

「パーソナルエージェント用自然言語インターフェース」 開発 ツールキット (PANLI toolkit)

定政 邦彦[†] 牧野 貴樹[†] 光石 豊[†] 鳥澤 健太郎[†] 松田 晃一[§] 辻井 潤一[‡]

[†]東京大学理学部 [‡]東京大学大学院理学系研究科

[§]ソニー株式会社 PSD センター UI 開発部

1 はじめに

本研究では、仮想世界におけるユーザーの行動を支援するパーソナルエージェント用の自然言語インターフェースを簡潔に記述するためのツールキット、PANLI toolkit (Personal Agent Natural Language Interface Toolkit) を提案する。近年のインターネットの発達とともに、インターネット上でチャットを楽しむ人々が多くなってきた。チャットの機会を提供する場の一つとして、インターネット上に構築された仮想世界が挙げられる。ユーザーは仮想世界内を自由に歩き回ることができ、他のユーザーとチャットを楽しむことができる。このような仮想世界の一つとして PAW[1, 2] (図 1) が知られている。PAW では、各ユーザーがログインした際に、パーソナルエージェント (以下 PA)¹ が与えられる。PA は、ユーザーの発言に反応して、何らかのアクション²を返す。我々の研究の最終的な目的は、PA の言語機能を司る自然言語インターフェースを強化することによって、仮想世界におけるユーザーと PA 間のチャットを円滑かつ豊かにすることである。本研究で提案するのは、そのための PA の自然言語インターフェースを簡単に拡張できるようなツールである。

本ツールの特徴は、必ずしも自然言語処理に詳しくない PAW のユーザー自身が PANLI 開発者となれるよう、PA の対話規則を簡単に記述できる枠組を用意したことである。具体的には、対話規則は、単語の意味クラスを表現できる変数を利用する他は通常の文と同じである表現 (本論文では「変数混じり文」と呼ぶ) で記述される。PA 内部では、日本語に特有の自由な語順に対処するため、入力文をパズルして得られる構文構造により処理を行うが、対話規則は直接構文構造では記述しない。代わりに、変数混じり文で記述された対話規則を前もって構文構造に変換して解析に使用する。こうすることで PAW のユーザーは、構文解析に関する特別な知識

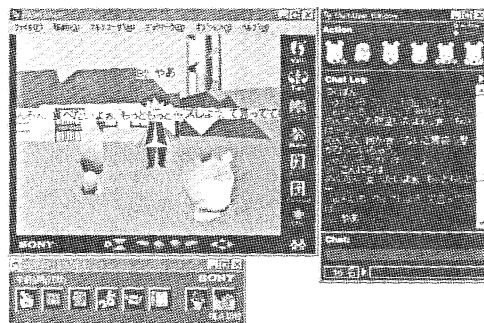


図 1: PAW³

がなくても、普通の文を書くような感覚で簡単に PA の自然言語インターフェースを開発できると考えられる。また、ユーザー同士で PA の賢さを競わせるといった過程で多様な対話規則を得ることもできると考えられる。

将来的には、我々はこのツールを拡張して、ユーザーの興味や意図を聞き出して適当な返答をしたり適切な行動をしたりするタスクを実行する PA を構築することを考えている。ユーザーの興味や意図を聞き出すことを利用する具体的なタスクとしては、ユーザーと共通の嗜好・趣味を持つ他のユーザーを探して対面させる、従来は口コミでしか伝わらなかったような「美味しいレストラン」の情報を聞き出す、ユーザーの好みそうなウェブページを WWW から探してくる、ユーザーをマーケットリサーチの対象とする、といった例が挙げられる。その際、例えば、ユーザーの発言中の単語の文字列をそのままコピーして他のユーザーに話しかける、というような単純な処理でも、ある程度使用に耐えうると我々は考えている。従って、現在の実装のような単純な機構からでもサービスを開始することができるであろう。本研究では以上のようなタスクを行う PA のプロトタイプとして、好き嫌い学習と伝言を行う PA を実装した (4 節)。

¹PAW では、PA は犬の形をした仮想動物であり、PAW のユーザーにとって仮想的なペットである。

²現在は、お手をやる、しりとりゲームをする、などの単純な機能のみとなっている。

³©Copyright 1997-98 Sony Creative Products Inc.
©Copyright 1998 Sony Corporation.

