

英語文章の大意生成

内海功朗、重永 実 (山梨大学、工学部)

1. はじめに

論理的な文章は通常、筆者の主張(文章の主題)を導くために順序づけられた文から構成されている。言い換えれば、文章は読者が連続した二文間の関係を一般的事実の知識を用いて捉えながら、文章の論理的展開を把握できるように文が並べてある。よって、読者と同じように一般的事実の知識や推論規則を自然言語処理のシステムに入れておけば、文章の要約を取り出すことも可能である。

物語文を対象にした要約抽出にはRumelhart [1] やSchankたち[2]の研究がある。前者は物語文法を使って解析した結果に要約規則を作用させて、物語文の要約を作るもので、後者はエピソードの鋳型であるスクリプトを使って解析し、それをもとにして要約を生成するものである。説明文を対象としたものにHobbs [3]の研究がある。この研究は説明文の各文を論理式に直し、それから並列や対比などの coherence 関係を基にして、要約表現を生成するものである。しかしながら医学書を処理する対象にしているため、論理的な展開に対しての対処はなされていない。

筆者らは論理的な文章の大意を生成するために、文章の構造を階層性のある表現にするために木構造を使用して表わした。文章を解析して得られた木構造の根から枝をたどり、到達した文をトピックセンテンスとして取り出し、それらを組み合わせるで大意とした。文章の木構造は命題の構成要素の類似や相連、推論規則、スクリプトを使って求めた文の接続関係をもとにして生成する。

2. システムの構成

英語文章の大意を求めるため、システム全体を構文解析部、意味解析部、文脈解析部、大意生成部の4部分に分けて処理する。処理の流れを図1に示す。

構文解析部では文に対してATN文法を使ってその構文木を求める。構文木はバックトラックを使って複数個求める。

意味解析部では構文解析部で得られた構文木を基にして節ごとに意味表現(格構造表現と簡略化したCD表現)を生成する。この二つの表現を合わせてイベントとする。意味表現は主動詞の格構造を基準にして求める。格構造表現は文の生成に、CD表現は二文間の接続関係を求めるのに使われる。CD表現は次のような形で表わす。

(pred +OR- (case1 arg1)(case2 arg2) ...)

predはプリミティブアクトまたは述語動詞、+OR-は

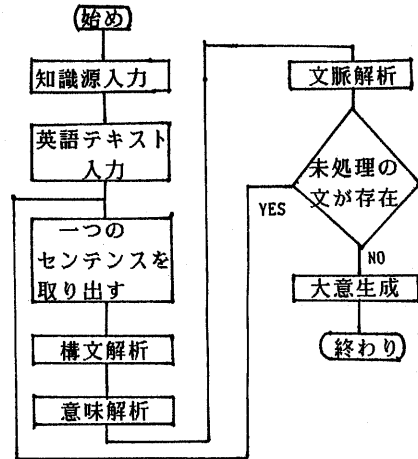


図1 処理の流れ

肯定または否定、caseは概念格、arg は引数を指す。

文脈解析部では意味解析部で処理した文と前文との間で接続関係を求める。二文間の接続関係は文の構成要素の概念関係や推論規則、スクリプトを使って求める。この接続関係の重みをもとにして拡張二分木の形で文脈を表現する。

文章中の全ての文について解析を終えると、作業を大意生成部へ移す。大意生成部では文脈解析部で得られた拡張二分木からトピックセンテンスとそれに関係のある文を取り出し、それらを組み合わせるでテキストの大意を生成する。

3. 文脈解析

3.1 文の接続関係

文の接続関係(文と文の論理的な関係)として市川[4]が提唱した類型を使う。この類型は二文間を結ぶことができる日本語の接続詞の意味から8種類に分類されている。各分類の内容とそれにあてはまる英語の接続詞を表2に示す。また、8種類の類型はつぎの3グループに分けられる。

論理的結合関係(順接型・逆接型)

二つの事柄を論理的に結びつけて述べる関係。

多角的連続関係(添加型・対比型・転換型)

二つ(以上)の事柄を別々に述べる関係。

拡充的合成関係(同列型・補足型・連鎖型)

