

文レベルでのトランスファのための 簡易な意味表現についての検討

橋本 和夫 鈴木 雅実 野垣内 出 神 博史
 KDD 研究所

1. はじめに

結合価文法と格文法は、文の構造のとりえ方が形態的か意味的かのちがいはあるが、共に動詞に着目して動詞に対する名詞の結合のし方を論じる点では共通点が多い。この類似性を利用すると、結合価文法レベルの統語構造から格文法レベルの意味構造への変換はかなり一般的な規則として整理しやすくなる。

筆者らは、結合価文法に基づく統語解析を前提とした深層格推定方法について調査を進めており、英文の深層格推定規則⁽²⁾については既に部分的な提案を行なっている。

しかし、同推定法は、統語的必須格だけからなる統語構造に対する推定手順であり、任意格を含む統語構造に対しては推定条件が不十分となる。

同推定方法を機械翻訳の中で採用するためには、深層格推定で用いる意味情報の付与のし方、意味情報の共起関係によって作る深層格推定条件などについて妥当性を検証する必要がある。

そこで筆者らは、深層格推定方法の実験的検証を目的として、英語統語構造から格文法レベルの意味構造に対応する中間構造を生成する小規模な実験システムを作成した。

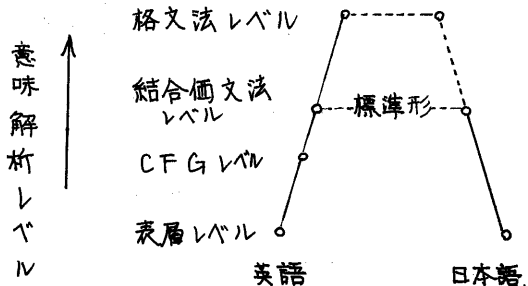


図1. 概念図

本稿では、実験システムの構成と、中間構造における意味表現の形式について報告する。

2. システム設計の基本的立場

筆者らは、日本語解析においても結合価文法を重視し、表層表現における多様な助詞構造を結合価文法が扱う統語的標準形の助詞構造に変換する過程⁽³⁾について検討を進めている。この実験システムは、英語側と日本語側からの解析結果を格文法レベルの意味的中間構造であらわし、双方向トランスファを行うことを想定して作成した。

2-1. 統語解析と意味解析の分離

統語解析では、結合価文法に基づき文の統語構造を動詞、統語的必須格及び任意格の組合せであらわす。ここでは暫定的に以下の統語格ラベルを採用する。

統語的必須格

Subject direct-Object

Complement

統語的任意格

Modifier

意味解析では上記の統語格ラベルを格文法の深層格ラベルに変換する。筆者らは、統語解析を意味解析のための前処理ととらえるため、両者は完全に分離する。

2-2. 中間構造の記述

意味解析結果は、語義ごとにその語義が関与する修飾・被修飾関係をまとめておくことにより、トランスファ・生成過程での負担を減らすことができる。意味ネットワーク⁽⁴⁾⁽⁵⁾は概念間の関係を概念ごとにまとめて記述する表現方

法として有力な一手法である。

本稿では意味ネットワークの記述法を用いて、語義をノードとしてその語義が関与する修飾・被修飾関係をリンクであらわす。語義選択・訳語選択の際に意味素性の共起関係が必要な語義同志は一回のリンクトラバースだけで互いにアクセスできるものとする。

2-3. 対象とする表現

形容詞、前置詞、Be動詞、動詞、名詞から構成される基本表現の一部を対象とする。以下に語義の表現について述べる。

A. 形容詞

形容詞は、〈属性〉〈対象〉〈値〉の関係を示す項述語とする。形容詞の語義選択の際には、〈対象〉にあたる名詞が形容詞の指定する属性を持つか否かを検査する。

〈対象〉は統語構造上必須であるが、〈属性〉〈値〉は任意である。〈属性〉〈値〉にはデフォルト値を与えておく。

B. 前置詞

前置詞は、時間・空間・抽象などの関係を示す述語と定義する。これにより、動詞句における必須格・任意格の推定、前置詞句に修飾される名詞句の意味解析などの問題を、前置詞の側から整理することによって同じ形式の問題としてとらえることができる。

