

2006年7月25日(火)
夏の集中勉強会の番外編

Generative Lexicon の概要

松本裕治

語の意味的曖昧性の取り扱い

意味の基本要素 (semantic primitive): 語の意味を記述するためにどのような基本要素を設定すればよいか。

曖昧性 (ambiguity): 同じ表記をもつ語が複数の語義 (sense) をもつことがある。

- 対立的曖昧性 (contrastive ambiguity)
意味的に関連のない語がたまたま同じ表記をもつ場合。同音異義語 (homonym) — bank(銀行) vs bank(川の土手)
- 相補的曖昧性 (complementary ambiguity)
意味関係を共有しつつも異なる振る舞いをする語。
bank(銀行の組織 vs 銀行の建物や場所)

曖昧性 (語の多義性) の取り扱い:

- 多義性の数え上げ (sense enumeration lexicon: SEL)
- core meaning を考えてそこから様々な語義を派生する

生成語彙が対象にする曖昧性の例

- “begin” は補語としてイベントを取るだけでなく一般名詞も取る？

(1a) Mary began to read the novel.

(1b) Mary began reading the novel.

(1c) Mary began the novel.

- “bake, cook” は , process と creation の2つの語義を持つ？

(2a) bake potatoes

(2b) bake a cake

(3a) cooked the carrots

(3b) cooked a meal

- “good” の意味が修飾する名詞によって異なる？

(4a) a good car

(4b) a good meal

(4c) a good knife

- 修飾語を伴う場合にだけ許される補語

(5a) ? Mary kicked him with her foot.

(5b) Mary kicked him with her left foot.

(6a) ? John buttered the toast with butter.

(6b) John buttered the toast with expensive butter from Wisconsin.

生成語彙 (Generative Lexicon)

- 動詞、名詞、形容詞などに共通した語彙表現を与え、いくつかの生成的な演算 (generative operation) を用意することによって、見かけ上の多義性の問題を解決しようという試み
- 生成語彙は、次の 4 つの表現の層からなる。

項構造 (Argument structure): 統語的に実現される項構造を記述する。

事象構造 (Event structure): 語が表す事象とその時間的な関係を記述する。

特質構造 (Qualia structure): 語の意味を表し、後に述べる 4 つの役割からなる。

語彙階層構造 (Lexical Inheritance Structure): 語が表す概念間の階層構造を記述する。

生成語彙における語の記述例

build

ARGSTR =

- ARG1 = 1 [**animate**
FORMAL = **physobj**]
- ARG2 = 2 [**artifact**
CONST = 3
FORMAL = **physobj**]
- D_ARG1 = 3 [**material**
FORMAL = **mass**]

EVENTSTR =

- E₁ = e₁ : **process**
- E₂ = e₂ : **state**
- RESTR = <_∞
- HEAD = e₁

QUALIA =

- create**
- FORMAL = **exist**(e₂, 2)
- AGENTIVE = **build_act**(e₁, 1, 3)

項構造 (Argument structure)

- 単語が取る意味的な補語の一覧を示す。
- 現在の Generative Lexicon では、平板な構造 (補語の羅列) という形をとる
- 補語が統語的にどのように実現されるかに応じて、3 種類に分類されている
 - 真の補語 (true argument): 意味的にも統語的 (英語の場合) にも必須なもの
John arrive late.
 - デフォルト補語 (default argument): 意味的には必要だが統語的には必須でないもの
John built the house out of bricks.
 - 影の補語 (shadow argument): 語の中に意味的に組み込まれてしまっているもの。特に情報を追加する場合以外は言及不要なもの
John buttered his toast with expensive butter.

項構造の記述例

$$\left[\begin{array}{l} \mathbf{build} \\ \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG}_1 = \mathbf{animate_individual} \\ \mathbf{ARG}_2 = \mathbf{artifact} \\ \mathbf{D-ARG}_1 = \mathbf{material} \end{array} \right] \end{array} \right]$$
$$\left[\begin{array}{l} \mathbf{butter} \\ \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG}_1 = \mathbf{human} \\ \mathbf{ARG}_2 = \mathbf{physical_object} \\ \mathbf{S-ARG}_1 = \mathbf{butter} \end{array} \right] \end{array} \right]$$

事象構造 (Event structure)

事象に対して語がもつ情報 (アスペクト情報)

