

文章にはアクターの行動を記述するものと、アクターの属性を陳述するものがあります。その他に、人間の場合、「思い・感想」とか「感覚・感情」があります。「思い・感想・感覚・感情」は行動というよりも陳述だと考えます。無論この陳述の中に行動が、そして動画システムがあることもあります。

(1) 行動記述の例文

「パンフレットを詰め込んだいくつもの封筒と脱いだスーツの上着を抱えたおじさんが、受付の台に立った。あともう少しでお昼の休憩という、前半でいちばんだるくなる時間帯でぼんやりと座っていたわたしと長田さんは慌てて立ち上がった。東京での展示会の仕事は5回目の長田さんが、笑顔を作って台の上の芳名録に手を添えた」(フルタイムライフ 柴崎友香著 マガジンハウス)

説明)「～した。～した。～した。」と文をつなげている。動画システムを形成するパターンです。

(2) 思い・感想の文

「実機を見たのは新入社員研修の工場見学と展示会の時だけで、もちろん実際に使ってみたこともないわたしは、どこがどう良くなったのか実感できなかつたけれど、同じシリーズの今までの機種よりは色も明るい水色で操作盤がすっきりして見栄えはよかった。こういう産業用の機械は受注先の要望や条件によって仕様が変更されるので、工場に完成品がずらっと並んでいるわけではない。それでも、製造途中の機械があちこちにありパイプや電線が壁や天井を走る工場の風景は、何度でもわくわくする」(フルタイムライフ 柴崎友香著 マガジンハウス)

説明)「～はよかった。～わけではない。～わくわくする。」と陳述が続く。それは、思い・感想・感覚・感情といったことを表す動詞が使われていることから言えることで、動詞がキーポイントであることを示す。これは動画システムを構成しない。単なる属性を提示する陳述になっています。追記しますと、この例では工場の中のオフィスがステージ(フレーム)であることが見えてきます。それは出てくる単語が個性的で、そこからステージを推測することが容易だからです。

(3) 事実記述の文

「だが、人間には、特別な事情がある。動物や植物、様々な自然現象、人間の作った人工的な機械の一部にも存在するかもしれないが、現時点では、私たち人間だけにしか存在することが確認されていない特殊事情がある。そう、私たち人間は、「心」を持っているということだ。朝目覚めると、それまで何もなかったところに、「私」の意識が生ずる。意識のレベルが上がってくるに従って、私の心は、はっきりとした形をとりはじめる。そして、私の心のなかには、様々な表象が現れはじめる」(クオリア入門 茂木健一郎著 ちくま学芸文庫)

説明)「～がある。～だ。動詞現在形の並び」と陳述が続く。動詞「だ」、「ある」など

は陳述を表すことを示しています。また動詞の現在形も陳述で動画システムを構成しないようです。というのは、動画システムを構成しているようにも解釈はできますから。多分、前後の状況から、判断されるのでしょうか。陳述モードなら現在形は陳述で、動画システムモードなら現在形は動画システムモードになる。

そもそも私の「動画システム」はアクターの行動を記述するのでも、陳述、考え、感情といったことを扱えないため、文章の意味理解はまだまだ考えていかねばならないことが多数あります。今回は意味理解に精一杯大きく踏み込んでいきたいと思います。

(4) 語りの文

「一七七七年、二十一歳のモーツアルトは、一家の希望を負い、音楽による名声獲得の為、母親と二人で大旅行の途につく。翌年の夏、パリ滞在中母親が死ぬ。

不幸のあった夜、モーツアルトは、同時に二通の手紙を故郷に書き送った。一通は父親宛、一通は友人のプリンガア宛である」(モーツアルト・無情という事 小林秀雄 新潮社)

説明)「～つく。～死ぬ」と現在形が続くがこれは陳述ではない。最後に「書き送った」と過去形で締めくくって、事実の記述であることを明記しているからです。この文を例にしたのは、さらに意味理解を深く議論したかったからです。「大旅行」が「パリ滞在」を含むことを認識できなくてはなりません。「母親が死ぬ」が「不幸」と結びつかねばなりません。「書き送る」が「宛」と同義であることを知らねばなりません。こういうことを知識ベースは記しておかねばならないのです。そんなことに注意しましょう。