基本的には前置詞は2項述語で、〈主テーマ〉〈従テーマ〉を構造として持つ。〈主テーマ〉は前置詞が修飾する意味を示し、〈従テーマ〉は前置詞の目的語の意味を示す。

述語としての意味判定は、〈主テーマ〉〈従テーマ〉の意味カテゴリ・意味特徴などの共起関係を利用する。

C. Be動詞

α Be β の統語形態を持つBe動詞表現は格関係としてはとらえにくい概念である。そこで、Be動詞を〈主題〉

〈叙述〉関係をあらわす述語 Copula として定義する。同様に、it that による強張文も〈主題〉を強張する述語として〈主題〉〈叙述〉の構造を持つ。

D. 動詞

ここでは、必須の統語構造として補文構造を持たない、行為概念に属する動詞について、格構造を仮定する。

3. システムの概要

ここでは、これまでに蓄積した統語解析の文法をそのまま利用することとした。そのため、基本的には統語解析と意味解析の2つの処理系から構成されるシステムが2段階の統語解析系を持っている。以下に各処理部の概要を述べる。

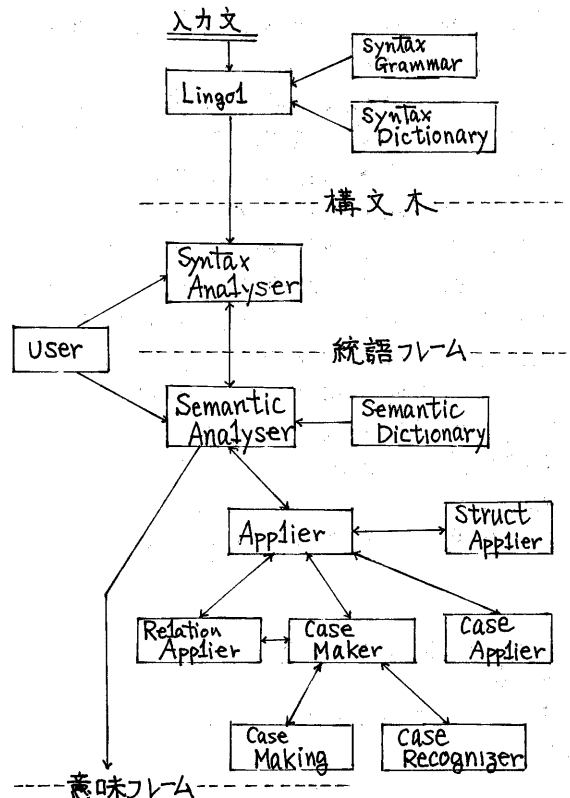


図2 システム構成

3-1. 〈統語解析1〉

処理: 入力文から構文木への変換

n-lingol⁽⁶⁾をパーザとして、Hornbyの動詞パターンを参照して自作した文法⁽⁷⁾により一つ又は複数の構文木を生成する。使用する文法は統語条件だけを用いて制限を行なう。同文法が持つ統語解釈のあいまい性は、CCITT勧告100例文(1文当り平均25単語)の解析を行なうと、平均3~4個の構文木を許す程度である。

lingolが用いる統語辞書は次の内容を含む。(参照図3.)

1. 意味辞書中でのindex
2. 意味辞書を引くときに満たさなければならない統語条件
3. 統語条件により、単語が固有に持つ、統語構造から述語構造への変換規則

```
(be VP1 ((be) 0)
  (copula ←1
2 → (syntax chk (not (emphasis check)))
3 → { (subject theme)
      (complement predication)
      (emphasis
        (syntax chk (emphasis check))
        (complement theme)
        (subject predication))
      (exist
        (syntax chk (not (emphasis check)))
        (subject object))
```

図3 統語辞書の形式

```
(VP6 break ←1
2 → { ((verb) pattern vp6a)
      ((time present)
        (voice active)
        (aspect affirmative))
3 → (break)
4 → { (subject NP (30002))
      (direct object NP (30004))
```

1. 統語フレームの Governor
品詞と Governor
2. Governorから抽出された情報
3. Governorの意味辞書中でのindex
のリスト
4. Governorの持つ統語構造を示す
スロット群

図4. 統語フレームの形式

§ The boy breaks the table.

