

物語理解におけるエピソード・ネットワークの構築†

野崎 広志** 中澤 俊哉** 重 永 実**

エピソード・ネットワークを形成し、質問に答え、いろいろな観点から物語のあらすじを作成することのできる日本語物語理解システム“JStory”が記述されている。JStoryでは物語理解は物語の筋の軌跡をエピソード・ネットワークとして記憶の中に残す過程とみなしている。その目的のために、意味解析部では格構造と概念依存理論とを使用し、各入力文の意味はイベントとして表されている。文脈解析部ではスクリプト、プラン、ゴール、因果関係のほか標準的なシーンは存在しないが、概念で連鎖関係を保持しているような状況に対応する“概念ユニット”が使用されている。システムはこれらの知識を使って、イベント間のより深い関係を推論し、入力物語の行間にいくつかの補助的な文を埋めることができる。その結果、システムは物語中に陽に述べられたイベントを捜すだけでなく、形成されたエピソード・ネットワークや時空マップやシステムの辞書内の知識を使って、質問に答えることができる。かつエピソード・ネットワーク内の重要なイベントを抽出して、物語のあらすじを出力することもできる。

1. はじめに

物語理解の研究は今までも数多くあり、特に先駆的なものは R. C. Schank ら^{1)~4)}の研究であり、理解における記憶の仕組みや階層構造に関連した様々なレベルの知識の必要性を明らかにした。一方、物語の構造に関する様々な研究がなされ、D. E. Rumelhart ら^{5), 6)}の物語文法は代表的なものである(ただし、物語文法は実際にはインプリメントされていない)。しかし、より深い理解のためには、なお一層の検討が必要だと考えられる。特に、日本語における物語理解システムが実際に計算機上にインプリメントされ、物語全体を通しての物語の構造とその理解の深さを示したものはないように思われる。

一般に物語の理解の程度を示すには、以下のような側面が挙げられる。(1)物語の内容についての質問に答えることができる。(2)物語のあらすじを言うことができる。(3)物語を言い替えることができる。(4)類似した物語を思い浮かべることができる。

本研究では人間の物語を理解する過程を、次のようにみなしている。

- (1) 読み手の持っている実世界に関する知識や常識、物語の次の展開についての予測などを使って、物語中のイベントを意味的に結びつけること
- (2) 物語中に陽に述べられていないことをも推論

により物語表現に組み込む

- (3) (1), (2)より得られる物語の展開をエピソードのネットワークとして記憶の中に残す

そして、先に挙げた物語理解の側面を物語表現(エピソード・ネットワーク)の利用により実現すべきであるという考え方に立って、物語理解システム JStory を構築した。JStory では、R. C. Schank らのスクリプト、プラン、ゴールはもちろんのこと、新たに提案する“概念ユニット”という概念の連鎖関係を汎用的に表す知識を使った推論を利用することにより、より充実したエピソード・ネットワークを作り出す。このエピソード・ネットワークを利用することにより、他の物語理解システムと比較して JStory は、行間を埋めることによる物語中のイベント間のつながりの流暢さ、および物語全体の構造の明確な表現、質問に対する応答のわかりやすさ、大事なイベントの抽出の適切さ、他の物語との類似点の抽出のしやすさ等の特徴を持っている。

本稿では紙面の都合上、主として今回新たに取り入れた概念ユニットという知識の紹介とエピソード・ネットワークの構築について述べる。

2. システムの概要

本システムは構文解析部、意味解析部、物語文脈解析部、質問文理解部、文生成部、あらすじ生成部からなる。構文解析部では、ローマ字で分かち書きされた入力文から句構造を抽出し、構文木を作る。意味解析部では、構文木を基にして、係り受け等の処理を行い、入力文の意味表現を作る。文の意味表現は格構造、テンス、アスペクト等の情報を取り出したもの

† Construction of Episode Network for Story Understanding by HIROSHI NOZAKI, TOSHIYA NAKAZAWA and MINORU SHIGENAGA (Department of Computer Science, Faculty of Engineering, Yamaguchi University).

