

省略表現を含む不完全テキストを知識を用いて復元する 英文補完システム†

唐 沢 博** 松 永 義 文**
小 川 均** 田 村 進 一**

日常会話においては省略表現が多用される。省略は、統語構造上の類似性をもとに復元が可能なものから文脈的な情報に依存するもの、知識を用いて推論することで復元できるものまで種々のレベルに存在し、その復元には、それぞれに応じた手法が必要である。本論文では、復元の手法を体系化し補完法と呼ぶ形式に整理した。さらに、この補完法を基礎においたコンピュータ・システムを作成してその評価を行った。同システムは、英文テキスト作成支援の目的をもつ。すなわち、利用者の頭に浮かんだ単語、句、文の羅列を入力することにより、システムは利用者の意図を解釈して、より完全なテキストを生成する。利用者は、その出力テキストをポスト・エディットして目的とする最終テキストを得る。このシステムの構成上要求される機能は、単語の並びを意味のある集りにまとめ上げていく、省略されている要素を見いだす、種々の知識を用いて省略を復元する、そして表層文を生成するなどの能力である。これらの機能を個々のプログラム・モジュールとして実現し、パーソナル・コンピュータ上に BASIC でインプリメントした。また、辞書構成においてコンパクトかつ完結性をもたせるために、850 語から成る Basic English の体系を導入した。

1. ま え が き

日常会話で用いられている言葉は効率的なコミュニケーションのために省略された表現となることが非常に多い。文法的にみると非文と考えられるような単語や句の羅列が、会話者間では理解され意思の疎通が実現している。こうした日常的な会話形式のコミュニケーションが計算機システムと人間との間で実現すれば、優れたマンマシン・インタフェースとなりうる。本論文では、人間が頭の中に描いている事柄を、表出された情報の集りからいかにして汲み取るかといった課題を中心にして、英文の断片情報からそれが全体として意味すると推測されるような内容を有するテキストを生成するシステムについて報告する。このシステムは英文テキスト補完システムと呼ぶもので、英文文章の構成に関して豊富な知識をもっている。本補完システムは、利用者の英文テキスト作成を支援することを目的とする。システムの入力情報は、利用者が思いつく単語、句、文などの並びであるので、文法的な誤りや意味的あいまいさ、統語構造の不明瞭さを含む可能性が高い。一般にそのような情報を機械処理の対象とするには困難な問題が多い。たとえば、LIFER¹⁾ の

入力解析における省略処理は不完全な入力文の断片を扱うための一つの手法であるが、それは構文構造上の類似性が成り立つ範囲内でのみ有効である。これに対し補完システムの入力解析は、積極的に意味のかたまりを認識し上位構造へまとめ上げていく。また、補完システムは、MARGIE²⁾ と多くの機能上の共通点をもつ。すなわち、構文構造に依存しない入力文の解析、推論知識を用いた事実の付加、テキストの生成などである。MARGIE の目的は入力情報を理解したことを示すことであるのに対し、補完システムは入力情報を骨組みとするより完全なテキストを復元するという点で異なる。復元されたテキストは、入力情報に対する一つの解釈とみなせる。この解釈の過程に理解が含まれる。一方、解釈は利用者の意図と極力一致していることが望ましい。その目安を得るためにシステムの補完能力の評価法を考案して適用した。その結果、質の良い復元テキストを得るためには推論知識が被う世界に偏りがなく、また辞書内の連想情報を適切に決める必要があることが明らかになった。この評価法は、補完システムを利用者の使用意図に適合させるのに有効である³⁾。換言すれば、利用者を教師とした知識の獲得のための基盤となりうる。

2. 英文補完システムの基本的考え方

システムが有する特徴を以下に示す。

(1) 入力情報は、英語の単語、句、文が入り混じったものである。

† English Text Complementing System Restoring an Incomplete Text with Ellipsis by Knowledge by HIROSHI KARASAWA, YOSHIFUMI MATSUNAGA, HITOSHI OGAWA and SHINICHI TAMURA (Department of Information & Computer Sciences, Faculty of Engineering Science, Osaka University).