一つの事柄に関して拡充して述べる関係。

各接続関係には前文、後文のどちらがより重要で

表2 文の接続関係の基本的類型

- 順接型 前文の内容を条件とするその帰結を後文に述べる型。
 順当[JS:JT] SO, THEREFORE (==>)
 逆接型 前文の内容に反する内容を後文に述べる型。
 反対[GS:HT] BUT, HOWEVER (==>)
 添加型 前文の内容に付け加わる内容を後文に述べる型。
 累加[TK:RK] AND (->)
 並列[TK:HR] OR, WHILE (<->)
 対比型 前文の内容に対して対比的な内容を後文に述べる型。
 比較[TH:HK] RATHER (->)
 対立[TH:TR] ON THE OTHER HAND (<->)
 転換型 前文の内容から転じて、別個の内容を後文に述べる型。
 転移[TN:TI] BY THE WAY (->)
 推移[TN:SI] WELL (->)
 同列型 前文の内容と同等とみなされる内容を後文に重ね述べる型。
 要約[DR:YY] NAMELY (->)
 詳述[DR:SJ] (<->)
 換言[DR:KG] IN OTHER WORDS (<->)
 例示[DR:RG] FOR EXAMPL (<->)
 反復[DR:HP] ……同じ文の繰り返し (->)
 換置[DR:KC] ……肯定と否定の置き換え (->)
 補足型 前文の内容を補足する内容を後文に述べる型。
 根拠[HS:KK] BECAUSE, FOR (<->)
 制約[HS:SY] PROVIDED THAT … (<->)
 連鎖型 前文の内容に直接結びつく内容を後文に述べる型。
 連係[RS:RK] ……解説付加、見解付加(?)
 応対[RS:OT] ……問答形式(=>)
 ※注 ?は後続の文によって決まる重みを指す。

あるかの標準的な指標を付け加えておく。表2において、(->、=>)なら後文の方が、(<->)なら前文の方がより重要であることを示す。->と=>はどちらも右向きであるが上のグループ分けの違いから区別している。また(<->、<=>)はどちらが重要か決められないことを示す。例えば、二文間の接続関係が補足型の根拠であるならば、前文の方がより重要であり、前文に重みがあることになる。

3. 2 接続関係を求める方法

文の接続関係は次の三つの方法を用いて求める。

- ① 命題の構成要素の概念関係から
- ② 動詞、名詞、形容詞が持つ推論規則から
- ③ スクリプトから

①の方法は前文、後文中の構成要素間に類似、相違などの関係がある場合、それらを基にして文の接続関係を求めるものである。この考えはHobbsが提唱した「コヒーレンス coherence」[5]に基づいている。コヒーレンスは文間の筋の通った意味的なつながりを指すもので、文の接続関係とそれほどかわりはない。Hobbsはこのコヒーレンスを命題概念に用いて規定し、人間がコヒーレンスを理解する過程のモデル化を行なった。例えば、一般化という関係は、前文から命題P0、後文から命題P1を推論することができ、P0とP1は同じ述語を持ち、P0のいくつか

の引数がP1の対応する引数の部分集合か要素で、他の対応する引数が同じ時と規定した。

この命題概念による規定を表2で示した文の接続関係に応用した。このシステムでは、まず、CD表現で表わされた二つの命題において、相対応する格の引数(名詞)の関係を名詞辞書の各スロットを使って求める。名詞間の関係は表3に示すように分類され、また、引数がイベントの場合をも考慮して、イベントとイベントの関係、イベントと名詞の関係は表4のように分類される。命題間の名詞の関係は表3の上から順にその内容が二つの名詞にあてはまるかどうかで判断する。次に、名詞間の関係にプリミティブアクトと肯定・否定を加えて、イベントの接続関係を判断する。表5に文の接続関係とプリミティブアクト、肯定・否定、名詞間の関係との対応を示す。

例えば、次のような連続した文があるとする。

S1 JOHN IS NOT A STUDENT.

S2 HE IS A TEACHER.

S1、S2をCD表現で表わしたものをそれぞれP1、P2とする。

P1 (EQ - (OBJ JOHN-1) (TO STUDENT-1))

P2 (EQ + (OBJ HE1) (TO TEACHER-1))

この二つの命題の構成要素を比較すると、表3からHEはJOHNを指示することができることよりEQLの関係にあり、STUDENTとTEACHERの上位概念は等しく職業であることよりEKOの関係にある。プリミティブアクトはどちらもEQで等しく、前文は否定で、後文は肯定である。これらのことより、表5からこの文の接続関係は補足型の換置(肯定と否定の置き換え)と判定できる。

②の方法は意味的な推論は動詞や名詞、形容詞の意味に基づいて、おもに順接型や補足型の接続関係を求めるものである。この推論規則を用いた推論は、接続関係を求める文に現われた動詞、名詞、形容詞

表3 名詞間の概念関係

関係名	内容(名詞1は前文、名詞2の後文の名詞)
EQL	名詞1と名詞2が等しい(ID名詞が等しい)
ANT	名詞1と名詞2の対の関係にある
ILR	名詞1が名詞2の上位概念
ABS	名詞1が名詞2の低位概念
EKO	名詞1と名詞2の上位概念が等しい
SYN	名詞1と名詞2が等しい(ID名詞は等しくない) または、名詞1は名詞2の同意語

表4 イベントと名詞、イベントとイベントの関係

ABS	前文の格の引数がイベントで、その格表現の動詞の名詞化が名詞2に等しい
CCT	後文の格の引数がイベントで、その格表現の動詞の名詞化が名詞1に等しい
SYN	前文のイベントと後文のイベントが等しい (独立した場合、同列型の反復の接続関係を持つ)
RVS	前文のイベントと後文のイベントが等しい (独立した場合、逆接型の反対の接続関係を持つ)

