

昔話「桃太郎」を対象とする 自然言語文の意味構造自動生成

上田世志、瀧口伸雄、小谷善行

東京農工大学工学部電子情報工学科

われわれは、日本語の物語文に対応する意味構造を自動的に生成するシステムELF(an Experimental natural Language understanding system for Fairy)についての研究を行っている。意味構造生成の対象とした自然言語文は、物語文である昔話「桃太郎」の一部である。ELFは、構文解析系、格フレーム生成系、意味構造生成系から構成される。ELFは、分かち書きされた対象文を入力として、対応する意味構造を出力する。生成される意味構造の表現形式は、フレーム表現を基にした物語知識ベースFAIRY(Frame system for Acquisition and Inference of Representation in a storY)^[1]の形式をとる。本稿では、ELFの構成および生成される意味構造について述べる。

Automatic Generation of Semantic Structure from a Fairy Tale 'Momotarou'

Hiroshi UEDA, Nobuo Takiguchi, Yoshiyuki KOTANI
Department of Computer Science
Tokyo University of Agriculture and Technology
2-24-16, Nakamachi, Koganei-shi, Tokyo, 184, Japan

We research an experimental natural language understanding system ELF which generates semantic structure automatically from sentences in a Japanese fairy tale, 'Momotarou'. The system is described to make semantics about some sentences in the fairy tale. ELF consists of the three parts: the syntax parser, the case frame generator, and the semantics structure generator. The semantic structure generated is described by the form of the knowledge base FAIRY(Frame system for Acquisition and Inference of Representation in a storY)^[1] based on frame system. This paper describes ELF's components and semantic structure.

1. はじめに

本研究では、物語の自然言語文に対応する意味構造を自動的に生成するシステムELF (an Experimental natural Language understanding system for Fairy)を作成した。意味構造生成の対象とした自然言語文は、物語文である昔話「桃太郎」の一部を用いた。

生成される意味構造の表現形式は、物語知識ベースFAIRY[1][2]の形式をとる。FAIRYは「人の手で物語を理解した形を表現する」という見地から知識ベースの設計を研究したものである。

さらに、本研究では「計算機プログラムによる意味構造の自動生成」を目的として、自然言語文からFAIRYの形式による意味構造を自動生成するシステムの作成と、その実験を行った。

2. システムELFの構成

ELFの構成を次の図1に示す。

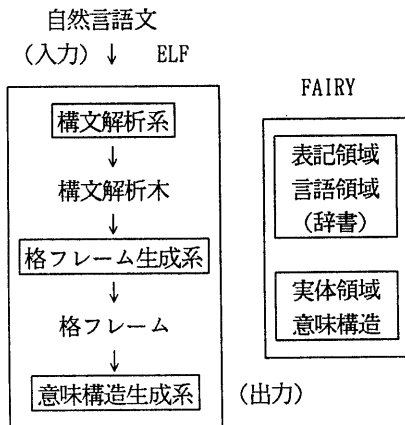


図1 システムELFの概要

ELFは大きく分けて、

- ・ 構文解析系
- ・ 格フレーム生成系
- ・ 意味構造生成系

の三つの部分から構成される。

各部分は、必要に応じてFAIRYから情報を引き出す。各部分の間でのデータの入出力を、次に示す。

- ・ 構文解析系は、自然言語文を入力し、構文解析木を出力する
- ・ 格フレーム生成系は、構文解析木を入力し、格フレームを出力する

・ 意味構造生成系では、格フレームを入力し、FAIRY形式の意味構造を出力する
すなわちELFに入力として「桃太郎」の文を与えると、FAIRY形式による意味構造の生成、出力を行う。

構文解析系は、文献[3]を基礎として拡張を行った。また各部分に必要なFAIRYからの情報は、第4章で述べるIPAL[4][5]を知識辞書としてFAIRYへ拡張したものである。

