頑健なHPSGパーザの出力のTDL意味表現への変換

佐藤 学† 戸次 大介‡ 宮尾 祐介* 辻井 潤一*§
† 東京大学 大学院情報理工学系研究科 コンピュータ科学専攻
‡ 東京大学 21 世紀 COE「心とことばー進化認知科学的展開ー」
* 東京大学 大学院情報学環
§ 科学技術振興機構、School of Informatics, University of Manchester
†*{sa-ma, yusuke, tsujii}@is.s.u-tokyo.ac.jp
‡bekki@ecs.c.u-tokyo.ac.jp

1 はじめに

実テキストに対しても頑健に統語解析を行う技術が発展し、情報抽出や情報検索、機械翻訳等に応用されている [1, 2]。これらの解析器は言語学の文法理論に則って文の構造を解析するが、その関心は主として統語解析に向けられており、意味解析については、単純な述語項構造を出力するにとどまっている。

Bos らは、実テキストから意味情報を得るために、頑健な CCG パーザ [1] の出力する構文解析結果から、述語論理に基づく意味表現を導出する手法を提案した [3]。この手法は、頑健な意味表現の導出を可能としたものの、その意味論的表現能力はいまだ古典的述語論理に基づいており、照応や前提、主題といった談話に関する問題を解決することはできない。

このような談話に関する問題を解決することが可能な意味論に、型付き動的論理 (Typed Dynamic Logic: TDL)[4] がある。TDL は型付きラムダ計算で記述される動的論理 [5] であり、量化・複数形・照応・前提といった意味的現象を表現することを可能とする。また、TDL は構成性を持つため、透過的に統語理論と結び付けられる。

本稿では、TDL に基づいた意味表現を実テキストから出力する手法を述べる。この手法は、頑健な HPSG パーザ [2] の出力を TDL 意味表現に変換するものである。また、手法を実装し、実テキストから TDL 意味表現を生成する実験を通して、その有効性を示す。

2 手法

HPSG[6] の構文木から TDL 意味表現を導出する手法は、[3] に倣う（図 1）。構文木の終端ノード（HPSG 語彙項目）に、対応する TDL 意味表現を割り当て（割り当て規則）、構文木に沿ってボトムアップに意味表現を合成していく（合成規則）。構文木の各ノードには、TDL Extended Structure (TDLES) という TDL 論理式を拡張した構造を付加する。

2.1 割り当て規則

HPSG 語彙項目への TDLES の割り当ては、テンプレートを記述して行う。“a”、“the”、“not” といった、クローズドクラスの単語には、単語ごとにテンプレートを記述する。

(1)
$$\begin{bmatrix} PHON & \text{“a”} \\ HEAD & det \\ SPEC & \langle noun \rangle \end{bmatrix}$$
$$\Downarrow$$
$$\left\langle \lambda x.\lambda s.\lambda\phi. \begin{bmatrix} \lambda n.\lambda w.\lambda e.\lambda s.\lambda\phi. \\ nx_1s\,[wx_1es\phi] \end{bmatrix}, \begin{matrix} \_empty\_ \\ \_empty\_ \end{matrix} \right\rangle$$

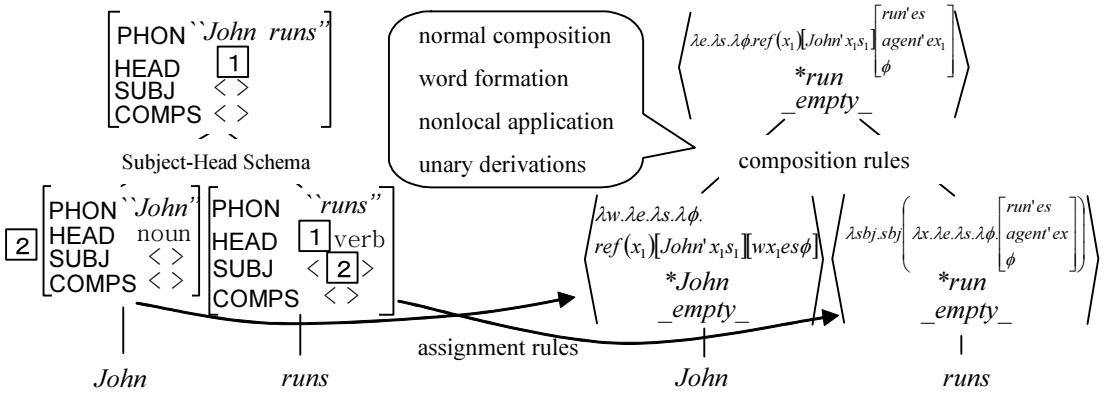

図 1: 割り当て規則・合成規則の適用の例

(1) は、不定冠詞 “a” のテンプレートである。(1) の上半分は HPSG 語彙項目であり、この語彙項目は、音声が “a”、POS が冠詞で、名詞にかかるということを示している。下半分がこの HPSG 語彙項目に割り当てる TDLES である。上から、ラムダ式で表される TDL の論理式（TDL 項）、TDL 項の中の主要述語を表すエントリー、長距離項を示すエントリーである。