(独白)同義語・異義語とはベクトルのように方向性をもって管理するとエレガントになるのではないのでしょうか。「深く議論した」と「注意しましょう」は同じ方向を向いていますね。「散漫な処置でいいです」となれば、別の方向(180度)を向いている。「神経を使う濃度が濃い」をベクトルAとすれば、(神経、薄い)は-Aと表現できます。

第1章 文章の分類

意味を理解するとは、アクターが何をするのかということとアクターの属性を陳述を理解することです。属性はアクターが何色で、何を考えたり感じたりするかとか、あれは何でこれは何でとかからなります。「リンゴの木に登った明ちゃん」ですと「リンゴの木に登る」ことも属性になります。「思う」や「想像する」はアクターの行動でなく属性であると分類するのは、事実と想像、感情といったものを分離した方が問い合わせに答えることを想定すると重要な事柄に成るからです。無論、「思う」も「想像する」も目的格に動画システムをとることは十分に考えられることです。

(例文)「多分彼女は映画を見た。それから食事をして、電車に乗った。私はそうにらんでいる。」

なんて文が考えられます。でも事実ではない。「私」の属性でしかないのです。属性の内部に動画システムが埋め込まれているのですね。

行動	陳述		
	陳述	意志	推量・感情
~した ~している ~する~した ~なる(モード 次第) ~する(モード 次第) ~したとき~ した ~するとき~ した	~だ ~だった ~なる(モード 次第) ~する(モード 次第) ~したとき~す る ~すると~する ~するならば~す る ~するならば~し た ~したならば~す る ~したならば~し た	~にする ~したい	~思う、思った ~感じる、感じた ~して欲しい たぶん~ もし~ 絶対~ ~かも ~か

表1.1 文の分類用キー単語

結局文とは次の枠組みを埋めて把握していくべきものではないでしょうか。

基盤ステージ	アクター	記述内容	モダリティ	テンス
時間 場所 論理	当事者 話題者	属性 行動	事実 意志 推量、感情	過去 現在 未来 完了 継続 表明だけ

表1.2 文の意味の枠組み

第2章 文章の認識の構造

文は名詞があり、動詞を要として格によって配置され、事象を陳述します。名詞は基本的にアクターとなります。椅子も、机もみなアクターです。文毎に名詞を認識するだけでは簡単なのですが、文章となりますと、文に使われている名詞の間で相同を認識する必要があります。更に知識ベースの名詞とも相同性を把握していかねば成りません。だから文章とは切り離して名詞をアクターとしてまとめて管理していく必要があるのです。動詞の相同については、意味の包含関係だけを考えればすみますから、結局、時系列の動詞とその格の固まり（以降「カット」と呼ぶことにします）とアクターを別にして管理していくという方略になります。