- Vendler (1967) による分類

	説明	図式	例
state	定常状態をあらわす	- - - -	live、住む、いる
activity	一般的な動作	~ ~ ~ ~	walk, run、歩く
accomplishment	最終点をもつ幅をもつ動作	~ ~ ~ ×	build (a house)、描く
achievement	最終点のみ、点的な動作	×	fall, 勝つ、たたく

- 単語が表現する事象に関する情報
- いくつかの部分事象 (subevent) を内部にもってもよい
- Generative Lexicon における基本的な事象は state, process, transition の3種類 (- - -, ~ ~ ~, × に対応) .
- accomplishment と achievement は基本事象として個別に表さず、辞書の内部構造の違いによって表現する

事象構造の記述例

$$\left[\begin{array}{l} \mathbf{build} \\ \\ \mathbf{EVENTSTR} = \left[\begin{array}{ll} E_1 = & \mathbf{e_1:process} \\ E_2 = & \mathbf{e_2:state} \\ \mathbf{RESTR} = & <_{\infty} \\ \mathbf{HEAD} = & \mathbf{e_1} \end{array} \right] \end{array} \right]$$

- build が 2 つの内部事象をもつことが表現されている
- RESTR は、制約条件を表す。上の例では、 e_1 が e_2 よりも前にあり、その間には含まれる事象がないということを表す順序条件を示している。
- HEAD は、主辞事象を示す。上の例では、 e_1 の方が中心的な事象であること。

事象構造で用いられる事象間の関係

\preceq partial order of “*part-of*”

$<$ strict partial order

\circ overlap

\sqsubseteq inclusion

e^* は, e が事象構造内の “head” であることを示す .

事象構造の簡易表現の例

build: $[e^\sigma e_1^* <_\alpha e_2]$

$<_\alpha$ のように, 関係記号に付与された α は, “exhaustive” な関係, つまり, e^σ が, e_2, e_2 以外の事象を含まないことを表す

関係の定義

[Pustejovsky 95] からの抜粋 (下線部ちょっと怪しい)

$[e_3 \ e_1 \prec_\alpha \ e_2] =_{\text{def}} \prec_\alpha (e_1, e_2, e_3)$ “exhaustive ordered part of”

$$\forall e_1 e_2 e_3 [\prec_\alpha (e_1, e_2, e_3) \leftrightarrow e_1 \preceq e_3 \wedge e_2 \preceq e_3 \wedge e_1 < e_2 \wedge$$

$$\forall e [e \preceq e_3 \rightarrow e = e_1 \vee e = e_2]]$$

$[e_3 \ e_1 \circ_\alpha \ e_2] =_{\text{def}} \circ_\alpha (e_1, e_2, e_3)$ “exhaustive overlap”

$$\forall e_1 e_2 e_3 [\circ_\alpha (e_1, e_2, e_3) \leftrightarrow e_1 \preceq e_3 \wedge e_2 \preceq e_3 \wedge e_1 \sqsubseteq e_2 \wedge e_2 \sqsubseteq e_1 \wedge$$

$$\underline{\exists e [e \sqsubseteq e_2 \wedge e \sqsubseteq e_2 \wedge e = e_3]} \wedge \forall e [e \preceq e_3 \rightarrow e = e_1 \vee e = e_2]]$$

$[e_3 \ e_1 \prec \circ_\alpha \ e_2] =_{\text{def}} \prec \circ_\alpha (e_1, e_2, e_3)$ “exhaustive ordered overlap”

$$\forall e_1 e_2 e_3 [\prec \circ_\alpha (e_1, e_2, e_3) \leftrightarrow e_1 \preceq e_3 \wedge e_2 \preceq e_3 \wedge$$

$$\underline{e_1 \circ e_2 \wedge \text{init}(e_1) < \text{init}(e_2) \wedge \text{end}(e_1) = \text{end}(e_2)}$$

$$\wedge \forall e [e \preceq e_3 \rightarrow e = e_1 \vee e = e_2]]$$

事象構造の例

build:	$[e^\sigma e_1^* <_\alpha e_2]$	←	accomplishment verb
arrive:	$[e^\sigma e_1 <_\alpha e_2^*]$	←	achievement verb
give:	$[e^\sigma e_1^* <_\alpha e_2^*]$		
buy:	$[e^\sigma e_1^* o_\alpha e_2]$		
sell:	$[e^\sigma e_1 o_\alpha e_2^*]$		
marry:	$[e^\sigma e_1^* o_\alpha e_2^*]$		
walk:	$[e^\sigma e_1^* < o_\alpha e_2]$		
walk home:	$[e^\sigma e_1 < o_\alpha e_2^*]$		

事象構造についての補足事項

- ここで扱っているのは、単語がもつ事象構造 (あるいは、アスペクト構造)
- アスペクトとは、出来事の時間的な流れの見方
- 単語が持つアスペクト情報を語彙アスペクト、文法的に決まるもの (進行形や完了形) を文法アスペクトという。ここでは、語彙アスペクトが対象
- 文構造がアスペクト構造を変える例
 - ? 1時間で行った。
 - 1時間で学校まで行った。
 - ? I walked in one hour.
 - I walked to school in on hour.