‡ 大阪大学基礎工学部情報工学科

* 現在 富士ゼロックス(株)

このような情報の集りを不完全テキストと呼ぶことにする。しかしながら完全な文章が入力されてもかまわない。その場合出力テキストは、より自然な言いまわし、付加的な情報の付与といった操作が加えられた結果となる。

(2) 入力情報は意味的な流れをもっていることを仮定する。

意味的にばらばらな情報の集りは、人間の場合もそうであるようにコミュニケーションを成立させない。したがって、そのような入力に対する出力テキストは生成はするが意味的な保証はしない。

(3) 出力テキストは入力情報から大きくかけはなれない。

補完の目的は、入力情報を骨格とした省略の復元にある。すなわち、完全なテキストには存在しながら入力情報では失われているような構造の復元の結果として出力テキストを得る。

(4) 出力テキストは文法的な誤りは含まない。

(5) 文法的に閉じた体系とするために Basic English の体系 (Ogden, C. K., 1930)⁴⁾ を採用している。

Basic English の体系は、850 語の単語とその構成指針とから成る。この単語数は日常的に使用される 2 万語の単語から精選されたもので、計算機上に辞書をコンパクトに作成するうえで都合がよい。また構成指針は、不完全テキストの解析手法にそのまま取り入れられている。

(6) マイクロ・コンピュータ上にシステムを実現している。

補完システムを利用者の身近なパートナーとして位置づけるためには、小規模なパーソナル・システムとして実現する必要がある⁵⁾。マイクロ・コンピュータという非常に制限された資源のうえにシステムを構成するためにも、前述の Basic English の体系が必要となった。

3. 補完システムの構成

入力された不完全テキストは、前処理、解析、深層構造への変換、補完、生成の過程を経て、より完全なテキストへと復元される。システムはそれぞれの処理に対応するモジュールから構成され、順にプリプロセッサ、アナライザ、トランスレータ、コンプリメンタ、ジェネレータと呼ぶ。モジュールと処理との関係を図 1 に示す。モニタ・モジュールは、補完能力評価のための種々の情報を集計して出力する機能をもつ。

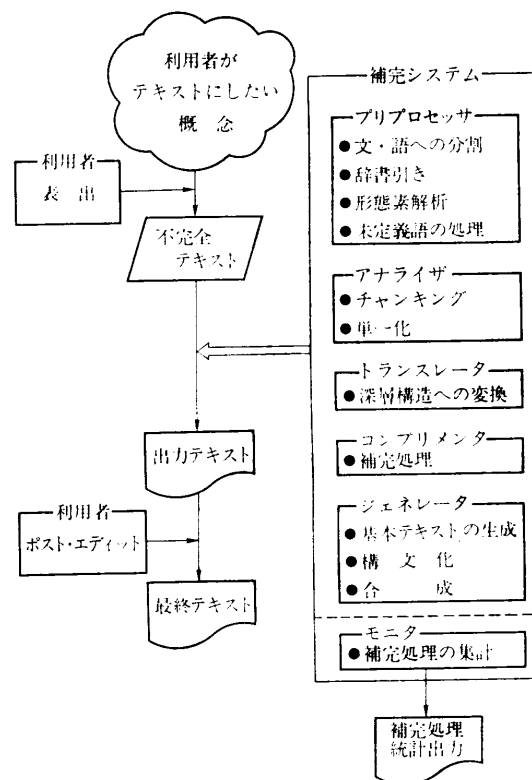


図 1 システム構成

Fig. 1 Construction of the system.

3.1 前処理

不完全テキストをその構成単位に分割して構造解析に備えるために、プリプロセッサは以下に述べるような 7 段階の処理を行う。

(1) 文の単位への分割

セミコロン、ピリオド、改行などを手がかりに文や句の単位でおおまかな分割を行う。

(2) 語の単位への分割

語や特殊記号 (ピリオド、コロン、疑問符など) を切り出す。

(3) 辞書引き

マイクロ・コンピュータの環境に辞書を作成するためには、アクセス効率に工夫を要する。この目的で、辞書構成をインデックスと記述本体とに分け、前者を内部メモリ上に置き後者をファイル上に置いた。

