

組合せ範疇文法に基づく漸進的な意味解析

加藤 芳秀[†]松原 茂樹[‡][†]名古屋大学情報連携統括本部[‡]名古屋大学大学院情報科学研究科

yoshihide@icts.nagoya-u.ac.jp

1 はじめに

漸進的な解析とは、自然言語文を単語の出現順序に従って解析し、文を途中まで読み込んだ段階でその構文構造や意味構造を捉える枠組みであり、同時通訳システムやリアルタイム字幕生成システムなどの実時間音声言語処理システムの実現に必要な要素技術の一つである。また、人間の言語理解の過程も漸進的であると考えられており、心理言語学的なモデルとしての研究も行われている。

本稿では、自然言語文を先頭から読み込み、読み込んだ文の断片に対して意味表示を割り当てる解析手法を提案する。本手法は、組合せ範疇文法 (Combinatory Categorical Grammar, CCG)[7] に基づき、意味表示を構成する。本手法は、CCG における導出を構文木とみなし、導出を表現する構文木を漸進的構文解析により生成する。解析過程で生成される部分構文木は、導出の一部を表現しているが、本稿では、この部分構文木から意味表示を合成する方法を CCG に沿って定義する。本手法は、CCG による意味合成にのみ依存しているため、CCG に基づく意味解析、例えば文献 [1] のような解析を流用し、漸進的な意味解析が実現できると期待できる。

2 関連研究

CCG に基づく漸進的構文解析がこれまでにいくつか提案されている [6, 3, 4]。CCG では、構成要素ではないような単語列に対しても、統語範疇を割り当てることが可能である。従来の CCG ベースの漸進的構文解析ではこの性質を活用し、先頭から読んだ文の断片に対して統語範疇が割り当てられるように、左枝分かれ構造の導出を使用している。しかし、Demberg が文献 [2] において詳細に論じているが、左枝分かれ構造の導出により表現できない文が存在する。このこと

は、従来の手法では、先頭から読み込んだ文の断片に対して統語範疇を割り当てられないケースがあることを意味する。これは同時に、意味表示が割り当てられないということをも意味する。

3 CCG に基づく漸進的な意味解析

本節では、従来の方法とは異なるアプローチにより、漸進的に意味表示を構成する手法を提案する。本手法では、CCG における導出を構文木とみなし、漸進的構文解析により、導出を表現する構文木を生成する。漸進的構文解析は、任意の文の断片に対してそれを覆う部分構文木が生成できる。導出を表現する部分構文木には、情報が確定していない部分が存在するが、本手法では、これを意味表示において変数として表現する。これらの変数は、情報が確定した段階で順次、確定した意味表示に置き換えられ、意味表示は漸進的に構成される。任意の文の断片に対して部分構文木が漸進的構文解析により与えることができ、かつ、それに対する意味表示を定めることができる。すなわち、従来の手法と異なり、任意の文の断片に対して意味表示を与えることができる。以下では、CCG、及び本手法でベースとする漸進的構文解析について説明し、続いて、意味表示の構成法について述べる。

3.1 組合せ範疇文法

CCG においては、統語範疇は、基本範疇と複合範疇からなる。複合範疇は、 X/Y あるいは $X\backslash Y$ のような形をとる。 X/Y は、その右側に統語範疇 Y があるとき、それと組み合わせると、統語範疇が X となることを意味する。 $X\backslash Y$ は方向が左側の場合である。図 1 に示すような規則に従って、統語範疇が組み合わせられる。これらの規則は、矢印の左側の要素を組み合わせると、右側の要素となることを表している。コロン

$X/Y : f$	$Y : a$	\Rightarrow	$X : fa$	
$Y : a$	$X \setminus Y : f$	\Rightarrow	$X : fa$	
$X/Y : f$	$Y/Z : g$	\Rightarrow	$X/Z : \lambda x.f(gx)$	
$Y \setminus Z : g$	$X \setminus Y : f$	\Rightarrow	$X \setminus Z : \lambda x.f(gx)$	
$Y/Z : g$	$(X \setminus Y) / Z : f$	\Rightarrow	$X/Z : \lambda x.fx(gx)$	
$X : a$		\Rightarrow	$T / (T \setminus X) : \lambda f.fa$	
$X : a$		\Rightarrow	$T \setminus (T / X) : \lambda f.fa$	
$X : f$	$CONJ : b$	$X : g$	\Rightarrow	$X : \lambda \dots b(g \dots)(f \dots)$