** 山梨大学工学部計算機科学科

と、CD (Conceptual Dependency Theory) 表現⁷⁾からなる。物語文脈解析部では、意味解析部でできた意味表現から適切に様々な文脈知識に起動をかけ、イベント間のつながり (エピソード・ネットワーク) を作る。質問文理解部では、質問文の類別や何を答えれば良いのかを判断し、質問文に対する答えを見つけるために、エピソード・ネットワークまたは辞書知識の中を探索する。文生成部では、質問文に対する応答文あるいは、あらすじ文を生成する。あらすじ生成部では、キー・センテンスの抽出と、あらすじの図式を使い、あらすじ文を生成する。

3. 構文解析

構文解析部で使用している日本語文法は、基本的には時枝文法における詞と辞の概念を取り入れており、本論文では詞に対応する語を概念語、辞に対応する語を機能語と呼び、両者の結合したものを句と呼び、解析の単位としている。

ローマ字によって分かち書きされた入力文を、構文状態遷移ネットワーク⁸⁾を初期状態から終了状態までバック・トラックしながらたどることにより、句単位に解析し、文の句表現を作成し、各句間の修飾関係から構文木を作る。構文解析部と意味解析部は完全に分離されているわけではなく、構文解析中にも意味情報を使って後続の単語予測を行う等して不必要な探索をしないように努め、できるだけ早く (可能な構文木をすべて作ってから意味解析に進むのではなく) 意味解析に持ち込み、係り受けが正しくなければ、その情報を次の構文木作成に生かすことができるようにして、構文解析の高速化を計っている。

4. 意味解析

構文解析で得られた句表現を構成するすべての句について句の性質を解析し、構文木に表された係り受け関係を調べて、意味的整合性のない構文木を排除し、文の意味表現を作る。意味表現は基本的には格構造⁹⁾、アスペクト等の法情報¹⁰⁾を取り出したものと、CD 表現からなる。文の意味表現に格構造表現と CD 表現の二つを使用しているのは、CD 表現を格構造表現より深層の表現とみなし、①その文に暗に示されている情報を示すことができる。②表層上は異なる言語表現であっても、意味的に同じ、または似た文は、同じ、または似た表現になる。③CD 表現を使うことにより、例えば ptrans (物理的な移動を表すプリミテ

ィブ) なら ptrans 用の推論規則を用意しておけば、「行く」と「来る」の動詞辞書に重複して推論規則を記述しておく必要がない、等の利点があるからである。

また、プリミティブには Schank らのもののほかに、JStory 独自のものもある。例えば、「何かを作り出す、または表現する」ことを示す product や、「ある出来事が起こるまで待つ」ことを示す wait などがある。

述語辞書には CD 表現のプリミティブ名と概念格、そして、その概念格に対応する格構造の格の名前が記述してある。格構造の要素を利用し、CD 表現を完成する。もしも埋め込み文がある場合には、その埋め込み文が修飾する名詞の属性に埋め込み文の意味表現を記入し、かつ、それをイベントとして陽に出している。

CD 表現を構成し終わると構文情報、格構造、CD 表現、述語辞書項目番号を一つのフレームにする。このフレームをイベントと呼ぶ。

5. 文脈解析

文脈解析部では、意味解析部で意味表現にまで落とした文と、前文までの意味表現との間になんらかの関係を求め、イベントとイベントとをエピソード・リンク (表 1 参照) でつなぎ、物語 (文章) 全体としての構造を抽出する。物語中のイベントとイベントとを意味的に結んでできるネットワークは、物語の進行の軌跡を残しており、これをエピソード・ネットワーク (以下、EPN と略記) と呼ぶ。文脈解析での目標は、大筋

表 1 エピソードリンクの分類
Table 1 Categories of episode links.

リンク名	記号	説明
planfor	p	あるゴールに対して使われるプランを示す
subgoal	s	あるプランを達成させるためのサブゴールを示す
instan	in	プランから行為イベントに向かい、そのプランの具体化したものを示す
motive	m	ゴールとそのゴールに動機を与えたイベントをつなぐ
reason	R	精神的事象または状態は行為の理由となり得る
initiate	it	ある客観的事象または状態は偶然に精神的事象を引き起こし得る
result	r	ある行為はある状態を引き起こす
enable	e	ある状態はある行為を可能にする
disable	da	ある状態はある行為を不可能にする
achieve	a	あるイベントがゴールを達成させる
fail	f	あるイベントがゴールを失敗させる
sum	sm	いくつかの複数の事象から合意される

表 2 知識の数
Table 2 The number of prepared knowledge.

