

知識ベースにおける動詞的概念の学習

岡田和久、高田正之、小谷善行

東京農工大学工学部数理情報工学科

自然言語における動詞は知識ベース上で動詞的概念として表現される。本報告は動詞的概念の表現方法と獲得方法について述べる。

知識ベースにおける動詞的概念は特有の属性値と、属性値に付随する確からしさの数値から表現されている。動詞的概念は前後の状態を変化させるものとしてとらえられている。この属性情報を入力するデータから獲得し、動詞的概念を構築する。格納するとき属性の出現回数から確からしさの数値を決定する。

Learning of Verb Concepts in Knowledge Base

Kazuhiisa OKADA, Masayuki TAKATA, Yoshiyuki KOTANI

Department of Information Science
Tokyo University of Agriculture and Technology
2-24-16, Nakawachi, Koganei-shi, Tokyo, 184, Japan

Each verb of natural language is represented as a frame of verbal concept in knowledge base. We propose a method of describing and acquiring concepts of verbs.

Verbal concepts in knowledge base is composed of attribute values and strength parameters of the links of the attribute values. The concepts are formulated as state changing by acting the action. The verbal class concepts are configurated by the information of instance attribute from input data. The system determines the link strength parameters by counting occurrence of attribute.

1. はじめに

動詞は自然言語理解において重要な意味をもつ。

動詞を表現する方法には最小単位となる要素を組み立てて表現するもの、機能により分類するものなどがある。これらの研究では動詞および動作は、人間がもつ概念を人間が計算機上に構築することで表現する。本研究では動詞的概念を一定の規則を用いて知識ベースに自動的に構築を行う。知識は一般的な概念をあらゆるクラス概念と個々の事柄を記述するインスタンスにわけることができる。本研究はフレーム型知識ベースにインスタンスとして入力された動詞的概念を一般化し、クラス概念に格納する。

2. 動詞的概念

われわれが設計を行っている知識ベース Fairy (Frame system for Acquisition and Inference of Representation in a story) は、物語の内容を表現することが可能な記述能力を備えることを目標としている [1][2]。

Fairy で物語を記述する場合は動作、具象物、場所などのフレームを使用し、それらのフレームとおたがいリンクすることにより表現される。

Fairy における「動詞的概念」とは、

- ・「いく」、「いる」などのように文法的に動詞に相当する
- ・「レストランで食事をする」といった特定の動作を表わしている

ものである。「いく」、「いる」などの動詞より「レストランで食事をする」は下位に位置し、サブクラスとして表現される。

動詞的概念は動詞的概念に特有な属性値と前後の動作とのリンクにより表現される。たとえば、

いく . . . あるところに移動する

なげる . . . 腕を振って、もっていたものを離れたところへ飛ばす

のような動詞的概念は、「行為者」や「終点」、「対象」などといった属性や連続した動詞的概念のつながりとして表現される。

本研究は前後の状態の変化に着目して設計された動作の表現方法を用いて動詞的概念を獲得する方針を考えた。

3. 動詞的概念の構築

動詞的概念は、物語を記述する上で重要な役割を演じる。動詞的概念は意味上、文章構造の中心に位置し、文章中の意味を表現するための、多くの属性をもつ。ここではこの動詞的概念を文章中から獲得し、文章の意味構造を構築するための全体像を示す。

動詞的概念構築の全体像は次の通りである。

- ・物語を動作を中心とした入力データに変換する
- ・入力データを知識ベースに格納する
- ・格納と同時に出現回数を変更する
- ・動詞的概念のもつ属性値を移動、変更する
- ・動的に変化する確からしさにより動作フレームの変更、構築を行う

4. 動作フレーム

動作フレームは、物語を記述するうえで、知識ベースの中心となる。動作以外の各フレームは、動作フレームの属性として、知識ベース上に記述される。動作以外のフレームも、たとえば、「人間」は、「動物」の中の一種類である、というような上下関係や、時間の前後関係などを記述してある。動作フレームも他の動作フレームの属性として記述されることがある。たとえば、「おばあさんは川へ洗濯をしにいきました」