図 1: CCG の規則の一部

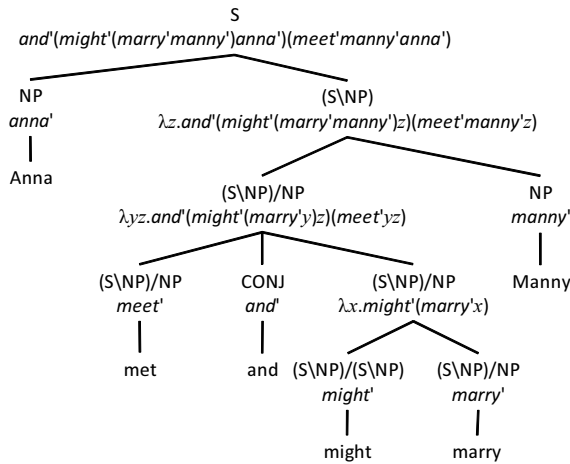


図 2: 導出の例

(:) の右側にあるのは、対応する意味表示であり、統語範疇の組み合わせ方に応じて、その意味表示を合成する方法が一意に定まっている。図 2 に導出を木構造で表した例を示す。

3.2 漸進的構文解析

本手法では、Kato らが提案する漸進的構文解析 [5] に基づき、導出を構文木とみなし、文を解析する。この手法では、allowable chain と呼ばれる要素を部分構文木に付加することにより、解析が進行する。allowable chain は単語と統語範疇からなる系列で、最後の記号が単語であり、それ以外は統語範疇である。これは、構文木中のあるノードから左端の子ノードを順に辿るときに得られるラベルの系列に相当する。この手法では、さらに、左再帰構造による局所的曖昧性の問題を回避するために、左再帰構造を接合する操作を導入している。図 3 に、図 2 の構文木を漸進的に生成する過程を示す。四角で囲まれた部分が allowable chain であり、丸で囲まれた部分が、接合操作により導入されたノードを表す。この例が示すように、漸進的構文解析では、任意の文の断片に対してそれを覆う部分構文

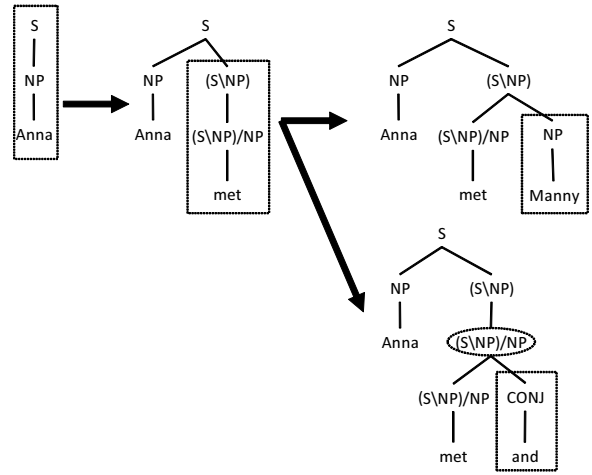


図 3: 漸進的構文解析の解析過程

木を生成できる。

3.3 意味表示の構成法

本手法では、前述の漸進的構文解析が生成した部分構文木に基づき意味表示を構成する。意味解析処理は以下の手順で進行する。

- w_1^i に対する部分構文木において新しく具現化されたノードから、遷移関数 t_i を構成する。
- w_1^{i-1} の意味表示 s_{i-1} に対して遷移関数 t_i を適用し、 w_1^i に対する意味表示 $s_i = t_i(s_{i-1})$ を得る。

以下では、 t_i を意味遷移関数と呼ぶが、ポイントは遷移関数をいかに構成するかである。以下で順を追ってそれについて説明する。

意味遷移関数は、部分構文木において新しく具現化されたノードの情報に基づき構成する。本手法では、2 項組 $\langle \alpha, M \rangle$ を用いてこれを実現する。 α は変数の系列であり、 M は意味表示である。変数系列 α は、 M において具現化されていない意味表示を示すための変数を保持する。また、系列における順序は、意味表示が具現化される順序を表している。 M は、解析処理が進んだことにより、新たに具現化された意味表示を表している。