表5 構成要素の概念関係と文の接続関係

文の接続関係	アクト	肯定・否定	対応する格の関係
逆接型の反対	対の関係	前文と後文が等しい	全ての格の関係がEQL, SYN, EKO のいずれか
同列型の換言	等しい	"	ある格の関係がCCT もう一つの格の関係がEQL, SYN, EKO のいずれか
同列型の要約	"	"	ある格の関係がABS もう一つの格の関係がEQL, SYN, EKO のいずれか
対比型の対立 逆接型の反対	"	"	ある格の関係がRVS, ANT もう一つの格の関係がEQL, SYN, EKO のいずれか
同列型の例示	"	"	ある格の関係がILR もう一つの格の関係がEQL, SYN, EKO のいずれか
添加型の並列	"	"	ある格の関係がEKO もう一つの格の関係がEQL, SYN のいずれか
同列型の反復	"	"	ふたつ以上の格の関係がEQL, SYN のいずれか
同列型の換置	両方EQ	前文は否定 後文は肯定	OBJ の格の関係がEQL TOの格の関係がEKO
逆接型の反対	"	前文は肯定 後文は否定	ふたつ以上の格の関係がEQL, SYN, EKO のいずれか
連鎖型の連係	異なる	関係なし	後文の主語の格と 前文のある格の関係がEQL

の辞書から推論規則を取り出し、それらを二つのイベントに適用させてイベントの接続関係を求める。

推論規則は辞書の中に全て以下のような形式ではいつている。

((推論規則の情報)

(述語 +/- (ACTOR #A) (OBJ #B))

(述語 +/- (ACTOR #A) (OBJ #B))

推論規則の情報には、その規則が前向きなものか、それとも後ろ向きなものかの区別や、規則を適用して得られる接続関係名などが記入されている。

推論規則はそこにある命題表現と接続関係を求めるイベントのCD表現とのマッチングによって行なう。推論規則の先頭の命題表現と前文の命題、二番目の命題と後文の命題表現、三番目以後は前文以前にあるイベントの命題、もしくは名詞辞書内にある一般的事実としての命題が対応する。推論規則は規則内の全ての命題が満足した時に、その規則の接続関係が有効になる。

以下、推論規則の適用例を説明する。次の二文間で接続関係を求めるとする。

S3 A DOLPHIN IS LIKE A MAN.

S4 IT HAS A LANGUAGE.

S3, S4のCD表現はそれぞれP3, P4となる。

P3 (STATE + (VALUE LIKE)(OBJ DOLPHIN-1)
(TO MAN-1))

P4 (AHOLD + (ACTOR IT1)(OBJ LANGUAGE-1))
形容詞LIKEの辞書から推論規則を取り出すと下のよ
うな規則を得る。

((-> < 0 (DR:RG NIL)(<))

(STATE + (VALUE LIKE)(OBJ #A)(TO #B))

(*PRD + (ACTOR #A)(OBJ #C))

(*PRD + (ACTOR #B)(OBJ #C)))

*PRDはプリミティブアクトの変数で、#A, #Bなどは格の引数の変数である。P3, P4を上規則にあてはめていき、推論規則の三番目の命題表現は各変数にその値をあてめると次のような命題を得る。

P5 (AHOLD + (ACTOR MAN-1)(OBJ LANGUAGE-1))

P5は”人間は言葉を持っている。”という当然の内容があらわされている。このような知識は当然どこかの辞書の中にはいっていないなければならない。そこで名詞LANGUAGEの中に次のような命題の形でいれておく。

P6 (AHOLD + (ACTOR #A (&NC (AKO HUMAN)))
(OBJ #B (&NC (INF LANGUAGE))))

&NC は変数が取り得る必要条件で、変数#Aの場合、取り得る名詞の上位概念はHUMAN でなければならず、また変数#Bの場合、取り得る名詞の概念がLANGUAGE と等しいものでなければならぬことを表わしている。二つの命題は等しい関係にあるので、推論規則内の3番目の命題が成立する。よって、この推論規則が有効になり、二つの命題の接続関係は同列型の例示(DR:RG)となる。このように名詞辞書の中には接続関係を求める推論規則の他に、その名詞に付随する一般的事実を命題の形で入れておく。

③のスク립トはSchankら[2]によって提唱したスク립トを使用したものである。スク립トは日常的な活動エピソードにおいて見られる具体的な連続したシーンの集まりである。推論規則に合わない二つの命題でも、スク립トを起動させることで、ある特定の状況や場所を設定して、その間の接続関係を求めることが可能となる。スク립トは”店で食事をする”や”犯人を逮捕する”などの一つの日常的なエピソードを表わすマクロ的のものから、”買う”のような一つの動作を一連のシーンに分解して並べたミクロ的なものまで用意しておく。シーンはその内容から4種類にわけている。

