

日常言語によるプログラミング

Programming in Everyday Language

杉本 徹 伊藤 紀子 岩下 志乃 菅野 道夫
Toru Sugimoto Noriko Ito Shino Iwashita Michio Sugeno

理化学研究所 脳科学総合研究センター

Brain Science Institute, RIKEN

In order to extend the application domain of natural language interfaces to more realistic tasks without the decrease of user's performance, it is desirable for users to be able to specify their requests as coherent texts consisting of more than one sentence, in other words, to write a program in everyday language. In this paper, we present a processing model of a natural language interface that accepts such an input text. Corresponding to the characteristics of task specification texts, it is comprised of paraphrasing, detection of loop structures, and program generation using rhetorical information. The algorithms explained in this paper have been fully implemented in our everyday language programming system that deals with personal email management tasks.

1. はじめに

近年、自然言語インタフェース技術の進歩により、多くの人がより簡単に情報システムにアクセスできるようになってきている。また、ユーザとシステムが協調して問題解決を行う対話型インタフェースの研究も進められている。しかし、ほとんどのシステムは、基本的にユーザの入力を1文ずつ順番に独立して処理するため、入力される一連のテキストの一貫性や相互関係を十分に扱うことができない。そのため、自然言語インタフェースをより複雑で現実的なドメインに導入する際に、ユーザとシステムのやり取りが増加してタスクの円滑な遂行の妨げとなる可能性がある。

この問題に対処するには、ユーザが、複雑な操作手順や実行条件を含む要求仕様を、複数の文からなる一貫した自然言語テキストの形で表現し、システムに入力できるようにすることが望ましい。自然言語で記述された要求仕様からのプログラム生成に関しては、これまでに事務計算などを題材として多くの研究が行われているが [Biermann 83, 原田 86], 受理可能なテキストの内容と形式は厳しく制約されていた。我々は、一般の人々が日常生活で使うようなより自然なテキストを理解し処理できるようにしたいと考えている。図1は、電子メール管理に関するそのような入力テキストの例である。このようなテキストを処理するためには、次の点について考慮する必要がある。

- (1) 理研の人から来たメールは「理研」フォルダに入れる。
- (2) 理研の人のメールアドレスは、末尾に「.riken.jp」が付く。
- (3) 研究室の人からのメールは、「研究室」フォルダに入れて自宅に転送する。
- (4) ただし、伊藤さん、高橋さん、小林さんからのメールでセミオティックベースに関するものは、「SB」という名前のフォルダに入れる。
- (5) この時、添付ファイルがあったらマイドキュメントの「SB」という名前のフォルダに保存する。
- (6) また、添付ファイルが文書ファイルの場合は、それを表示する。

図1: 日常言語プログラムの例

連絡先: 杉本徹, 理化学研究所 脳科学総合研究センター 言語知能システム研究チーム, 〒351-0198 和光市広沢 2-1, sugimoto@brain.riken.jp

1. 入力テキストに、コマンドと直接対応しない語句が含まれる可能性がある(例えば、文(1)における「理研の人」や「来た」)。
2. 暗黙的なループ構造が存在する可能性がある。例えば文(1),(3),(4)は、ユーザが受信したすべてのメールに対する条件判断および操作の繰り返しを表わしている。
3. 文の意味解釈は、隣接する文の影響を受ける。例えば「研究室」という語が理研内の研究室を指すという仮定の下で文(1)と(3)を素朴に解釈すると、研究室の人からのメールに対する相反する操作(「理研」フォルダと「研究室」フォルダに入れる)を表わすことになる。したがって、研究室の人からのメールは対象から除外するように、文(1)の解釈を変更する必要がある。

本論文では、このような複数の文からなる一貫した入力テキストを日常言語プログラムと呼び、日常言語プログラムを受理するインタフェースのための処理モデルの提案を行う。上述の1, 2の点に対処するため、コマンドに関する知識と換言を利用して、入力された日常言語プログラムの各文を正規化し、同時にループ構造の検出を行う。また、3を扱うために、入力プログラムの修辞構造 [Mann 88] を同定し、修辞関係を考慮しながら正規化の結果を統合し、実行可能プログラムを生成する。本論文では、対象ドメインとして電子メール管理を取り上げ、処理モデルの概要と実行例の説明を行う。