「歩く」は、process (activity) としてのアスペクト構造しか持たないのに、終点が明示されることにより、accomplishment としてのアスペクト構造をもつようになった。

特質構造の内容

構成役割 (Constitutive role): 材料、内容、などの構成物を表す
(the relation between an object and its constitutions or proper parts) – material, weight, parts and component elements

形式役割 (Formal role): 一般に上位の概念を表す (That which distinguishes the object within a larger domain) – orientation, magnitude, shape, dimensionality, color, position

目的役割 (Telic role): その概念を用いる目的や機能を示す動作を表す (Purpose and function of the object) – purpose, built-in function

主体役割 (Agentive role): その概念を産み出す動作や原因を表す (Factors involved in the origin or “bringing about” of an object) – creator, artifact, natural kind, causal chain

- ‘book’ を表現する特質構造

$$\left[\begin{array}{l} \mathbf{book}(x, y) \\ \mathbf{CONST} = \mathbf{bound_pages}(x) \vee \mathbf{disk}(x) \\ \mathbf{FORMAL} = \mathbf{information}(y) \\ \mathbf{TELIC} = \mathbf{read}(T, w, y) \\ \mathbf{AGENTIVE} = \mathbf{artifact}(x) \wedge \mathbf{write}(T, z, y) \end{array} \right]$$

Generative Lexicon における生成的な演算

タイプ強制 引数のタイプが関数の予測するものと異なる場合、引数のタイプを強制的に変更するそうさ。例えば、目的語が動詞の予測するタイプのものでなかった場合、動詞の予測するタイプに合うように目的語を変化させる操作

共構成 引数が関数の予測するタイプのもものと異なる場合に、引数と関数の両方の変更を試みて、両者が合うように変更する操作

選択束縛 引数が関数の直接の引数になるのではなく、関数の内部のある構造の引数として扱われる操作。

- 小さい象、大きい鼠、速いタイピスト、など

タイプ強制 (Type coercion)

(1) John began a book.

(2) John began reading a book.

(3) John began to read a book.

- これらは類似の意味内容を表しており、これらの間の関係を正しく理解することが重要。
- 特質構造の情報と強制という考え方をを用いることにより、それが可能

$$\left[\begin{array}{l}
 \mathbf{begin} \\
 \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG1} = \mathbf{x} : \mathbf{human} \\ \mathbf{ARG2} = \mathbf{e}_2 \end{array} \right] \\
 \mathbf{EVENTSTR} = \left[\begin{array}{l} \mathbf{E}_1 = \mathbf{e}_1 : \mathbf{transition} \\ \mathbf{E}_2 = \mathbf{e}_2 : \mathbf{event} \\ \mathbf{RESTR} = <_{\infty} \\ \mathbf{HEAD} = \mathbf{e}_1 \end{array} \right] \\
 \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{FORMAL} = \mathbf{P}(\mathbf{e}_2, \mathbf{x}) \\ \mathbf{AGENTIVE} = \mathbf{begin_act}(\mathbf{e}_1, \mathbf{x}, \mathbf{e}_2) \end{array} \right]
 \end{array} \right]$$

- ‘begin’ が直接目的語に取るのは事象のタイプ (ARG2=e₂ : event)
- (2) と (3) の文では、意味上のタイプの不整合なし。
- (1) の文では、実際の目的語 (a book) と begin が要求する目的語のタイプ (事象タイプ) の間に不整合がある

タイプ強制: 関数の引数のタイプを、その関数が予測するタイプへ変更する操作。

- 関数とその引数の間にタイプの不整合がある場合、タイプ強制により、引数側のタイプを関数が要求するタイプに変更する。
- (1) の文では、目的語 (a book) のタイプを事象タイプに変更するため、TELIC あるいは AGENTIVE 役割の情報を利用する。
- TELIC 役割を用いる場合は、‘a book’ を ‘to read a book’ に、AGENTIVE 役割用いる場合は、‘a book’ を ‘to write a book’ に変更することになる。

共構成 (Co-composition)

共構成とは、関数と引数の両者が何らかの変更を強制される操作

(1) John baked a potato.

