

統計的手法による 音声対話制御

基
般

南 泰浩 (NTT コミュニケーション科学基礎研究所)

統計的手法による音声対話制御とは

統計的手法による音声対話制御は、大量の学習データから音声対話制御機構を自動的に学習する制御手法である。しかし、このタイトルで執筆を始めて、はたと考えてしまった。実は、現状の音声対話システムにおいては、ルールやオートマトンの状態遷移を人手で記述する対話制御手法が主流であり、統計的な対話制御手法は、データ数が十分ではない点、目的関数が明確でない点から、研究者の間でも、その方向性さえコンセンサスがとれていない。このことに気が付いたからである。したがって、本稿のタイトルは、筆者がさまざまな音声対話システムを作ってきた経験に基づき、これからの対話制御手法の目指す方向性を考慮し、付けたものに過ぎない。ただ、筆者らの研究の方向性を叙述することが、読者の研究方針の参考にはなるだろうと考え執筆することにした。

筆者が目指す方向性とは、音声での対話を通して、ユーザをサポートし、かつ、ユーザが対話そのものを楽しむ対話制御である。これまで、電話番号案内システム、天気予報案内システム、新聞記事からの質問応答システム、クイズ対話システムなどの複数の対話システムを作成してきたが、これらほとんどが、一問一答型を拡張した対話制御であり、対話を楽しむということからはほど遠い感じがした。筆者は、この原因が、主に、(1) 複数知識源を同時に適切に扱えない、(2) 認識誤りが生じた場合に適切に

対話制御ができない、(3) 対象タスク以外の自然な対話（非タスク達成型対話）を適切に制御できないことによると考えている。本稿ではこれらの点に焦点を当てる。ここで、(2)、(3)の問題は、統計的手法による対話制御の対象である一方、(1)はプロダクションシステムやオントロジーなどのAI分野で古くから議論されている複数の知識源をどう扱うかという問題である。したがって(1)は統計的手法の対象とはいえないかもしれない。しかし、現状解決できていない課題であり、かつ、対話システム全体を概観する上で重要であると考えたため、本稿では取り上げることにした。

以上で挙げた3つの点について個々に説明する。

■ 複数知識源への対応

筆者らが複数の対話システムを作成していて、問題と感じたことは、それらの対話システムのいくつかを組み合わせて1つのシステムを作成することの困難さである。一般的に、対話システムを構成する際に用いる知識源は、別々の独立に開発されたプログラムから作成されることが多い。このため、知識源ごとにそのデータ構造およびインタフェースが大きく異なる。対話制御系において、この違いを吸収するためのプログラムを書くことは根気強い作業を必要とする。このため知識源をどのように統一するかが大きな問題である。

■ 認識誤りに対して適切な対話制御

現状で、音声認識の認識誤りを0にすることはできない。認識誤りの精度の違いは、音声対話の制御に大きな影響を及ぼす。たとえば、10項目の中から1つを選ぶような簡単なタスクの対話では、音声認識率が低い場合（たとえば80%）、確認の対話を入れる必要があるが、認識率が高い場合（たとえば98%）、確認の対話を入れる必要はないであろう。このような認識率の違いによる最適な対話制御を人手で作成するのは、莫大な労力を必要とする。これを自動的に設定するための手法が必要である。

■ 非タスク達成型対話の制御

ユーザがコンピュータに接するときには、何らかの意図を持っている。たとえば、天気予報を聞きたい、“対話制御”という言葉の関連情報を検索したいなどである。音声対話も含めて、今まで利用されてきたコンピュータによる人間とのインタラクション手法では、このようなユーザの意図を達成させる“タスク達成型”の制御を行うことがほとんどであった。ところが、最新の音声対話システムによるインタラクションでは、この様相が異なる。このことは、音声対話システムに関する最近の雑誌の記事のほとんどが、情報検索などのタスクを達成する対話の紹介だけでなく、システムが対応可能な雑談対話も紹介していることから分かる。たとえば、“Siriは賢いね”と言え、ば、“ほめてもらとうれしいですね”と答えてくれることを面白いと記載する記事がある。このようにユーザが対話そのものを楽しむ対話を、筆者らは“非タスク達成型対話”と呼んでいる。音声対話システムは、コンピュータシステムへの単なる機械的な入力手法でありながら、ユーザは、タスク外の対話そのものを試したくなる。これは、音声対話システムの特徴であろう。このように音声対話システムでは、非タスク達成型の対話への対応が重要である。