2. 日常言語プログラムの構造

人は、操作の手順や実行条件を日常言語で表現する際、必ずしもそれらを明確で論理的に正しい形で表現するとは限らない。Paneら [Pane 01] は、プログラミングの未経験者が複雑な情報処理課題の内容について記述したテキストの構造を調査した。その結果によれば、人は複雑な論理的条件や場合分けの組み合わせ、繰り返しの構造を明示的に表現しない傾向がある。明示的に表わされなかった情報は、テキスト内の文の間に存在する修辞関係によって補完されると考えられる。

我々は、日常の電子メール操作の内容を記述してもらうアンケート調査を行い、その結果の分析に基づき、このドメインにおいて重要と考えられる修辞関係を特定した。我々は修辞構造理論 [Mann 88] に従い、テキストの構造を、さまざまな種類

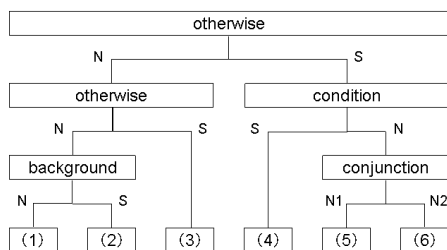


図 2: 日常言語プログラムの修辞構造

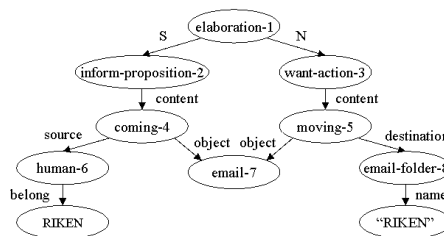


図 4: 文 (1) の概念構造

の修辞関係により結び付けられたテキスト範囲からなる木構造と見なす。それぞれの修辞関係は、主要素 (Nucleus) と従属要素 (Satellite), または複数の主要素を持つ。操作 (実行のための条件を伴うこともある) 間の関係としては、次の 4 種類が重要である。

conjunction 複数の主要素をすべて実行する (順序は自由)。

sequence 複数の主要素を指定された順序で実行する。

otherwise 従属要素の実行条件が成立しない場合のみ主要素を実行する。

condition 従属要素の実行条件が成立する場合のみ主要素を実行する。もし主要素と従属要素の操作が両立不可能ならば、主要素の実行を優先する。

また、操作と事実、あるいは事実間の関係として、次の 2 種類を考える。

background 従属要素は、主要素の意味を解釈するために利用される知識を表わす。

circumstance 従属要素は、主要素の記述内容を補足する。

図 1 のプログラムの修辞構造は、図 2 のように表わされる。

3. 処理モデル

3.1 概要

日常言語プログラムの処理モデルの概要を図 3 に示す。この処理モデルは、日常言語を使って情報処理を行う日常言語

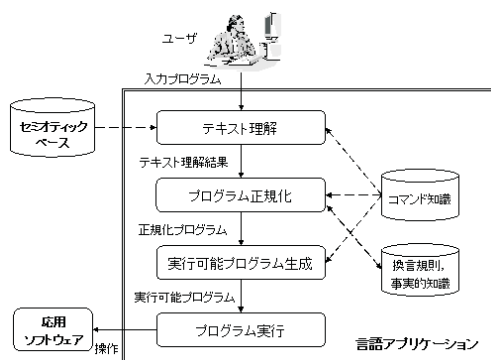


図 3: 日常言語プログラムの処理モデル

コンピューティング [岩爪 03] の枠組を利用しており、特に、言葉を使って応用ソフトウェアを操作する言語アプリケーションの一機能と位置付けられている。処理の途中結果や用いられる知識はすべて、語彙文法特徴、意味特徴、概念構造と結び付けられた日常言語テキストとして表現される。

ユーザが入力した日常言語プログラムに対して、まずセmiotickベース (Semiotic Base) と呼ばれる言語資源を使ってテキスト理解処理を行い、各文の語彙文法的、意味的、概念的構造と、プログラム全体の修辞構造を同定する [伊藤 03]。入力されたプログラムは一般に、実行可能な形、すなわち言語アプリケーションのコマンドと明示的な制御構造の組み合わせの形にはなっていない。そこで、コマンドに関する知識を参照して各文の実行可能性を検査し、必要に応じて、換言規則と事実的知識を用いた換言の処理を行う。このプログラム正規化の過程で、文中の暗黙的なループ構造の検出も行われる。その後、入力プログラムの修辞構造を考慮しながら各文の正規化の結果を統合し、実行可能な形をしたプログラムテキストを生成する。最後に、言語アプリケーションのコマンド実行機能を使って、応用ソフトウェアの操作を実行する。