① 基本シーン……スク립トの内容を表わす基本的なシーン。

② 条件シーン……基本シーンが成立するための条件を具体的に表わしたシーン。

③ 内容シーン……基本シーンを構成するシーン。

④ 結果シーン……基本シーンが成立したために得られた具体的なシーン。

各シーンにおいては時間的順序に命題が並べてある。スク립トは動詞と名詞の辞書の中に入れてお

```

($NOTIFY1
(SCENE-NUMBER 3 4)
(BASIC-SCENE
(O - (NTRANS + (ACTOR *B) (OBJ *B) (TO *C)))
)
(COND-SCENE
(1 - (MHOLD + (ACTOR *A) (OBJ *B)))
(2 - (MWANT + (ACTOR *A)
(OBJ (SPEAK + (ACTOR *A))))
)
)
(CONT-SCENE
(3 - (SPEAK + (ACTOR *A) (OBJ *B) (TO *C)))
(4 - (ATTEND + (ACTOR *C) (OBJ *B) (FROM *A)))
)
)
(RSLT-SCENE
(5 - (MHOLD + (ACTOR *A) (OBJ *B)))
)
)

```

図6 スクリプトの例

り、名詞の場合は格の条件に合っていれば、スクリプトを起動させることができる。スクリプトの例を日本語で図6に示す。このスクリプトは”知らせる”という行為を時間の順に具体的なシーンで示している。なお、SCENE-NUMBERにはそのスクリプトの重要なシーンを表わす番号が入っている。スクリプトを用いて接続関係を推論するために、命題のあてはまるシーンとシーンとの関係が接続関係とどう結びつくかを決めておく。

以下、スクリプトを使った接続関係の推論の例を示す。今、連続した二つの文S7、S8があるとすると

S7 I KNOW THE TRUTH.

S8 I SAY IT TO HIM.

これをCD表現で表わすと、P7、P8のように表わせる。

P7 (MHOLD + (ACTOR I-1)(OBJ TRUTH-1))

P8 (SPEAK + (ACTOR I-1)(OBJ IT1)(TO HE1))

P7はスクリプトNOTIFY1の条件シーンと合い、P8は内容シーンと合っている。条件シーンと内容シーンが連続していることより、この二文間の接続関係は順接型の順当になる。このシステムでのスクリプトは単に二文間の接続関係を求めるためにのみ使われて、スクリプトそのものが文脈解析によって得られる文章構造の基盤になることはない。

3.3 代名詞の処理

代名詞の処理は意味解析の時点とこの二文間の接続関係を求める時点の二段階で処理する[6]。意味解析の時点で全ての代名詞の処理を行わないのは、代名詞の使用という情報を損なうことになり、また、誤った推論をすることが多くなると考えられるからである。よって意味解析では、代名詞が指す名詞の候補を同じ文中で代名詞以前に出現した名詞に限って処理をし、一文内での処理で照応関係が明らかにされなかった代名詞に対しては、接続関係を求めると同時に処理を行なう。

二段階で処理するのは次の例から考えても妥当である。例えば、文S9、S10、S11があるとすると

S9 SCIENTISTS DISCOVERED DOLPHINS HAD A LANGUAGE.

S10 THEY TALKED WITH EACH OTHER.

S11 THEY DISCOVERED MONKEYS HAD ONE, TOO.
S9の後にS10が続く場合、代名詞THEYはDOLPHINSを指すが、S11が続く場合、THEYはSCIENTISTSを指す。この違いが出るのはS9とS10、S9とS11で得られた二文間の係る部分が異なるからである。このように意味解析だけでは決して正しく照応関係を処理することはできない。意味解析での代名詞の処理は次のような文に行なわれる。

S12 WHEN A RABBIT SEES SOMETHING STRANGE, IT RUNS AWAY.

この場合、代名詞ITが指示する名詞の候補はRABBITとSOMETHINGである。RABBITは動詞RUNの主語として意味的に適当であり、代名詞の数ともあうのでITはRABBITを指していると判定することができる。

意味解析の時に処理できなかった代名詞はそのまま意味表現の中に入れておき、イベント間の接続関係が有効になった場合、イベントの命題表現(格構造表現とCD表現)内の代名詞をそれに対応する名詞に置き換える。

代名詞は名詞句ばかりでなく、文やあるまとまった文脈を指す場合がある。この場合も名詞を指す場合と同じような処理が考えられる。

S13 I WANT TO SAVE MUCH MONEY.

S14 BUT THIS IS NOT EASY.