3-2. <統語解析2>

処理: 構文木から統語フレームへの変換

構文木構造に対して統語解釈を行ない、統語構造上の格を決定する。同処理での主ルーチンSyntax-Analyzerは<統語解析1>により生成された複数の構文木を一つづつ取り出しトップダウンに統語フレームへの変換を行なう。統語フレームは図4の構造である。

3-3. <意味解析>

処理: 統語フレームから意味フレームへの変換

A. Semantic-Analyzer

主ルーチンSemantic-Analyzerは、<意味解析>全体を管理する処理系で以下の機能を持つ。

A-1. 入出力

統語フレームを受け取る。意味辞書を引く。意味フレームを出力する。

A-2. 解析処理部の駆動とback-track

解析処理部Applierに対し、意味的修飾関係を検査すべき意味フレームを渡し、その処理結果を受け取る。Applier又はUserからfailを受け取るとback-trackにより次に検査すべき意味フレームをApplierに渡す。ここでのすべての処理に失敗すると、<統語解析2>のSyntax-Analyzerにfailを返す。これにより、Syntax-Analyzerは次の構文木に対する統語フレームを作り、Semantic-Analyzerに渡す。

B. Applier.

解析処理部Applierは統語的修飾関係を持つ意味フレームの関係を調べ、ネットワーク関係を作る。

Applierは修飾される側の意味フレーム(以後Governorと記す。)のタイプにより、次の4つの処理系の一つを選択する。

B-1. Relation-Applier

Governorが述語構造を持たない場合(名詞に対する修飾関係の処理など)は、Governorの被修飾関係を示すrelationスロットに修飾する側の意味フレームをうめ込む。これは意味ネットワークの逆リンクである。

B-2. Struct/Case-Applier

統語条件により、統語構造から述語構造への変換規則が指定される場合で

a) Struct-Applier

Governorが、格以外の述語構造を持つ場合(前置詞, 形容詞, Be動詞, 冠詞)の処理

b) Case-Applier

Governorが、格構造を持つ場合の処理に分かれる。

ここで扱う述語に対しては統語構造から述語構造への変換規則を統語辞書内に記述する。(図3参照)

ApplierはGovernorが変換規則を持っていないか、Struct/Case-Applierを呼び出し、Struct/Case-Applierは変換規則で指定される述語構造のスロットを作る。

B-3. Case-Maker

ApplierはGovernorが変換規則を持たない場合、次の2つの処理のいずれがえ選択する。

a) Case-Recognizer

統語構造上の必須格を処理する場合、深層格推定規則に基づいて格構造を推定する。

b) Case-Making

統語構造上の任意格を処理する場合、この一歩前のプロセスでStruct-Applierにより前置詞をGovernorとする意味フレームは出来上がっている。前置詞句の意味フレーム中のCase-makingスロットには、同フレームがGovernor(動詞)側から解析される格ラベルとその条件を記述しておく。Case-Makingはこのスロットを解釈して格ラベルを決定する。

B-4. 以下にApplierの処理手順を説明する。The boy breaks the table.の統語フレーム(図4)に対してApplierは次の手順でagent格を作る。

a) ApplierはGovernorのbreakが統語構造から述語構造への変換規則を持っていないので、Case-Recognizerの処理を選択する。

b) Case-Recognizerは、

- the boyは統語構造上では主語であること
- boyとbreakの意味特徴とを使つて格推定を行ない、the boyの意味フレームをきう入したagentスロットを作る。

c) Applierは、Case-Recognizerから受け取ったagentスロットを、breakの意味フレームの格構造としてうめる。

C. 意味辞書

意味辞書は次の内容を含む。

1. 意味カテゴリ
2. 意味特徴
3. 述語構造
4. 述語構造を作るときの条件

(Build

- 1 → (sem_category complex_activity)
- 2 → (sem_feature (volition (ergative)))
- 3 → (case (agent (or (KDP (animate)) (object))))

