

自然言語入力に対する知的コマンドインターフェイスICIにおける入力文の意図理解

2F-4

美馬 秀樹, 青江 順一
徳島大学工学部

1. まえがき

近年の情報システムの高度化とも相まって誰もが容易に使える情報システムが望まれているにもかかわらず、利用環境が向上しているとは決して言い難いのが現状である。そこで、本報告の目的は、コマンド機能に関するユーザーインターフェイスを改善することであり、自然言語入力に対してユーザの意図を理解し、情報システムの実状況に適合したコマンドを自動実行させる知的コマンドインターフェイス(ICI, Intelligent Command Interface)の実現を目標とする。

2. システムの概要

- ICIは図1のような解析部と知識ベースで構成される。
- 1)形態素解析部は、語彙知識辞書により、入力文の分類、語彙の属性、語彙とコマンド概念の対応を決定する。
 - 2)構文・意味解析部は、ユニフィケーション文法に基づく拡張LRパーサで実現され、1)から送られる属性構造に基づいて解析を行う。
 - 3)実状況知識獲得部は、システムの現在の状況の理解と、入力文の名詞句の実体が情報システムに存在するの否かをチェックし、その属性と属性値を決定して1)、2)から得られた静的知識に、この動的知識を融合する。
 - 4)文脈知識は、入力文が不完全で付加情報を必要とする場合、属性構造を格納してインクリメンタルな増強を行うための一時的な知識である。
 - 5)コマンド理解部は、構文・意味解析結果とコマンド知識を参照して、正しいコマンドの自動生成と実行が可能かどうかを決定するものであり、それが不可能な場合は、ユーザに付加情報の入力を催促する応答文を生成する。
 - 6)コマンド知識は、コマンドが実行できる実状況の制約条件を保有し、コマンド生成と応答文の情報を与える。さらにコマンド推論における、発話意図に関するコマンドの関連情報を与える。
 - 7)文生成知識は、5)の応答文を生成する文生成のパターンであり、属性構造の不足情報や制約外の属性値について回答し、新しい情報の入力を要求する。

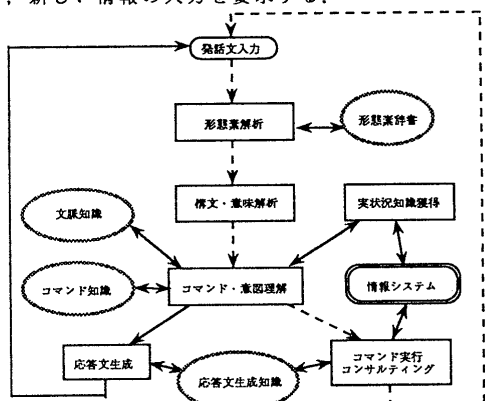


図1 ICIの構成

3. 実状況の理解

ユーザが要求するコマンド機能を理解・実行するためには、入力文の名詞句の、対象世界(情報システム)での指示実体(referent)を同定・理解し、システムの状況に依存したコマンドを実行する必要がある。

Understanding Demands and Intentions in the ICI
Intelligent Command Interface using Natural language
inputs, Hideki MIMA, Jun-ichi AOE, Tokushima University

このような名詞句の実体とシステムの状況を理解するために、情報システムに現れうるすべての事物の集合(UNIVERSE)を対象とし、そのサブセットである個々の対象世界を $DOMAIN(i, j)$ (i :対象世界の実体名, j :対象世界の属性と定義する。そして、名詞句と対象世界における指示照応関係(reference)を吉浦ら^[3]の分類(表1)を利用し、I, AT, B, NA, Nを考慮した対話構造を考える。

表1 指示照応関係の分類

分類	内容	例
Identifiable(I) in DOMAIN(j)	聞き手がDOMAIN(j)に実体を同定しうる	△△を削除したい △△がDOMAIN(#current_directory) カレント・ディレクトリに唯一つある
Anaphoric Total(AT) in DOMAIN(j)	候補が複数あり、実体はDOMAIN(j)内の全てである	△△ファイルを全て削除する
Erroneous(E) in DOMAIN(j)	話し手は、聞き手が実体を同定できると考えているが聞き手は同定できない	△△を消したい △△がDOMAIN(#current_directory) に見つからない
Nonspecific(N) in UNIVERSE	実体は、UNIVERSE(全対象世界)に存在しない。実体が不特定である。	△△ファイルを削除する(△△ファイルがない) △△ファイルの削除方法(△△ファイル名が示されていない)