PA への入力チャット文であるが、チャット文を扱うことの利点は、チャット文が話し言葉に近いため、ユーザーが自分の興味や意図をより多く発言する可能性があるということである。また、解析に曖昧な点・不明な点があったときに、聞き返しを行うことによって多くの情報をユーザーから引き出すという方法が考えられるが、この聞き返しも、チャット文を使用することにより、スムーズに行える可能性がある。

言語处理的な観点から見ると、チャット文の特徴として以下の3点が挙げられる。

- 新聞の文のような定型的な言い回しが少ないため、1つの事柄に対する表現の幅が広がっていく傾向がある(例えば、「好きだ」に対して、好き!、愛してる、マイブーム、など)。
- 多くの文は比較的短く、文構造が単純である。
- 「ぞーさん」(「ぞうさん」の変形)、「ちょーっと待つてっ」(「ちょっと待つて」の変形)等、形態素が変形された表現が多用される。

第一の特徴である表現の広さについては、言い換え規則を導入すること、およびすでに述べたように簡潔な規則記述の枠組みを提供し、多くのPAWのユーザーに規則を記述してもらうことにより解決する。第二の特徴により、比較的単純な機構のパパーでもある程度処理が可能である。さらに、解析が部分的にしか成功しない場合も、すでに述べた聞き返しによってある程度解決することができると思われる。第三の特徴については、チャットに特化した形態素解析⁴[3]を用いることで解決する。

2 規則の記述

現在のプロトタイプのパは、入力文を形態素解析し、構文解析(3節)を行い、言い換え規則によって構文木を変換し、結果を応答規則と比較して対応する結果を返す(図2参照)。この処理の中で、PANLI開発者が記述する必要があるのは、言い換え規則、応答規則、応答規則中の関数から参照されるフレームの3つ(2.2節)である。これら3つをあわせて対話規則と呼ぶ。これらは変数混じり文(2.1節)を用いて記述される。

2.1 変数混じり文

プロトタイプの各規則は、変数混じり文を用いて記述することができる。変数混じり文とは“_MHは_Mが好きだ”のように、通常の文の中に意味クラスを持った変

⁴現在は使用していないが近いうちに導入する予定である。

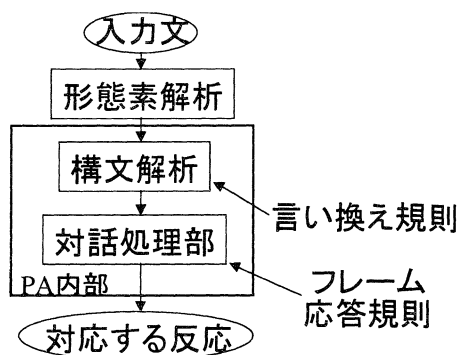


図2: PAの処理の流れ

_M	名詞とマッチする
_MH	「人」の意味をもった名詞とマッチする
_MB	「場所」の意味をもった名詞とマッチする
_MZ	「時間」の意味をもった名詞とマッチする
Xする	用言とマッチする
END	文末表現であることを示す
_S	全ての表現とマッチする

図3: 意味カテゴリ変数一覧

数(意味カテゴリ変数)が含まれる形式をいう。意味カテゴリ変数は自分と同じ意味クラスの単語とのみマッチする。以上により、規則適用の汎用性は残したまま、自然言語処理に詳しくない人間も対話規則を記述できるようにした。現在は図3に示す意味カテゴリ変数が使用できる。