話を単純にするために、まず、接合操作を使用しない場合の遷移関数の構成法を述べる。上述の 2 項組は、部分構文木に新たに導入された allowable chain の各ノードに対して割り当てられる。その手順を以下に示す。

- 前終端ノード (単語の親ノード) に対して、 $\langle \varepsilon, \omega \rangle$ を与える。ただし、 ω は単語 w_i の意味表示である。

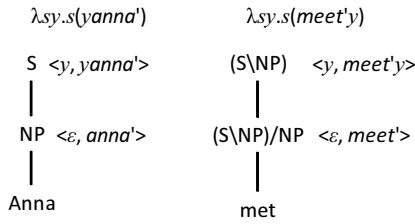


図 4: 意味遷移関数の構成 (接合操作を許可しない場合)

- ノード n の子ノードに, $\langle \alpha, M \rangle$ が与えられているとする. また, n は規則 R により導出されたノードとする.
 1. R が型繰り上げ規則 ($> T, < T$) ならば, n に $\langle \alpha, \lambda f.fM \rangle$ を与える.
 2. それ以外の規則ならば, n に $\langle \alpha y, C_R[M, y] \rangle$ を与える. ここで, $C_R[x, y]$ は, x と y をこの順番で規則 R により意味合成した結果を表す.

allowable chain の最上位のノードに割り当てられた 2 項組 $\langle \alpha, M \rangle$ から以下の意味遷移関数を構成する.

- $\lambda s\alpha.sM$

意味遷移関数の構成の例を図 4 に示す.

漸進的な意味構成の例として, “Anna met Manny” に対する解析の過程を説明する. 開始時の意味表示は, 恒等関数 $\lambda x.x$ である. “Anna” に対しては, 図 4 に示した意味遷移関数が構成される. これを開始時の意味表示である恒等関数に適用すると, “Anna” に対して以下の意味表示が得られる.

$$\lambda y.(yanna') \quad (1)$$

図 4 に示した “met” について構成された意味遷移関数を (1) に適用すると, “Anna met” に対する意味表示として

$$\lambda y.meet'yanna' \quad (2)$$

が得られる. 最後に, “Manny” の意味遷移関数 $\lambda s.smanny'$ を適用すると, CCG の導出により得られる意味表示と同一の意味表示

$$meet'manny'anna' \quad (3)$$

が得られる.

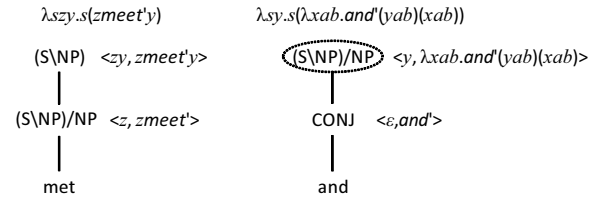


図 5: 意味遷移関数の構成 (接合操作を許可する場合)

3.3.1 接合操作に対応する意味表示の構成

ノードに対して接合操作を許可する場合, そのノードに割り当てる 2 項組を次のように変更する.

- R が型繰り上げ規則の場合は, $\langle \alpha z, z(\lambda f.fM) \rangle$ を割り当てる.
- R がそれ以外の規則の場合は, $\langle \alpha y z, zC_R[M, y] \rangle$ を割り当てる.

変数 z は, 左再帰構造が構文木に接合された場合に, 対応する意味表示を導入できるように用意した変数である. 解析過程において接合操作を適用しなかった場合は, z を恒等関数 $\lambda x.x$ で置き換えればよい. すなわち, 接合操作が適用しなかったノードの数だけ, $\lambda s.s(\lambda x.x)$ を意味表示 s_{i-1} に適用し, その後, 意味遷移関数 t_i を適用する.

n を接合操作により導入されたノードとするとき, n に対しては, 以下のように 2 項組を割り当てる.

- n に付加される allowable chain のルートに $\langle \alpha, M \rangle$ が与えられているとする. また, n は規則 R により導出されたノードとする.
 - R が等位接続規則 (Φ) であり, n に接合操作を許可するならば, n に, $\langle \alpha y z, \lambda x.z(\lambda \dots (M(y \dots)(x \dots))) \rangle$ を与える. 接合操作を許可しないならば, $\langle \alpha y, \lambda x.(\lambda \dots (M(y \dots)(x \dots))) \rangle$ を与える.
 - それ以外の規則について, n に接合操作を許可するならば, $\langle \alpha z, \lambda x.zC_R[x, M] \rangle$ を与える. 許可しないならば, $\langle \alpha, \lambda x.C_R[x, M] \rangle$ を与える.