構文解析系、格フレーム生成系、意味構造生成系、知識ベースFAIRYについての説明を簡単に行う。

(1) 構文解析系

構文解析系は、BUP[6]を中心に構成する。構文解析時に、構文解析の曖昧さを少なくするため、格支配を用いた意味チェックを同時に行う。

(2) 格フレーム生成系

格フレーム生成系は、構文解析系から与えられる構文解析木を入力として、意味構造生成の骨組みとなる格フレームを生成するプログラムである。

(3) 意味構造生成系

意味構造生成系は、格フレームを入力として、動詞の意味構造を生成するプログラムであり、この生成のためにIPALのデータを用いる。

(4) 知識ベースFAIRY

知識ベースFAIRYは、表記領域、言語領域、実体領域、管理領域の四領域と推論系からなる。

本研究のシステムで扱うFAIRYは、その初期状態として表記領域と言語領域だけを与える。表記領域、言語領域は、構文解析系、格フレーム生成系、意味構造生成系における辞書情報である。本研究では、この部分にわれわれがもつ既存の辞書を用いるほかに、意味構造生成系に必要な、IPALによる知識辞書をここに作成、格納し、利用する。

実体領域には、本システムの最終出力である意味構造が格納される。

3. 知識ベースFAIRY

システムELFが最終的に出力する意味構造は、このFAIRYの形式をもつ。FAIRYの表現形式における大きな特徴を次に示す。

- ・ FAIRYは、すべてPrologで記述されている。

- ・FAIRYの表現形式はフレーム構造である。
- ・FAIRYは、表記領域、言語領域、実体領域の三つで表現される。

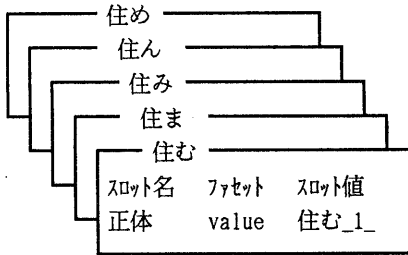
FAIRYの三つの領域について次に説明する。

3.1 表記領域

表記領域は、単語の読み（つまり文字列）の情報を表現するフレーム群が属する領域の名称である。例えば、「住め」、「住む」...などという文字列の表記領域のフレームは、図2のように示される。ここで「住む_1」とは、次節で説明する「住む」の言語領域におけるフレーム名である。

```
住め(正体, value, 住む_1, _).
住ん(正体, value, 住む_1, _).
住み(正体, value, 住む_1, _).
住ま(正体, value, 住む_1, _).
住む(正体, value, 住む_1, _).
```

(a) Prolog表記



(b) フレーム表記

図2 表記領域の例「住む」

3.2 言語領域

言語領域は、単語の文法的知識を表現するフレーム群が属する領域の名称である。例えば、動詞「住む」の言語領域のフレームは、図3に示される。表記を簡潔にするため図3(b)の波線部は省略化してある。ここで「実_住む_1」とは、次の節で説明する「住む_1」の実体領域におけるフレーム名である。

辞書項目とその値は、スロット名とスロット値によって表現される。

名詞と動詞の言語領域による辞書項目を、表1、表2に示す。

名詞に関して、例えば「おじいさん」の言語

領域には、その属性は「人間名詞」であるといったことを記述する。

```
住む_1(名称, value, 住む, _).
住む_1(未然1, value, 住ま, _).
住む_1(未然2, value, 住も, _).
住む_1(音便, value, 住ん, _).
: : : :
住む_1(品詞, value, 動詞, _).
住む_1(活用型, value, 五段, _).
住む_1(格支配, value,
[[ニガシ, ガ], [ハシヨ, ニ]], _).
: : : :
住む_1(実体, value, 実_住む_1, _).
```

(a) Prolog表記

住む_1		
名称	value	住む
未然1	value	住ま
未然2	value	住も
音便	value	住ん
:	:	:
品詞	value	動詞
活用型	value	五段
格支配	value
:	:	:
実体	value	実_住む_1

(b) フレーム表記

図3 言語領域の例「住む_1」

表1 名詞の辞書項目

スロット名	スロット値
品詞	名詞
格支配構造	格支配構造の記述 (サ変名詞に限って記述)
補助属性	普通名詞、動作名詞、形式名詞 副詞的名詞、人称代名詞、 疑問代名詞、指示代名詞
意味属性	人間名詞、動物名詞、抽象名詞 場所名詞、具象名詞、時間名詞 方向名詞、etc.