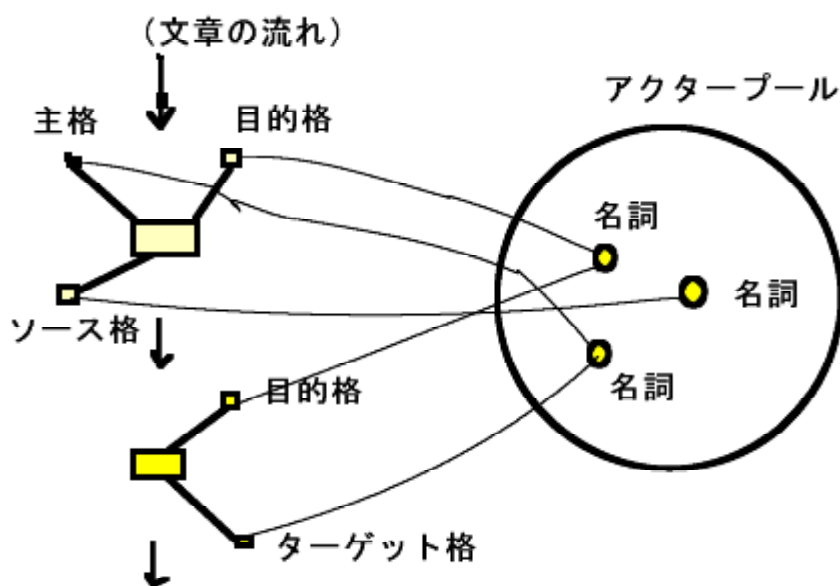


図2.1 文章の要素の管理

アクターの管理は注意を要する。同一性の範囲というものがあるからです。「私」とか固有名詞ですと、どんなステージでも同一性は単語が等しいかどうかで判定できますが、「椅子」とか「テーブル」、「人物」という一般名詞はステージとかシーンに固有になるでしょう。ステージが変われば、別のものを指すことになります。

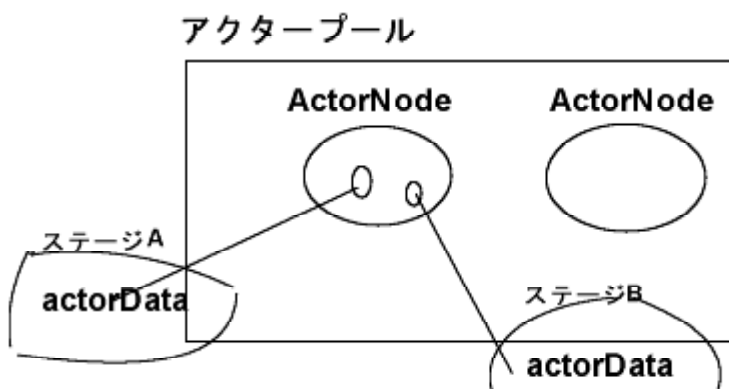


図2.2 アクターの相同性の管理方法

動画システム内のアクターはアクタープールでは ActorNode というエンティティを設けて管理します。同じアクターは同一の ActorNode に含めるようにすれば、簡単に相同性が管理できます。

文章を認識するために必要な知識とはどんなものがあるのでしょうか。動画システムから展開しますと、ステージ、シーン、カットの知識ということになります。文章がどういう状況で語られているかについては、ステージとそれに付随する、フレームが重要でしょう。欠落語、曖昧語の認識には(ステージ、シーン)フレームの下のカットの情報が必要でしょう。ステージの切れ目、シーンの切れ目は明確でないかもしれませんから、ステージ、シーン、カットの単語、格構造から推定していくことも必要になります。そんな目的のためのデータベースを構築していかなければなりません。知識の体系は次の図のようになります。

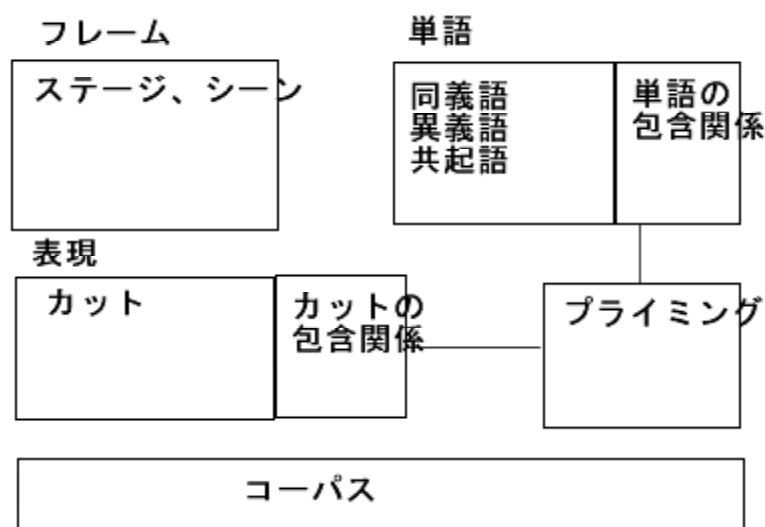


図 2 . 3 文章理解の為の知識の体系

2 . 1 フレーム

行動なのか、陳述なのかの違いは把握できるようにしてなくては行けません、基本的に動画システムを形作ると考えられます。それは文の並びというものを表現している必要があるからです。行動はフルスペックの動画システムを必要としますが、陳述は簡易版でいいでしょう。そして、フレームは知識フレームのバックアップを受けるようになっていなくてはなりません。単語とか表現とかからどんなフレーム(レストランとか学校、車中など)であるかを認識する必要があるからです。表現の曖昧性をなくすとか、ステージの変わり目を解析したりして、質問に的確に答えられるようにするためです。

