考察：「オントロジーと自然言語処理」

「オントロジー工学」を読んでいて、自然言語処理を素直に展開したら面白いなと感じて、この小論を作りました。

参考にした文献は下記のものです。
・オントロジー工学　　溝口理一郎著　オーム社
・セマンティック・ウェブのための RDF/OWL 入門　神崎正英著　森北出版

第１章　はじめに

オントロジーとは「存在概念」といいますか、「意味が明確に定義された語彙体系」という意味です。「オントロジー工学」では、「オントロジーとは、現実世界に存在するものを説明するためのカテゴリーの体系である」としています。

図１．１ オントロジー世界

オントロジーとして、上位オントロジー（哲学の世界からの視点でもののカテゴリーを整理した)、デバイスオントロジー（人工物の機能カテゴリーを整理した）とか幾つかが作られているそうです。

上位オントロジーの項目を少しあげてみますと、

基底
　・空間　　・時間　　・物質
実在物
　・具体物
　　・物
　　　・形態物
　　　　・立方体
　　　・機能物
　　　　・生命体

・人工物
・意志所有物
・社会
・プロセス
・抽象物

（等々）

表１．１　上位オントロジー（一部）

これ、著者が「意味処理の考察」であげた、意味プリミティブと同じような思想で作られていると思いませんか。それで、興味を持ってオントロジーの勉強をしたのです。その想いは、オントロジーを自然言語処理の中に組み込めないかなというものです。オントロジーは結局カテゴリーシステムなのですから。

オントロジーにおいて語彙の定義は、RDF という XML 表現で、主語、述語、目的語の三組で行います。例えば
（例）「田中家はペットに猫を飼っている」
というのを RDF で表現しますと、

```
<rdf:Description rdf:about="田中家">        ・・・・・・（主語）
    <ex:hasPet>                              ・・・・・・（述語）
      <rdf:Description rdf:about="http://www.animal/cat"・・・・・（目的語）
      </rdf:Description>
    </ex:hasPet>
</rdf:Description>
```

ここでの rdf:Description、ex:hasPet、rdf:about などのタグや値は、URI によって示される名前空間によって、意味づけられます。意味プリミティブという概念はありません。意味が一意であることを保証するだけです。

セマンティックウェブというものがあります。ウェブサービスのインテリジェント化ということで、メタデータ（XML タグ）をウェブ文書に徹底させて、計算機の処理の知的レベルを向上させるというものです。このメタデータとしてオントロジーを導入するのです。それで、オントロジーとセマンティックウェブは対で語られる場合が多いのですが、オントロジーはウェブに閉じた物でなく、明確に用語を用いたいところに導入されています。

つまるところ、セマンティックウェブとは、次のような検索というか、問いをウェブに発することができるシステムといえると思います。

（例１）英国首相官邸にいた猫の名前は？
（例２）地球の年齢は？

例１だと、ウェブの文書の中に、次のような記事を見つけて、意味解析をすることになるはずです。

Humphrey,a stray cat who wandered into the official residence of Britain's prime minister in 1989 and caused a scandal when he "retired" in 1997,has died,a spokesman for Prime Minister Tony Blair said.

・・・・・・・・・

He had wandered into No.10 Downing Street under Margaret Thatcher and remaind throughout the tenure of John Major. But he was sent away to live with a civil servant in "retirement" months after Tony Blair was elected in 1997.

