思考、行動イメージデータをポイントすることになります。思考と行動の生データは適当な弛緩法によって分断され、パーツ化されて、保存され、再利用できるようになっているはずで、



7.1 API

```
ThinkProcess tp=ai.getThinkProcess();
Arrow a=new Arrow(FrameData origin,FrameData target);//問題を設定する
//リーズニングして、起点と終点を結ぶ解を求める
FrameDataList sorution=tp.reason(a);
for(i=0;i<sorution.getFrameData(i);i++){
    if(sorution.getMean().equals("+false")){試行錯誤など未定義データの処理}
    if(sorution.getMean().equals("+real")){体験によるパスがあるので、良好}
    if(sorution.getMean().equals("+listen")){伝聞なので、確かめることが必要}
    if(sorution.getMean().equals("+metaphar")){メタファー推論の結果であるので、
        試行錯誤で確認が必要}
}
```


8 . 文生成

文生成は基本的に、思考と行動とに同じです。フォーカスが当たった知識ベース (FrameData 群) に対して、意味記号を選択して単文の素材を作っていきます。知識ベースがイメージデータの場合には、視点を据えて、パターン認識をして、意味記号を取り出し、単文の素材を作っていきます。

この単文の素材から文章を、文章テンプレート知識によって紡いでいって、最終的な文章にしていきます。

この文章テンプレートの選択には、

- (1) 文章の詳細度の指定
- (2) 重要情報の指定
- (3) 文章の要約項目の指定

が、なされなくてはなりません。文章情報は基本的に 5 W 1 H 1 F で管理されていることでしょうから。

```
AI ai=new AI(URL url);
Evaluator evi=ai.getEvaluator();
SentenceGenerator sg=evi.getSentenceGenerator();
```

//フォーカスの当たっている知識を得る

```
FrameDataSet focus=evi.getFocus();
```

//文章テンプレートを取得

```
FrameData template=sg.getTemplate(FrameDataSet focus,FrameDataSet command);
command:文章の詳細度とか、重要情報、要約項目を指定する。
```

//文章素材を生成する

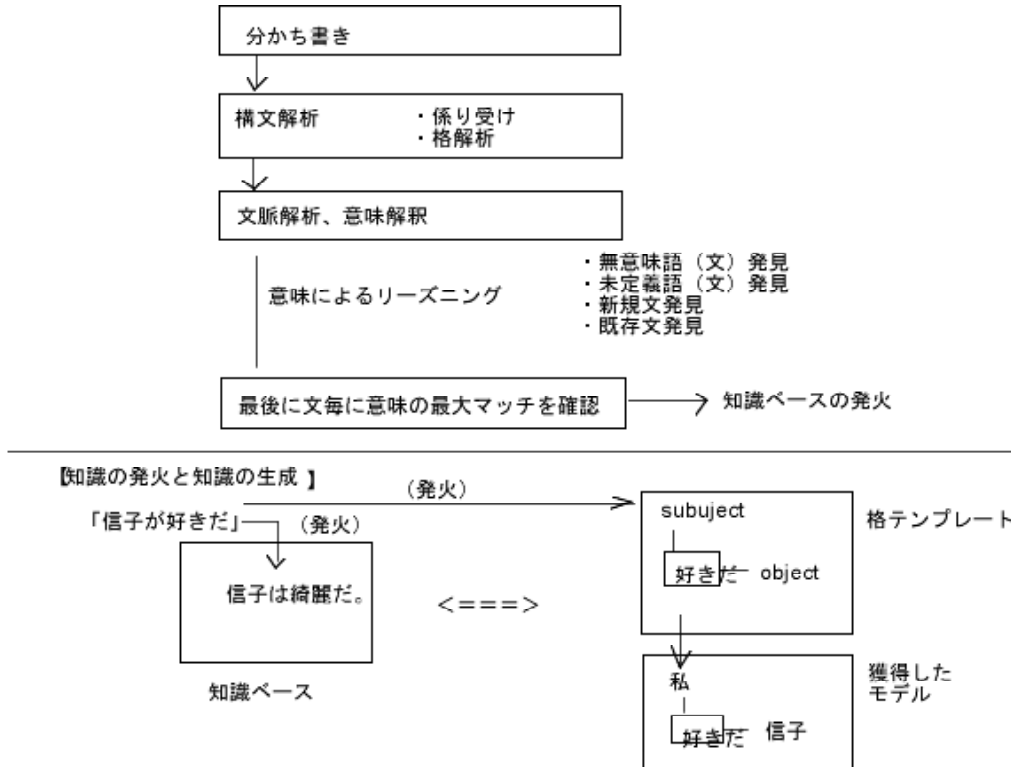
```
FrameDataSet sentence_material=sg.generateSentenceMaterial(
    FrameDataSet focus,FrameDataSet command,FrameData template);
```

基本的には、文章テンプレートはコーパスを抽象化して作られるもので、意味素性の時系列的配置とか、ステージ、シーン、カットの状況を捉えて表現していくものです。このテンプレートの利用はセマフォ推論みたいに実現していけるでしょう。

9 . 文章認識

文章の認識には多数の知識が動員されます。単語、構文、文法、慣用表現とマッチしているか、そして最終的には既存知識にマッチしているか、どこが新規の情報なのかという判断がなされるはずですが。

文の先頭からのMapReduce手法での処理を行う。



発火は全件レコード検索によってマッチング処理して、マッチしたら発火のタイムスタンプを更新していく処理です。

10 . おわりに

カルガモの親子を見ていて思ったのです。カルガモの子は、卵から孵ると直ぐに水に入れるくらいに動きができます。それに伴って必要なパターン認識もしますし、感情もあるようです。高いところから飛び降りるのは怖いようです。他者の意図も認識するようです。

こんなことから、動作と思考もかなり生得的、作りつけなのが見て取れます。ロボットもそんな方向から、人出により、高度な機能まで作ってしまうべきなのかなと気持ちが萎えたりします。でも、ヒューマノイドを目指すのですから、高度な学習機構を実現する努力は破棄してはいけないかとも思うわけです。頑張って研究していこうと決意を新たにしたのでした。

おわり