(3-1) 辞書引きのプロセス

辞書を効率的に参照するために 2 段階に分けて実行する。

step-1 インデックス上で辞書引きし、該当語があればその語と対になっているファイル登録番号を取り出して **step-2** に進む。該当語がなければ接尾辞処理

を行い、再び step-1 を試みる。
それでも該当語がなければ未定義語の分類を与える。

step-2 登録番号を用いてファイル・アクセスを行い、記述本体をとり出す。

(3-2) インデックスの構造

辞書のインデックスは、

語、登録番号

の羅列から成り、語をキーとして登録番号を得る。

(3-3) 辞書ファイルの構造

辞書ファイル内の記述本体は、

語、カテゴリ、属性 1, ...,

属性 n

の構造をとり、登録番号と 1 対 1 の対応をもつ。

(4) 形態素の解析

(5) イディオムの認識

例 1) for good→for-good

例 2) because of→because-of

イディオム中に付加されたハイフンは、そのイディオムが認識され構造化されたことを示す。

(6) 誤り綴りの処理

辞書引きで未定義語扱いにされた語について、おもに発音上の思い違いに由来する誤り綴りを訂正する。

(7) 未定義語の処理

この段階でなお未定義語とされている語は、接尾辞の処理と同様な操作を受けてカテゴリ化される。ただしこの場合は文字並びからの類推にすぎず、したがって辞書ファイルにも対応する記述はない。この処理でもなおカテゴリが定まらない語は、強制的に名詞として分類する。

以上の前処理が行われた結果を図 2 の b) に例示する。

3.2 不完全テキストの解析

プリプロセッサによって意味的なまとまりに分割さ

I GOT UP AT 7, AND SCHOOL AT 8
MATHEMATICS AND ENGLISH
AT HOME 4. SUPPER

a) 入力テキスト

1 I GET-UP AT 7 , AND SCHOOL AT 8
2 MATHEMATICS AND ENGLISH
3 AT HOME 4 .
4 LAST-MEAL-OF-THE-DAY

b) 前処理後のテキスト

1 I GET-UP AT 7 : T10
2 SCHOOL AT 8 : , RPA
3 MATHEMATICS-AND-ENGLISH
4 AT HOME 4 .
5 LAST-MEAL-OF-THE-DAY

c) 解析後のテキスト

1 S:I V:GET-UP TENSE:T10 TIME(1):AT-7
2 RELATION:RPA PLACE(1):AT-SCHOOL TIME(1):AT-8
3 O:MATHEMATICS-AND-ENGLISH
4 PLACE(1):AT-HOME TIME(1):AT-4-0'CLOCK
5 O:LAST-MEAL-OF-THE-DAY

d) 補完前の GTI

1 S:I V:GET-UP TENSE:T10 PLACE(1):AT-HOME
TIME(1):AT-7 STARTING-POINT:FROM-THE-BED
2 S:I V:GO TENSE:T10 RELATION:RPA PLACE(1):AT-SCHOOL
TIME(1):AT-8 STARTING-POINT:HOME TERMINAL:SCHOOL
WAY:BY-SOME-WAY
3 S:I V:HAVE-EDUCATIONS-OF O:MATHEMATICS-AND-ENGLISH
TENSE:T10 TIME(2):WHILE-I-BE-AT-SCHOOL COMMENT(1):MOD/DIFFICULT
COMMENT(2):GTIRED
4 S:I V:GO-BACK TENSE:T10 PLACE(1):AT-HOME
TIME(1):AT-4-0'CLOCK STARTING-POINT:SCHOOL TERMINAL:HOME
WAY:BY-SOME-WAY
5 S:I V:HAVE O:LAST-MEAL-OF-THE-DAY TENSE:T10
PLACE(1):AT-TABLE TIME(1):IN-THE-EVENING COMMENT(2):FHAPPY

e) 補完後の GTI

I GOT UP FROM THE BED AT 7, AND I WENT BY SOME WAY TO SCHOOL FROM HOME AT 8. I HAD EDUCATIONS OF MATHEMATICS AND ENGLISH THAT WERE DIFFICULT WHILE I WAS A SCHOOL, AND I GOT TIRED. I WENT BACK BY SOME WAY TO HOME FROM SCHOOL AT 4 O'CLOCK. I HAD LAST MEAL OF THE DAY AT TABLE IN THE EVENING, AND I FELT HAPPY.

f) 出力テキスト

図 2 処理例

Fig. 2 An example of the processing.