4

図5. 意味辞書の形式

4. 中間構造の形式

中間構造は、単語の語義に対応するノードを意味的な修飾・被修飾関係に従つてリンクさせた意味ネットワークである。ネットワークのノードは、以下の内容を持つフレームである。

4-1. 意味フレームの形式

A. 意味カテゴリ

概念間の識別のためのカテゴリ

B. 意味特徴

動詞の格推定に用いる情報

動詞	}	+affect	+benefactive
		+causality	+direction
		+ergative	+experiential
		+mutuality	+relation
		+volition	
名詞		+animate	+human

C. self

意味辞書中でのindex

D. information/ modal

名詞 数(単・複) 性格 属性
 形容詞 比較(原級・比較級・最上級)
 節/文 時間, アスペクト etc.
 などの情報

E. 変換規則(mapping rule)

意味構造を得るために適用された変換規則を保存しておく。

F. 構造

case: 動詞の格構造

OAV: 形容詞の〈対象〉〈属性〉〈値〉
関係

struct: Be動詞, 前置詞, 冠詞の構造

relation: 被修飾関係

4-2. ネットワーク表現

A. 表現能力

現段階では以下の文型に対し中間構造を作ることができる。

A-1. 冠詞, 形容詞, 前置詞句により修飾される名詞句

A small box on the table.

A-2 必須格, 任意格を含む動詞句

The boy broke the window by the hammer.

A-3. Be動詞表現

a) NP BE ADJECTIVE

The boy is short.

b) NP BE PP (前置詞句)

The box was on the table.

c) IT-THAT による強調文

It was by the hammer
that the boy broke the window.

A-1, A-3-c) のネットワークの例を付録1, 2に示す。

B. 必須格と任意格によるネットワークのちがひ

統語構造上の任意格を構成する前置詞句は、意味構造上で必須格になるか任意格になるかにより、異なるネットワーク構造を作る。以下に前置詞byの例(agent格, location格, 〈命題-命題関係を示す述語〉)を説明する。

必須格の場合は動詞と名詞の意味関係は格ラベルにより明確にとらえられる。したがって図6-aではなくRelationのノードのあらわす意味は格ラベルagentと等しく、動詞側から見た前置詞byの意味は単に格マーカアしかない。しかし任意格の場合は、格ラベルよりは前置詞のノードのほうが細かく意味を指定している。動詞の側から前置詞が規定する詳細な関係にアクセスするためには、前置詞のノードを格ラベルの下下部構造として解釈する必要がある。

そこで、前置詞句が深層構造上の必須格と解釈される場合は、動詞の必須格リンクは前置詞句側のsub-themeリンクと同じものを示すこととし、任意格と解釈される場合は、動詞の任意格リンクは前置詞句のノード自身を示すこととする。

これを行なうためbyの意味辞書は次のように記述する。(図6-c参照)
case-making というスロットに、動詞側

The box was broken by the boy.

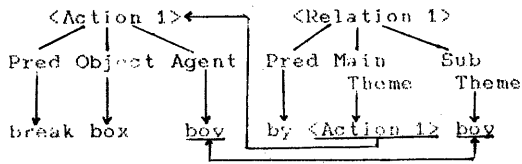


図 6-a 必須格 Agent

The boy stopped by the corner.