知識名	数	知識名	数
スクリプト	11	因果ルール	22
プラン	19	概念ユニット	5
ゴール	16		

としてこの EPN を作り上げることである。このために、概念ユニット、スクリプト、プラン、ゴール、因果規則、物語の時空マップ、物語構造知識等の知識を利用し(表 2 に現在準備している各知識の数を示す)、

(1) 物語中のイベントとイベントの関係を探る

(2) 物語テキストのなかで陽に述べられていないイベントを推論し、EPN の中に組み込む

といったことを行っている。

スクリプト、プラン、ゴール、因果規則は Schank のものを幾分拡張しているが、おおむね同様の処理を行っている。ここでは文脈解析知識の一つとしての概念ユニットについて述べる。

5.1 概念ユニット (conceptual unit)

R. C. Schank の提唱したスクリプトは、記憶ないしは知識の表現に対して有用な構造である。しかし物語の流れにはスクリプト的概念には似ているが、スクリプトの処理能力を越えた、明らかに一つの巨大な因果連鎖により結びついている状況が存在していることが多い。例えば、「恩返し」とか「脅迫して何かをさせる」といったものであり、これらは関係する動作主間での価値観や、そのときの状況などにより様々な手法で行われ、スクリプトのような標準的シーンというものが存在しない。しかし、動作主間での関係の概念が連鎖関係を保持しているのが特徴である。この考えは、W. C. Lehnert¹³⁾ の提唱した Plot Units に近いものである。しかし、彼のように Primitive Plot Units まで細かく分けるのではなく、標準的な概念関係を組織化し、知識として準備しておくことにより、記述および処理面での簡約化を図っている。

この知識には以下のような利点がある。

- (1) 物語の中心概念に着目することができる。
- (2) 物語内の文と文の結合性の強弱まで把握することができる。
- (3) 要約時に使用することができる。
- (4) 長期記憶として残るのはこの部分である。
- (5) 物語間の類似性を捉えることができる。

この知識を概念ユニットと呼ぶことにする。この知

識は他の文脈解析の知識の処理とも結びついていて、柔軟性のある知識となっている。概念ユニットとしては動作主間でポジティブ、ネガティブな行為が行われると、それにより以後どのような行為が行われるのか、それによりどのような感情の想起が行われるのかに着目したもの (pos-effect, neg-effect)、出発-行動-帰路といった大きな一連の動作を表現したもの (nanika-sini-iku)、脅迫することにより相手に何かをさせるといったことを表現したもの (threaten) などが現在準備されている。

各ユニットの適用方法は、スクリプトの適用とほとんど同様に行える。文をこの処理に渡すと、文を構成する語、この文以前に入力された文との関係から、概念ユニットが適用できるか、適用するとしたらどのユニットかを決め、目的となるユニットを活性化させる。そして、文の表現している概念が活性化されたユニット内のどの概念と一致しているかを調べる。これは単なるマッチングだけで済むときもあれば、ユニット内に登録されている特別な条件審査をするときもある。マッチした概念よりも以前にあるだろうと予測されている概念がユニット内に登録されていれば、それを陽に出して文間を埋める。活性化したユニットの最後の概念がマッチするまで、そのユニットを活性化しておき、次の文以降活性中のユニットであるということでユニット内の概念と文の関係を順次調べて行く。

具体例として、広義の「恩返し」を表現した概念ユニット pos-effect を説明する。pos-effect は「人 1 (*psn 1) が相手 (*psn 2) に好印象を与えたら、*psn 2 は *psn 1 にとってポジティブ (positive; +) なことをする」という概念関係を表現している。すなわち、図 1 に示すようにこの場合の各概念は、大きく分けると次の①～③の三つのグループからなる。