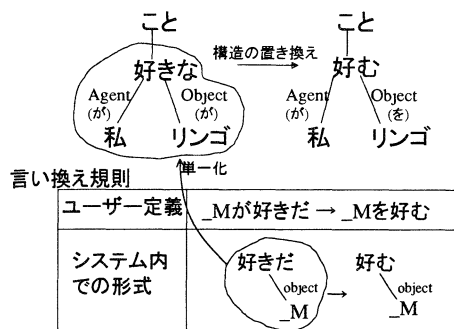
2.2 対話規則

以下ではプロトタイプでの対話規則記述の枠組みについて述べる。

言い換え規則 図4は言い換え規則の例である。一つの言い換え規則は、「言い換え元」と「言い換え先」の2つによって定義される。言い換え規則は変数混じり文によって記述されるが、システム内で前もってパーズされ、構文木の形で保持される。入力文の構文木中に言い換え元の構文木が含まれるならば、その部分を言い換え先の構文木によって置き換える。また、言い換え元と言い換え先の両方に同じ意味カテゴリ変数が出てくる場合には、内容がコピーされる(図5参照)。言い換え規則によって等しい意味を持つ表現の数が絞られるので、応答規則を記述する労力が軽減される。

_Xして下さる -> _Xする
 _Xします -> _Xする
 _Xして _END -> _Xしろ
 _Xするのは _MB でだ -> _MB で _X する
 // 左から右に言い換えが行われる

図 4: 言い換え規則の例



言い換え元「Mが好きだ」がパーズされた入力文の一部と単一化され、言い換えが起こる

図 5: 言い換え規則適用の様子

応答規則 一つの応答規則は入力文と比較する表現 (入力表現) とそれに対する返答の組で定義される。図 6 は応答規則の例である。入力表現と返答は変数混じり文で与える。また返答としては、文字列による返答と外部関数呼び出しをサポートしている。外部関数は PANLI toolkit で提供する JAVA のクラス Action のサブクラスとして定義され、外部関数を記述しさえすれば、WWW から情報を拾ってくる等、複雑な処理への応用も可能である。また、応答規則においても言い換え規則の所で述べた意味カテゴリ変数を用いることができる。さらに、応答規則は簡単なオートマトン機能として、各応答規則に対して元の状態と遷移後の状態を定義することができ、元の状態とシステムの状態が一致するときのみ規則が選択されるようにできる。

```

# suki1 suki1
  _MH は _M が好きだ // 入力表現
  わーい、_M、_M、 // 返答
// システムの状態が suki1 であるときのみ適用
#
  _MB に連れて行って
  call: navi(_MB)
// 外部関数 navi を引数 _MB で呼び出す
入力文と単一化した入力表現に対応する返答を返す
  
```

図 6: 応答規則の例

```

frame event(place, time)
slot place(_MB)
_MB で _X する
mod: _MB で _X する _M
slot time(_MZ)
_MZ に _X する
end
  
```

図 7: フレームの例

フレーム 名詞の持つ付加的な情報を拾うために定義する。第 4 節の実行例では、場所、時間のスロットを持った event というフレームを利用している (図 7 参照)。

各規則について、適用できる規則が複数ある場合は、最も限定性の高い規則を選択する。

3 構文解析

プロトタイプでは、単語 1 つ 1 つにクラスオブジェクトを割り当てて行うオブジェクト指向パーズングを採用している。パーズを制御するために外部から与える情報としては、品詞情報、簡単な名詞の意味クラス、そして少数の格フレームのみを採用し、パーズアルゴリズムとしては (1) 語の修飾先が 2 通り以上ある場合には一番近い単語を選ぶ、動詞の格フレームが埋まっているときのみ外側の単語を選ぶ、(2) チャット文で頻繁に起こる助詞落ち、副助詞による格の曖昧さは、動詞の格フレーム情報を用いて、空いている格を順番に埋めていく、という単純なヒューリスティクスを採用している。しかしこれらは暫定的な処置で、将来的には汎用性の高い文法 [8] を使用したパーザーに置き換える予定である。