“met” の親ノードに接合操作を許可した場合の遷移関数の構成の例を図 5 に示す. 例として, 文の断片 “Anna met” について考える. 接合操作を許可した場合の “met” に関する意味遷移関数を図 5 に示したが, これを, “Anna” に対する意味表示に適用すると,

$$\lambda zy.zmeet'yanna' \quad (4)$$

が得られる。ここで、次の語が“Manny”であったとすると、 $(S \setminus NP) / NP$ に対して接合操作は適用されず、(4)に $\lambda s.s(\lambda x.x)$ を一度適用する。その結果は(2)と同一であり、最終的には、“Anna met Manny”に対する意味表示として(3)が得られる。

次に、“Anna met”に続く語が“and”であった場合について考える。この場合、解析は図3における下側のような解析過程をたどることになる。“and”について構成される意味遷移関数は、図5のようになり、これを(4)に適用すると、

$$\lambda yx.\text{and}'(yx\text{anna}')(\text{meet}'x\text{anna}') \quad (5)$$

が得られる。さらに、“might marry Manny”が後に続くと、順次得られる文の断片に対する意味表示は以下ようになる。ただし、新しく具現化したノードについては接合操作を許可しないものとしている。

$$\lambda yx.\text{and}'(\text{might}(yx)\text{anna}')(\text{meet}'x\text{anna}') \quad (6)$$

$$\lambda x.\text{and}'(\text{might}(\text{marry}'x)\text{anna}')(\text{meet}'x\text{anna}') \quad (7)$$

$$\text{and}'(\text{might}(\text{marry}'\text{manny}')\text{anna}')(\text{meet}'\text{manny}'\text{anna}') \quad (8)$$

4 議論

従来の CCG ベースの漸進的構文解析では、左枝分かれ構造の導出を用いる。“Anna met Manny”における“Anna met”に対しては、 S/NP といった統語範疇が割り当てられる。“Anna”と“met”を組み合わせたこのような解析は、“met”が等位項である場合に対応できない。“met”が等位項であるような解析のために、“Anna”と“met”を組み合わせない別の導出を作らなければならない。一方、本稿で提案した手法では、図3で示したように、“Anna met”に対する単一の部分構文木から、両者のケースに対応することが可能である。このため、本手法により、漸進的な解析処理において問題となる局所的曖昧性を抑えることが可能となると期待できる。

5 おわりに

本稿では、漸進的に意味表示を構成する方法を提案した。従来の CCG ベースの漸進的解析が、左枝分かれ構造の導出を用いるのに対して、本手法では、導出を構文木として表現し、その部分構文木を用いる。意味遷移関数を定義することにより、任意の文の断片に

対して意味表示を与えることができる。本手法では、意味表示の漸進的な構成法を CCG に沿って定義したが、具体的な意味論について今後検討したい。

謝辞

本研究は一部、科研費基盤研究(B)(No. 22300051)により実施した。

参考文献

- [1] Johan Bos. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pp. 277–286, 2008.
- [2] Vera Demberg. Incremental derivations in CCG. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 11)*, pp. 198–206, 2012.
- [3] Hany Hassan, Khalil Sima'an, and Andy Way. A syntactic language model based on incremental CCG parsing. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pp. 205–208, 2008.
- [4] Ahmed Hefny, Hany Hassan, and Mohamed Bahgat. Incremental combinatory categorial grammar and its derivations. In *Computational Linguistics and Intelligent Text Processing*, pp. 96–108. Springer, 2011.
- [5] Yoshihide Kato and Shigeki Matsubara. Incremental parsing with adjoining operation. *IEICE Transactions on Information and Systems*, Vol. E92-D, No. 12, pp. 2306–2312, 2009.
- [6] David Reitter, Julia Hockenmaier, and Frank Keller. Priming effects in combinatory categorial grammar. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 308–316, 2006.
- [7] Mark Steedman. *The Syntactic Process*. The MIT press, 2001.