(2) John baked a cake.

これらは類似の行為を表しているが、(1)では、ポテトが焼かれてその状態が変化する対象であるのに対し、(2)では、ケーキは焼くことによって作り出される対象である。その意味で、前者は状態変化 (change of state) の動詞、後者は創造 (creation) の動詞として働いている。

生成語彙では、‘bake’の語義を状態変化の動詞と見なし、創造の動詞としての意味は cake の記述と共構成することによって得られると考える。

bake の記述

$$\left[\begin{array}{l}
 \mathbf{bake} \\
 \mathbf{ARGSTR} = \left[\begin{array}{l}
 \mathbf{ARG1} = \boxed{1} \left[\begin{array}{l}
 \mathbf{animate} \\
 \mathbf{FORMAL} = \mathbf{physobj}
 \end{array} \right] \\
 \mathbf{ARG2} = \boxed{2} \left[\begin{array}{l}
 \mathbf{mass} \\
 \mathbf{FORMAL} = \mathbf{physobj}
 \end{array} \right]
 \end{array} \right] \\
 \mathbf{EVENTSTR} = \left[\begin{array}{l}
 \mathbf{E}_1 = \mathbf{e}_1 : \mathbf{process} \\
 \mathbf{HEAD} = \mathbf{e}_1
 \end{array} \right] \\
 \mathbf{QUALIA} = \left[\begin{array}{l}
 \mathbf{state_change} \\
 \mathbf{AGENTIVE} = \mathbf{bake_act}(\mathbf{e}_1, \boxed{1}, \boxed{2})
 \end{array} \right]
 \end{array} \right]$$

cake の記述

$$\left[\begin{array}{l} \mathbf{cake} \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG1} = \mathbf{x} : \mathbf{food} \\ \mathbf{D_ARG1} = \mathbf{y} : \mathbf{mass} \end{array} \right] \\ \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{CONST} = \mathbf{y} \\ \mathbf{FORMAL} = \mathbf{x} \\ \mathbf{TELIC} = \mathbf{eat}(\mathbf{e}, \mathbf{z}, \mathbf{x}) \\ \mathbf{AGENTIVE} = \mathbf{artifact}(\mathbf{x}) \wedge \mathbf{bake_act}(\mathbf{e}', \mathbf{w}, \mathbf{y}) \end{array} \right] \end{array} \right]$$

共構成によって得られる解釈

1. potato の FORMAL 役割のタイプは自然物 (natural_kind) であり、これは bake の AGENTIVE 役割の述語 (bake_act) の引数である y のタイプ mass と矛盾しない。
2. cake の FORMAL 役割を見ると x は、AGENTIVE 役割の記述により人工物 (artifact) であり、bake の引数と整合しない。
3. 引数である cake の情報によって共指定が起こり、bake の事象タイプ (process) から変化のタイプ (process + state) への変更が起こる。結果として、bake は、過程の結果として得られる状態を含む “creation” の意味を持つ語として定義し直される。

$$\left[\begin{array}{l}
 \mathbf{bake\ a\ cake} \\
 \\
 \text{ARGSTR} = \left[\begin{array}{l}
 \text{ARG1} = \boxed{1} \left[\begin{array}{l}
 \mathbf{animate} \\
 \text{FORMAL} = \mathbf{physobj}
 \end{array} \right] \\
 \text{ARG2} = \boxed{2} \left[\begin{array}{l}
 \mathbf{cake} \\
 \text{CONST} = \boxed{3} \\
 \text{FORMAL} = \mathbf{food}
 \end{array} \right] \\
 \text{D_ARG1} = \boxed{3} \left[\begin{array}{l}
 \mathbf{material} \\
 \text{FORMAL} = \mathbf{mass}
 \end{array} \right]
 \end{array} \right] \\
 \\
 \text{EVENTSTR} = \left[\begin{array}{l}
 \text{E}_1 = \mathbf{e}_1 : \mathbf{process} \\
 \text{E}_2 = \mathbf{e}_2 : \mathbf{state} \\
 \text{RESTR} = <_{\infty} \\
 \text{HEAD} = \mathbf{e}_1
 \end{array} \right] \\
 \\
 \text{QUALIA} = \left[\begin{array}{l}
 \mathbf{create} \\
 \text{FORMAL} = \mathbf{exist}(\mathbf{e}_2, \boxed{2}) \\
 \text{AGENTIVE} = \mathbf{bake_act}(\mathbf{e}_1, \boxed{1}, \boxed{3})
 \end{array} \right]
 \end{array} \right]$$