・・・・・・・・

（Asahi-Weekly）

この問いに答えるには単なる語彙の検索ではできません。語彙の意味を計算機が認識できて、「言い換え」にも対応できなくてはなりません。

オントロジーを自然言語処理に融合して、セマンティックウェブを実現するとなると、次のような技術体系に解を与えることになります。

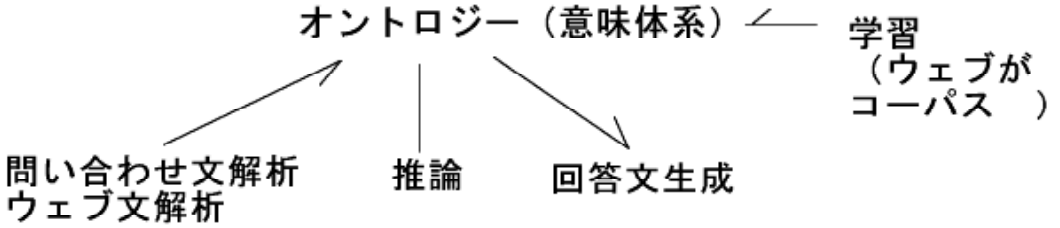

図１．２　セマンティックウェブ技術体系

自然言語処理ですから、語彙の定義（学習）、文解析、文生成、文脈処理、推論を考察していかねばなりません。

以下に、オントロジーで考察すべき事柄として、以上のことを詳述していきます。

第２章　文脈定義

文脈を規定するものは、次のものが考えられます。

・表現に関する状況

文頭、文末、句切り記号などとの関連。例えば、

（例１）３以上の数値

（例２）以上のことから・・・

（例３）・・・・　　以上

例１、例２、例３の「以上」は表現から解析すべき意味の違いがあります。

・意味に関する状況

前後の意味との関連。例えば、

（例４）ダイヤモンドを買った。

（例５）ダイヤモンドを一周した。

例４では、「買った」から、鉱石のダイヤモンドで、例５では、「一周した」から、野球のダイヤモンドであることが分かります。

・プライミング

前後の語彙との関連。例えば、

（例６）空をかける。

（例７）運動場をかける。

「翔る」、「駆ける」の違いですね。意味に関する状況とも言えますけれど、脳の重要な機能なので、別立てて考えていきたく思います。

文脈を表現したい。どんな方式があるか。考えられるのは、

（１）疑似自然言語表現（タグ付きの情報豊富な表現）

文の表現をそのまま保存しているので、文脈の「表現に関する状況」に対応しやすいという特徴がある。一方で、語順が不要な抽象化に難がある。

（２）フレーム表現

「表現に関する状況」に対応できないが、抽象的なため、推論に向いている。

（３）語彙セット

プライミング処理に適する。

ということで、３つとも必要で、それを作業品詞（文脈空間）に表現してブラックボックス化することが使いやすさの面で優れているようです。

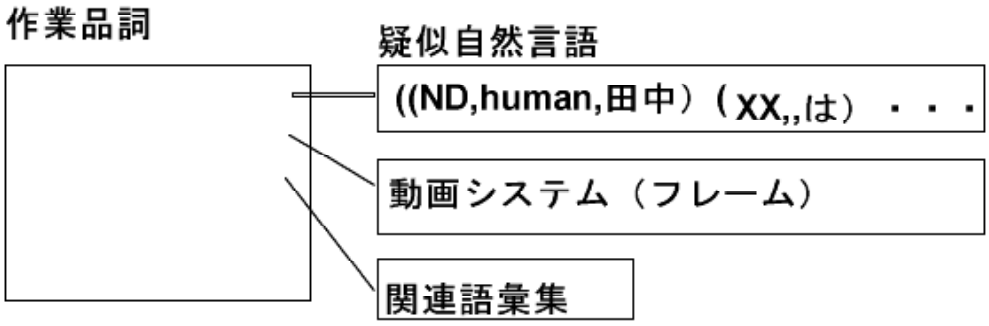

図２．１　文脈の表現

疑似自然言語は XML でも、フレームで表現してもいいのですが、語順が保たれていることが一義的に重要です。ここでは、タップルで表現しています。
タップルの意味：　（品詞、付加情報、語彙；決定情報）
ここで、決定情報は曖昧性の絞り込み情報などの、解析プロセスで生成する情報を格納するものです。

フレーム内の記号は原則的に意味プリミティブですが、記述が重たくなるので、マクロというか、意味プリミティブのフレームで表現された拡大した意味要素も定義して、利用できるものとします。

猫：生き物、肉食、小柄なほ乳類
ですと、
cat:life
:get meal　　meal: material that is part-of animal
:small mammal　　mammal:animal that make decendant in one's body