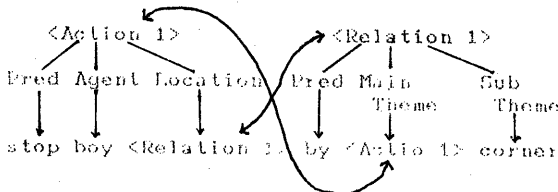


図 6-b 任意格 Location

- (by
- (sem_category relation)
- (sem_feature agent)
- (main_theme (volition))
- (sub_theme (animate))
- (case_making
- (sub_theme agent))
- (by 3
- (sem_category spatial)
- (sem_feature location)
- (main_theme)
- (sub_theme)
- (case_making
- (self location))

図 6-c by の意味辞書

The boy broke the window
by {throwing} the hammer.
using

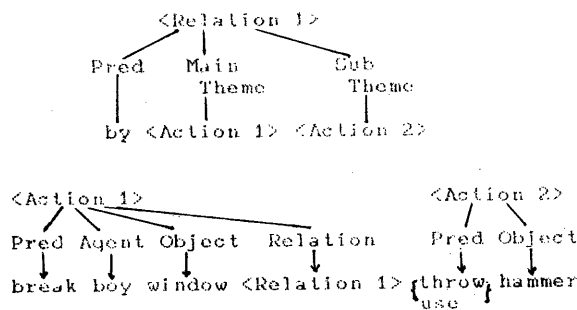


図 6-d <命題-命題関係>

から前置詞句を格要素として解釈する際の処理規則が書いている。図6-c上の例は、前置詞句の意味構造の Sub-theme を agent と解釈する規則である。

一方場所格など任意格と解釈される場合は、self という指示によって前置詞のノード全体を任意格として解釈する。(図6-c下参照)

5. 検討

5-1. 修飾する側から文の意味構造をとらえる意義

翻訳に利用される主な関係は、主節から見た関係であるが、ここでは中間構造の記述に意味ネットワークを採用して、修飾する側の構造も作っている。

図6-a, b, d では修飾側の意味構造の重要性が a, b, d の順で増している。

図6-a では、<Relation 1> ノードは格ラベル agent と等しい意味しか持たない。

図6-b では、<Relation 1> ノードは、格ラベル location の意味の他、location 格の下部構造の意味を含む。

図6-d では、<Relation 1> ノードは、命題と命題の論理関係を示す述語としての性格を強く持ち、「<Action 1> に対して <Action 2> はそのやり方を説明する」という関係を示す。これを強いて主節の側からとらえる場合には、「instrument 格の概念を拡張して命題を下部構造として許すようにする」又は「手段格という格ラベルを設ける」などの操作が必要である。

この傾向から「意味構造上の任意性が強まるにつれ修飾する側からも文の構造をとらえておく意義が生ずる。」と考えることができる。

5-2. ネットワーク記述の利点

中間構造を意味ネットワークで記述することにより、翻訳システムにとって以下の利点が生ずる。

A. 文の意味構造が主節の動詞について整理されている他、主節の動詞に対する個々の修飾要素についても整理されており、種々の視点から見た情報をトランスファ、生成の際に利用できる。

B. 文のトップレベルの構造として格関係以外の関係を許すことができる。図6-dでは、〈Action 1〉ノードではなくbyの持つ命題-命題関係を示す〈Relation 1〉ノードを文のトップレベルの構造としていいる。これは、「修飾-被修飾関係が格関係としてとらえにくい場合には、修飾語が持つ抽象関係の構造を文の意味構造として主節の意味構造をその中に吸収する。」ことにより得られる。これにより、意味解析に対してかなり柔軟な態度をとることができる。

6. まとめ

本稿では、結合価文法に基づく統語構造から深層格推定を行なう実験システムを作成した。本システムでは、

1. 一般的格推定規則
2. 1.以外の任意格処理
- 3 統語条件から単語固有の格構造を指定する処理

の3通りの格推定を行なうことができ、これらをあわせて用いることにより、実用レベルの格推定手順を記述できる。これにより一般的格推定規則の能力を実験的に検証することが可能となった。現在のところ、1の格推定規則としては、既に提案したものをそのまま使用している。今後は、格推定のできるだけ多くの部分を一般的格推定規則で行なえるよう補強を計る。