知識フレームは素材名詞セットと基本アクションセット、それと時系列情報を持つことになります。あとは、目的(レストランだと食べることが機能です)が重要になりますか。5 W 1 H な問い合わせに答えられるデータは必須でしょう。

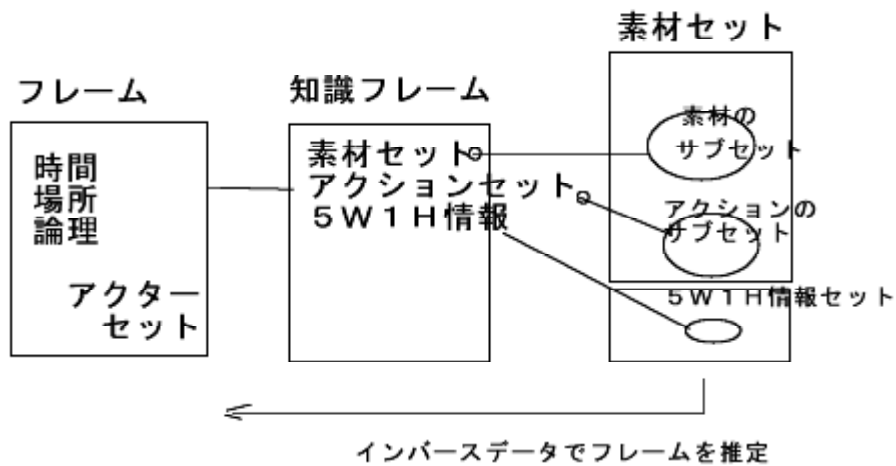


図2.4 フレーム関連データ群

2.2 カット

カットは基本的に動詞情報です。格とその格にぶら下がる名詞を管理します。従いまして、カットはステージやシーンに依存します。同じ「食べる」でもレストランで食べるものと、家で食べるものは違うでしょう。無論、「食べる」を抽象化して管理することは未知の状況に対応するために必要なことです。

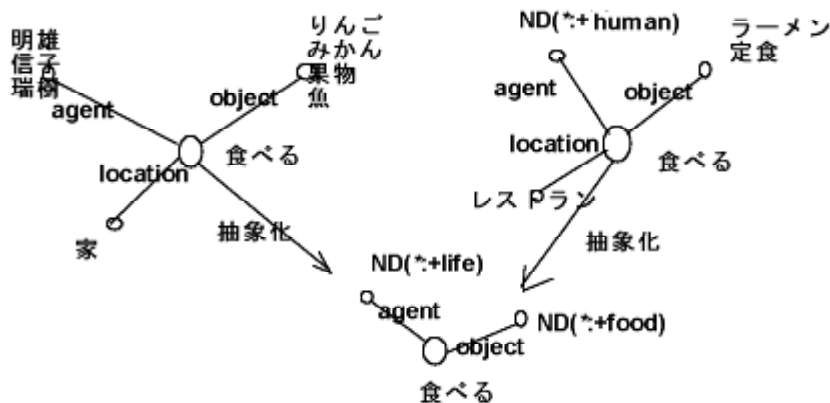


図2.5 カットの知識情報の構造

2.3 コーパス

コーパスは基本的にカットと同じであるべきです。できるだけ単文にして、複文は単文合成属性を付加して並べることになります。つまりカットへのポイントと付加属性で構成するのが理想です。使えるコーパスとはそういうものです。生文のコーパスとこうした加工したコーパスを並記していくのが更に理想なように思えます。