こんな情報を意味記号定義辞書を創設して格納し、利用していくことになります。

フレームは時間、場所、状況描写、イベント描写という、ステージ、シーン、カットからなる動画システムを適用します。マップシステムを使うこともあるでしょう。

語彙セットは内容検索できるようにインデックスを張られて、まとめて管理されている関連語彙テーブルです。

第３章　語彙定義

語彙の定義は次の体系となるはずです。

（１）文法定義

・名詞、動詞、形容詞、形容動詞、副詞とかいった品詞情報

・修飾・被修飾情報

（２）意味定義

・プリミティブ意味

（例１）歩く：move

・文章による意味定義

（例２）水泳：水の中を浮いて移動すること

・イメージ（マップ）による意味定義

（３）文脈定義

・文脈情報を持つ。文脈パターンをキーとして、その時の意味をオブジェクトとしてテーブルで持つ。

文章による意味は抽象度の高いフレームで持つのがよい。推論がしやすいから。

第４章　文パターン定義

文パターンは語義を決定するので、文脈情報の一部であるが、特徴が顕著なので別項目としてみました。次のものが有ります。

（１）SV、SVC、SVO、SVOC、SVOO、SVA、SVOA などの文型（英語の流用）

「てにおは」のパターンで、例を挙げると次のようなものです。

・猿はバナナが好きだ。(SVC)

・ツバメは空を飛んだ。(SVA)

・母はナイフでリンゴを切った。(SVOA)

・君にラジオをあげよう。(SVOO)

・・・・・・・・・・

このパターンは主に格情報を決定します。しかし、曖昧性を持っているため、意味は複数取れること、そして、漸近的に曖昧性を取り除けるようにデータエリアの表現を工夫していく必要があります。

（２）連結動詞パターン

「いる」、「いく」、「くる」、「みる」などの補助動詞のパターンです。

・用意してくる

・こうしてやっていく

（３）文法化パターン
「～なければならない」、「かもしれない」などのパターンが決まった表現です。

（４）文章パターン
文章パターンは文体とも呼ばれます。文章解析、文章生成には必須の情報です。文章から動画システムを作るとき、逆に動画システムから文章を組み立てるときに、意味を配置する設計書となるものです。

これらのパターンは語彙と同じレベルまたはその上位レベルの情報として利用します。パターンに語彙を埋め込んだものがパターンのインスタンスで、このインスタンスを評価するのですね。

パターンは文脈定義で示した疑似自然言語表現です。

（例１）　パターン識別子：SVO
パターン記述：((ND,human,;actor+focus)(XX,,は;))((ND,object,;)(XX,,を))
(VB,,;)
【説明】「～は～を～している」は「～は」の名詞が主語になったり、フォーカスになったりする・・・ということを示している。

（例２）　パターン識別子：している
パターン記述：((VB,,；-ing)(XVB,,している))
((VB,,；completion)(XVB,,している))
【説明】「～している」は動詞が進行形になるか、完了形になるかのどちらかであることを示している。

第５章　文解析

文解析は基盤層（文法層）と意味処理層、文脈処理層に分けるのが良いようです。

（１）基盤層（文法層）

格の配置、修飾・被修飾、並置をシステマティックに処理します。基本的にはデフォルトの解釈をしますが、特別な記号を語彙が持っていたら、その指示にしたがった処理をします。主にそれは、修飾・被修飾関係と並置関係を変更して再処理するのに使います。

（２）意味処理層

意味の的確性を判断する層です。事実データベースや文脈情報にマッチする意味となる文、句、単語を的確します。非文となるならば、文法層の再処理を依頼します。

（３）文脈処理層

文章の文脈を紡ぎます。必要ならば文章を再処理して文脈が一貫性を持つように保ちます。

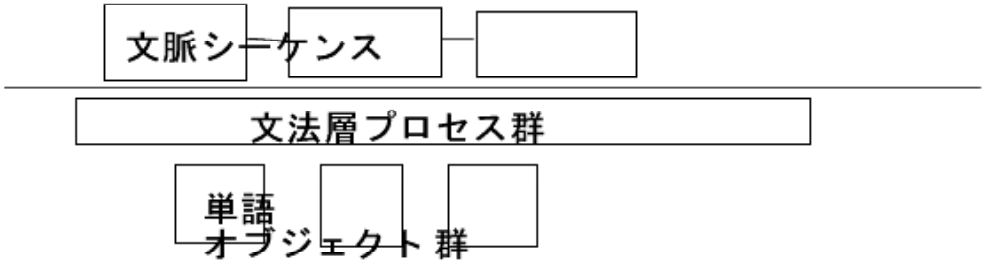

図５．１　処理構造

意味処理層で行う処理は、修飾先の変更が主なものです。修飾・被修飾のパスは入れ子構造が厳しく、クロスしないように再度全体のパスを張り直すことになるはずです。

また、意味層は擬人化とか、視点の位置、スケールといった文脈を考慮した判断が求められることがありますので、文脈層と深く関わってきます。文脈層のデータエリアも見るようにしなくてはなりません。