4. コマンド知識

本システムでは、知識の表現形式として、UC^[1]のKODIAK(Keystone to Overall Design for Integration and Application of Knowledge)を利用する。

コマンド知識とは、正しいコマンドを生成するためにコマンド理解部から参照される知識ベースで、その機能的な概念により分類され、意味解析部で得られるコマンド機能概念により選択される。そして、コマンド理解部から送られるコマンド、引き数等の情報を用いて正確なコマンドを生成する。また、引き数に不足情報や誤った情報があればその情報をコマンド理解部へ送る。さらにICIでは、正しいコマンド実行に導く補足入力を催促するために、実行のためのreferenceも含めた制約条件を保有する知識が必要であるので、実状況知識をも考慮した知識表現を構築する。

5. 意図理解

ユーザーとシステムとの対話において、ユーザーの真に意図する内容が表層文中に陽に表現されていない場合、もはや表層的な意味解析だけではユーザーの意図の決定的な解釈は不可能となる。例えば、

「他人に読まれたくないファイルがある。」
のような発話では、その意図するコマンドは表層文中には示されてはならず、

「ファイルが消えている。」
のような表現では、表層の主動詞「消す」とは全く逆の意図である「消したくなかった」が発話者の真の意図であると思われる。このように、発話に表れる言語表現に発話者の意図がすべて明示的に示されているとは決して言い難い。

そのような問題に対処するために、文中に表れるテンス・アスペクトおよび、ムード表現^[4,5]に注目し、質問応答における一般的な発話文約300例について、特に意図抽出に関係があると思われるテンス・アスペクト・ムード表現の抽出の結果、それらの表現に対して、表2のように発話文の表層に現れない真の意図が考えられる。

表2 意図に関する表現と意図リンク

Tense/Aspect/Mood	Intention	Intention Link
ている, た, ていた	表層のアクションに対して意図はその逆になる	contrast
てしまった		
てしまう	意図は表層のアクションに対する拒絶である	refusal
たくない, ほしくない		
な	意図は表層のアクションに対する苦情である	complaint
できない		
ない		
なかった		

しかし、文中に現れる全てのテンス・アスペクト表現が重要な意図を持つわけではない。例えば、「保存しているファイルの内容を見たい」のような文では「ている」は静態を表し、相対テンスは前なのでコマンドの推論にはそれほど重要ではない。しかし、状況によっては動態未完了になり、その場合は特定のファイルを指すので、実状況の取得が必要となる。また、「行番号が付いたファイルを印刷したい」でも2つの解釈が可能になる。つまり、行番号を付けるのが発話時点に対して前ならば、特定のファイルのことを示しており、後ならば行番号を付けて印刷するという意味になる。このような発話文中の意図に関する表現の有効性を適切に判断するためには、個々の動詞の特性により焦点との相対的な時間関係を理解しなければならず、また、実際のシステムの状況による判断も必要となる。このような、テンス・アスペクトの表現に基づく時制・焦点の解析と、ムードによる心的態度の付加によるコマンド意図の解析結果、さらに、コマンドに対する要求の動作特性により、各コマンド間に存在する関連性を発話者のコマンドに対する意図によるもので定義し、それらを意図リンクによって表現したコマンド知識を提案する(図2)。さらに、その知識表現から発話者の真に意図するコマンドを推論する手法を述べる。

6. コマンド推論

まず、ユーザーの発話文より、テンス・アスペクト・ムード表現を抽出する。そして、上記のような意図解析により表2より決定した意図情報を付加する。さらに、発話文の構文・意味解析における、ユニフィケーション文法による属性構造とにより、意図するコマンドを次のようなアルゴリズム

- アルゴリズム
- ステップ1:ユーザーの発話文 s に対して、EFFECT_VALUE(s)の値が一つに同定されるよう、ユーザーとの対話によって絞り込む。ここで、EFFECT_VALUE(s)の値を x とする。
 - ステップ2:入力文 s に対する意図を表2より決定し、意図解析により意図リンク INTENTION(s)を y とする。
 - ステップ3: y が null の場合は、ステップ6へ進みコマンドの決定を行う。そうでない場合は次のステップへ進む。
 - ステップ4: x より上位概念リンク(Dリンク)を辿り、 y の値を持つリンク(意図リンク)があれば遷移する。 y のリンク先の EFFECT を z とする。
 - ステップ5: z よりさらにDリンクを逆に辿り(下位概念)入力文 s に対する f-構造の情報を使得って新しいEFFECTに遷移する。その値を w とする。
 - ステップ6: w よりeffectリンクを逆遷移しEXECUTE-COMMAND概念へと遷移する。この時点でコマンドの推論は一応完了するが、format等の知識と実状況との関連によりステップ5へのバックトラックまたは、ユーザーとの対話処理が起こり得る。