2.4 システムイメージ

トータルなシステム構成としては次のようになるはずですが。

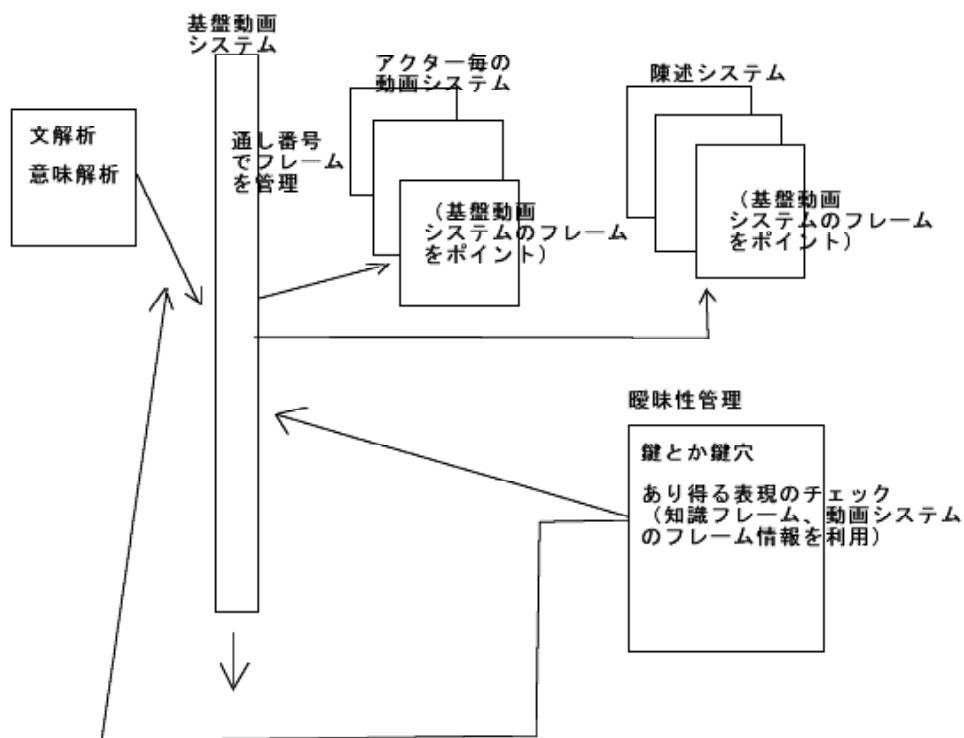


図 2.6 自然言語解析システムの構成

第3章 意味理解の手順

意味理解は、人からの問いかけに、いかに答えられるかという能力として評価しましょう。今まで、意味解析について述べてきたものの他に多くのデータとプロセスを用意しなくてはならないことを、以下に示していきます。結局どれだけデータとプログラムを用意すれば良いのでしょうか。これからの研究が待たれるところです。

(例文1) 朝起きて、ニラをきざみ、みそと一緒に鍋にいれた。みそ汁を作ったのだ。ご飯は電気釜で炊いてあった。そして、8時に食事した。

問い) 食事は朝食か夕食か？

朝の家庭のフレーム情報から朝食という記述を取り出す必要があります。

問い) 食事の準備はどんなだったか？

食事の準備フレームと対照しながら、「ニラをきざみ」「鍋にいれ」「みそ汁を作った」「電気釜で炊いた」を選択していくことになります。「煮ている間に本を読んだ」なんて文が挿入してあっても、フレームを参照して、的確に削除しましょう。

(例文2) 腹減った。弁当食べよう。

問い) なぜ弁当を食べたか？

これも「食べる」のフレームの因果を示すスロットを見て、それと「腹減った」を照合して、答えることになります。

(例文3) 瑞樹と公園に行った。瑞樹の笑顔をひたすら追った。これはというところを写真に納めた。母はこれは良く撮れているねと言った。

問い) どうして写真はすばらしかったのか？

これには、「瑞樹の笑顔を追った」「これはというところを写真に撮った」という、普通の行動ではないところを抽出するフィルターが無いと答えられません。そんな事にもアクションフレームを活用していく必要があるのです。アクションフレームには日常の当たり前のアクション列が記されているはずだからです。