4 PA の対話例

PA のタスクの実行例を示す。これらは PANLI toolkit のプロトタイプを用いて記述された対話規則を元に動作している。ここでは、ユーザーの好みを学習して、後のチャット時に茶々入れする例 (例 1)、他のユーザーに伝言を頼み、欠落した情報を PA に拾わせる例 (例 2,3) を取り上げた。

例 1 「好き」を表す単語群「好き、好きだ、大好き、愛している、マイブーム」などを言い換え規則で置き換え。

USER: 僕、リンゴって好きだな。

PA: ふんふん

(数日後)

USER: 昨日リンゴが送られてきてさあ。

PA: ご主人さまはリンゴがすきだっぴよね

例2 フレームを用いた聞き返しの例

USER: 伝えたいことがあるんだけど。

PA: 誰に伝言するの?

USER: 佐藤さんに伝言して

PA: 佐藤しゅんに伝言するのね。伝言する
メッセージを教えて♪

USER: 今日、野球の試合をしようぜ。

PA: 野球の試合ってイベント?

USER: そう

PA: 野球の試合の場所は?

USER: 近くの公園で

PA: 「今日、野球の試合をしようぜ」だっ
てっぽ。これって場所は近くの公園だっ
てっぽ♪

例3 上記とは欠けているフレームが異なる

USER: 鈴木さんに伝言がしたいな。

PA: 鈴木しゅんに伝言するのね。伝言する
メッセージを教えて♪

USER: 渋谷であったコンサート、面白かつ
たよ。

PA: 渋谷であったコンサートってイベント?

USER: そうだよ。

PA: それって、いつの話?

USER: 昨日の話

PA: 「渋谷であったコンサート、面白かつた
よ。」だってっぽ。これって昨日の話
だってっぽ♪

5 結論

本研究ではパーソナルエージェントの自然言語インターフェースを簡潔に記述するためのソフト、PANLI toolkit を提案した。その特徴は、全ての規則を、普通の文に近い変数混じり文で記述できることで、それによって自然言語処理に詳しくない人間でも PANLI のための規則を記述できるようにした。この形式は正規表現よりも簡潔であるため、より多くの人がルール記述に参加できるようになり、今まで多大な労力を必要とした対話規則記述の一人当たりの労力削減につながると思われる。また、以上の特徴を生かしてプロトタイプを作成し、その上でユーザーの好き嫌い学習、伝言機能を実装した。今後は、パーザーを改良して、より広いカヴァレッジ・より高い精度で文章を解析できるようにし、より複雑なタスクを実装していく予定である。また、多くの人によって将来大量に記述されるであろう対話規則は、無作為に集められたコーパスに比べ、動詞の格構造、使われる文脈が容易に分かるので、将来的にはそれらの対話規則を文法のチューニングに役立てるという可能性も探求する予定である。

参考文献

- [1] SONY “Personal Agent World” <http://www.son-net.ne.jp/paw/index.html>
- [2] 松田 晃一 “不思議な島をペットと歩こう！ インターネット上の共有仮想世界 PAW”, *bit*, Vol. 30, No. 9, pp. 2-10, 1998
- [3] 風間 淳一, 光石 豊, 牧野 貴樹, 鳥澤 健太郎, 松田 晃一, 辻井 潤一 “チャットのための日本語形態素解析”, 言語処理学会第 5 回年次大会, 1999
- [4] 大澤 一郎, 米澤 明憲, “オブジェクト指向方式による対話システム”, コンピュータソフトウェア, Vol. 2, No. 1, pp. 11-28, 1984
- [5] 大澤 一郎, 米澤 明憲, “対話システム” *NLC86-3*, 東京工業大学, 1986
- [6] 渡部 卓雄, 大澤 一郎, and 米澤 明憲, “対話領域の独立性を指向した日本語対話理解” 人工知能学会誌, Vol. 2, pp. 70-77, June 1987
- [7] 寺村 秀夫, “日本語のシンタクスと意味” 1982
- [8] Y. Mitsuishi, K. Torisawa and J. Tsujii, “HPSG-Style Underspecified Japanese Grammar with Wide Coverage”, In *COLING-ACL '98*, pp. 876-880, 1998