名詞や動詞といったオープンクラスの単語には、語ごとではなく、統語範疇ごとにテンプレートを記述する。

(2)
$$\begin{bmatrix} PHON & P \\ HEAD & noun \\ MOD & \langle\rangle \\ SUBJ & \langle\rangle \\ COMPS & \langle\rangle \\ SPR & \langle det\rangle \end{bmatrix}$$
$$\Downarrow$$
$$\left\langle \begin{array}{c} \lambda x.\lambda s.\lambda\phi.\begin{bmatrix} *P'xs \\ \phi \end{bmatrix} \\ *P \\ \_empty\_ \end{array} \right\rangle$$

(2) は普通名詞のテンプレートである。HPSG 語彙項目は、音声が変数 P、POS は名詞で、項をとらず、他の語を修飾もしないことを示している。変数 P には、実際に入力される語の音声が代入される。これにより、TDL 項の述語 P' が生成される。

動詞への TDLES の割り当ては、動詞の基本形に対応する TDLES に、語彙規則を適用することにより行う。これは、HPSG の語彙規則に対応するものである。詳細はスペースの都合上省略する。

2.2 合成規則

合成規則は 3 種類ある。関数適用規則は、標準的な合成規則、語形成規則は、複合語を構成するための規則、長距離依存規則は、関係代名詞等による長距離依存現象を扱うための規則である。

Definition 2.1 (関数適用規則) $Type\,(T_L) = \alpha$, $Type\,(T_R) = \alpha \mapsto \beta$ のとき、

$$S_M = \left\langle \begin{array}{c} T_R T_L \\ p_R \\ union\,(n_L, n_R) \end{array} \right\rangle$$

$Type\,(T_L) = \alpha \mapsto \beta$ *and* $Type\,(T_R) = \alpha$ のとき、

$$S_M = \left\langle \begin{array}{c} T_L T_R \\ p_L \\ union\,(n_L, n_R) \end{array} \right\rangle$$

Definition 2.1 において、$Type(T)$ は TDL 項 T の型を返す関数である。また、$union\,(n_L, n_R)$ は以下のように定義される。

$$union\,(n_L, n_R) = \begin{cases} empty & if\ n_L = n_R = \_empty\_ \\ n & if\ n_L = n,\ n_R = \_empty\_ \\ n & if\ n_L = \_empty\_,\ n_R = n \\ undefined & if\ n_L \neq \_empty\_,\ n_R \neq \_empty\_ \end{cases}$$

この関数は HPSG の SLASH 素性の $union$ に相当する。

Definition 2.2 (語形成規則) $Type\,(T_L) = \omega$ のとき、

$$S_M = \left\langle \begin{array}{c} T_R \\ concat\,(p_L, p_R) \\ n_R \end{array} \right\rangle$$

$Type\,(T_R) = \omega$ のとき、

$$S_M = \left\langle \begin{array}{c} T_L \\ concat\,(p_L, p_R) \\ n_L \end{array} \right\rangle$$

Definition 2.2 において、$concat\,(p_L, p_R)$ は、p_L と p_R の接合を返す関数である。

Definition 2.3 (長距離依存規則) $Type\,(T_L) = (\alpha \mapsto \beta) \mapsto \gamma$, $Type\,(T_R) = \beta$, $Type\,(n_R) = \alpha$ であり、*HPSG* 構文木で *Filler-Head Schema* が適用されているとき、

$$S_M = \left\langle \begin{array}{c} T_L\,(\lambda n_R.T_R) \\ p_L \\ \_empty\_ \end{array} \right\rangle$$

2.3 単一導出規則

TDL では、語句が空範疇（音声を持たないが、統語的・意味的機能を持つ範疇）と合成され、型変換されることがある。多くの型変換規則は HPSG パーザでは扱われないため、HPSG 構文木とは独立に TDL 意味表現を変換する単一導出規則を定義した。下は、句 “this year” が名詞句から副詞句に変換される例である。(3) が型変換され、(4) になる。

$$(3)\ \left\langle \begin{array}{c} \lambda w.\lambda e.\lambda s.\lambda\phi.ref\,(x_1)\,[*year'x_1s_1]\,[wx_1es\phi] \\ *year \\ \_empty\_ \end{array} \right\rangle$$

$$(4)\ \left\langle \begin{array}{c} \lambda v.\lambda e.\lambda s.\lambda\phi. \\ ref\,(x_1)\,[*year'x_1s_1]\left[ves\left[\begin{array}{c} mod'ex_1 \\ \phi \end{array}\right]\right] \\ *year \\ \_empty\_ \end{array} \right\rangle$$