表2 動詞の辞書項目

スロット名	スロット値
品詞名	動詞
活用型	五段サ行型、上下、カ変型、サ変型、アル型、ウル型 etc.
活用形	未然1、未然2、連用、終止、連体、假定、命令
格支配構造 (注)	(a)人間、具象物、場所、抽象行為、不定 (b)が格、を格、に格、から格、と格

(注)格支配構造のスロット値は、(a)と(b)との組合せによるPrologのリスト表現によって記述される。

3.3 実体領域

実体領域は、ものごとの概念を表現するフレーム群が属する領域の名称である。実体領域は一般的な概念を表すフレーム群の総称であるクラスと、個々の特定の概念を表すフレーム群の総称であるインスタンスに分かれる。

クラスとインスタンス間は、上位・下位の継承関係で結合され、各領域間のフレーム群も互いに結合されている。例えば、「おじいさん」という概念は、図4のように各領域で示され、互いに結合される。図4の四つのフレームは、上から順にそれぞれ次の領域に属する。

(a)実体領域クラス

「おじいさん」に関する一般的な概念。
例：「人間である」、「高齢である」といった概念。

(b)実体領域インスタンス

ある特定の「おじいさん」に関する概念。
例：「あるおじいさんが、ある場所である時に何を行ったか」といった概念。

(c)言語領域

単語「おじいさん」の文法的属性。

(d)表記領域

「おじいさん」という表記。

実際の例として、「おじいさんが山に出かける」という概念を表すインスタンスフレーム群の例を図5に示す。図5のフレームは、互いにスロットによって結ばれている。

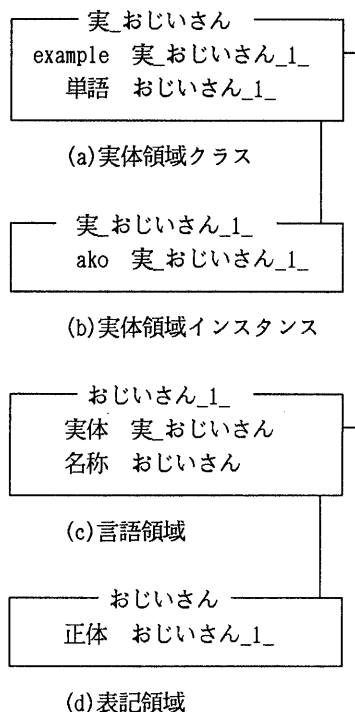


図4 三領域にまたがる例

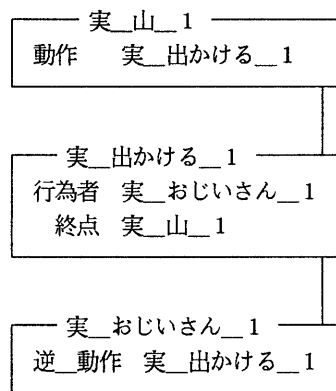


図5 実体領域の例

4. IPALによる動詞知識辞書

構文解析系や意味構造生成系に使われる辞書とその扱いについて述べる。ELFにおいて辞書とは、FAIRYの表記領域、言語領域のことを指す。

本研究においては、これまでわれわれが開発してきた辞書に加えて、知識辞書IPAL[5][6]を使用する。IPALは、構文解析、意味解析、意味

構造生成のための情報として十分であると思われる、市販のマシンリーダーダブルな磁気テープに納められた、動詞に対する大規模な辞書である。

構文解析において、解析できる文を増やすには、そのための辞書も増やさなければならない。また、意味構造生成をより適切に行うには、充実した意味情報をもつ辞書が必要である。人が自ら意味構造を生成する場合は、その裏にある膨大な知識を利用している。システムにも同様にそのための知識が必要となる。すなわちシステムの各部分に必要な情報として、充実した意味的知識をもった大規模な辞書が不可欠となる。すなわち、自然言語理解システムを構築する上で、大規模な知識辞書に対する要求がある。