次章からは、ここで挙げた3つの点に関する技術について説明していく。

複数知識源の統一化による対話制御

AppleのSiriの原型を作ったSiri社は、CALO (Cognitive Assistant that Learns and Organizes) プロジェクト¹⁾を主導したSRI国際ナショナルから派生したことが知られている。CALOは、アメリカ国防高等研究計画局 (Defense Advanced Research Projects Agency: 通称DARPA) が企画したPAL (the Personalized Assistant that Learns) の2つのプログラムのうちの1つであり、2003年から2008年まで続いた。CALOは、オフィス環境における知的な個人秘書の構築を目指す巨大なプロジェクトである。CALOでは、オフィスでの会議に関する情報も対象としており、会議室の情報として、画像、音声、ノート、スケッチ、ソフトウェアとのインタラクションがすべて記録されている。この会議に関する情報を引き出したり、スケジュールの設定をしたりすることも目標に掲げている。

ここでは、CALOにおける複数の知識源を用いた対話制御に関する技術について述べる。特に、CALOプロジェクト内のシステムである「質問管理 (Query Manager)」と「計画実行補助 PEXA (Project Execution Assistant)」を取り上げる。

■ 質問管理 (Query Manager)^{1), 2)}

Query Managerは、CALOが管理する知識源中の情報に対しユーザからの質問に答えるためのシステムである。特にミーティング環境での質問に答えるシステムを想定している。たとえば、“ある会議に参加している人は誰?”などと質問すると、Query Managerは、会場の画像処理結果などのさまざまな知識の中から、その人物の名前を探してくる。

これを実現するためのアーキテクチャを図-1に示す。ユーザの質問は、KIF (Knowledge Interchange Format) というフォーマットで記述される。これは、たとえば、(*CurrentCaloUser ?user*) のように、述語とそれに付随するいくつかの変数で表現される。ここでの述語は *CurrentCaloUser* であり、変数は *?user* である。ユーザの質問は、この