3 実験

我々は、以上に述べた規則を実装した。詳細は表 1 に記されている。この手法を評価するにあたり、実テキストに対して本手法を適用し、TDL 意味表現を出力することに成功した被覆率を計測した。これは、[3] に倣ったものである。正確に手法を評価するためには、文章の著者・話者の直感と、意味表現の真理値の一致を計測する必要があるが、現状、それを可能とする資源が存在しないため、意味表現の生成の被覆率を計測する。

表 1: 実装した規則とその数

割り当て規則	
HPSG-TDL テンプレート	50
- クローズドワード	13
- オープンワード	37
動詞の語彙規則	27

合成規則	
二項合成規則	3
関数適用規則	
語形成規則	
長距離依存規則	
単一導出規則	8

表 2: Penn Treebank 23 (2,122 文) に対する被覆率

covered sentences	88.3 %
uncovered sentences	11.7 %
assignment failures	6.2 %
composition failures	5.5 %
word coverage	99.6 %

実験には、Penn Treebank [7] を用いた。22 節 (1,527 文) を手法の構築に、23 節 (2,144 文) をテストに用いた。まず、コーパス中の文を頑健な HPSG パーザー [2] で解析した。出力された 2,122 文 (98.9%) の構文木に対して、我々の手法を適用した。表 2 に結果を記す。未見の文の 88.3%に TDL 意味表現を割り当てることに成功した。これは、既存研究 [3] の 92.3%という被覆率にやや劣るが、本手法では TDL に基づいた表現力の高い意味表

```
ref($1)[]
   [lecture($2,$3) &
    past($3) &
    agent($2,$1) &
    content($2,$4) &
    ref($5)[]
       [every($6)[ball($6,$4)]
             [see($7,$4) &
              present($4) &
              agent($7,$5) &
              theme($7,$6) &
              tremendously($7,$4) &
              ref($8)[]
                 [ref($9)[groove($9,$10)]
                     [be($11,$4) &
                      present($4) &
                      agent($11,$8) &
                      in($11,$9) &
                      when($11,$7)]]]]]
```

図 2: *"When you're in the groove, you see every ball tremendously," he lectured.* に対する出力の例

現を出力していることを考慮すれば、評価できる結果であると言えよう。

図 2 は PTB 22 節の文に対する出力の例である。変数`$1,...,$11` は entity、event、situation を示す指標である。`every($6)[ball($6,$4)][see($7,$4)...]` は一般化量化詞 "every ball" を示す。`ref(...)[...][...]` は照応表現を示す。このように、イベントや状況、量化や照応が表現されていることがわかる。

4 結論

本稿では、頑健な HPSG パーザー [2] の出力する構文木を TDL[4] の、動的意味論に基づく意味表現に変換する手法を述べた。また、その手法の実装が、Penn Treebank コーパス中の実テキストに対して、90%程度の高い被覆率を達成し、結果の意味表現が量化・複数形・照応・前提等の言語現象を表現できることを示した。

我々が取り組んだ課題は、頑健性と意味論の表現能力の両立にある。この課題はこれまで実現がされてこなかったが、それは、本質的にそれらが対立しているからではない。統語理論と (動的) 意味理論を透過的に扱う理論によって解消されうるのである。

参考文献

[1] Julia Hockenmaier and Mark Steedman. 2002. Acquiring Compact Lexicalized Grammars from a Cleaner Treebank, In *Proc. LREC-2002*, Las Palmas.

[2] Yusuke Miyao, Takashi Ninomiya and Jun'ichi Tsujii. 2005. Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank, in *IJCNLP 2004, LNAI3248*, pp.684-693. Springer-Verlag.

[3] Johan Bos, Stephen Clark, Mark Steedman, James R. Curran and Julia Hockenmaier. 2004. Wide-Coverage Semantic Representations from a CCG Parser, In *Proc. COLING '04*, Geneva.

[4] Daisuke Bekki. 2000. Typed Dynamic Logic for Compositional Grammar, Doctoral Dissertation, University of Tokyo.

[5] Jeroen Groenendijk and Martin Stokhof. 1991. Dynamic Predicate Logic, In *Linguistics and Philosophy 14*, pp.39-100.

[6] Carl Pollard and Ivan A. Sag. 1994. Head-Driven Phrase Structure Grammar, *Studies in Contemporary Linguistics*. University of Chicago Press, Chicago, London.

[7] Mitch Marcus. 1994. The Penn Treebank: A revised corpus design for extracting predicate-argument structure. In *Proceedings of the ARPA Human Language Technolog Workshop*, Princeton, NJ.