本研究では各処理を行うべき必要な情報として、IPAL[4][5]を取り入れ利用する。そのためにIPALのデータからFAIRYの辞書への変換方法について検討し、IPAL変換系を作成し、その実行によりFAIRYの形式をもつ知識辞書に変換した。この辞書は大規模なファイル（大きさは合わせて約17Mbyte程度）であり、工夫しなければProlog上で扱いきれない。それらの辞書ファイルを実際にProlog上から読出し可能な辞書ファイルの構成形式を検討し、それらの形式による辞書の構成、辞書引き方式も検討、実現した。辞書構成と辞書引きについては、文献[9]に示されるものを基礎として拡張した。

```
でかける_1_1(品詞, value, 動詞, _).
でかける_1_1(仮定, value, 出掛けれ, _).
でかける_1_1(意味, value, 'ある目的でどこかに行く.', _).
でかける_1_1(格1, value, ガ, _).
でかける_1_1(格2, value, 'ニ/へ*', _).
でかける_1_1(格3, value, ニ, _).
でかける_1_1(述語素, value, 'LG1/DR1, MT1, A', _).
でかける_1_1(素1, value, 'HUM', _).
でかける_1_1(素2, value, 'LOC', _).
でかける_1_1(素3, value, 'ACT', _).
:
:
```

図6. IPALからの辞書作成例

これにより、FAIRYにおける

- ・ 基礎的な知識辞書の拡充
- ・ 辞書環境に対する整備

が実現された。

IPALの情報から作成されたFAIRY形式の動詞辞書（言語領域）「でかける_1_1」の一部を図6に示す。今までFAIRYになかった辞書項目は、IPALのものをそのまま取り入れている。

5. 構文解析系

構文解析系は、インタフェース、構文規則変換系、BUP[6]、意味チェッカからなる。

インタフェースは、操作者とELF間の情報のやりとりを行うプログラムであり、構文解析木表示プログラムも含まれる。構文解析系以降の意味解析系および意味構造生成系を自動化しているため、構文解析系内のインタフェースはシステム全体のインタフェースとなる。

構文解析系には、それを支援するものとして、構文規則変換系をもち、DCGによる記述の構文規則ファイルを実行可能なBUPに変換する。

意味チェッカは、「名詞+格助詞」とそれが係る用言の格関係が正しいかどうかをチェックするプログラムである。これは、構文解析時に実行される。

5.1 構文規則の検討

本システムの構文規則はDCGで記述され、青島[2]の研究で作成された構文規則をもとにして作られているが、「桃太郎」に出てくる文章に対応するために発展させた。

構文規則は、文献[8]の構文規則をもとにして作られたが、本研究で扱う文章の構文解析を行える構文規則を作成するために、

- (a) せりふに対する構文規則
- (b) 「にしては」等の助詞として扱われる連語に対する構文規則

についての検討を行う必要があることが分かった。

(a) について、「桃太郎」の文章には、せりふの文が多数ある。せりふを扱うのに、次の図7に示す構文規則を導入した。構文規則は、説明のために簡略化するが、本来のものは、各々の（非）終端記号がPrologの複合項の形をとり、構文解析に必要な情報を構文解析時に伝達する

仕組みになっている。

文	→ 引用部, 結び.
文	→ 接続詞, 引用部, 結び.
引用部	→ 括弧1, せりふ, 括弧2.
括弧1	→ [' 「 '].
せりふ	→ [_ 文字列].
括弧2	→ [' ' '].
述句	→ 引用部, 引用結合子, 述句.
引用結合子	→ [と].

図7 せりふの構文規則

文中で、括弧は特殊なものであるため、プログラムを構文規則内に記述し、処理を行う。

せりふ文を含む文の構文解析の例を図8に示す。

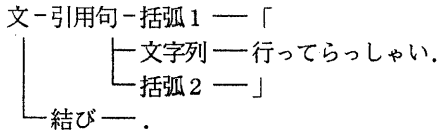


図8 構文解析木の例

(b)については、例えば「桃にしては大きい」という文に存在する「に・し・て・は」のような助詞がらみの連語の構文規則を検討した。このような助詞的な連語には、「において、に関して、によって」などがある。「にしては」を、単語レベル、すなわち品詞の観点で分けると、

に：格助詞

し：動詞「する」の連用形

て：接続助詞

は：係助詞

となる。しかし、通常これらは、ひとまとまりの連語とされる。各単語部分「に・し・て・は」によって表される意味を考えるだけでは、この単語群の本当の意味をとらえることはできない。すなわち個々の単語がもつ意味を繋ぎ合わせてその全体の意味を表せるレベルではなく、このまとまりが意味の最小単位である。ここで、

「桃にしては大きい」といった場合、単に「桃が大きい」ということのみを意味するのではなく、「桃という条件を考えにいれると…」あるいは「桃の割には…」などのニュアンスが付加されている。すなわちこれを、いわゆる様態としてとらえることができ、意味構造生成の立場からは、そのための情報を記述する必要がある。

よって、構文解析系においては、助詞を含む連語は、図9に示すように、構文規則中に分かち書きされた形で明示的に記述し、また意味構造生成に必要と思われる様態情報を記述するための情報を辞書として図10に記述できる形式を定めた。

情況句 → 名詞句, 情況格.