① initiate

1) 人 1 (*psn 1) の何か (*koto) の様子が普通よりも良い、あるいは上手である (manner > *normal)。

② effect-on-person

これはさらに図 1 中の番号 (概念番号) の 2)~4) の三つの概念に分かれる。

2) ①を見るか、聞くかしていた相手 (*psn 2) が喜ぶ (ment-st[mental state] の value が 10 になる)。

3) 次の 4) をしようとその人は思う (*psn 2 が *mobj[mental object] を mbuild する)。

4) 最初は直接記述されていないが、マッチングの条件 (& match に記述しておく) を満たす CD 表現

```

($pos-effect
  (type affect) ;ユニットのタイプ
  (trig t)      ;ユニットの起動条件
                ; (この場合はここには記述されていない)
  (roles (value ($psn1 hit1) ($psn2 #aitel)));登場人物
  (props (perse));道具 (この場合は何もなし)
  (concept      ;概念群
    (initiate
      (1 (state (obj #koto (#part #psn1)) (manner > #normal)))
      (effect-to-person
        (2 (change (obj #psn2) (from (ment-st #?))
                   (to (ment-st ($val 10))))))
        (3 (mbuild (actor #psn2) (obj (to #mobj))
                   (from (mloc (itm #psn2))) (to (mloc (im #psn2))))))
        (4 #mobj)
      )
      (effect-to-another-person
        (5 (change (obj #psn1) (from (ment-st #<))
                   (to (ment-st (> #<))))))
      )
    )
  (link (initiate 1 2) (motiv 2 3) ;各concept間のエピソード・リンク
        (reason 3 4) (initiate 4 5))
  (&match (4 (pos-effect?)));各conceptのマッチング条件
)

```

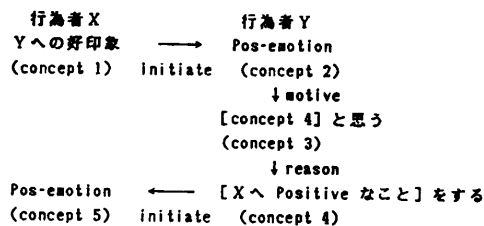


図 1 概念ユニット pos-effect の内容

Fig. 1 Content of "pos-effect" conceptual unit.

が *mobj のところに入る。ここでマッチング条件とは、*psn 2 のすることが *psn 1 にとってポジティブなことではなければならない。ここで *psn 1 にとってポジティブなことというのは、*psn 1 が持っているゴールに *psn 2 が貢献するようなことであり、このユニットはこの部分でプラン・ゴールによる処理を使用している。つまり、pos-effect の概念 4) は特定の文で表したのではなく、ポジティブなことをしたという概念を内在した文であれば、どのような文でも良い。

③ effect-on-another-person

5) 人 1 は喜ぶ。 ((to(ment-st(> * <))); [* < は現状])

このユニットの起動法には以下の 2 通りがある。

(1) 「人 1 が相手に好印象を与える。」: これは人 1 の行動などにより、相手の気持ち (印象) が通常よりも良くなったということであり、例えば、人 1 の何かが「上手」だ、といったことにより起動される。

(2) 「人 1 が相手の欲求 (ゴール) を満たした。」:

これは後で述べる EPN の achieve リンクを調べて行う。

ユニット内の概念間のリンクはそのユニットのスコット link に記述してあり、ユニットの各概念が実体化 (インスタンス) されたとき、EPN の中でそのリンクが張られる。

例えば、「こぶとりじいさん」では「おじいさんの踊りが上手だったので、鬼がおじいさんのこぶを取った」というところで pos-effect が使われている。「上手だ」という語の関連項目に pos-effect と記述されているので、pos-effect が候補として上がり、pos-effect のシーンと入力文を CD 表現にしたものとのマッチングをとると、「おじいさんの踊りは上手だった」という入力は、概念 1) とマッチすることがわかる。そこで、pos-effect が活性化される。次に、「鬼はおじいさんの醜いこぶをとった」が入力されると、pos-effect が活性化中なので、その中のシーンとのマッチングをとると、シーン 2) ~5) のいずれともマッチしない (概念 4) は CD 表現が記述されていないので当然マッチしない)。しかし、マッチング条件 (& match) に記述されているポジティブな行為かどうかを調べる関数 (pos-effect?) を実行すると、「おじいさんはこぶを取りたい」というおじいさんのゴールに対して鬼が貢献していることがプラン・ゴールの処理を利用することによりわかり、鬼がおじいさんにポジティブなことをしたいということになる。したがって、入力文は概念 4) とマッチしたことになり、*mobj はマッチした CD 表現に置き換えられる。そして、*mobj を使用している概念 3) の *mobj (「鬼はおじいさんのこぶを取りたい」) を今マッチした CD 表現に置き換えたものと (概念 3) は、結局「鬼はおじいさんのこぶを取ろうと思った」ということになり、概念 2) (「鬼は喜ぶ」) が補間データとしてデータ・ベースに登録される。概念 5) は、結局最後までどの入力ともマッチせず残ったままになるが、物語の終了をシステムに知らせると、活性化したままのユニットでまだデータ・ベースに登録されていない概念があれば、それらを登録してからシステムは終了するので、「おじいさんは喜んだ」という補間データがデータ・ベースに登録される。