という二つの文は下に示した形容詞EASYの辞書に入っている推論規則を使うと代名詞THISはTO SAVE MUCH MONEYを指すことになる(WANTにも係るが、普通、願望に対して困難ということはないので、*PRDにはWANTを取らないという規則を設けておく)。

((<- < 0 (GS:HT NIL)(<))

(*PRD + (ACTOR *DMY)) (DMY=dummy)

(*STATE - (OBJ (*PRD +)))

しかしながら、実際の文章では文や節を指す代名詞を処理するのは非常に困難である。

3.4 接続関係の決定

三つの方法でイベント間の接続関係を求めるのであるが、それぞれの方法から異なった接続関係が得られることも少なくない。そこで一つに絞るために下のように優先順位を設けておく。

構成要素の概念関係の方法 <

推論規則の方法 <

スクリプトの方法

これまでの議論は主節と主節との関係を対象にしている。しかし、必ずしも主節間で接続関係が得られるとは限らない。前文の主節と後文の目的節との間で得られたイベントの接続関係がそのまま二文間の接続関係になることもある。そこで、主節同士以外の節は以下の規則に基づいて接続関係を得る。

1) 目的語となる節は主節と同じように扱う。

2) 前文の主節と後文の副詞節との間で、あるいは、前文の副詞節と後文の主節との間で文の接続関係が成立するのは、二つの節の接続関係が同列型のものに限る。この時、原則として二文間の接続関係は得られた接続関係に従う。

3) 前文の関係詞節と後文の主節との間で得られた接続関係は、前文の主節がBE動詞の文で、その主語か補語に関係詞節が係っている場合か、関係詞節と主節の間で得られた接続関係が同列型のもののみ、有効となる。

なお、ひとつの方法で二つ以上の接続関係が得られた場合、つぎのような優先順位を設けて接続関係を一つに絞る。

主節同士 > 主節と目的節 >

目的節同士 > 主節と副詞節

以上二つの優先順位から絞られた接続関係が二文間の接続関係となる。ただし、後文が接続詞を持つ場合は異なる。接続詞は確かに接続関係を求める際に重要な働きをするが、単に後文が接続詞を持つからといって二文間に係り受けの関係があると断定できない。なぜなら、文章中においては必ずしも後文が前文に係るとは限らないからである。文脈によっては後文が前文よりも前の文に係ったり、それまでの文脈全体に係ったりする。後文に接続詞がある場合、上の三つの方法で得られた接続関係の内、その接続詞が持つ接続関係と合わなければ二文間には係り受け関係がなかったものとして処理する。

3.5 表現形式

時枝は入子型構造形式を文章内の段落間の関係にまで適用している[7]。入子型構造形式はもともと文における句と句の関係で、句が句に含まれて次第に大きな句となり、全体が完結形式をとった句となって文をなすような日本語の文構造のことを言う。時枝は文と句の関係を、文章と段落の関係に結びつけている。

そこで、この入子型構造形式を文のレベルで使い、文の接続関係と結びつけることを考えた。接続関係の重みが--> または==> であるものは前文の命題を受けて後文の命題が展開する関係にあり、重みが<-であるものは前文の命題に対して後文の命題が解説または説明している関係にある。従って、接続関係の重みが--> または==> であるものは後文が前文を

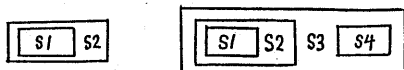


図7 入子型構造形式の文脈例

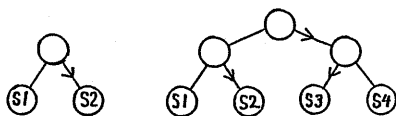


図8 二分木・拡張二分木の例

含む関係にあり(図7、左)、重みが<-であるものは前文が後文を含む関係にある。この包含関係は段落と同じように、二文間ばかりでなく文章全体にまで拡張することができる。その結果、図7右では、外の枠組みである文ほど文章の中心文に近く、中に入っている文ほど中心文から遠い関係になる。

文脈解析後の表現は単に文と文の間をリンクで結んだ鎖状にするのではなく、個々の文が文脈をどのように構成しているか、あるいは文脈においてどのような位置にあるのかを明らかにするものでなければならぬので、図7のような入子状態を表わせるのが望ましい。そこで図8のように二文間の関係を二文木の構成で表わすことにする。また、三文以上の文と文の関係を表わすために拡張二分木の構成を取る。

3.6 文章構造の作成

システムは文章の先頭から一文ずつ処理しながら、それまでの木構造(既に処理された文脈)の根あるいは葉に部分木を付け加えて拡張二分木の文章構造を作成していく。この拡張二分木の組み方を決定するのが、3.1で述べた文の接続関係の重みである。一文を処理する場合、既にある拡張二分木の最右端から中間順に文(イベント)を辿り、接続関係が得られるかどうかを試みる。得られた接続関係の重みが<-か<->である場合、相手のイベントまたはそのイベントが支配する部分木との間で二分木を構成し、その相手のイベントまたは部分木と置き換える。"部分木aを支配するイベントb"とは部分木aから接続関係の重みにそって枝を辿った時に到達するイベントbのことを言う。反対に、接続関係の重みが-->か==>、<=>である場合、相手のイベントの場所から根の方へ辿り、部分木を支配するイベントとの間で接続関係を求める。そして再びその重みが-->か==>、<=>であればまた根の方へ辿るという具合に、接続関係の重みが-->か==>、<=>である限り、木を根の方向に向かって辿る。重みが<-か<->である場合、あるいは木の根に到達した場合、イベントをその場で部分木と二分木を作り、文脈の木に組み込む。