情況格 → [に, し, て, は]_.

図9 助詞的連語の構文規則の例

— 助詞連語_にしては —	
名称	にしては
品詞	助詞
属性	連語
様態	にしては
分離形	[に, し, て, は]
:	:

図10 助詞的連語の辞書の例

6. 格フレーム生成系

格フレーム生成系では、意味構造の基礎となる格フレームを生成する。格フレームは、文の構造を述部と名詞との依存関係によって記述したものである[7]。

構文解析木は、Prologの複合項によって、文の構造を示しているが、構文解析系で使われる処理用の引数が残っていることと、その表現形式自体がその後処理しにくい形式であるので、そのままの形では扱いにくい。よって、構文解析木からではなく、中間段階として格フレーム生成系により生成された格フレームリストを用いて、その後の意味構造生成に備える。この格フレームリストを、意味構造生成系に入力して

意味構造を生成する。格フレームはリスト構造であるが、フレーム表現の中間形式としてとらえることができる。「おじいさんが山へ柴刈に出かけた」を例として、その格フレームを図11に示す。

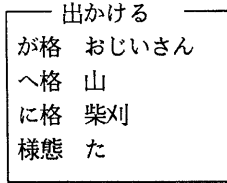


図11 格フレームの例

7. 意味構造生成系

意味構造生成系は、格フレームを入力として、意味構造を出力する意味構造生成プログラムである。作成する意味構造の情報は、格フレームと辞書情報である。これらの情報をもとにして意味構造を生成する。これら処理して、意味構造を生成する手続きの方針を次に示す。

- (1) 格情報からインスタンス動作フレームの生成
動詞について、格支配による情報、ここではIPALの述語素による表とFAIRY辞書の情報を用いて動作フレームのスロットを決定し、インスタンスの動作フレームを作る。
- (2) 品詞からフレームへの対応づけ
文中の単語の品詞をもとに、各々の単語の実体領域が属するクラスフレームを決定し、インスタンスフレームを作る。
- (3) スロット表による同領域におけるフレーム間の結合
各フレーム間の結合を行う。
- (4) 三領域間の結合
クラスフレーム、言語領域などの階層間(縦)の結合(ako、exampleスロット生成およびクラスフレーム生成・マッチング処理)を行う。
- (5) 意味構造生成のための推論
指示語や省略されているものの推論すなわち、
・ 代名詞、指示連体詞が示す対応物

を以前の意味構造から推論

- ・ 省略されているものを以前の意味構造から推論

- ・ 文の順番が時間の流れに対応しているものという事実からの推論

を行う。

(1)において、動詞の格情報は、動作フレームを生成する最も重要な要素である。

名詞の意味属性と格の組合せにより、その名詞の実体領域の概念が割り当てられるスロットを決定する。この格とスロットの対応は、個々の動詞によって異なる。

この問題は、IPALの動詞辞書を用いて解決することができる。IPALは、個々の動詞に対してスロットに対応するものとして、述語素という辞書項目をもっているからである。すなわち、IPALの述語素を用いてスロットを決定する。この述語素に対する、訳語(述語素の意味を表す日本語)を基礎として、スロットの生成を行う。

IPALの扱う動詞は、語彙体系上ならびに使用頻度上重要であると考えられる基本的な861個の動詞である[4]。そのため、本研究で扱う対象文に存在しない動詞がある。本研究では、IPALに含まれる動詞は、IPALの辞書情報を使い、それ以外の動詞はFAIRYの辞書情報を用いた。

FAIRYの辞書情報とIPALの辞書情報の並存を可能にするために、名詞の属性と動詞の格支配情報をまとめた。FAIRYの格支配構造、意味属性に、IPALの文型、意味素性を対応させる。格支配構造はそのまま取り入れた。名詞の意味属性とIPALの意味素性は表3のように対応がとれる。