つまり、「こぶとりじいさん」では pos-effect によって以下の概念関係ができあがったことになる。

「おじいさんの踊りは上手だったので、鬼は喜び、鬼

はおじいさんの醜いこぶを取ろうと思い、こぶを取った。するとおじいさんは喜んだ。」

この概念ユニットは物語（昔話）の骨格や、それが内在している教訓を抽出するのに有効な知識であると考えている。さらに、適切なあらすじを生成する手がかりになる。

5.2 物語の時空マップ

物語の時空マップとは、物語中で起こった出来事の場所関係や時間関係を表したもので、EPN 中の各イベントに対するポイントを含んでいる。マップ情報は例えば「雨が降る」事象が起これば、雨を避けるゴールは建物の中にはいない登場人物に対してのみ適用するというように、推論規則を適用する人物を限定し

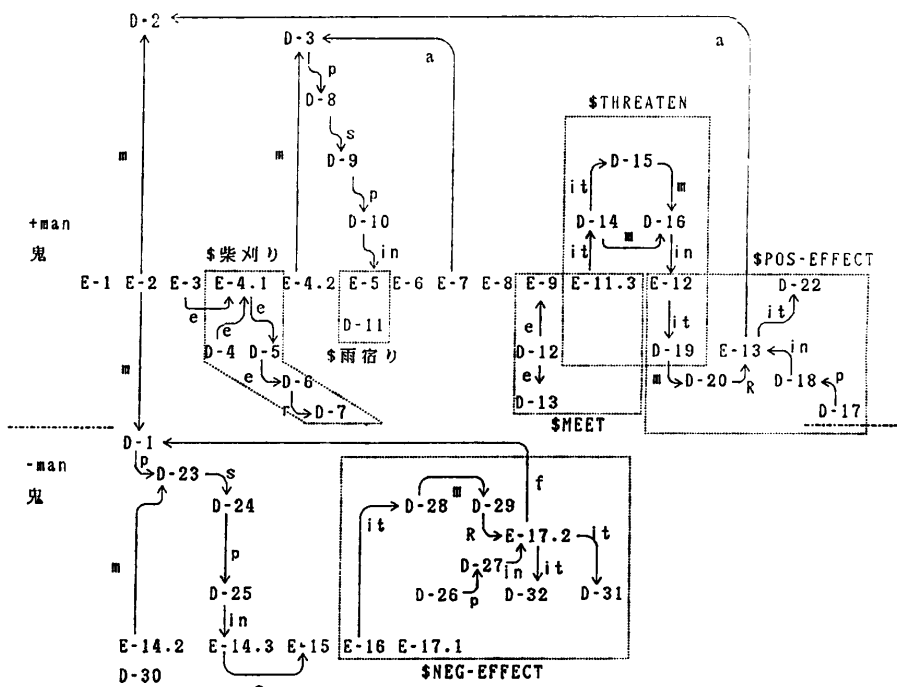
たり、また時間や場所に関する質問の処理に利用している。

5.3 物語の意味表現

物語をどのように表現したら良いかという問題は、物語が記憶の中でどのように表現されているかという問題と同じく難しい。人間の記憶の中では単にイベントをリンクでつないだだけとは思えない。しかし、現在のところ、文脈解析によりイベントをリンクでつないでできあがった EPN は物語の展開の骨格をなしており、これを物語の内部表現としている。