という文章は、「いく」フレームの

- ・「行為者」スロットに“おばあさん”フレーム
- ・「終点」スロットに“川”フレーム
- ・「目的格」スロットに“洗濯する”フレーム

が格納される。

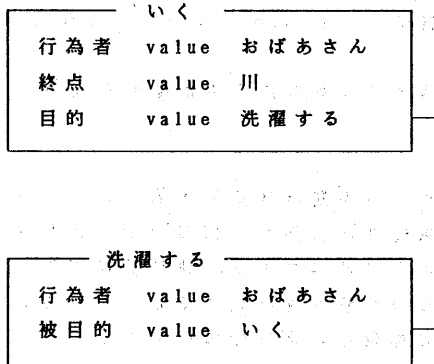


図 動作「いく」と「洗濯する」のインスタンス

動作“洗濯する”は、動作“いく”と従属関係となり“いく”フレームと“洗濯する”フレームは、「目的」スロットと「被目的」スロットという属性で結ばれる。

動作フレームは、

- ・動作名（フレーム名）
- ・深層格（スロット名）
- ・属性値（ファセット名）
- ・値（データ名）
- ・デモン（手続き名）

を組みあわせることで構成されている。深層格と属性値は、格支配構造を用いた。

実体領域中の動作フレームの動作名は、そのままフレーム名として使用されている。動作フレームの動作名は、文章中の動作から得られる。

深層格とは、動作に要求される属性であり、たとえば「おじいさんは山へ柴刈

りにいきました」という文章なら、

- ・「おじいさんは」から行為者が、
- ・「山へ」から終点が、
- ・「柴刈りに」から目的が、

獲得することができる。これら属性の名前の多くは、あらかじめ人間が決定したものを使用する。他の実体領域のフレームが、深層格の値としてリンクされる。深層格は入力データから得ることができる。入力文を格納した知識ベースを調べることにより、動作の深層格がどのように使用されているかを推定することができる。

属性値は、格納するフレーム中の値の属性である。実体領域中の具象物フレームの上下関係を検索することで、判定できる。たとえば上の文章の動作「いく」の「行為者」「おじいさん」は“人間”の一種だとわかる。

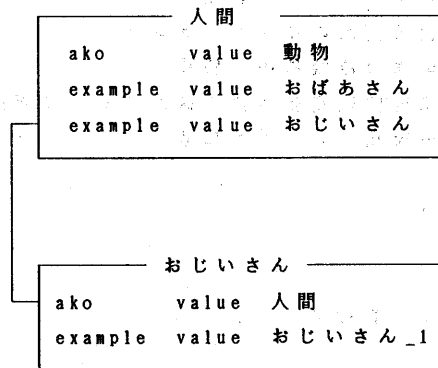


図 具象物“人間”と“おじいさん”の上下関係

このことから、動作や値の一般化を行い、一般的な「いく」という動作の「行為者」には“人間”が格納される、と記述する。この値を属性値として require ファセットを設け格納する。属性値は入力データの情報から動作フレーム同士の関連を発見するときや、入力データに誤りがないかをチェックするときなどに役立つ。人間も「豚が空を飛ぶ」を見た場合、この文章は冗談だ、

と判別することができる。それは、動作「飛ぶ」の行為者の属性となるための要素が、“豚”にはないということを知識として保持しているからである。

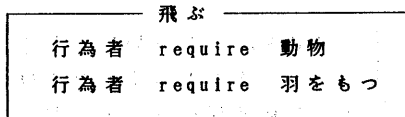


図 動作クラス「飛ぶ」の例

本研究では、知識ベースに上の構成要因のほかに出現回数を取り入れ、値が何回出現したかをデモンを使用することにより記録した。

デモンには、知識ベースに入力データが格納されたときに起動する if_added デモンがある。それらは

- ・ 知識ベースの一般化
- ・ 出現回数のインクリメント
- ・ 動作の深層格の一般化
- ・ 出現回数変化に伴う確からしさの変更
- ・ 推論用フレームの作成

を行う。

4. 1 動作名

知識ベースにおける動作名は任意なので、入力文中の動作名をそのままフレーム名として使用する。そのあと知識ベースに格納するために変換を行う。

4. 2 深層格

特有な属性を深層格と属性値を組み合わせ

- ・ 格支配構造
- ・ require ファセット

を使用することで表現する。

スロットの種類は、インスタンスフレーム設計時に決定しているのでそれに従う。

4. 3 属性値

属性値は、

- ・ 深層格とペアとして使用される。
- ・ 深層格の require ファセットの値となる。
- ・ 格納する値を限定する抽象的な値を示す。

入力文から深層格は獲得することができるが、属性値は獲得した情報の中には存在しないことが普通である。属性値は獲得した情報の中の値を抽象化することで得られる。この抽象化は実体領域中のクラスフレームを参照し、親フレームを検索することで行われる。たとえば、「桃太郎は鬼退治に出発しました」という文章から属性値を得るためには「行為者」、「目的」スロットの値を参照し、そのクラスフレームを検索する。「行為者」「桃太郎」の上位概念は“人間”になっている。よって動作「出発する」の「行為者」スロットの属性値は“人間”となる。「目的」スロットの属性値も同様に“鬼退治”の上位概念になる。