表3 IPALとFAIRYとの属性の対応

IPAL	FAIRY
CON	具象名詞
ANI	動物名詞
HUM	人間名詞
ABS	抽象名詞
LOC	場所名詞、方向名詞
TIM	時間名詞

これらを下位・上位も含んだ属性にまとめ、FAIRYの辞書において、IPALとFAIRYのものを共存させた。

意味構造生成系について、格フレームのもつ

格情報からインスタンス動作フレームの生成を行う部分を実現した。格フレームに、IPALの情報を用いて、動作フレームのスロット名とスロット値の割り付けを行い、意味構造を生成する。

「おじいさんは山へ柴刈に出かける」の格フレームを例に説明する。「出かける」における辞書情報として述語素は、

LG/DR1, MT1, A

LG/DR1, MT2, A

の二つがある。IPALによるこれらの述語素の訳語と、関連する格形式をそれぞれ表4に示す。

表4 「出かける」における述語素の情報

述語素	訳語	関連する格形式
LG/DR1	空間的着点/方向	ガ/ガ・ヘ
MT1	動機	ガ・ニ
A	動作主	ガ
MT2	動機	ガ・ニ

また、IPALでは、名詞句に対する制限として、その名詞が「動物」や「人間」などといった意味をもっているということを示す「意味素性」という情報をもっている。

例えば、動詞「出かける」については、

HUM (人間) ガ

LOC (場所) ニ/ヘ

ACT (動作) ニ

といった情報をもっている。すなわち、名詞句で「か格」を取る場合には、その名詞は「人間」の属性をもたなければならないことを示している。

図12の格フレームのスロット値の名詞の属性を調べ、動詞「出かける」の意味素性と述語素を調べ、マッチングを行い、次の動作フレームを得る。

出かける_1_	
動作主	おじいさん_1_
方向	山_1_
空間的着点	柴刈_1_
動機	柴刈_1_

図12 生成される動作フレームの例

8. まとめ

本研究では、「計算機プログラムによる意味

構造の自動生成」という立場で、自然言語文からFAIRYの形式による意味構造を自動生成するELFの開発、評価を行った。このために文献[3]の構文解析系を拡張し、格フレーム生成系を実現し、意味構造生成系について設計とその一部を実現し、その際にELFに必要となる動詞の知識辞書を設計、実現した。

意味構造自動生成の実験の結果、人手で解析して知識を表現することを意図したFAIRYに対して、システムによる自動生成によって、ある程度の意味構造を生成できた。

なお本システムの実現と実験評価は、SONY-NEWS上でK-Prologを使用した。

9. 参考文献

- [1]佐々木泰, 西明美, 高野啓, 青島正明, 高田正之, 小谷善行: 物語の意味構造を可能な限り記述する実験, 情報処理学会第34回全国大会講演論文集6L-5, pp. 1651-1652(1987)
- [2]岡田和久, 佐々木泰, 高田正之, 小谷善行: 物語知識ベース—クラス概念の設計—, 情報処理学会第36回全国大会講演論文集3P-8, pp. 1433-1434(1988)
- [3]青島正明, 高野啓, 高田正之, 小谷善行: 自然言語文から意味構造を生成する方式, 情報処理学会第34回全国大会講演論文集4X-1, pp. 1321-1322(1987)
- [4]情報処理振興事業協会技術センター: 計算機用日本語基本動詞辞書 I P A L (Basic Verbs) —解説編—, 情報処理振興事業協会(1987)
- [5]情報処理振興事業協会技術センター: 計算機用日本語基本動詞辞書 I P A L (Basic Verbs) —辞書編—, 情報処理振興事業協会(1987)
- [6]松本裕治, 田中穂積: Prologに埋め込まれたbottom-up parser: BUP, 自然言語処理34-6(1982)
- [7]草薙裕, 南不二男, 中野洋, 吉田夏彦: 朝倉日本語新講座4 文法と意味II, 朝倉書店(1985)
- [8]西村恕彦, 水谷静夫, 尾上圭介, 田中幸子: 日本語基本文法単文編, 電子技術総合研究所研究報告783(1978)
- [9]上田世志, 岡田和久, 高田正之, 小谷善行: 物語知識ベースFAIRY—構文解析系における辞書の設計—, 情報処理学会第40回全国大会講演論文集7F-2, pp. 501-502(1990)