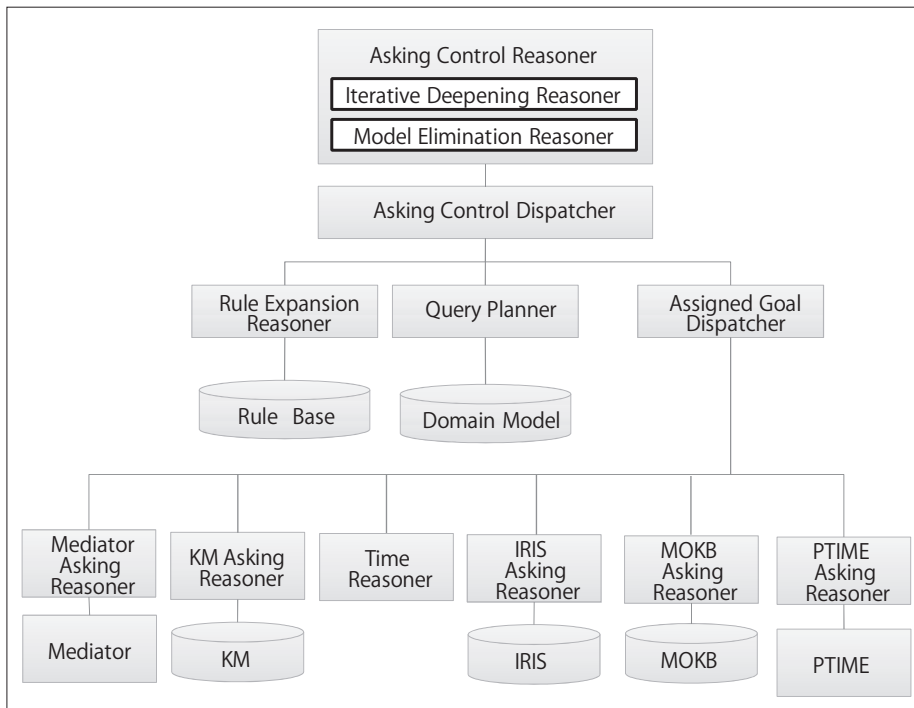


図-1 Query Manager のアーキテクチャ

ような個々の質問を and あるいは or で結び付けた簡単な論理式で表されている。変数がバインドされていないならば（すなわち、未設定ならば）それが質問項目になる。バインドされていれば（設定されていれば）、その変数は条件となる。バインドされていない変数に対し、この条件を満たす解を複数の知識源から探してくること（バインドすること）が、ユーザの質問へ回答することとなる。

CALO では、図-1 に示すように、KM (Knowledge Machine), Time Reasoner, IRIS (Integrate. Relate. Infer. Share.), MOKB (Meeting Ontology Knowledge Base) などの知識源が用意されている。各知識源には Reasoner すなわち推論器が具備されていて、Reasoner は入力された質問の変数をバインドして返す。各知識源は以下のような特徴を持つ。知識源 KM では iCalendar などのオントロジーのうち、人、会社、カレンダー、ミーティング、連絡、スケジュール、タスクと作業に関するものからオントロジーを作成している。IRIS は、オフィス関係の情報オブジェクトの“個人マップ”を作るためのフレームワークである。これは、具体的には、個人の電子メールやカレンダーの面会予約などを階層的な知識として表現するフレームワークである。知識

源 MOKB はマルチモーダルの談話解析結果に基づく会議の情報をオントロジーで表現したものである。Time Reasoner は時刻や時間に関する簡単な演算を実行する（ここでは知識源として扱う）。

以上の知識源のほか、Mediator と PTIME というシステムが用意されている。Mediator はインターネットから情報を取ってくる情報統合システムである。PTIME は会議のスケジュール

管理を行う汎用の推論システムである。これらのシステムにもシステムから情報を探索するための Reasoner が付与されている。すべての Reasoner は、各知識源開発者の共同作業により決定された CALO オントロジーと呼ばれる共通化されたオントロジーを実装するようにしている。ユーザの質問はいくつかの個々の質問に分解でき、個々の質問を順番に Reasoner によって処理することにより、最終的にユーザの質問に答える。

質問が入力されてから回答するまでの操作を図-1 により説明する。最初に、Iterative Deepening Reasoner は、入力された質問に再帰的ルールがある場合に終了を保証する。Model Elimination Reasoner は一階の述語論理で個々の質問が構成されているかをチェックする。これらの処理の後、Rule Expansion Reasoner は、個々の質問に、あらかじめ登録してある置換えルールを適用して、複数の質問を新たに生成する。Query Planner では、これらの個々の質問に回答できるように複数の推論プランを設定する。プランの作成では、個々の質問を複数の知識源に問い合わせることになるため、かなり多くのプラン候補が生成される可能性がある。これに対処するため、Query Planner では、個々の質