例えば、これまで処理した文脈の木構造が図9左のようになっている場合、#2と#3の間に重みが右向きの接続関係が得られれば図9中のように#3を組み合わせ、左向きの場合は図9右のように組み合わせる(#iはイベント番号)。

一つの文が等位接続詞で結ばれている場合、その

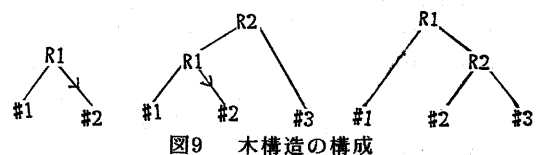


図9 木構造の構成

接続詞の辞書からその意味に合う接続関係を取り出し、節を葉にして二分木を作る。それまでの文脈と接続関係を求める場合、接続関係の重みの方向にある命題を先にする。

3.7 特別な処理

1) 話題イベントの設定

文章中に同じ内容の文が離れた場所に現われることも少なくない。そういった文は通常、その文章の話題になっている場合が多い。そこで、新しく意味解析を終えた文（イベント）は接続関係を求める前に文脈解析を終えた各イベントとの間で動詞や必須格のマッチングを取り、それらが等しければその二つのイベントの和集合を取り、その和集合を一つの独立したイベントとする。このイベントとそれを作り出した二つのイベントの間でアークを結ぶ。このイベントを使うことによって、大意の生成時に離れた場所からの情報を取り出すことができる。

2) 文脈内での命題の生成

連続した二つの命題から一つの命題を次の二つの方法により生成する

① 動詞の上位概念から求める

二つのイベントの接続関係の重みが \leftrightarrow の場合、主動詞の上位概念が等しく、一方の格の名詞が対応する格の名詞自身、または同意語、上位概念と等しい時に限り、動詞の上位概念を述語動詞にした命題を生成する。

② スクリプトから求める。

スクリプトを使って、接続関係を求めた時、二つの命題が内容シーンにあてはまり、得られた接続関係が添加型の累加であれば、基本シーンに二つの命題の名詞を入れて命題とする。

この新しく得られた命題はCD表現で表わされているので、そのCD表現から格構造表現を生成する。そして、二つの表現で表わされた命題を接続関係が記述されている場所に収める。この命題はその後、他の命題と同じように接続関係を求める時に使われたり、大意を生成する時に取り出されたりする。このことによって文脈間の係り受けも可能となる。

3) 判断の後処理

連続した二文において、後文の主語が前文のある格の名詞と同じである場合、その二つの文の接続関係は連鎖型の連係になる。これは二つの文の間には何か関係があるが意味的に明瞭でない接続関係で、それだけでは前文、後文のどちらに重みがあるかを定めることが出来ないものである。この接続関係の場合、後文は前の文脈全体に対して関係がなかったものと仮に決めておき、後文より後の展開によってその重みを判断する。以下、具体的に示す。

図10において、#1、#2、#3は既に文脈解析を終えたイベントで、#4は#3と連鎖型の連係の接続関係を持ち、#5は未処理なイベントとする。

#4と#5の間で接続関係が得られた場合、重みが \leftarrow

でない限り、得られた重みをそのまま#3と#4の間の接続関係の重みにする。重みが \leftrightarrow の場合、接続関係がなかったものとして#4と#5の間で二分木を作る。

#4と#5の間で接続関係が得られなかった場合、#4を飛び越して#3と#5の間で接続関係を求める。接続関係が得られれば、自動的に#3と#4の間の接続関係の重みを \leftarrow にして、木構造に加える。

4) 木構造の変換

ひとつのイベントに対して、同じ接続関係で二つのイベントと関係している場合、木構造の作成段階で、係る二つのイベントを接続関係が添加型の並列である二分木と作って結ぶ。そして、その二分木と最初のイベントの間で接続関係によって繋ぐ。例を図11に示す。

5) 要約・詳述の処理

論説文や説明文では、ある一つの命題の内容を複数の命題で解説することがある。その内、解説される命題の存在によって初めてその文脈の内容がわかる文脈がある。言い換えれば、文脈自身によって一つの命題を作りだせないし、文脈自身にその命題の手掛かりとなるものを持っていない文脈である。この後者の場合、スクリプトの特殊な使い方で処理をする。スクリプトは一つの命題を複数命題によって説明したものと言うことができる。あるスクリプトを起動させて、そのシーンがそれまでの文脈を構成する命題と重なってれば、その文脈はそのスクリプトの基本シーンとなる命題を解説していることになる。そこで、文脈解析において拡張二分木を辿っている際中、その根に至り、かつ意味解析によって得られた新しい命題がそれ自身基本シーンとなるスクリプトを持っている場合、そのスクリプトを起動させて、拡張二分木の各命題がスクリプトの骨格となるシーンにあてはまるならば、その拡張二分木（処理された文脈）は新しい命題を解説していると判断する。この場合、処理された文脈と新しい命題との間の接続関係は同列型の要約となる。反