第６章　文生成

文生成は次のような手順で知識から、文章を組み立てます。知識は意味フレームで表現されているとします。

（１）語ることを選別します。選別した物をフォーカスとします。

（２）文章パターンによって、フォーカスの移動計画を作ります。

（３）フォーカスの当たったフレームから文法パターンにより単文を生成します。

（４）単文を結びつけます。このとき、文を変形したり、特別なパターンに則った語を生成したりします。

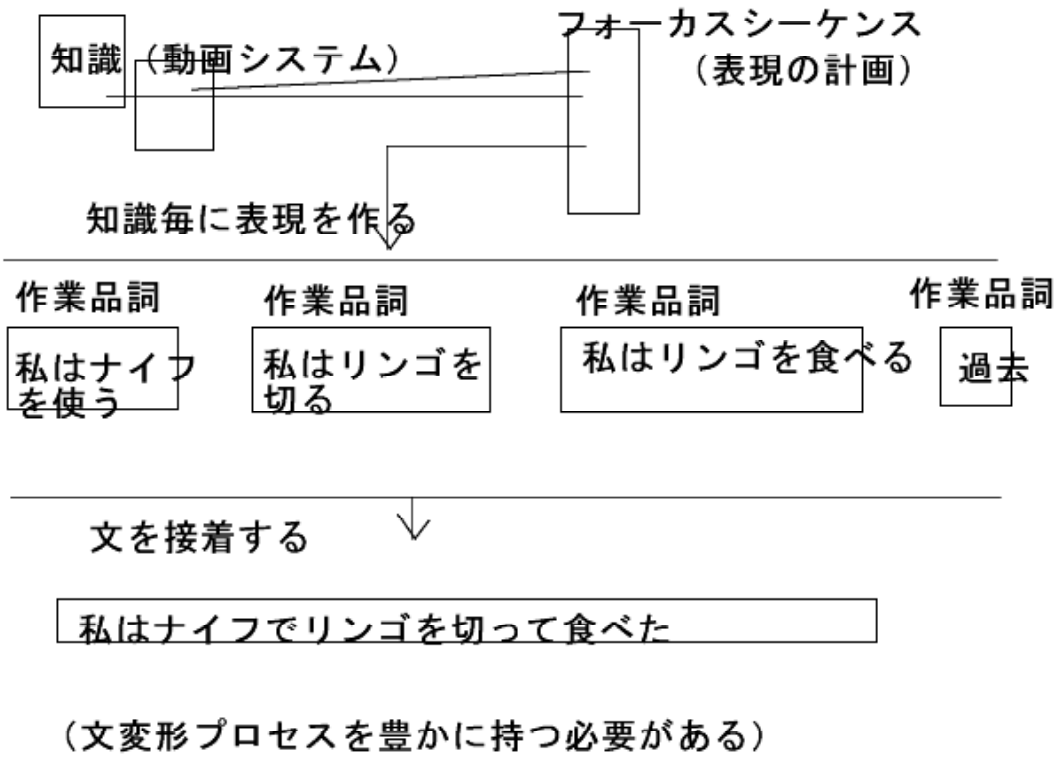

図６．１　文生成プロセス

第７章　推論

推論は、語彙文法学習時とか、文章解析とか、問い合わせ回答、文章言い換えのときに必要となる機能です。

（１）語彙文法学習時

パターンやカテゴリーを作りながら、その相同性によって新しい語や文法の意味を推測します。

（２）文章解析、問い合わせ回答時

意味フレームをたどって必要な情報を得ます。

（３）文章言い換え

意味プリミティブを中心に解析していき、求められるフレームを新規につくることを行います。

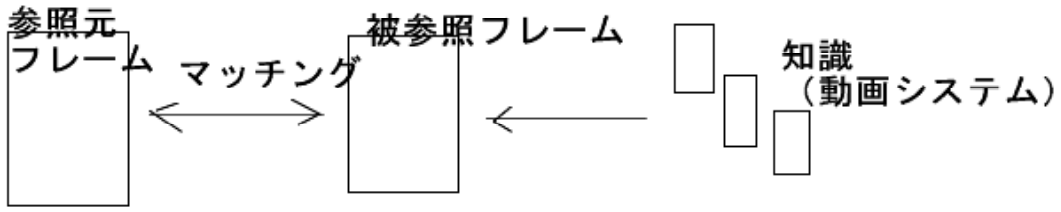

図７．１　推論の基盤（マッチング）

文章言い換えとは、次の例のようなものです。

（例文）私は長野で新幹線に乗った。列車は軽井沢を通って、大宮についた。大宮から東北新幹線に乗り換えて、仙台に行った。仙台は七夕祭りの最中だった。綺麗な短冊がいっぱい道に溢れていた。

この例文を次のように新しく表現するのです。

「私は新幹線で仙台の七夕を見に行った。」

第８章　学習

学習はウェブ文章と辞典をコーパスとして実現します。

文章は、状況文と定義文、条件文からなりますから、この３パターンを基本手がかりとして、学習を進めることになります。

（１）状況文

・私は学校へ行った。

・花は美しく咲いていた。

（２）定義文

・学校は勉強するところである。

・花は美しい。

・動物にはイヌ、猫がある。

（３）条件文

・雨が降ったら傘をさす。

・午後になったら、お弁当だ。

定義文は語義に直接影響しますが、状況文、条件文は補助的な例題として語義に添えられるものです。

８．１　語彙の学習

著者はパターン認識による語彙の獲得が本筋と信じるものですが、インターネットのテキストデータをコーパスと位置づけ利用することも有用と考えるところです。そこで、コーパスからの語彙獲得を具体的に考察してみます。