3.2 テキスト理解

ユーザが入力した日常言語プログラムの意味は、選択体系機能言語理論に基づく言語資源であるセmiotickベースを使って理解される [伊藤 03]。各文に対して、形態素解析、係り受け解析、汎用辞書検索を行った後、語彙文法ベースと意味ベースに納められた選択体系網 (system network) と具現規則 (realization statement) を利用して、その語彙文法構造と意味構造を同定する。さらに、コンテキストベース内の概念辞書を参照して、概念構造を同定する。理解結果の例として、文 (1) の概念構造を図 4 に示す。なお、本研究ではこの他に、省略や代名詞を扱うための簡単な談話解析も用いている。

文単位での理解を行った後、入力プログラム全体の修辞構造を同定する修辞構造解析 [Marcu 00, 新森 04] を行う。我々は、2つの隣接するテキスト範囲の間に成り立ちうる修辞関係の種類を求めるために、文の過程構成型や接続表現などの語彙文法的特徴と、次節で説明するコマンド知識から得られる操作間の両立可能性についての情報を利用する。例えば、 S_1 と S_2 がともに操作を表わす文を先頭要素とする隣接するテキスト範囲であり、 S_2 が「また」という接続詞で始まるならば、 S_1 と S_2 は、これらの中で指定された操作が両立可能かどうかに応じて、「conjunction」または「otherwise」の関係にあると推測される。

テキストの修辞構造を同定するには、修辞関係の種類を求めるだけでなく、テキスト範囲のグループ化の順序を決める必要がある。このため、テキスト範囲の一貫性を、対象スロットの内容と概念構造の類似性に基づいて評価し、最も一貫性の高い構造を選ぶ。

表 1: コマンド知識の例

種類	コマンド	必須スロット	
		単一値	複数値可能
操作	入れる	移動先	対象
	転送する	—	対象, 宛先
条件	等しい	—	対象 1, 対象 2
	末尾に付く	—	対象, 文字列
モノ	差出人アドレス	—	メール
	メールフォルダ	—	名前

表 2: 換言規則の例

換言前	[メール]が[人]から来た
換言後	[メール]の差出人が[人]と等しい
換言前	[メール]の差出人が[人]と等しい
換言後	[メール]の差出人アドレスが[人]のメールアドレスと等しい
換言前	[人]をメンバーとする組織の人
換言後	[人]

3.3 プログラムの正規化

入力プログラムに含まれる文の中で、事実を表わす文に対しては、何も処理を行わず事実的知識のリストに追加する。それ以外の、操作を表わす文に対しては、コマンド知識を参照して実行可能性の検査を行う。コマンド知識は、言語アプリケーションの各コマンドに対して、それを実行する際に値を指定する必要のあるスロットのリストを明示したものである。コマンド知識の例を表 1 に示す。あるテキストの実行可能性を検査するには、そのテキストの主要部（例えば、節の場合は主動詞）がコマンドに対応しており、そのコマンドのすべての必須スロットの値が指定されていて、かつ実行可能であることを確認する。必須スロットの中には、複数の値を取ることが許されるものもあり、そのようなスロットに対しては、値が 1 つに特定できなくても（つまり、実行可能でなくても）、その値が属すと考えられる母集合が文脈から特定できればよい。この場合、その母集合のすべての要素に対して条件判断または操作を繰り返すループ構造が生成されることになる。検査の結果、テキストが実行可能でないことが判明した場合は、換言規則（表 2 参照）と事実的知識を用いてそのテキストを換言し、再度実行可能性の検査を行う。最終的に、文の正規化の結果として、実行可能な条件部分、実行可能な操作部分、および（もし存在する場合は）ループ範囲とループ要素が求まる。