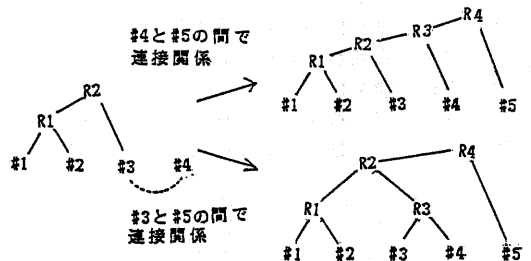


図10 判断の後処理

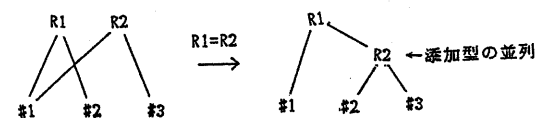


図11 木構造の変換の例

対に、ある一つの命題を後ろから説明する場合も同様な処理をする。この時の接続関係は同列型の詳述である。

4. 大意の生成

トピックセンテンスは木構造の根から接続関係の重みにそって辿り、到達した命題とする。そして、トピックセンテンスと関係のある命題を組み合わせて大意を構成する。トピックセンテンスには要約的中心文と結論的中心文の二通りがある[4]。要約的中心文は一つの段落(文章)の中心的内容を要約的に示している文で、繰り返しの部分、付加的な部分を取り除いてとらえられる。結論的中心文は一つの段落(文章)の中心的内容を結論の形で示している文で、論理の道筋を辿ったり、いろいろな説や事実を検討したりしたうえで、その行き着くところとしてとらえられる。大意を構成するためにはトピックセンテンスがどちらの型の中心文に相当するかを判断して、その型にふさわしいトピックセンテンスと関係のある文を取り出さなければならない。この判断は木構造の根の接続関係が論理的結合関係のもの(重みが \Rightarrow)であればトピックセンテンスを結論的中心文とし、それ以外は要約的中心文とする。トピックセンテンスが要約的中心文である文章の場合、トピックセンテンスと関係のある文は最初の根に対して接続関係の重みの方向と逆の方向を辿り、それ以後接続関係の重みの方向に辿って得られた命題になる。この二つの命題を根の接続関係の意味を持つ接続詞によって結ぶ。結論的中心文の場合、トピックセンテンスと関係のある文は、論理的結合関係である接続関係で結ばれた全ての命題である。各命題間は接続関係の意味を持つ接続詞によって結ばれる。

二つの命題を求めて枝を辿っている際中、添加型の並列の接続関係を通じた場合、両方の枝を辿り到達したイベントの命題を取り出す。この場合、二つの命題が文章中で等位接続詞で結ばれている場合はその接続詞で、木構造の作成段階で命題が結ばれた場合ORで両方を結ぶことにする。

文章構造が二つ以上の根によって構成された場合は、それぞれの根から接続関係の重みの方向に従って辿り、到達したイベントの命題を取り出し、それらを並べて大意とする。

5. 処理例

二つの文章の処理例を示す。処理例文1はそのトピックセンテンスが要約的中心文で、処理例文2は結論的中心文である。文番号は文章の先頭から順にS1、S2、S3...と付けられている。各文番号内のイベント番号は主節のイベント、次に副詞節あるいは目的節のイベントの順に付けられている。

6. おわりに

現段階において作成したシステムが処理できる文章の範囲は限られている。この理由として次のようなことがあげられる。

- 1) 推論規則が一階の述語論理式の形で記述されているため、助動詞や副詞、前置詞句などが文におよぼす意味的な役割を取り入れることが出来ない。
 - 2) 抽象名詞などは、その性質をスロットによる階層構造で表わすことが難しい。また、その性質を推論規則に取り入れることが難しい。
 - 3) 前文の主節と後文の副詞節に関係が得られた時など、主節同士以外の関係から前文と後文の接続関係を判定する基準を設けることが難しい。
- 以上の問題を解決することがこれからの課題となると思われる。

文脈が持つ情報を文の接続関係の決定や接続関係の重みの決定に使うという形であればトップダウンの処理をこのシステムに付け加えることが出来る。このシステムでは文の接続関係の重みを固定したまま処理しているが、接続関係の重みは文脈によって変化するべきものである。また、接続関係の決定に際しては、文脈の流れを閉ざさないためにも、全体との関係をも考慮にいれるべきである。トップダウンの処理の一つとして、キーワードの利用が考えられる。キーワードによって論理的展開とは別に、話題や視点の移動を捉えることができる。キーワードはまた、大意の生成においても単語を取り出す判断の大きな要因になる。