てにをはパターンが手がかりとなります。

（([名詞]は[名詞]が[名詞]を[名詞]に[名詞]で[名詞]へ・・・）[動詞][助動詞]）

（１）種となる単語とその品詞、文法情報、意味プリミティブを人手で定義していく。３０００語彙は必要かもしれません。

（２）未知の語彙の品詞を、コーパスの中での文例パターンに当たって、助詞、助動詞と活用語尾を手がかりに推定していく。

（３）未知の語彙の文法情報を、コーパスの中でのパターンの相同性（既知パターン語彙のパターンとの）を解析し、推定していく。

（４）未知の語彙の意味を、コーパスの中でのパターンの相同性を解析し、推定していく。必要ならば新規カテゴリーを定義していく。

（５）最後には教師によるレビューを受けて、品詞、文法情報、意味、カテゴリーを確定する。

「魚」という抽象概念に至る解析手順を考察してみます。次の表のようなコーパスがあるとします。

【海コーパス】海にイルカがいる。
海にイワシがいる。
海にタイがいる。
【川コーパス】川にコイがいる。
【イルカコーパス】海にイルカがいる。
イルカは泳ぐ。
イルカは体温を持っている。
【イワシコーパス】海にイワシがいる。
イワシは体温を持たない。
イワシは泳ぐ。
【タイコーパス】海にタイがいる。
タイは泳ぐ。
【コイコーパス】川にコイがいる。
コイは泳ぐ。
【ライオンコーパス】ライオンはジャングルにいる。
ライオンは歩く。
ライオンは走る。

表８．１　分類されたコーパス

コーパスパターンの比較によって、次のことが言えます。
（１）イルカとイワシの相同：海にいる
泳ぐ
イルカとイワシの相異：体温をもつ/もたない
（２）イワシとタイの相同：海にいる
泳ぐ
（３）イワシとコイの相同：泳ぐ
イワシとコイの相異：海にいる/川にいる
（４）イワシとライオンの相同：なし
イワシとライオンの相異：泳ぐ/(歩く、走る；＝意味プリミティブ move)
・・・・・・
【解析】
「海にいる」と「泳ぐ」の相同でカテゴリーを作れます。
「泳ぐ」という相同でカテゴリーを作れます。
「(歩く、走る)」という相異でもカテゴリーを作れます。

・・・・・・・・

こうしてみますと、カテゴリーを沢山作っていけば、様々な視点が取れるようになります。多くは語彙の分類としてはゴミですが、知識としては保持していて損はないはずです。メモリ食いますが、思考をプログラムで実現しようとしたときベースとなるデータを形作ります。