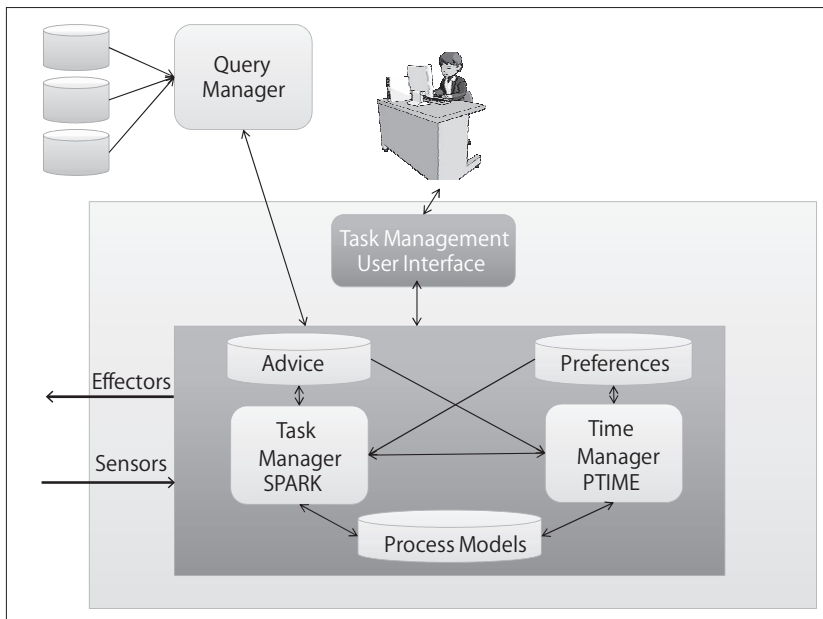


図-2 PExA アーキテクチャ

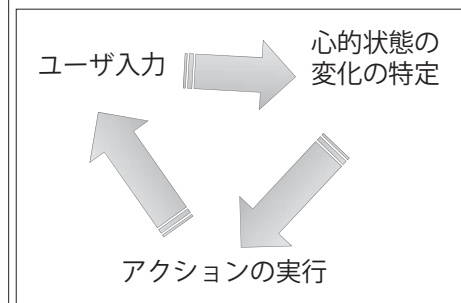


図-3 BDIモデルの実行サイクル

問がどの知識源で答えられるかという知識を利用し、むやみに候補の生成を増加させないようにしている。Assigned Goal Dispatcherは、以上のようにして得られたプランに従って、順番に並べられた個々の質問を各Reasonerに割り振る。これらのすべてのインタフェースは、統一的に扱えるように設計されている。しかし、従来からある知識源とこのインタフェースとの間を完全には埋めることはできなかった。これに関しては、アダプタというコードを書いて対処している^{1), 2)}。以上のようにCALOでは、各知識源の共有化に多くの労力を払っている。一般に、すべての知識源を統一化することは莫大な労力を必要とする。CALOの資料は、読者が音声対話システム作成する上で、どの程度知識源の統一を図れば良いかの指針になると思われる。

■ プロジェクト実行補助 (PExA)^{1), 3)}

前節では、ユーザの質問に答える機構の説明を行った。CALOでは、これとは別に、システム側がユーザに働きかけるアクションを実行するための機構として、PExAというアーキテクチャを用意している。PExAは、Task ManagementとTime Managementという2つのモジュールからなる。図-2にこのアーキテクチャを示す。

Task Managementは、個人のあるいはプロジ

ェクト内でのタスクの立案、実行、監視を行う。Time Managementは、ユーザによる現在あるいは将来の約束事（たとえば会議の予約など）の管理を補助する。この処理には、会議や面会のスケジューリングだけでなく、リマインダの生成や作業負荷を均衡させる処理も含まれている。この2つのモジュールは、Query Managerと統一的なインタフェースでつながっている。

Task ManagementはBelief-Desire-Intention (BDI) モデルという複数エージェントを制御する手法を使ってアクションを決定している。BDIモデルは心理モデルであり、信念、願望、意図という心的状態を持っている⁴⁾。図-3にBDIモデルの実行サイクルを単純化して示す。このサイクルでは、心的状態に従ってアクションを実行する。そして、ユーザからの入力を受け取り、心的状態が変化する。CALOでは、この枠組みをSPARK (SRI Procedural Agent Realization Kit) という言語で記述している。この言語で書かれた最もシンプルなエージェントのプロシージャは $\{defprocedure \text{名前} cue: イベント precondition: \phi body: \tau\}$ という形をしている。このプロシージャは、 ϕ という条件（心的状態）が成立していて、かつ、イベントが起こったときにアクション τ を実行する。イベントとしては、たとえば $[newfact: \xi]$ などのようなステ