れた入力情報は、アナライザに渡されてさらに詳細に構造の検出が行われ再構造化が行われる。前者の構造の検出はチャンキングと呼ぶプロセスで、後者の再構

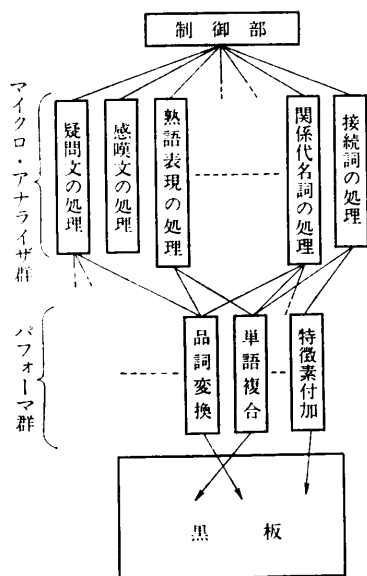


図3 アナライザの構造
Fig. 3 Structure of ANALYZER.

造化は単一化と呼ぶプロセスで実現される。

(1) チャンキング

Basic English の構成論において、チャンクと呼ぶ概念がある⁴⁾。チャンクは意味をもった小さなかたまりであり、文法的には句や節が対応する。このチャンクを認識する過程はチャンキングと呼ばれる。チャンキングの例を次に示す。

〔原文〕

I had come with the idea of sleeping at his house that night.

〔名詞句 チャンクの検出後〕

I had come with *the-idea-of-sleeping* at *his-house* *that-night*.

〔動詞句 チャンクの検出後〕

I *had-come* with the-idea-of-sleeping at his-house that-night.

(2) 単一化

重文、複文、混文のような複雑な構造をもつ文をすべて肯定単文に分解して次段のトランスレータの負担を軽減する。

(3) 解析の手順

入力テキストの解析には黒板モデル (Blackboard Model) を導入した。これはチャンキングや単一化に関する多くの専門家手続きのモジュール性をよくする目的があり、機能向上のための部分的な、手直しを容易にしている。解析は bottom-up に進行する。チャ

表1 GTI の構成
Table 1 Grammatical terms inventory.

項	文 番 号	
	1	2, ..., n
主 要 項	主 部	I
	動 詞 句	PUT
	直 接 目 的	
	間 接 目 的	A-LETTER
	主 格 補 語	
特 殊 項	目 的 格 補 語	
	時 制	past
付 加 項 (26 項目)	前文との関係	
	文 の 種 類	
	場 所	TODAY
	時 間	
	方 法	
	所 有	
	量 的	
	目 的 的	
	コ メ ン ト	
	目 的 的	
	コ メ ン ト	

ンキングや単一化はマイクロ・アナライザと呼ぶ一群の手続きによって実現される。各手続きはパフォーマンスと呼ぶ副手続きによって実際のテキスト処理を遂行する。各パフォーマンスは黒板を介して相互作用しながら各自の役割を果たす。図3はその様子を示している。制御部はマイクロ・アナライザの起動順序を指示する機能をもつ。前処理された入力テキストはチャンキングと単一化の手続きが繰り返し作用しながら形式的な構造へ変形されていく。図2の c) に解析例が示されている。

3.3 深層構造への変換

アナライザによって整形された入力テキストは、次にトランスレータにより深層構造に変換される。トランスレータの入力はチャンクの集合とみなすことができるので、その各チャンクと GTI (Grammatical Terms Inventory) と呼ぶ深層構造との対応づけを行う。

(1) GTI の構成

GTI は拡張された格表現の一種と考えられ、主要項群、特殊項群、付属項群とから成る。アナライザが番号つけた文単位を縦軸に、前述の各項を横軸にもつ2次元の表が GTI の構造である。表1に示したように、主要項は文法的な単位であり、特殊項は文の属性と文間の関連を示す。また付属項は各種格情報の集