具体例として、文 (1) の正規化について考える。文 (1) は複文であり、主節が操作部分、埋め込み節が条件部分に対応する。まず、埋め込み節「(メールが) 理研の人から来た」の実行可能性を調べると、「来た」が電子メール管理アプリケーションのコマンドでないため、この埋め込み節は実行可能でない。そこで、換言規則を用いて「メールの差出人が理研の人と等しい」と換言し、再度実行可能性を調べるが、「人」がコマンドでないため、これも実行可能でない。さらに、換言規則と文 (2) から得られる事実的知識を用いて「メールの差出人アドレスの末尾に『.riken.jp』が付く」と換言する。「差出人アドレス」と「末尾に付く」はコマンドであり、「メール」はコマンドではないが、その母集合が「ユーザの受信メールボックス内のメール」であると判断できるので、全体として実行可能となる。主節に

表 3: 文 (1) の正規化の結果

条件部分	メールの差出人アドレスの末尾に「.riken.jp」が付く
操作部分	メールを「理研」フォルダに入れる
ループ範囲	受信メールボックス
ループ要素	メール

対しても、同様に処理を行う。正規化の結果を表 3 に示す。

3.4 実行可能プログラムの生成

次に、入力プログラムの修辭構造を考慮しながら、各文の正規化の結果を統合し、実行可能な形をしたプログラムテキストを生成する。この統合の処理は、修辭関係の種類に応じた統合のための規則を、修辭構造の末端（つまり、文）からルートに向かってボトムアップ的に適用していくことにより行われる。統合規則の例をいくつか挙げる。

- もし 2 つのテキスト範囲が “conjunction” または “sequence” の関係にあるならば、これらのテキスト範囲に対応する実行可能プログラムを「 P そして Q 」という形につなげたテキストを作る。
- もし 2 つのテキスト範囲が “otherwise” の関係にあるならば、その主要素および従属要素に対応する条件部分、操作部分をそれぞれ C_1, A_1 および C_2, A_2 とすると、「もし C_2 ならば A_2 , そうでなければ、もし C_1 ならば A_1 」という形のテキストを作る。
- もしこのノードが、同じループ範囲 R , ループ要素 E を持つすべての文を含む最も下位のノードならば、生成したテキストの先頭に「 R の中のすべての E に対して次の処理を繰り返す」という形のテキストを追加する。

図 1 のプログラムの処理結果として生成される実行可能プログラムを図 5 に示す。

3.5 実行可能プログラムの実行

生成された実行可能プログラムは、次の 2 つの機能を利用してアプリケーションの操作命令に逐次変換、実行される。

コマンド実行機能 実行可能プログラムに含まれる操作、条件、モノを表わす表現から、それに対応する言語アプリケーションのコマンドを呼び出して、ソフトウェアの操作を実行する。

実行制御機能 次に挙げるようなコマンドの実行順序の制御を指定する表現を解釈して、指定された順序でコマンドの実行を行う。「 P そして Q 」、「もし C ならば A , そうでなければ P 」、「 R の中のすべての E に対して次の処理を繰り返す。 P 」。

4. 議論

4.1 評価実験

本論文で提案した処理モデルが日常用いられる言語表現にどの程度対応しているか評価するために、次のような実験を行った。我々は、6 人の被験者に対して、図 1 に示したような電子メール管理タスクを特徴付けるような見本の電子メールとそれに対する操作の組のリストのみを渡して、このタスク

受信メールボックスの中のすべてのメールに対して、次の処理を繰り返す。
 もし、そのメールの差出人が伊藤さんと等しい、または...、かつそのメールがセミオティックベースに関するならば、次の処理を行う。
 そのメールを「SB」フォルダに入れる。
 もし、添付ファイルがあるならば、その添付ファイルをマイドキュメントの「SB」フォルダに保存する。
 もし、添付ファイルが文書ファイルであるならば、その添付ファイルを表示する。
 以上。
 そうでなければ、もし、そのメールの差出人が菅野先生と等しい、または...ならば、次の処理を行う。
 そのメールを「研究室」フォルダに入れる。
 そのメールを自宅に転送する。
 以上。
 そうでなければ、もし、そのメールの差出人アドレスの末尾に「.riken.jp」が付くならば、そのメールを「理研」フォルダに入れる。
 以上。