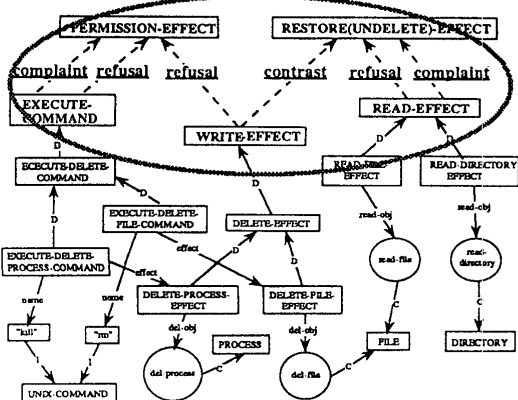


図2 コマンド知識における意図リンクの例

以下に「ファイルを消されたくない。」を例にし、コマンド推論のプロセスを上記のアルゴリズムに基づいて詳述する。

- 入力: $s =$ "ファイルを消されたくない。"
- ステップ1: $x =$ EFFECT_VALUE(s) より、 $x =$ DELETE-FILE-EFFECT を得る。
- ステップ2: $y =$ INTENTION(s) より意図リンク refusal を得る。ステップ3: $y \neq$ null より次のステップへ。
- ステップ4: x よりDリンクを辿り、WRITE-EFFECTで意図リンク refusal と y がマッチングする。よって、その遷移を辿り $z =$ PERMISSION-EFFECT を得る。
- ステップ5: z より再びDリンクを逆に辿り、入力文 s に対する f-構造の obj-FILEよりCHMOD-EFFECTに遷移する(図3参照)。よって $w =$ CHMOD-EFFECT となる。
- ステップ6: w より effectリンクを逆遷移、意図するコマンドを同定する。この時点よりFORMAT情報などを用い、コマンド"chmod"の実行に満足すべくユーザーとの対話-コマンド作成処理を行う。

「ファイル(プロセス)が消えている」などの、結果残存の表現でも意図リンク"contrast"により同様のプロセスで意図したコマンドを推論することができる。

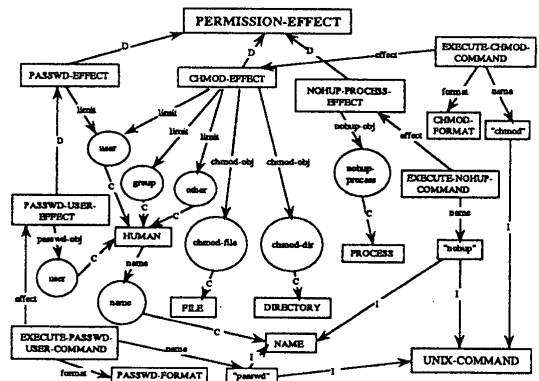


図3 コマンド知識の例

7. 現在のシステムと今後の課題

UNIXシステムを対象に、実際の使用環境に基づいてテストを行った結果、初心者には、コマンド推論に対して不要な語がより多く見られる傾向があるが、対話によるものも含め入力文の約81%(内16%は意図解析による)に適切な出力が得られた。

- また、今後の課題としては、
- (1)コマンドのオプションの処理。
 - (2)コマンド間の因果関係による連続コマンド実行への拡張。例えば、モードが原因でコピーできない場合は、モード変更コマンドを実行する。
 - (3)ユーザーレベルの認識とレベルに応じた応答の出力。などきりが無いが、実用化を目指すには可能な限り多くのコマンドを実行できる環境を作り上げ、試験運用により地道に拡張していくことである。

文 献

[1]R. Wilensky and D. N. Chin, "The berkeley UNIX consultant project", Computational Linguistics,14, 4,pp.35-84, Dec. 1988.
 [2]山岡, 飯田, "文脈を考慮した音声認識結果絞り込み法", 情処研資自然言語処理, 78-16, 1990.
 [3]吉浦, 片山, 中西, 平沢, "日本語質問応答システムにおける質問の曖昧性を解消する意味解析方式", 情処論, 27,3, pp.321-329, 1986.
 [4]情報処理進事業協会技術センター, "計算機用日本語基本動詞辞書 I P A L (Basic Verbs)-解説編-", 昭和62年2月
 [5]水谷,石綿,荻野,賀来,草薙, "文法と意味", 朝倉書店