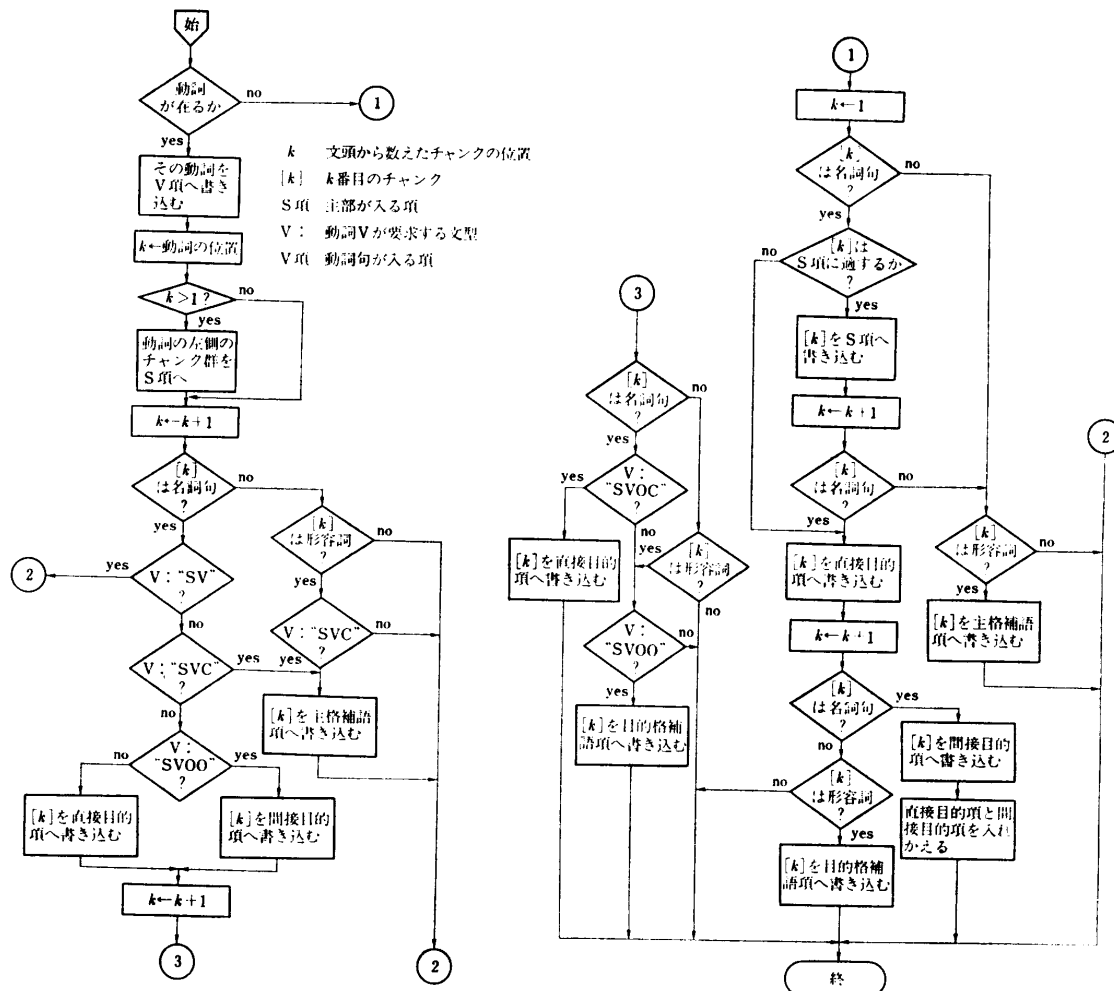


図 4 主要項の決定アルゴリズム (後半部)
Fig. 4 Decision algorithm for main terms (latter part).