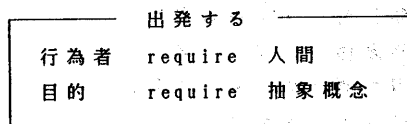


図 動作クラス「出発する」の例

4. 4 値

値は、入力文から得ることができる。入力文中の値は、すべて他の実体領域中のフレームを指しており、作成されるインスタンスフレームに使用されるファセットは、

value となる。値は属性値を決定する。

4. 5 デモン

本研究では、入力データが物語文を変換したものであり、その中にデモンは含まれていない。また入力データの性質からデータに手続きを記述することはできない。知識ベースにデモンを記述するには、入力文を知識ベースに格納するための入力用プログラムに記述しなければならない。入力用プログラムには、デモンは require ファセットを格納するときと一緒に if_added ファセットとして格納するように記述してある。

5. 入力

入力文は、「桃太郎」[3]とする。本研究は自然言語文からの構文解析、意味解析は範囲外としたので、入力には自然言語文を文法、意味解析したあとの深層格構造を使用した。

「桃太郎」は、以前から研究に使用されており他の研究での知識ベースと比較しやすいという利点がある。

5. 1 入力形式

入力データの最小単位は、動作に依存する。多くの動作は「桃太郎」の一文の中に一つである。「おばあさんは川へ洗濯をしに出かけました」という文章は、動作「出かける」を中心として構成される。「おばあさん」は行為者となり、「洗濯」は目的となる。

入力形式は、一文の情報を動作を中心に深層格構造にし、リスト構造に変換したものを使用する。この形式により知識ベースに格納するフレーム構造も「桃太郎」の一文が単位となる。「おじいさんは山へ柴刈りにいった」という文章は、構文・意味解

析され、次のリストに書き換えられる。

```
[[いく, [[行為者, おじいさん_1],  
[終点, 場所_1],  
[目的, 柴刈り_1]]]]
```

本研究では、動作フレーム以外のクラスフレームの構築は考えていない。動作以外を入力データは、実体領域にすでに存在するフレーム名に変換する。「いく」の「行為者」となる「おじいさん」は実体領域で「おじいさん」をあらわす「おじいさん_1」に変換する。

物語「桃太郎」には、深層格を省略してある文章もある。上の文章には、「始点」となる格は、陽には記述されていない。しかし、暗黙として「始点」は「家」となるところである。しかし、陽に記述されていないので入力データには変換されていない。

「桃太郎」において、文章中に暗に記述される情報は、

- ・ 時間、場所に関する深層格
- ・ その他の動作属性として省略可能なもの

がある。「時間」、「場所」に関するものは自然言語文では非常に頻繁に省略される。かえって記述されているとそれは強調を意味する場合もある。しかし「時間」や「場所」の属性は動詞的概念を記述する上で非常に重要なので、本研究では「時間」と「場所」スロットについての情報は補うことにした。「時間」スロットは、動作フレーム同士の前後関係を記述するため、「場所」スロットは、動作フレームと状態フレームの関連を記述するためである。しかし、その他の文章中に陽に記述のないものは、その属性は省略することも動作の使用法だと判断し、補うことはしない。「場所」フレームについては、文章中から判断できるものとできないものがあるので、人間が判断して補った。また、物語「桃太郎」は、一つの時間軸であらわすことができるので、「時間」フレームは知識ベース格納時につ

け加えた。これにより物語「桃太郎」に出現する動作は、一連の前後関係をつけ加えることができるようになった。

5. 2 入力文情報

物語「桃太郎」の全文章数は、205文であり、その文章を入力形式に変更すると271個になる。これは、物語文章中の動詞的概念の数が、一文中に複数個入っていることによる。たとえば、「ふたりはお湯をわかして、赤んぼうにうぶ湯をつかわせました」という文章には、「わかす」と「つかわす」二つの動詞的概念が出現する。この文章を入力形式に変更するときは