5.4 物語構造についての知識

JStory では物語を物語=設定+展開という構造で見ようとする。物語文法^{5),6)}のようにさらに下位レベル



「こぶとりじいさん」テキスト
 昔々あるところに悪いおじいさんと悪いおじいさんがいました(E-1)。悪いおじいさんにも悪いおじいさんにも顔に醜いこぶがありました(E-2)。ある日、悪いおじいさんは山へ仕事に行きました(E-3)。芝刈りをしていると、雨が降って来ました(E-4.1 E-4.2)。おじいさんは急いで大きな木の陰に入り込みました(E-5)。夜になって、雨がやみました(E-6 E-7)。その時、おじいさんの傘が壊れました(E-8)。おじいさんは傘を見つけてしまいました(E-9)。もしもおじいさんが傘を見つけたら、鬼はおじいさんを食べるとおじいさんに言いました(E-11.1 E-11.2 E-11.3)。しかし、おじいさんの顔は太く上ずったので鬼はおじいさんの顔からこぶを取ったことを悪いおじいさんは、次の日、山に行きました(E-14.1 E-14.2 E-14.3)。悪いおじいさんは鬼と一緒に顔を見ました(E-15)。おじいさんの顔は太く下ずりました(E-16)。鬼は悪いおじいさんの顔からこぶを取った。こぶを悪いおじいさんの顔につけてしまいました(E-17.1 E-17.2)。

- D-1 -manはこぶをとりたい
 - D-2 +manはこぶをとりたい
 - D-3 +manは雨を避けたい
 - D-4 +manが仕事場へ行く
 - D-5 +manが傘を囲む
 - D-6 +manが傘を入物に
 - D-7 +manは傘を持っている
 - D-8 +manの雨宿りプラン
 - D-9 +manが木に近づく
 - D-10 +manは自力で移動する
 - D-11 +manは暗れを待つ
 - D-12 +manと鬼は近くにいる
 - D-13 +manは鬼を見る
 - D-14 +manが恐怖を感じる
 - D-15 +manは鬼に食べられたくない
 - D-16 +manは驚くことにする
 - D-17 鬼はこぶをとりたい
 - D-18 (鬼) とるプラン
 - D-19 鬼は喜ぶ
 - D-20 鬼はこぶをとることにする
 - D-22 +manは喜ぶ
 - D-23 -manは鬼に自分のゴールを代行させたい
 - D-24 -manは山へいきたい
 - D-25 -manは自力で移動する
 - D-26 鬼は何かつけたい
 - D-27 (鬼) つけるプラン
 - D-28 鬼は怒る
 - D-29 鬼はこぶをつけることにする
 - D-30 -manは山からどこかへ行く
 - D-31 -manは怒る
 - D-32 -manは泣く
- 注: +man 良いおじいさん, -man 悪いおじいさん

図 2 「こぶとりじいさん」のエピソード・ネットワーク

Fig. 2 An episode network constructed for a fairy tale "Kobutori jiisan (Old men with a wen)."

にまでは展開しようとはしていない。なぜなら Black ら¹²⁾の指摘のように物語を一文一文処理している理解過程時に、それが物語文法規則のどのカテゴリーになるかを決定するのは難しいと思えるからである。JStory ではこの構造は単に物語全体を設定部と展開部に分ける以外には使っていない。これは、あらずじ生成の処理時に、物語の定型、つまり物語=設定+展開を用いてあらずじを生成しようとするためである。

処理は次のようにして設定部と展開部とに分ける。

- (1) 物語は展開部が始まるまでは設定部である。
- (2) 物語の最初に設定部がなく、最初から登場人物が何か行動をするときには、その行動の前提を推論し、それを設定部とする。
- (3) 展開の始まりの識別は登場人物の行動の記述や“ある日”、“やがて”等のキー・ワードに頼る。
- (4) 状態性の述語文や、登場人物の行動を記述した文以外の文が続いているならば、まだ設定部が続いているとみなす。