この他にコーパスから、「泳ぐこと」が生活の基盤である生物があることを検出することができれば、それを「魚」というカテゴリーを創設することができるわけです。どの語彙が重要な概念であるかを計測していく必要があります。単語の共起頻度分析でいいとおもいます（学術文献のコーパスは信用できるが物語はフィクションがあって、学習には適さないというような考慮も必要でしょう）。

「魚」というカテゴリーが既にあり、「タイ」というカテゴリーが無かったとします。「魚」と「タイ」のコーパスの一致度が高いので、「タイ」カテゴリーは「魚」カテゴリーの一つとして、タイという語彙の意味に魚の意味プリミティブもしくは、意味マクロを振ります。と同時に、カテゴリーツリーを作ります。

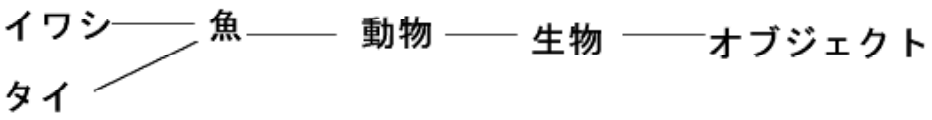

図８．１　カテゴリーツリー

意味記号には、カテゴリーツリーの識別子と最下位のカテゴリーの識別子の組で管理するのが良いでしょう。カテゴリー分類上での推論が行えるようになるからです。カテゴリーツリーは辞書に登録し、更新があっても古い物はそのまま残し、どのカテゴリーツリーでのカテゴリーであるかが分かるようになっていることが重要です。メンテナンスは複雑にしたくないものです。

カテゴリーに付属するコーパスも核部分と追加部分に分けて管理します。核部分のコーパスはこのカテゴリーの正規の定義項目として利用し、追加部分は核で考慮していなかったものになるので、将来のカテゴリー内容変更のための資料にしていきます。

８．２　文脈の学習

文脈とは、文章の曖昧性を処理するための機構です。文章の曖昧性とは、異なったことを言おうとしたとき、結果として同じ文章になるという現象から生まれます。これは、言語表現パターンが意味表現パターンより少ないことによります。言語表現パターンは物理的な制約がありますが、意味表現は制限がありません。だから納得いくところです。

例えば、意味「信子は明雄のことが好き」というのを言語表現で、

（１）信子は明雄が好き。

とします。
　一方で、意味「明雄は信子のことが好き」というのを言語表現で、
（２）信子は明雄が好き。
と強調点を「信子」にして表現ができます。ニュアンスは確かに違うのですが、結局（１）と（２）とでは同じ表現となり、曖昧性を生じてしまいます。

　このように、文脈の学習は本質的に、文を生成していく体験が必要なのです。自然言語処理システムが獲得した文法パターンを用いて文章を生成するときに、獲得したパターンからは表現の重複があると気づいたとき記録していき、それを分析して文脈情報としていくのです。自動学習はそうするしかありません。だから、文脈学習には、生成した文とその意味をコーパスとして蓄えて行くシステムが重要ということが言えます。

第９章　セマンティックウェブの処理例

　（例）英国首相官邸にいた猫の名前は？

　手順としては次のようになります。
（１）「英国首相官邸」の記事を探します。
（２）選んだ記事からさらに「猫」のことを言っているらしい記事を選別します。
　「猫のことを言っているらしい」というのは、代名詞をたどるとかの高度な意味処理が必要になります。
（３）「猫」の「名前」を言っている文をさがします。
　「名前を言っている」のは、「his name」とか「is Xxxxx」とかの表現を探して判断します。プリミティブな意味を汲まないと実現できないところです。
（４）名前を同定します。これは文章の意味解析を行って実現します。

第１０章　まとめ

　セマンティックウェブを実現するには今のオントロジーではできなくて、もっと深くて大きなシステムが必要であることがご理解頂けたでしょうか。特に、意味の基本定義を URI による名前空間としていては、汎用的な意味処理プロセスを作ることができません。ここは意味プリミティブを導入すべきところでしょう。意味プリミティブは上位オントロジーを拡張して、哲学でなくて、人間の脳の仕組み（クオリア）に立脚したものにすべきです。

　さらに、自然言語処理として地道で忍耐強い研究が待たれます。

以上