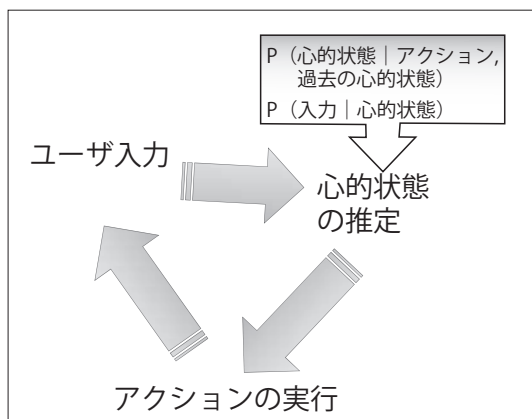


図-4 統計的対話制御のサイクル

ートメントが使われる。これは、 ξ が知識源に付加されたことを表す。

このように、PExAも特定のタスクを想定したアーキテクチャではなく、オフィスでのさまざまなタスクに柔軟に適応できるようなアーキテクチャを採用している。

統計的手法による対話制御へ

CALOはCognitive Assistant that Learns and Organizesという名前に示されているようにさまざまな適応的学習機能を有している。知識源も、大量のデータから取得されるか、あるいは、学習により構成される。しかし、対話の制御に関しては、BDIモデルによるルールで記述することを前提としている。ここで重要なのは、タスク達成型の対話、すなわち、知識源が明確なデータから情報が取られるような対話では、対話制御の目的が明確なため、BDIモデルのルールを書くことは比較的容易であるが、認識誤りに適切な対応をする必要のある対話や、非タスク達成型の対話では、ルールを書くことが難しい（非タスク達成型でも一問一答型のものであれば書き下せる可能性はある）ということである。このため、より自然な対話を実現するためには、対話制御を自動的に獲得することが必要不可欠である。

このような対話制御を構築する試みとして、筆者らはPOMDP (Partial Observable Markov Decision Process) を利用する対話制御を提案している。POMDPの対話サイクルを図-3と対比させて図-4

に示す。図-3との違いは、図-4では心的状態はシステム側ではなくユーザ側の意図を表しており、直接観測できないということである。このため、心的状態は、ユーザの入力から推定しなければならない。それに加え、入力の推定結果には認識誤りがあると仮定している。これらの不確定性を扱うために、 $P(\text{心的状態} | \text{アクション}, \text{過去の心的状態})$ 、 $P(\text{入力} | \text{心的状態})$ という確率を導入している。このような枠組みを用いて認識誤りと非タスク達成型対話に対応している。

■ 認識誤りへの適切な対応をする対話制御の実現

ここで、ファイル消去を実現する簡単な対話タスクを考えよう。まず、ユーザの心的状態として2つの意図“ファイル消去しない”と“ファイル消去する”を設定する。システムはこのユーザの心的状態を推定しアクションを実行する。システムのアクションは、“ファイル消去しない”、“ファイル消去する”、“確認する”の3種類だけである。“確認する”はユーザの発言を促す。

ユーザからの入力は、“ファイルを消去しないでください”と“ファイルを消去してください”の2つである。これらの入力は心的状態から出力されると仮定する。ただし、音声認識結果に認識誤りがあるため、システムは真の心的状態を観測できない。ここでユーザの真の入力を生成する確率、 $P(\text{心的状態} | \text{入力})$ を0.7と設定する（これは直接の認識率ではないが、そう捉えてもよい）。また、簡単化のため、 $P(\text{心的状態} | \text{アクション}, \text{過去の心的状態})$ による状態の変化は起きないものとする。

もし、ユーザの心的状態の推定が最初どちらも50%だとすると、システムがユーザの“ファイルを消去してください”という発声を聞いたとき、ユーザの心的状態が“ファイル消去する”である確率は70%となる（確率 $P(\text{心的状態} = \text{“ファイル消去する”})$ を計算するのであるが、ここではその詳細は省略する）。ファイルを誤って消去する確率は30%