5.5 物語のエピソード・ネットワークの例

図2に物語テキストに対して JStory の作った EPN を示す。EPN 図中の、文番号に E がついている文は物語テキスト中に陽に現れているイベント（入力文）番号を示す。また、文番号に D がついている文は物語テキスト中には陽に現れていないが、システム内部で推論されて組み込まれたイベントを示す。E-*, * と記述されているイベントは、それが埋め込み文を持っている、あるいは埋め込み文であるということを示している。しかし、「て」接続型の埋め込み文に関しては、「て」接続は前文と後文が時間的に同時に起こったことなのか、別々に起こったことなのかのいずれかを表すが、同時であれば E-n, E-n+1 とし、別々であれば E-n. m, E-n. m+1 (m=1, 2, ...) としている。同時であるかどうかの判断は、その判断法が難しいために、ユーザに問い合わせることにより処理している。その他の記号については表1を参照されたい。

点線で囲まれている部分は、そのテキスト理解において起動された概念ユニット（概念ユニット名の前には \$ 記号がついている）のうちのいくつかで覆われたノードを示す。一点鎖線での区切りは各命題により、どの登場人物が動作主になっているかを示している。

EPN 内には何もリンクを持たないイベントがあるが、EPN で表現する範囲ではつながりがないということであり、実際には、時間的、空間的、あるいは語

表3 物語と使用概念ユニット
Table 3 Conceptual units used in stories.

物語名	使用概念ユニット
こぶとりじいさん	pos-effect neg-effect threaten nanika-sini-iku
舌切り雀	pos-effect neg-effect nanika-sini-iku
大きなかぶ	coop
笠地蔵	pos-effect nanika-sini-iku

と語の関係（概念関係）などで関連があり意味を持っている。また、埋め込み文であるために、他のイベント内で使用されていて EPN 内には存在しないものもある。

5.6 物語間の類似性について

人間が物語を読み進めているときに、今読んでいる物語と似た筋の展開を持った過去に読んだ物語を思い出すことがある。そしてその思い出した類似の物語の筋の展開から物語の次の展開を予想したりする。もしも複数のテキスト間で類似した筋の展開があるならば、それらのテキストの物語表現の中にもいろいろな形で反映があつて然るべきであろう。事実、表3に JStory で処理した物語名とその物語で使用された概念ユニット名を示すが、「こぶとりじいさん」「舌切り雀」「笠地蔵」が勧善の点で、特に前二者はさらに懲悪の点でもよく似ており、人や動物が協力して (coop 概念ユニット) 大きなかぶを抜く「大きなかぶ」は他の三者と全く異なる展開であることがわかる。

6. エピソード・ネットワーク (EPN) の利用

EPN を用いて質問応答やあらずじ生成を行うことができるが、紙面の都合上、質問応答について簡単に説明する。なおあらずじ生成などの詳細は文献13)を参照していただきたい。

6.1 質問応答

「こぶとりじいさん」の場合、「鬼はなぜ良いおじいさんの顔からこぶを取りましたか」という質問を JStory に入力すると、図3-①のような出力が得られる。これは、「鬼はおじいさんの顔からこぶを取った」という図2の中の E-13 が前提となっていることが、質問文の分類^{11,14)}、および文の構成要素などからわかる。したがって、E-13 から後向きのリンクをたどって答えとなるノードを集めていく。さらに、E-13 は pos-effect の概念となっているので、答えとして E-12, D-19, D-20 が選ばれる。そして、各ノード間の関係はその間のリンクの種類により判別できるの

- ①
 *** Question is ... ***
 木ノ陰ニヨイオジイシノカカラコトヲトリアシカ
 *** Answer is ... ***
 ヨイオジイシノトトリノシヨウスダツタナ
 木ノ陰ニヨイ
 ヲレテヨイオジイシノミニカイコトヲトッテアケヨクトモッタカラテス
- ②
 *** Question is ... ***
 オジイシノカキノカニハイリマシカ
 *** Answer is ... ***
 ヨイオジイシノアメニヌレタカクツタナ
 アマトトリヨシヨクトモッタカラテス

図 3 質問応答の例

Fig. 3 Examples of question and answering.