選択束縛 (Selective Binding)

修飾語 (句) が修飾先の表現の意味構造の適切な部分と結びついて、選択的な解釈を生み出す生成的演算。形容詞や副詞等の解釈を説明する。

形容詞の解釈の例

1. I like to buy a fast car.
2. Mary is a fast typist.
3. Fast drivers will be caught.
4. John started a long speech.
5. Put the tea leaves into a boiling pot.
cf. Boiling water is in the pot.

typist の記述例 :

$$\left[\begin{array}{l} \mathbf{typist} \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG1} = \mathbf{x} : \mathbf{human} \end{array} \right] \\ \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{FORMAL} = \mathbf{x} \\ \mathbf{TELIC} = \mathbf{type}(\mathbf{e}, \mathbf{x}) \end{array} \right] \end{array} \right]$$

fast typist:

$$\left[\begin{array}{l} \mathbf{typist} \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG1} = \mathbf{x} : \mathbf{human} \end{array} \right] \\ \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{FORMAL} = \mathbf{x} \\ \mathbf{TELIC} = \mathbf{type}(\mathbf{e}, \mathbf{x}) \wedge \mathbf{fast}(\mathbf{e}) \end{array} \right] \end{array} \right]$$

選択束縛の操作

α が $a \rightarrow a$ のタイプの句であり、 β が b のタイプの句とする。 β 中にタイプ a の特質構造の要素 q_β がある時、 $\alpha\beta$ の表現は、 $\beta \cap \alpha(q_\beta)$ となる。(つまり、 α が β を修飾してできる表現は、 α が β 中の q を修飾したものと解釈される)

fast typist:

$$\left[\begin{array}{l} \mathbf{typist} \\ \mathbf{ARGSTR} = \left[\mathbf{ARG1} = \mathbf{x} : \mathbf{human} \right] \\ \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{FORMAL} = \mathbf{x} \\ \mathbf{TELIC} = \underline{\mathbf{type}(\mathbf{e}, \mathbf{x}) \wedge \mathbf{fast}(\mathbf{e})} \end{array} \right] \end{array} \right]$$

Lexical Conceptual Paradigm (lcp)

- systematic polysemy の表現のための新たなデバイス
- ある語が二つ以上のタイプ（意味の側面）を持つときに，それらを合わせ持つタイプ (dotted type) を定義する
- 例えば，ドアは，物理的な物質と出入り口という2つの意味を合わせ持つ

`phys_obj.aperture_lcp`

`= { phys_obj.aperture, phys_obj, aperture }`

lcp を用いた記述例

$$\left[\begin{array}{l} \mathbf{newspaper} \\ \mathbf{ARGSTR} = \left[\begin{array}{l} \mathbf{ARG1} = \mathbf{x:org} \\ \mathbf{ARG2} = \mathbf{y:info.physobj} \end{array} \right] \\ \mathbf{QUALIA} = \left[\begin{array}{l} \mathbf{org.info.physobj_lcp} \\ \mathbf{FORMAL} = \mathbf{y} \\ \mathbf{TELIC} = \mathbf{read(e2, w, y)} \\ \mathbf{AGENTIVE} = \mathbf{pubish(e1, x, y)} \end{array} \right] \end{array} \right]$$

1. The newspapers attacked the President for raising taxes.
2. Mary spilled coffee on the newspaper.
3. John get angry at the newspaper.

lcp によって表現することができるその他の例

- Count/Mass alternations: *lamb*
- Container/Containee alternations: *bottle*
- Figure/Ground alternations: *door, window*
- Product/Producer alternations: *newspaper, Honda*
- Process/Result alternations: *construction, merger*

construction の例

1. The house's construction was finished in two months.
2. The construction was arduous and tedious.
3. The construction is standing on the next street.

その他 GL が対象とする生成的意味

- 動詞の意味解釈の際の contextualization が目的語よりもむしろ主語から来る場合
 1. Most pilots prefer Kennedy to Logan.
 2. Most pilots prefer New York to Boston.