「ふたりはお湯をわかす」
「赤んぼうにうぶ湯をつかわせる」

の二つに分割して処理する。

出現する具象物は、「おじいさん」や「桃」など、56個あり、これらは知識ベースに具象物フレームとして格納される。これらは知識ベース上で上下関係でリンクされた階層構造をとる。そのほかにも、具象物や動作などを修飾する形容・様子フレームや、場所フレームなどを物語文から読み取り、知識ベースに格納しておく。

5. 3 知識ベースへの影響

本研究は、物語「桃太郎」からの情報が、どれだけ知識ベースを構築することができるか、というものである。「桃太郎」は、日常生活がコンパクトにまとまっており、自然言語理解のモデルとして適切である。しかし、一つの物語から得られる情報は、知識ベースを構築するための情報量としては遙かに少ない。また、この構築される知識ベース中の知識の構造は、入力文として使用される物語の情報に依存する。

ここで知識ベース中で動詞的概念以外の情報は、物語を読んだ人間がもっている常

識的な知識をすでに格納してあるものとする。

入力文を格納する以前の知識ベースの動作フレームは、ほとんどなにもない状態である。この知識ベースに動作を含む文章を入力することにより、その物語特有の動作構造が構築される。

6. 抽出情報

ここでは、入力文から獲得する情報について記述する。

動作に関する情報で入力データから得られるものには、次の五つがある。

- ・動作名
- ・深層格
- ・格属性
- ・値
- ・前後動作

これらの出現回数も得ることができる。出現回数は、動作フレーム中の任意のスロットの第4引数に格納する。入力文から取り出されたデータ中の深層格は、そのままスロット名として扱われる。そのスロットのrequireフェセットに、格属性が格納される。格属性は、深層格の取りうる値を制限する。たとえば、「おじいさんは山へ柴刈りにいった」という文章から格属性を取り出すと、「終点」スロット値は、“場所_1(=山)”である。この動作「いく」フレーム中の「終点」スロットの格属性は、“場所_1”の上位概念スロットであり、この場合“場所クラスフレーム”となる。よって、このフレームの「終点」スロットのrequireフェセットには、“場所クラスフレーム”が格納される。深層格と格属性の組合せからの動作に必要な属性を取り出す。前後動作は、文中の動作と前文、次文中の動作とをリンクしたものである。前後の文章とのつながりを動作フレームに絞って記録する。

この前後動作リンクは、出現回数として

も、“確からしさ”を表現するときに使用する。

7. 動詞的概念の格納

ここでは、フレーム構造に変換されたデータが知識ベースに格納されるまでを記述する。

「おじいさんは山へ柴刈りにいった」の入力例は、

実_いく_1	
ako	value 実_いくクラス_1
行為者	value 実_おじいさん_1
終点	value 実_場所_1
目的	value 実_柴刈り_1
時間	value 実_時間_1
前動作	value 実_住む_1
後動作	value 実_帰る_1

図 「おじいさんは山へ柴刈りにいった」の動作インスタンスフレーム

とインスタンスフレームに変換され、知識ベースに格納される。このとき、深層格の「行為者」、「終点」、「目的」と、それらの値は、そのまま知識ベースのスロットとデータとして使用される。「前動作」と「後動作」には、前後の文章中の動作を格納する。

さらに「いく」のクラスフレームもインスタンスフレーム入力時に書き換える。動作インスタンスフレームは、格納されたときに、ako-example リンクによりそのクラスフレームと結合するが、結合時に if_added デモンが起動し、動作クラスフレーム「いく」の出現回数をインクリメントする。

8. 出現回数

「あのあとにはこういうことがあったな」といったことを回数も含めて表現するため

に出現回数を取り入れる。確からしさが特定の動作の前後に何が起こったかを確率で表現したことに對し、出現回数はその確からしさの確率を知識ベース中の各フレーム、スロットの登録回数から獲得する、という違いがある。確からしさが静的なものであるのに対し、出現回数は動的に変化するものという違いもある。たとえば、動作「怒る」の次に動作「泣く」があった場合、動作フレーム「怒る」と「泣く」は結合され次にまたその順番で出現したらその結合はさらに強力になる。

出現回数の獲得は、フレームを知識ベースに格納するときに行う。フレームを格納したとき、使用したフレーム、スロット、格属性（require ファセット）の出現回数をインクリメントする。出現回数はフレーム表記における第4引数で表現する。

```
frame(slot, facet, data, 出現回数).
```

入力文中の動作に特定の深層格があった場合、知識ベースはその動作クラスフレームの出現回数とその深層格（スロット）と格属性（require ファセット）の組の出現回数をインクリメントする。「太郎は山へ柴刈りにいく」という文章なら「いく」の「行為者」の属性“人間”の出現回数をインクリメントする（“太郎”の上位概念は、“人間”）。出現回数を含めたフレームの例は、次のとおりである。