7. 謝辞

日頃御指導いただくKDD研究所 鍛冶所長、野坂副所長、橋松所長、武田第一特別研究室長に感謝する。プログラム上の御協力をいただいた

(株)SCC 井上氏に御礼申し上げる。

文献

- 1 D. J. Allerton: Valency and English Verb Academic Press, 1982
- 2 鈴木, 橋本他: 英文の深層格推定のための動詞分析, 情報処理学会第28回全国大会, 6L-4, 1984
- 3 野垣内, 橋本他: 日本語文の助詞の機能の多様性とその解析について 情報処理学会第28回全国大会, 6L-8 1984
- 4 Simmons, R.F., Semantic Network: Their Computation and Use for Understanding English Sentences, "Computer Models of Thought and Language" Shank and Colby Freeman 1973
- 5 Woods, W. A. WHAT'S IN A LINK Foundations for Semantic Networks, "Representation and Understanding", Bobrow/Collins Academic Press, 1975
- 6 敵見 lingol の m 進木への拡張 東工大 1980
- 7 橋本他: 英語構文解析のための文法作成, 昭和58年度電子通信学会総合全国大会, 1331.

付録1. 名詞句

```

/---(OBJECT 1 box)-----
      information:
      number: singular
      gender: neuter
      case: objective
      sem_category: produc
      self: box
      relation:
        deictic == (STATE 1 a)
        attribute == (STATE 2 small)
        spatial == (STATE 3 on)
/---(STATE 1 a)-----
      sem_category: deictic
      self: a
      struct:
        determinee == (OBJECT 1 box)
/---(STATE 3 on)-----
      sem_category: spatial
      sem_feature: location
      self: on
      struct:
        main_theme == (OBJECT 1 box)
        sub_theme == (OBJECT 2 table)
/---(OBJECT 2 table)-----
      information:
      number: sing
      gender: neut
      case: object
      sem_category: p
      self: table
      relation:
        deictic == (STATE 4 the)
/---(STATE 4 the)-----
      sem_category: deictic
      sem_feature: +definite
      self: the
      struct:
        determinee == (OBJECT 2 table)
/---(STATE 2 small)-----
      information:
      comparison: positive
      sem_category: attribute
      sem_feature:
      OAV-relation:
        attribute: size
        value: low
        object == (OBJECT 1 box)
      self: small
Text 1
A small box on the table
    
```

付録2 IT THAT 強調構文

```

/---(ACTION 1 break)-----
      modal:
      verb-pattern: vp0a
      time: past
      voice: active
      aspect: affirmative
      sem_category: impact
      sem_feature: +volition +ergati
      case:
        self == break
        agent == (OBJECT 1 boy)
        instrument == (OBJECT 3 hammer)
        object == (OBJECT 2 window)
/---(OBJECT 1 boy)-----
      information:
      number: singular
      gender: masculine
      case: nominative
      sem_category: human
      sem_feature: +animate
      self: boy
      relation:
        deictic == (STATE 2 the)
/---(OBJECT 2 window)-----
      information:
      number: singular
      gender: neuter
      case: objective
      sem_category: products
      self: window
      relation:
        deictic == (STATE 3 the)
/---(OBJECT 3 hammer)-----
      information:
      number: singu
      gender: neute
      case: objecti
      sem_category: pr
      self: hammer
      relation:
        deictic == (STATE 5 the)
/---(STATE 1 be)-----
      modal:
      verb-pattern:
      time: past
      aspect: affir
      sem_category: st
      self: emphasis
      struct:
        predication == (ACTION 1 break)
        theme == (STATE 4 by)
/---(STATE 4 by)-----
      sem_category: relation
      sem_featute: instrument
      self: by-2
      struct:
        main_theme == (ACTION 1 break)
        sub_theme == (OBJECT 3 hammer)
Text 3-c
It was by the hammer
that the boy broke the window.
    
```