りとみなせる。

(2) GTI への変換手続き

まず副詞と副詞句の切り出しを行う。これは各チャンクのカテゴリを確認することで容易に達成される。次に残ったチャンク群から前置詞句を切り出す。この前置詞句は、前置詞と名詞句としてチャンクされた単位とから成り、前置詞の種類に従って、そのチャンクが割り当てられる GTI の項が決められる。さらに残ったチャンクは、主要項とみなされる。主要項の決定アルゴリズムを図 4 に示す。動詞が要求する目的語や補語が欠けている場合、トランスレータは直接目的語の代りに something を、間接目的語の代りに someone を、補語の代りに in some situation を GTI にマーカとして書き込む。この操作を“第 1 次補完”

と呼ぶ。後述するように補完は第 1 次補完と第 2 次補完とに分けられる。トランスレータが受け持つのは第 1 次補完のみであり、主要項を対象とした文法的色彩の濃い補完である。図 2 の d) に例文が GTI に変換された様子が示されている。

3.4 補完処理

入力情報が不完全テキストである限り、GTI も値の欠落した項を含むことになる。そのような欠落項を文法的知識、文脈情報、一般的常識等を用いて補う過程が補完処理である。入力された不完全テキストに内在する意味の流れを把握し、より完全なストーリーとして理解するためにさまざまな種類の知識を用いて推論が行われる。補完は第 1 次補完と第 2 次補完の 2 段階から成るが、第 1 次補完は前節で述べた。より実質

的な補完は第2次補完が行い、コンプリメンタが受けもつ。第2次補完は補足型と付加型に大別される。補足型は、主要項、特殊項、付属項の補完にさらに分けられる。主要項に関しては、第1次補完で付したマークを推論可能な範囲内で具体的な語に置きかえる。特殊項や付属項の補完は、それを実現する推論知識の存在に依存する。付加型は、本来冗長な情報の付与を行うもので、単語からの連想や、項目間の関係から新しい文情報を付加したりする。

第2次補完を行うために、次に述べる4種類のタイプの補完規則を導入した。

(1) 文脈型

冗長な繰返しを避ける目的で省略された情報は、前後の文から復元できることが多い。たとえば、主語が省略されていればその前文の主語と同じである可能性が高い。動詞に関しても同様なことがいえる。

例)

{He went to school. → {He went to school.
I to church. → I went to church.

文脈型規則は、どのような場合に省略が行われるのかに関する知識と考えられる。

(2) 連想型

GTI 内に存在している語をキーとして連想されるさまざまな種類の情報を、その語の付属情報から取り出して GTI の欠落項に埋める。連想される情報はいくらかでも考えられるが、キーとなる語とともに辞書中に置かれるので、辞書のサイズ、アクセス効率などと照らし合わせて決める必要がある。

例)

lunch→at noon (time), have (verb).
get-up→from the bed (starting-point),
in the morning (time).

(3) 推論型

GTI の各項間や文間の関係から推測される情報を欠落項に対して補う。このために多くの世界知識や常識的知識が適用される。各知識はプロダクション・ルール形式をとり、GTI を作業領域とみなして推論を実行する。ルール表現された知識はモジュール性がよいので、知識の追加、変更が容易に行える。この性質は、出力テキストの内容の質を向上させるうえで重要である。

例)

I went out at 8. station at 9.