である。このとき、システムは、ファイルを消去すべきだろうか？ 多分すべきではないだろう。もし、この心的状態の確率を100%にしたいならば、システムは“確認する”を無限に繰り返すことが必要である。しかしそれは非現実的である。そこで、ここでは、確率ではない人の評価すなわち報酬を導入する。たとえば、もし誤ってファイルを消してしまったら、負の報酬-100を獲得するとする。それとともに、“確認する”を何回も聞かれると煩わしいので、このアクション1回ごとに-10の報酬を獲得するようにする。これらの負の報酬のバランスによりシステムが人の価値観に従って動作するように設定する。この値の設定はヒューリスティックに実施されるが、認識率に対応するルールを網羅的に記述するよりは、労力が少ない。実は、このように報酬を設定すると、心的状態の確率分布が分かれば、将来どれぐらい平均的に報酬を獲得できるかを自動的に計算することができる。この平均報酬を最大とするアクションがいま選択すべき最適なアクションとなる。このアクションは、 $P(\text{心的状態} | \text{アクション}, \text{過去の心的状態})$ 、 $P(\text{入力} | \text{心的状態})$ が決まっていれば、一意に決まる。これは、将棋などのゲームで、ある程度先まで読めれば、次の手が決まってしまうのと同じだが、将棋と大きく違うのはPOMDPではユーザが最適な手を打つとは限らないという点である。このため、ユーザの最適な手だけでなく、可能なすべての手を計算しその平均の報酬を計算する。このアクションの選択方法は方策と呼ばれる関数 $\pi(P(\text{心的状態}))$ で表現される。これは、心的状態の確率分布の値を入力すると、結果としてアクションを返す関数である。

ここで述べたさまざまな確率が既知でない場合は、ユーザとのやりとりからこれらの確率を自動的に学習する手法がとられる。

■ 人の評価による非タスク達成型対話の実現

非タスク達成型の対話に対し、従来型の対話システムでは、人手によりルールを記述するのが普通である。対話が一問一答であれば、ルールを記述する

発話	対話行為
S:こんばんは	挨拶
L:こんばんは	挨拶
S:テーマは食事をお願いします 今夜の夕飯はカレーでした Lさんはカレーは好きですか?	挨拶 自己開示 - 事実 質問 - 評価
L:よろしくお願いします 何カレー?	挨拶 質問 - 事実
S:自宅カレーです	自己開示 - 事実
L:カレー大好きです!	共感・同意
S:隠し味などナッスィングなカレーです	自己開示 - 事実
L:ナッスィングなカレーですね!	繰り返し
S:自宅カレーです	自己開示 - 事実

表-1 典型的な聞き役対話 Sは話し役, Lは聞き役。

のは可能であるが、流れのある非タスク達成型対話をルール化するのは難しい。そこで認識誤りと同様に、人の評価によって自動的に対話制御を学習することを筆者らは提案している。興味深いことは、冒頭で述べたように、Siriを扱った雑誌の記事では、雑談のような対話に対して、面白いとかユニークだという評価が下されている点である。もし、このような対話の流れを評価に基づいて自動的に学習できれば、所望の非タスク達成型の対話の実現できるだろう。

この考えのもと、筆者らは、POMDPを用いて聞き役対話という非タスク達成型の対話システムを実現しようとしている。ここでは、評価として、満足度という尺度を用い、それを報酬に変換し、方策を決定する手法を提案している。聞き役対話の例を表-1に示す。左欄が発話を示し、右欄はこの発話の対話行為を記述している。左欄のような発話を生成できればよいが、現状の技術ではそこまでできない。現在は、右欄に示す対話行為までを生成するシステムを作成している。

図-5に筆者らが実現しようとしている対話処理の全体像を示すが、ここで扱う対話行為の生成はこのうち点線で囲まれた部分を実現するものである。

前章で述べたように、POMDPでは、さまざまな確率および心的状態を決定する必要がある。ここでは、その構造そのものも統計的手法であるDBN(Dynamic Bayesian Network)を用いて学習している。



このシステムを作成し、人による評価を行ったが、システムが生成した評価用の対話行為列を見て、直接評価を行うことは困難である。そこで、一度人手で自然な対話文に直し、それを評価した。自然文に直す際には、まず、それぞれの実験参加者に日常的な状況を提示した。具体的には、時間・食べたもの・一緒に食べた人を提示した(例:「夕食」に「肉じゃが」を「家族」と食べた)。そしてその状況に自分が置かれていると

想定して、2人の Wizard (コンピュータ役の人間)が文章を作成し、14人の実験参加者が評価を行った。

この実験により、POMDPによる手法すなわち統計的な手法とルールベースによる手法がほぼ等価な評価結果を示すことが分かった。ルールによる手法に比べ統計的な手法は人手を介する労力がきわめて少ない。