第4章 意味システム

自然言語処理には、次の部分システムがあるでしょう。

- ・意志システム

自然言語処理の方向性とか価値観を決めていく。

- ・論理システム

因果とか、集合論、論理を展開して推論を実現する。

- ・意味システム

単語やアクションなどの意味を管理して、他のシステムが使えるデータを紡いでいく。

- ・陳述的知識システム

物事の知識を保持する。

- ・アクター毎の動作記憶システム(動画システム)

基本的にエピソードを保持する。

ここでは文解析に一番関連の深い、意味システムを考察していきます。

意味の働きとは何でしょうか。文解析にとって意味というものに期待することはなんでしょう。

(1) どうゆう行動を起こせば的確かの判断の基準になる。

(2) 前に言った事と同じ内容であるとか違う内容であるとか分かる。

この2つになるでしょう。推論も行動と見なします。つまり、ある記号の意味とは、アクションが添付されて、意味記号集合の中で相同・相違・包含関係が分かる仕組みのバックアップのあるもの・・・それが、意味であるといえます。そんなシステムを意味システムと呼びましょう。

これまで著者はクオリアをベースにした、プリミティブ記号を提唱してきました。また、単語の意味を素材となる単語の集合(素材集合)で定義すると相同性、相違性、包含関係が簡単に管理できることを述べてきました。

それから、上下、左右とかの物事には配置があり、視点によって単語が変わる(「見上げる」「見下ろす」の対など)ことを注意してきました。

これらを表にして見ますと、意味システムとして何を実現しなくてはいけないかが整理されるでしょう。

<ul style="list-style-type: none"> ・意味の相同・相違・包含関係解決ツール 	<ul style="list-style-type: none"> ・意味処理ツール 	<ul style="list-style-type: none"> ・素材単語フレーム集合管理 	<ul style="list-style-type: none"> ・名詞、名詞意味フレーム ・動詞、格フレーム ・フレーム構造定義
		<ul style="list-style-type: none"> ・プリミティブ単語フレーム集合管理 	
<ul style="list-style-type: none"> ・視点変換ツール 	<ul style="list-style-type: none"> ・視点変換単語フレーム集合 		
<ul style="list-style-type: none"> ・パターン類似の検査ツール 			

表4.1 意味システムの構成

意味システムで基盤になるクラス(属性とメソッドの固まり)は次のものでしょう。

(1) メリットとデメリットを意味するクラス(メリットは何で、デメリットは何で、総合的にメリットがデメリットかといったことを記します)

(2) 相同・相違・包含関係を意味するクラス

(3) 新規・既存を意味するクラス

(4) 陳述クラス群(事実、虚構、想像、感覚、感情)

(5) アクションクラス(動画システム)

(6) エピソードクラス(文、文章コーパス)

- (7) 視点演算クラス
- (8) 論理演算クラス
- (9) 意志・評価クラス

これらは、基本的にクオリアを想定して想像しました。必要なクラスは大体挙げられていると思います。

単細胞生物の時代から、自分に有利な事象か不利な事象かは重要な情報ですし、クオリアは感覚、感情で体现されています。そして、記号システムの大きな働きは、相同・相違・包含関係の特定にあります。

意志や評価はどんな事象に価値を置くか、何を追求していくかを管理するために必要になってきます。

視点を変えるのは3次元ボクセル上の幾何演算になるでしょう。論理は if-then-else、とか否定、三段論法などがあります。Prolog で実現できるようなシステムです。

これらを総合的に扱う、パターン生成、マッチング処理なんかも必要になるかも知れません。

連想は、単語とか記号の包含関係で実現するとよいと思います。あまり似たような機構を沢山作るのには感心しないわけで、プライミングも同様です。特に比喻なんかはそうですね。比喻専門のツールを作るのは、それはそれで意義ある事だとは思いますが。