参考文献

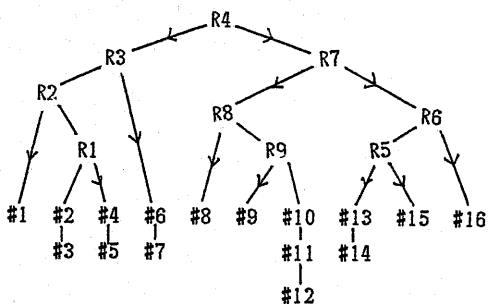
- [1] Rumelhart, D.E.: "Notes on a schema for stories." In Bobow, D.G. & Colling, A. (Eds.), Representation and Understanding, Academic Press, (1975).
- [2] Schank, R.C. & Abelson, R.P.: Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, (1977).
- [3] Hobbs, J.R., Walker, D.E. & Amsler, R.A.: "Natural Language Access to Structured Text", Proc. 9th int. Conf. on Computational Linguistics (COLING-82), 127-134, (1982).
- [4] 市川 孝: 国語教育のための文章論概説、教育出版、(1978).
- [5] Hobbs, J.R.: "Coherence and Coreference", Cognitive Science, 3, 69-90, (1979).
- [6] Hobbs, J.R.: "Selective Inferencing", Proc. 3rd National Conf. of Canadian Society for Computational Studies of Intelligence, 101-114, (1980).
- [7] 鈴木一彦: 日本文法本質論、明治書院、(1976).

処理例文 1

A bear sleeps all winter. When it gets very cold, most bears crawl into their dens. And they sleep until the warm weather comes. Why don't they get hungry or thirsty during their sleep? Bears eat a great deal of food in summer and fall. This food is stored in their bodies as fat. They live upon this stored fat as they sleep away the cold weather. During their sleep they hardly move or breathe. So their bodies do not use much food.

文番号(Si)と	
イベント番号(#j)の対応	
S1 (#1)	S6 (#9)
S2 (#2 #3)	S7 (#10 #11 #12)
S3 (#4 #5)	S8 (#13 #14 #15)
S4 (#6 #7)	S9 (#16)
S5 (#8)	

文脈解析によって得られた木構造



各接続関係及びコメント

(接続関係の番号は生成順)

- R1: 添加型の累加 (" 眠る " スクリプトによる)
- R2: 同列型の詳述
- R3: 転換型の転移 (#1と#7が同列型の反復)
- R4: 連鎖型の応答 (#6は疑問文)
- R5: 添加の並列
- R6: 順接型の順当
- R7: 添加型の並列 (木構造の変換による)
- R8: 連鎖型の連係 (判断の後処理による)
- R9: 連鎖型の連係 (判断の後処理による)

出力結果

* OUTLINE

MOST BEARS DO NOT GET HUNGRY OR THIRSTY WHILE THEY SLEEP. FOR MOST BEARS EAT A GREAT DEAL OF FOOD IN SUMMER AND FALL, OR THEIR BODIES DO NOT USE IT IN WINTER.

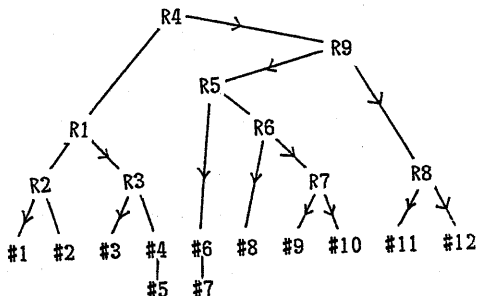
TIME 362.5839 S

処理例文 2

We have been using language for a long time. We have been communicating with others by language. At first, there were no words. Scientists tell us that our earliest ancestors had no language. Therefore they probably made noises that meant something. A growl meant anger. A screech might have meant danger. A grunt might have meant hunger. They probably made gestures and made faces that meant things, too.

文番号(Si)と	
イベント番号(#j)の対応	
S1 (#1)	S6 (#8)
S2 (#2)	S7 (#9)
S3 (#3)	S8 (#10)
S4 (#4 #5)	S9 (#11 #12)
S5 (#6 #7)	

文脈解析によって得られた木構造



各接続関係及びコメント

- R1: 逆接型の反対
- R2: 連鎖型の連係 (判断の後処理)
- R3: 同列型の換言
- R4: 順接型の順接
- R5: 同列型の例示
- R6: 添加型の並列
- R7: 添加型の並列
- R8: 添加型の並列
- R9: 添加型の並列

出力結果

* OUTLINE

WE HAVE BEEN USING LANGUAGE FOR A LONG TIME. BUT THERE WERE NOT WORDS AT FIRST. SO OUR EARLIEST ANCESTORS PROBABLY MADE NOISES WHICH MEANT SOMETHING. MADE GESTURES OR MADE FACES WHICH MEANT THINGS.

TIME 175.8159 S