これまでの実験から、自然な対話系列の対話制御まで統計的手法によって実現できることが分かってきた。

これからの対話処理

本稿では、統計的な対話手法として、複数の知識源を利用する対話制御と、認識誤りに強い対話制御、非タスク達成型の対話制御について説明を行ってきた。今後、これらの手法を統合することにより、より自然な対話を行えるシステムが実現できると思われる。しかしながら、統計的手法による音声対話システムを実現するためには、対話制御系だけでなく、図-5に示す理解・解釈や言語生成のモジュールの開発が必要である。これらのモジュールの実現には、統計的手法によりどのように意味理解を実現するか

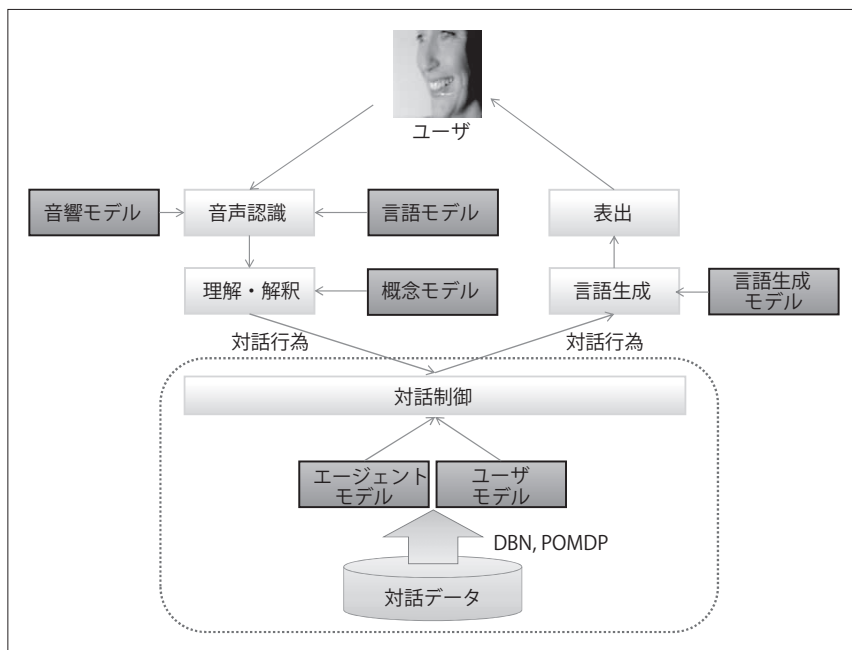


図-5 筆者らが目指す統計的対話制御

という困難な問題を伴う。これは、今後の研究課題である。

筆者は、冒頭で、統計的手法による対話制御は、確立された技術ではないと述べたが、統計的な手法を使った人工知能の研究は、音声認識、統計翻訳と徐々に人間のコミュニケーションにかかわるさまざまな領域に拡大しつつある。対話処理もその例外ではないだろう。この分野が今後注目されることは間違いないと筆者は感じている。

参考文献

- 1) <https://pal.sri.com/Plone/framework>
- 2) Ambite, L. J., Chaudhri, K. V., Fikes, R., Jenkins, J., Mishra, S., Muslea, M., Uribe, T. and Yang, G. : Integration of Heterogeneous Knowledge Sources in the Calo Query Manager, SRI Technical Report (2005).
- 3) Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, L. D., Morley, D., Pfeffer, A., Pollack, M. and Tambe, M. : An Intelligent Personal Assistant for Task and Time Management, AI Magazine, Vol.28, No.2, pp.47-61 (2007).
- 4) Rao, S. A. and Georgeff, P. M. : BDI Agents : From Theory to Practice, ICMAS, pp.312-319 (1995).

(2012年7月2日受付)

南 泰浩 (正会員) minami.yasuhiro@lab.ntt.co.jp

1986年慶大・理工・電気卒業。1991年同大博士課程修了。現在NTTコミュニケーション科学基礎研究所主任研究員。音声認識、音声対話処理、環境知能、幼児言語獲得の研究に従事。