ということで、基本となる知識ベースの例を以下に示します。

(1) フレーム記述

リンゴの意味を示す素材集合とか、レストランというフレーム構造を定義したり、表現の有意性の検出(「鯨が空を飛ぶ」なんて事はないことを発見するためのコーパスとして)、動詞の格構造を定義したりするデータベースのテーブルです。

key	frame-data
Frame(花)	{ND(はな),VB(さく1)・・・}
ND(はな)	{Mean(+flower), pretty, favorite,・・・}
VB(さく1)	{agent.[はな、りんご],location_AT.[庭,花壇],verb.[さく,萌える]}

表 4 . 2 フレーム構造

(2) 重要事項フレーム

文章を要約するときには、キーポイントとなる単語を探していきます。ある場合は時間だったり、ある場合は場所だったり、アクションだったりします。特に、日常と変わったことをポイントにすることが多いように思えます。そんなことを管理するのが、この重要事項フレームです。

key	frame-data
Frame(レストラン)	{location(*:*:50),time(*:*:50),actor(*:+focus:60),ND(*:+food:60)}

表 4 . 3 重要事項フレーム

表 4 . 3 はレストランでは時間と場所とフォーカスの当たっているアクターと食べ物が必要ということを示しています。

(3) 視点演算

視点を変える演算は。まず起点となるアクターを提示し、上下左右などの関係で他のアクターを配置していった、視点をどのアクターにするかというところで、配置情報を返すという演算になります。

MapSystem ms=new MapSystem(); マップシステムの定義

ms.setInitial(actor0); 基準アクターの設定

ms.setActor({near/far},{up,down,left,right,front,back・・・},this_actor,refer_actor);

アクターの関係の設定

String[] position_inf=ms.getRelation(this_actor,refer_actor);

アクターの関係情報を返す

(4) 意味管理 (含、同義語・異義語情報) 辞書

パターンに意味を管理します。

stext	mean	vector
AV(良く)_VB(考える)	+deep.in.thought+human+concept	+thought.density
AV(ゆっくり)_VB(思考する)	+deep.in.thought+slow+human+concept	+thought.density
AV(ざっと)_VB(考える)	+shallow.in.thought+human+concept+rapid	-thought.density

表 4 . 4 意味管理辞書

(5) 関連語辞書

「私は空腹だ」、「何か食べたい」の2文を結ぶ関係を推測するには、「空腹」と「食べる」の間に原因結果関係があることを知らねばなりません。また、「空腹」から「怠い」というようなことを推測したいものです。そんなことを実現する辞書です。

stext	relation_type	target_patrn
BE(くうふく)	cause	[VB(食べる),ND(*:+food)_XX(*)_VB(買う),VB(飲む)]
BE(くうふく)	lack	[ND(体力),ND(栄養),ND(食べ物)]
BE(くうふく)	same	[BE(渴く),BE(だるい)]
BE(くうふく)	anti	[BE(満腹),AV(はつらつと)]

表4.5 関連語辞書

(6) コーパス

「私は学校へ行った」の意味は次の2つが考えられます。

- ・「私は学校という建物に行った」
- ・「私は学校で学びに行った」

原文のコーパス記述は、

- ・「ND(私)_XX(は)_ND(学校)_XX(へ)_VB(行く:+past)」

意味記述は、

mean	vector	case
行く(私,学校)	+learn+move	VB(行く:+move),agent.ND(私),target.ND(学校),location_at(学校)
行く(私,学校)	+learn	VB(*:+learn),agent.ND(私),location_at.ND(学校),object(*:+subject)

表4.6 コーパスの記述(「行く」)

この「mean」は Prolog を参考にしました。これですと、意味推論が展開できるようになるのです。

「ND(私)_XX(は)_ND(学校)_XX(へ)_ND(公園)_XX(を)通って)_VB(行く:+past)」
は、「mean」を

行く(私,学校)^通る(私,公園)

と表せます。「行く(私、学校)」は定義済みです。そこで、「通る(私、公園)」を定義するだけです。

mean	vector	case
通る(私,公園)	+move	VB(通る:+move),agent.ND(私), location_through(公園)

表4.7 コーパスの記述(「通る」)

おわり