で、それに従ってこれらを組み合わせ、連続文として応答文を作成して行く。

また、「なぜおじいさんは木の陰に入ったか」という質問に対しては、この前提は、「おじいさんは木の陰に入った」(E-5)であるので、E-5 からリンクをたどり、E-4, D-3, D-8, D-9, D-10 というプラン・ゴールの一連の連鎖が選ばれ、それから連続文を生成すると図 3-② のようになる。

7. おわりに

物語理解を単にテキスト文の意味ネットワーク表現だけでなく、十分その行間を埋めて深層の意味をも表現した EPN を構築できるようにした。また、文脈解析知識として新たに概念ユニットを取り入れた。そして四つの物語を処理し、できた EPN を使って質問応答や新しい観点からのあらすじ生成等が可能になっている。

物語理解過程の解明は容易ではなく、かつ、本システムの程度では理解したとは言い難いかもしれないが、EPN の形成は、理解の程度を幾分深めた表現となっているのではないだろうか。

今後、(1)以前に行われた推論結果を修正できる機構、(2)予測破壊の処理とそれに伴う動的予測の機構の付加などが必要であり、これらの点に対する処理をも考えながら、より多くの物語を処理し、本手法の汎用性と理解の深さを高めて行きたい。

参考文献

- 1) Dyer, M.: *In-Depth Understanding*, MIT Press (1983).
- 2) Schank, R.C.: *Scripts, Plans, Goals and*

Understanding, Lawrence Erlbaum Associates (1977).

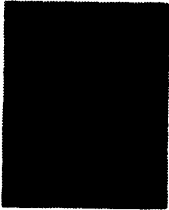
- 3) Schank, R.C.: *Reminding and Memory Organization: An Introduction to MOPs*, Research Report #170, Yale University Department of Computer Science (1979).
- 4) Schank, R.C. and Riesbeck, C.K.: *Inside Computer Understanding*, Lawrence Erlbaum Associates (1981).
- 5) Mandler, J.M. and Johnson, N.S.: *Remembrance of Things Parsed: Story Structure and Recall*, *Cognitive Psychology*, Vol. 9, pp. 111-151 (1977).
- 6) Rumelhart, D.E.: *Notes on a Schema for Stories*, in Bobrow, D.G. and Collins, A. (eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press (1975).
- 7) Schank, R.C.: *Conceptual Information Processing*, North Holland (1975).
- 8) 関口芳廣, 来喜 宏, 重永 実: 日本語文章の音声認識システムにおける構文情報について, 電子通信学会論文誌, Vol. J65-D, No. 6, pp. 782-789 (1982).
- 9) 田中春美: 格文法の原理, 三省堂 (1975).
- 10) 内藤昭三, 島津 明, 野村浩郷: 日本語文における法情報の解析, 情報処理学会自然言語処理研究会資料, 26-2 (1981).
- 11) Lehnert, W.G.: *Plot Units and Narrative Summarization*, *Cognitive Science*, Vol. 4, pp. 293-331 (1981).
- 12) Black, J.B. and Wilensky, R.: *An Evaluation of Story Grammars*, *Cognitive Science*, Vol. 3, pp. 213-230 (1979).
- 13) 野崎広志, 重永 実: エピソード・ネットワークを用いた物語理解, 情報処理学会知識工学と人工知能研究会資料, 51-4 (1987).
- 14) Lehnert, W.G.: *The Process of Question Answering*, Lawrence Erlbaum Associates (1978).

(昭和 63 年 10 月 17 日受付)
 (平成 元年 6 月 13 日採録)




野崎 広志 (正会員)

昭和 37 年生。昭和 60 年山梨大学工学部計算機科学科卒業。昭和 62 年同大学大学院修士課程修了。同年日本アイ・ビー・エム(株)東京基礎研究所入社。現在、同研究所において日本語テキスト入力ユーザインタフェースの研究に従事。

中澤 俊哉 (正会員)

昭和40年生。昭和63年山梨大学工学部計算機科学科卒業。同年同大学大学院工学研究科修士課程入学。物語理解の研究に従事。

重永 実 (正会員)

大正13年生。昭和22年京都大学工学部電気工学科卒業。金沢工業専門学校、金沢大学工学部を経て、昭和32年より山梨大学工学部電気工学科、46年より計算機科学科勤務。

工学博士。音声認識、物語理解、自動プログラミング、障害児用学習システムの作成等に従事。電子情報通信学会、日本音響学会、人工知能学会、日本認知科学会等の会員。