↓

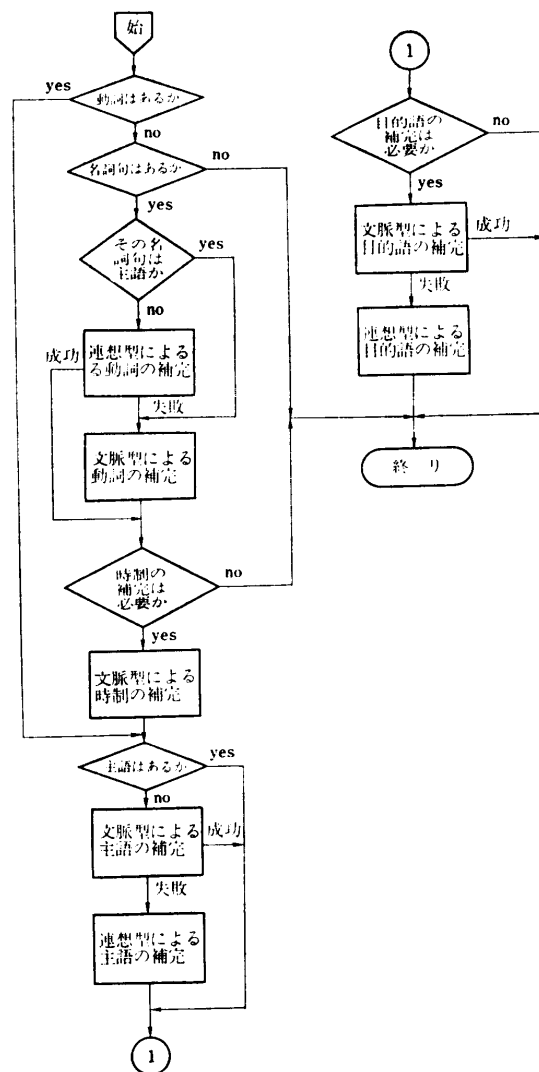


図5 主要項と時制に対する補完アルゴリズム
Fig. 5 Hokan algorithm for main terms and tense.

I went out at 8. I arrived at the station at 9 by some way.

(4) デフォルト型

文脈型、連想型、推論型のいずれの規則でも動詞の欠落項が埋められなかった場合、関連する他の項の内容に適合するような範囲の情報を補う。

例1) 未定義語の場合

psychology→something be done to the psychology.

例2) 形容詞のみの場合

happy→something be happy.

例3) 副詞のみの場合

quickly→something be done quickly.

GTI は最終的には少なくともその主要項に一つの欠落項も含まない内容となる。コンプリメンタの核となる部分の補完アルゴリズムを図5に示す。このアルゴリズムは、主要項および時制を示す項(特殊項の一つ)を文脈型と連想型の補完規則により求めるものである。その後、推論型、デフォルト型の順で補完が行われる。意味的整合性のチェックがデフォルト型補完の直前に実行される。これは、多くの規則が入り組んで適用された結果、各項間や文間の意味的な関係が乱れる危険性に対する処置である。

デフォルト型補完を導入したことによってコンプリメンタの補完能力は非常に強力となった。冠詞のみ、特殊記号のみ、接続詞のみといったほとんど意味のない入力を除けば、GTI は英文テキストを合成するために十分なものとなる。ただし、出力テキストの内容の質的な問題は、コンプリメンタ内の知識および辞書記述の内容と量とに依存する。

3.5 英文テキストの生成

コンプリメンタによって補完が施された GTI は、次にジェネレータによって英文テキストに変換される。変換された英文テキストの内容は、入力情報内に存在した文法的な誤りが除去され、ポスト・エディット時に利用者の注意を喚起し知識を引き出すような情報が付加されている^{6),7)}。もちろん文を構成するうえで必要な語はすべて補われている。また、よりエレガントな表現、言いまわしとなるような文の生成を実現するための機能をもたせた。出力テキストの生成は、基本テキストの生成、構文化、合成の3段階からなる。以下に各段階について説明する。

(1) 基本テキストの生成

形態素の合成、格変化の処理、冗長項の除去、第1次補完によって生じた不整合関係の改善等を経てから実際上の生成が行われる。

(2) 構文化

基本テキストは単文の集りである。しかし単文間の関係がみつければ、構文化の処理を行ってより読みやすい文となりうる。この操作は、3.3 節で述べた単一化と対照をなすものである。

(3) 合成

接続詞を用いて重文、複文、混文を可能な限り作り出す過程である。ただし、むやみと長い重文などはかえって読みづらいので、等位接続詞は2個以内に限定した。

生成された出力テキストは、入力情報に対する補完システムの解釈とみなすことができる。出力テキストに対するポスト・エディットは多かれ少なかれ必要であるが、システム利用者が入力情報に託した意図とシステムの解釈とができる限り近いことがエディット量の軽減という点で望ましい。解釈の良否はコンプリメンタの補完能力と関係しており、ジェネレータは文法上のチェックと読みやすい文体の生成とによってポスト・エディットを援助する。

3.6 補完操作の統計能力

補完能力の評価を行うための基礎データとして、次の3種類の情報を出力させるようにした。

(1) 第1次補完がいくつの項を補完したか。

(2) 第2次補完を構成する文脈型、連想型、推論型、デフォルト型の各手続きが何回起動されたか。

上記(1),(2)については、起動回数を示すヒストグラムと各補完型の起動比率が出力される。この出力情報は、現在処理したばかりのテキストに関するものと、それまでの処理のトータルとの両方が各処理ごとに出される。以上の機能はモニタと呼ぶモジュールが行する。