実_いくクラスフレーム_1			
ako	value	実_動作_1	5
行為者	require	実_人間_1	4
行為者	require	実_サル_1	1
始点	require	実_場所_1	3
終点	require	実_場所_1	4
前動作	value	実_住む_1	4
前動作	value	実_食べる_1	1
後動作	value	実_帰る_1	3

図 “いく”の動作クラスフレーム例

ako スロットの第4引数の“5”は、動作「いく」が今までに5回あらわれたことを示す。同様に、「いく」の「行為者」が“人間”であったことは、今までに4回あり、“サル”であったことが1回あった。動作「いく」の「行為者」にはだいたい“人間”がくるが、たまには違う動物がくる、と判断できる。“前動作”は、“住む”と“食べる”とあるが、“食べる”より“住む”のほうが高い確率で出現する、と推定することができる。

確からしさを、出現回数の値によって決定する。確からしさは、「この動作のあとに、あの動作が起こる確率は50パーセントぐらいだろう」という、動作間の関係を記述する。文献[1]では、この確率を人間の主観によって決定していたが、本研究では、これを出現回数の数値から確からしさを計算することで行う。

確からしさは、なにが起こったとき、その次に起こることを記述する。そこで出現回数から確率を計算する。この計算は、次の式であらわす。

$$\text{確からしさ} = \frac{\text{require 属性出現回数}}{\text{フレーム出現回数}}$$

出現回数と、確からしさの違いは、前者は回数で記述し、後者はパーセントで記述することである。出現回数は、クラスフレームに記述し、確からしさは、クラスフレームと ako-example リンクするインスタンスフレームに記述する。

9. リンクの変更

ここでは、獲得したデータをクラスフレームに格納する知識として一般化する手続きを記述する。

9. 1 出現回数の変更に伴う一般化

通常、複数の動作が何度もつながってあ

らわれた場合、その複数の動作で一つの働きをするものとしてとらえることがある。本知識ベースにもこの考えを導入した。

ある規定値以上に出現回数が増加すると、デモンが起動し一般化を行う。この一般化は、動作同士のつながりをより強力にし、動作についての情報を増やすことができる。この一般化はすべての属性について行う。

前後動作スロットについては、格納された動作フレームの出現回数が規定値以上に高まると、その動作同士を、独立した動作ではなく、複合して使用するものと認識して扱う。認識された動作同士は、より強力なリンクを使用し結合する。この場合、動作に要求される属性にとらえ、require ファセットを使用して結合する。

実際には、複数の動作が連続して出現し、前後動作スロットをインクリメントするとき、第四引数の値が規定値を超えている場合デモンが起動し、その動作特有の概念として動作フレームを書き換える。たとえば、図 4.6.1 の「住む」フレームが書き換えを行うと、「いく」の「前動作」スロットのファセットを value から require に変更し、「住む」を「前動作」として、必要なフレームとして位置づける。

同様に他の属性も出現回数が増加すると、一般化を行う。たとえば、動作「いく」の「行為者」が“人間”だと複数回入力された場合などである。

9. 1. 1 一般化に要求される出現回数

深層格と属性値のペアが、その動作に要求される属性か、または必要とする属性に変更してもよいかを決定するには、そのペアが入力文に何度出現したか、による。

変更するために必要な出現回数は、ほぼ入力文に使用する物語の長さ、動詞的概念の出現数に依存する。変更するための出現回数の値が十分に大きいと、一般化があまり行われなくなり、逆に非常に小さいと、一般化は始終行われる。変更のための出現回数の規定値は、「桃太郎」の文章の長さ、

動詞的概念の出現数によって決定する。

物語「桃太郎」に出現する動詞的概念の数は、知識ベースに格納する動詞的概念の数と等しく、143個である。その出現数は、277である。この数は、常識的に、動詞的概念を構築する上では少ないと思われる。ここで動詞的概念を、少ない出現数から構築するために、一般化を変更するために要する規定値を小さな値にする。

実際に、物語「桃太郎」に出現する動詞的概念は、平均すると2回前後であり、1回しか出現しない動詞的概念も多い(90程度)。逆に、動作「いう」などのように、50回以上も出現するものもある。これから、次の数値が一般化を決定づける数値となる可能性が高い。

