

知識ベースへのプラグインの一つの例として日本語処理を考えてみたい。日本語処理としては次のような機能がある。

- (1) 文章認識
- (2) 問い合わせ回答
- (3) 文章生成
- (4) イメージ（ボクセルデータ表現）をパターン認識して意味記号化する
- (5) 意味記号からイメージ（僕セルデータ表現）を生成する

1 . 文章認識

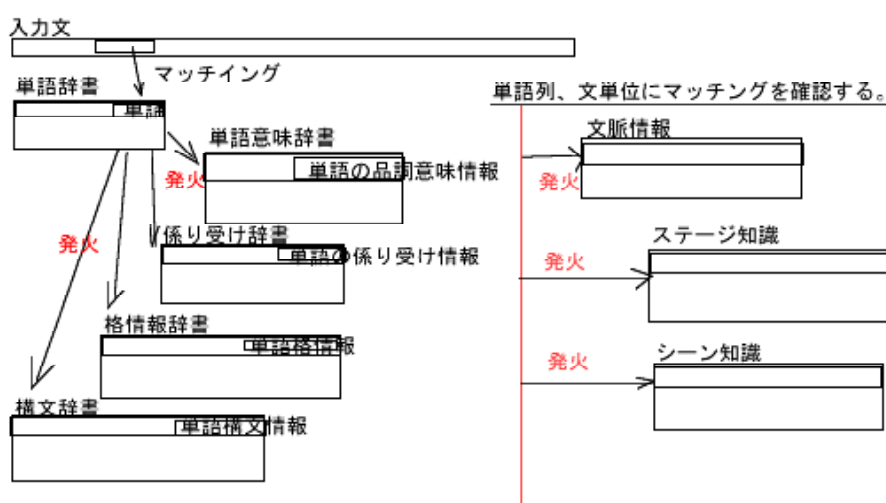


図 文章認識概要図

文章認識も「発火」を基本に構築していく。長文や長い文章を解析するときのメモリ制約や処理時間制約を考えると、文頭から逐次、文脈やステージ、シーン知識と照合し、現在の状況を実時間で把握しながら文章を解析していく。できるだけ早い段階で、文意を決定していく。

最終的にはコーパスモデルを作っていく事になる。知識の取り出しはコーパスからのオンデマンド解析によって必要に応じて行っていくものとする。カテゴリーなどはこのオンデマンド解析で充実していくことになる。また、ステージやシーン、アクターのモデル知識の充実もオンデマンドでやっていくことにする。

文脈は、欠落語や指示代名詞の同定、表現の微妙な違いの対応に必要な知識となる。文脈はカレントの知識、アクティブ（発火時点に近い）な知識であるから、他の知識より優先されるのである。

【API】

```
Model sentenceRecognizer=new SentenceRecognizer(AI ai);  
//文章認識プラグインモジュールを定義する
```

文章認識プラグインの中では、以下のモジュールを使っていくことになる。

```
SentenceAnalyzer sa=new SentenceAnalyzer(AI ai);// 1文解析モジュールの設定
```

FrameData には、文章も XML 形式で入っているので、それを 1 文ずつ取り出す必要がある。

```
Iterator sitr=fd.getSentences();  
while(sitr.hasNext()){  
    String sentence=sitr.next();  
    FrameData fd=sa.analyze(String sentence)// 1文を解析する  
}
```

2 . 問い合わせ回答

学校に行って、校庭で遊んでいたよね。A 君に何貰ったの？

というような文章では、「学校に行って」「校庭で遊ぶ」が文脈構築である。この文脈から過去のステージ、シーンモデルを発火させて、答えに結びつくコーパス文章・・・知識部分を発火させることが出来なくてはならない。知識の中に、

「A 君にボールをぶつけられた」「A 君は私にハンカチを渡してくれた」などの知識があれば、「渡す」=「貰う」の方向違い・・・というような知識によって、「A 君は私にハンカチを渡してくれた」を選択し、そこから知識を抽出して、「A 君にハンカチを貰った」という文を組み立てて表出する。

このように、問い合わせ回答でも、「発火」と基本的な知識加工プロセスがあって実現できる事がわかる。知識加工プロセスは学習によって充実していくべきで、コマンドパターンからの実行が可能なプロセスも作り込む必要がある。コマンドパターンは思考システムで生成していくことになる。

【API】

```
Module question_answer=new QuestionAnswer(AI ai);  
//問い合わせ回答プラグインモジュールを定義する  
question_answer.execute(FrameData question);//問い合わせ実行  
//問い合わせの結果は意志システムの待ち合わせに回答として保存する(問い FrameData に結果として回答 FrameData を連想させる)
```

3 . 文章生成

知識要素に振られている重要度の高い物を選んで文章化していく。また、文脈に登っている知識を優先に文章化していく。文章化か知識記号を単語に変換したり、文法、構文を連想発火して取り出し、シームレスになるようにシステミック文法で繕って、最終文章の形を決定し、表出していく。

システミック文法としては、

- ・重要事項を先に置く
- ・時間、場所、事象の順に言葉を紡ぐ
- ・主題は助詞「は」を用いる
- ・疑問は「かもしれない」を用いる

などなどである。係り受け、係り結びに注意して単語列を決定していかねばならない。

【API】

```
Module sentence_generator=new SentenceGenerator(AI ai);
```

```
//文章生成プラグインモジュールを定義する
```

```
sentence_genertor.execute(FrameData fd);//FrameData fdは文生成の種になる知識要素。ここから連想によってどんどん物語を広げていく事になる
```

4 . イメージをパターン認識して意味記号化する。

ステージ、シーンを解析していかねばならない。また、物の形状とか位置関係を解析していかねばならない。イメージの全ての状況、情報を記号としてもつのは不可能である。情報は無限にあり、視点によって情報は異なる記号で表現されもするからである。「与える」と「貰う」の関係とか、「右」と「左」の関係とかいろいろ視点のもつ問題はある。これらはオンデマンドでイメージを解析くることによって記号化し、文脈によって記号発火されねばならない。

5 . 意味記号をイメージ化する。

視点変換の為に、いったん意味記号をイメージ化しなくてはならない。

イメージ構成は、イメージ要素をアセンブラしていく過程でもある。また、イメージはロボットが生活していく中でコーパスモデルとして取り込んでいく視覚画像でもある。2次元イメージを3次元イメージ(ボクセル)に構成して、知識化していく過程も学習システムにあるということである。

6 . おわりに

文章認識も、文章生成も、イメージをパターン認識して記号を得ていく過程も、意味記号からイメージを得ていくことも、学習システムによって能力を充実していくことが必要なため、コマンドパターンで動くプロセスを設計しておく必要がある。そうして、コマンドパターンを得ていく過程は思考システムである。思考システムをトレースするデータをコーパスとして収集していき、それを抽象化して、コマンドパターンにしていくのである。

おわり