図 5: 生成された実行可能プログラム

の内容を自然言語で記述するように求めた。この実験で収集したテキストを分析した結果、本論文で説明したアルゴリズムによって、そのほとんどを正しく処理できることが分かった。例えば、図 1 のテキストに含まれる 2 つの “otherwise” の関係が、被験者の回答の中でどのように表わされたか調べてみると、次の 3 種類の表現方法が観察された。

1. 「ただし」などの接続表現を使う。
2. 文の順序により暗黙的に示す。すなわち、一般的な場合を先に述べて、次に例外的な場合について述べる。
3. “conjunction” の関係を、明示的な条件の指定（例えば、「研究室の人以外の理研の人」とともに用いる）。

このうち 1 と 2 は、我々のアルゴリズムで扱うことができる。また 3 も、明示的な否定の扱いを導入することにより、容易に対処することができる。

4.2 対話機能

ユーザによって入力された日常言語プログラムは、しばしば必要な情報を欠き曖昧であったり、ユーザの意図を正しく表現していないことがある。また、システムの側の解析能力にも限界があるため、利便性の観点からユーザとシステムの間で対話を行う機能は不可欠である。我々の日常言語プログラミングシステムでは、次の 4 種類の対話機能が実装されている。

1. システムの処理結果がユーザの意図に合ったものであることを確認するために、処理の各段階での結果をテキストの形で出力する。もし問題が判明した場合は、ユーザはプログラムを書き換えて再入力する。
2. 必須スロットの値が特定できないためプログラムの正規化に失敗した場合、その値をユーザに問い合わせる応答を生成する。
3. 現在作成中の、または前回作成したプログラムの内容に関するユーザからの質問を受け付ける。
4. 現在作成中の、または前回作成したプログラムに対するユーザからの内容変更の要求を受け付けて、その要求を満たすようにプログラムの内容に最小限の変更を行う。

5. まとめ

本論文では、日常言語で書かれた複数の文からなるタスク指示を理解して、応用ソフトウェアの操作を実行する手法を提案した。本手法は、ソフトウェア機能に対する間接的な言及や暗

黙的なループ構造、および修辭構造を扱うことができるため、一般のユーザによる広範な入力に対応できると考えられる。現在は電子メール管理を対象ドメインとしているが、テキスト理解に用いているセミオティックベースが状況によって分類された資源を持ち、状況に応じた処理が可能であることと、その他に利用しているコマンド知識や換言規則がドメインごとに独立して用意できることから、他のドメインにも容易に適用できると考えている。なお、本論文で述べたアルゴリズムはすべて実装されており、電子メール管理に関するユーザの指示を理解して Microsoft Outlook の操作を実行する。

参考文献

- [Biermann 83] Biermann, A. W., Ballard, B. W., and Sigmmon, A. H.: An Experimental Study of Natural Language Programming, *International Journal of Man-Machine Studies*, Vol. 18, No. 1, pp. 71-87 (1983)
- [原田 86] 原田 実, 篠原 靖志: 部品合成によるプログラム自動生成システム ARIES/I, *情報処理学会論文誌*, Vol. 27, No. 4, pp. 417-424 (1986)
- [伊藤 03] 伊藤 紀子, 杉本 徹, 高橋 祐介, 小林 一郎: セミオティックベースを使ったテキスト処理アルゴリズム, 第 17 回人工知能学会全国大会, 新潟 (2003), 3B1-01
- [岩爪 03] 岩爪 道昭, 小林 一郎, 杉本 徹, 岩下 志乃, 高橋 祐介, 伊藤 紀子, 菅野 道夫: 日常言語コンピューティング (第 2 報) — 日常言語に基づく計算機資源の管理・実行環境を目指して —, *人工知能学会論文誌*, Vol. 18, No. 1, pp. 45-56 (2003)
- [Mann 88] Mann, W. C. and Thompson, S. A.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *Text*, Vol. 8, No. 3, pp. 243-281 (1988)
- [Marcu 00] Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press (2000)
- [Pane 01] Pane, J. F., Ratanamahatana, C. A., and Myers, B. A.: Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems, *International Journal of Human-Computer Studies*, Vol. 54, No. 2, pp. 237-264 (2001)
- [新森 04] 新森 昭宏, 奥村 学, 丸川 雄三, 岩山 真: 手がかり句を用いた特許請求項の構造解析, *情報処理学会論文誌*, Vol. 45, No. 3, pp. 891-905 (2004)