(3) 各補完型が起動された時点での補完の種類、補完した項とその内容および補完のために参照した項とその内容。

これは推論過程の説明機能と考えられる。この機能によって、個々のテキストが補完されていく過程と各補完型の寄与状況が明確に把握できる。

3.7 ハードウェア構成

32k バイトのプログラム領域を有するマイクロ・コンピュータ(富士通 FM-8)に、1ドライブ・ミニフロッピー・ディスク、ラインプリンタ、CRT ディスプレイが接続されている。補完システム・プログラムは F-BASIC でインプリメントされている。プログラム・サイズが大きい(約100k バイト)ので、各モジュールをチェイニングして実行する。

4. 実験と検討

8人の利用者が、一般的な話題を内容とする40の人力テキストをシステムに与えて、その出力テキストの質を検討した。出力テキストの質の問題は、補完能力の評価として扱った。システムの補完能力の評価は、そのために開発した手法³⁾を用いて、以下の2点に関して行った。

(1) 各補完手続きの起動率について、利用者の意

図をよく反映した質のよい出力テキストと、そうでない出力テキストの相異がどこにあるか。

(2) 補完システムと英語圏の人間とに同一の入力不完全テキストについて補完を実行させ、それらに要するポスト・エディット量の比をシステムの総合的な補完力の目安とする。

上記(1)の結果、質のよい出力テキストを生成した場合には、連想型と推論型の起動率が高かった。また(2)に関しては、人間の補完能力に近い例ほど、連想型と推論型の起動率が高かった。1テキスト当りの処理時間は、図2の例の場合、各モジュールのメッセージ出力および印刷をすべて抑制しないで実行した場合で約10分、抑制した場合で約5分であった。

5. む す び

省略の加わった文法的に不完全な英文テキストを入力し、それが何を表そうとしているのか解釈した結果を、文法構造の整った、より多くの情報を有するテキストとして生成する自然言語処理システムについて論じた。このシステムの構成にあたり、欠落情報を補うための補完法という手法を体系化し、その評価もあわせ行った。その結果、深層レベルにおける知識を多く活用する推論型、連想型と呼ぶ補完型が生成テキストの内容の質に大きく関与していることが明らかとなった。この事実は補完体系の性質を示すものであると同時に、システムの調整にとっても重要な基礎を与える。一方、英文断片の羅列から成る入力テキストを形式的な構造へと徐々に変換していく手続きを、不完全

テキストの解析手法として確立した。以上の成果をふまえて、今後は日本語不完全テキストの処理手法についても同様の考察を行う予定である。

謝辞 最後に、本研究の補完能力評価において協力していただいた研究室の諸氏に感謝いたします。また、本研究の一部は文部省科学研究費「特定研究」による。

参 考 文 献

- 1) Hendrix, G. G.: Human Engineering for Applied Natural Language Processing, Proc. of IJCAI-5, pp. 183-191 (1977).
- 2) Schank, R. C.: *Conceptual Information Processing*, North-Holland, New York (1975).
- 3) 唐沢, 田村, 松永: 英文補完システムの補完能力評価, 情報処理学会第27回全国大会予稿集, 4 D-3, pp. 1139-1140 (1983).
- 4) Lockhart, L. W.: *Basic Picture Talks*, The Basic English Publishing Co., Cambridge (1942).
- 5) John, K.: *Experiments in Artificial Intelligence for Small Computers*, Howard W. Sams & Co., Inc., Indiana (1981).
- 6) 松永, 小川, 田中: マイクロ・コンピュータ上での補完的英文生成システムの実現, 情報処理学会知識工学と人工知能研究会資料, 28-2 (1982).
- 7) 松永, 小川, 田中: 補完手法を用いた英文生成について, 情報処理学会第25回全国大会予稿集, 7 H-9, pp. 1047-1048 (1982).

(昭和59年5月1日受付)

(昭和59年10月18日採録)