- ・ 動詞的概念の出現回数が2回
- ・ 属性ペアの出現回数が2回

この数値は暫定的なもので、変更することによって調整が可能である。

9. 2 属性値の一般化

ここでは、複数の動作が同じ属性を所有する場合の処理を示す。

深層格と属性値のペアは動作の概念をあらわす。たとえば動作「いく」の「終点」スロットとその属性値の“場所”から、「いく」が場所に関する動作だと推定することができる。

しかし、「いく」だけが「終点」スロットと属性値“場所”を保有するのではない。他の動作フレームにも移動に関するものであれば「終点」と“場所”をもつことができる。

本研究で動作フレームを構築するにあたり最初動作フレームはほとんど空の状態であり、わずかに動作クラスフレームが存在するだけである。この状態から一つずつ入力された動作が格納される。

最初に「いく」が格納され、その深層格として「終点」があり、その属性値が“場

所”であり、そのあとに「帰る」が格納されたとする。「帰る」の深層格と属性値が同じなら、両方の動作フレームにそのペアを記述する必要はなく、その上位概念に記述し、二つの動作をその上位概念の下位に位置づければよい。たくさんの動作が特定の深層格・属性値のペアを使用するならそのペアは非常に一般的なものであると判断できる。「行為者」などは常に動作に含まれているので、非常に一般的で重要だと考えることができる。これはたとえば「計算機を飛ばした」と友達に話せば、その友達はたぶん「だれが飛ばした?」と質問することからも明らかである。

知識ベースでの実現方法は、デモンにより記述する。そのデモンは入力データを知識ベースに格納するとき起動する。

10. デモン

ここでは、各デモンの説明をする。

10. 1 インクリメントデモン

各フレーム、スロットの出現回数をインクリメントするためのデモンである。フレームが知識ベースに入力されたときに起動する。入力された動作の、クラスフレームや、スロットとデータの組の出現回数をインクリメントする。インクリメントするのは、

- ・ ako-example スロット
- ・ 深層格と属性値のペア
- ・ 前動作と後動作スロット

である。ako-example スロットのインクリメントは、そのフレームが何回出現したかを記録するためのものである。

深層格と属性値のペアは、そのフレームにどのような属性が何回出現したかを記録する。前動作と後動作スロットは、動作の前後関係を記録する。

10. 2 一般化デモン

そのフレームに必要な属性を獲得するためのデモン。

インクリメントデモンにより、出現回数がある規定値以上に高まると、起動する。スロットの出現回数がインクリメントされ規定値以上になると、スロットの値はその動作に要求される属性に変更される。具体的には、value ファセットとして登録されていた属性を、require ファセットとして記述する。たとえば通常、動作「いう」の「内容」スロットには、「せりふ」フレームが格納されているが、「いう」が出現するほとんどのときに「内容」スロットは存在する。何度も「いう」フレームが入力されると、「内容」スロットの出現回数も増えていく。デモンは、フレームの出現回数が規定値以上になったときに、フレームの出現回数とスロットの出現回数を比較する。もしその出現回数が非常に近いかまたは同じなら、value ファセットの値を、require ファセットに書き換える。こうして動作に要求される属性を獲得する。

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] 坪田譲治：
新百選日本むかしばなし
新潮社 (1957)
- [4] R.S.Michalski 他編：
電総研人工知能グループ訳
知識獲得入門
共立出版 (1987)
- [5] R.S.Michalski 他編：
電総研人工知能グループ訳
概念と規則の学習
共立出版 (1987)
- [6] 仁木和久、石崎俊：
概念の機能的学習
人工知能学会誌
Vol.3 No.6 pp.695-703 (1988)
- [7] 村木新次郎：
述語素による動詞の分類
情報処理学会研究報告 自然言語処理研究会 85-NL-48-5 (1985)
- [8] 田中卓史：
物語理解のメカニズム
情報処理学会研究報告 自然言語処理研究会 85-NL-50-5 (1985)
- [9] 富浦洋一、吉田将：
動詞の多義性とその記述について
情報処理学会研究報告 自然言語処理研究会 86-NL-55-2 (1986)
- [10] Mark Stefik
AN EXAMINATION OF A
FRAME-STRUCTURED REPRESENTATION
SYSTEM
IJCAI-79 PP.845-852 (1979)
- [11] EXPERIENCE WITH KRL-0 ONE CYCLE
OF A KNOWLEDGE REPRESENTATION
LANGUAGE
DANIEL G. BOBROW, TERRY WINOGRAD,
KRL research group
IJCAI-77 pp.218-222 (1977)