

対話システムにおける知識の相違および認識誤りの解消

渡辺 太郎 荒木 雅弘 堂下 修司
京都大学大学院 工学研究科 情報工学専攻
〒606-01 京都市 左京区 吉田本町
e-mail: taro@kuis.kyoto-u.ac.jp

あらまし ユーザとシステムとが目的指向型対話を行うとき、知識の相違のため両者で競合が生じる可能性がある。また、音声言語をチャネルとすると、認識誤りのためにシステムがユーザの発話を誤解し、このために知識の競合が生じる。この両者の競合を解消するために、システム—ユーザ間での協調的交渉を行う必要がある。本研究では、これらの競合の検出および解消の過程について検討する。そして、交渉の過程として対話をモデル化し、知識の相違による競合が生じたとき、これを解消する過程を示す。また、音声認識による誤解をシミュレーションにより分析し、解消過程のモデル化を試みる。

和文キーワード 音声認識誤り, 誤解, 知識の相違, 競合解消.

Conflict Resolution in Difference of Knowledge and Recognition Errors in Dialogue System

Taro Watanabe Masahiro Araki Shuji Doshita
Department of Information Science, Kyoto University
Sakyo-ku, Kyoto 606-01, Japan
e-mail: taro@kuis.kyoto-u.ac.jp

Abstract During the engagement of task oriented dialogue between user and system, some conflicts might arise because of the difference of knowledge. And other types of conflicts could be aroused from recognition error when using speak language as a channel. To avoid these two types of conflicts, user and system should be evolved in the process of a collaborative negotiation. In this paper, we will argue the issues of detecting and resolving conflicts due to these conflicts. We treat dialogue as a process of collaborative negotiation, and show how to resolve the conflicts aroused from difference of knowledge. And also, we analyzed misunderstandings due to recognition errors through computer simulation, and attempt to make a model of resolving this type of conflicts.

Keywords recognition error, misunderstanding, difference of knowledge, conflict resolution

1 はじめに

目的指向型対話を行っているシステム—ユーザ間で協調的問題解決が行われ、対話により両者がプランを形成する。そしてシステムがその形成されたプランの実行を行う。プランニングの過程において、両者の知識の相違のため、食い違いが生じる。この食い違い、衝突を解消し、互いのプランの一貫性を保つためには両者で交渉を行い、プランの訂正を行う必要がある。システムは、ユーザの発話を調べ、システムの知識との一貫性が保たれるかを調べる。そして、ユーザの発話を受諾できない場合、単に「いいえ」と返答するだけでなく、互いに受諾できる命題を述べ、対話を継続する必要がある。

また、音声言語をチャンネルとした対話システムを実現する上で、音声認識誤りが障害となり、上記の協調的対話を単純に継続することができない。認識誤りが生じるたびにユーザに対して発話を再度要求するのは、ユーザに対して過度の負担を負わせることになり、協調的対話の実現が不可能である。音声認識器において部分的に解析できた結果を利用するのが妥当であり、システムが認識できなかった部分については、システムが質問する、あるいは後のユーザ発話から推測するのが妥当と思われる。また、システムが誤って理解した発話については、ユーザに確認を行うことにより、対話の継続が可能である。ユーザがシステムの誤りを指摘した時点で、システムはどの知識が誤っているかを認識し、知識の訂正を行い、プランニングの再実行を行わねばならない。

本稿では、上記の知識の相違によるシステム—ユーザ間の衝突、および、システムの認識誤りによる勘違いから生じた衝突の競合を検知し、解消する過程の対話モデルを提唱する。これは、対話を交渉を行っている場として捕らえ、対話参加者がある内容について提案、受諾、拒否を行うことにより目的を達成する。システムは、ユーザの発話を評価し、知識の競合が生じた場合に、ユーザの知識を訂正する発話を生成することにより競合の解消を行う。これは、システムがユーザに対して知識の相違を指摘することになる。また、認識誤りによる誤解については、これを知識の相違とみなし、ユーザがシステムの認識誤りを指摘した場合に、その発話は、システム自身の知識を訂正する発話として解釈し、状態の修正を試みる。こうして、システム発話、「もう一度最初からお願いします」の回数を最小化することを目的とする。

以下、2節では関連研究について述べ、3節において対話モデルを提唱し、知識の相違を解消する過程をシステム—システム対話により示す。また、4節では3節で用いた対話システムをベースにシミュレーションによりノイズを混入させたシステム—システム対話により分析を行う。そして、音声認識誤りによる誤解を解消する過程につ

いて概説する。

2 関連研究

現在の対話システムの研究において、システム—ユーザ間の競合の解消についての研究は多くなされている [1][2]。Chu-Carrollらは、エキスパートシステムとそのユーザとの間での知識、プランの競合を扱い、システムがユーザとの競合を解消するために、適切な発話を選択する過程のモデル化をおこなった [1]。彼女らは、対話を提案—受諾、提案—拒否の系列とみなし、対話を「提案—評価—修正」のサイクルでモデル化した。これは、対話相手のある提案に対して受諾できるかを評価する。拒否するならば、それに対して修正をし、対話相手に伝える。また、対話モデルを、「ドメイン、問題解決、信念、談話」という4つのレベルに分割した。この枠組みをプランスキーマで表現し、協調的交渉における発話生成の過程を示した。この研究では、協調的な発話を生成する過程に着目しており、特に信念、問題解決における競合を解消する枠組みを提案している。このモデルの利点として、各対話モデルの階層に知識が分類されており、競合あるいは誤解が生じた場合にその分類が可能であり、各レベルに対して異なった対処法を選択可能である。本研究でもこの枠組みを用いて、まず、知識の相違による競合解消の過程をモデル化する。そしてこのモデルを認識誤りによる競合に適用する。

また、誤解の解消を行う研究として、石崎らの研究があげられる [3]。彼らは経路探索問題において、システム—システム対話を行う際に、パーザの言語知識の不足から生じた誤解を解消する手法を提案している。誤解に気付いたシステムが相手のシステムに対して状態の修正を行い、両者が共有している状態、つまり、互いに共有信念が確実に確立されている状態へとバックトラックする。これは、複雑な誤理解消の手法をとらず、コミュニケーションにより解消するというものである。これは、対話システムを実現する上で、ノイズを考慮にいたったものであり、知識の種類がかなり限定されたタスクについては有効といえる。

本研究では、誤理解消を一般的な枠組みで捕らえ、特にスケジューリングタスクに絞り、詳細化を行う。また、システム—システム対話により様々な現象を分析し、知識の相違による競合、および認識誤りによる誤解をモデル化する。

3 競合の解消

3.1 対話のモデル化

対話をモデル化するにあたって、Chu-Carrollらと同様に、以下のような、4つのレベルを設定した(図1参照)。

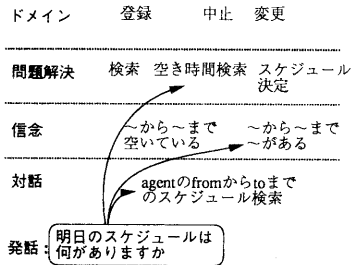


図 1: 対話モデルの 4 つのレベル

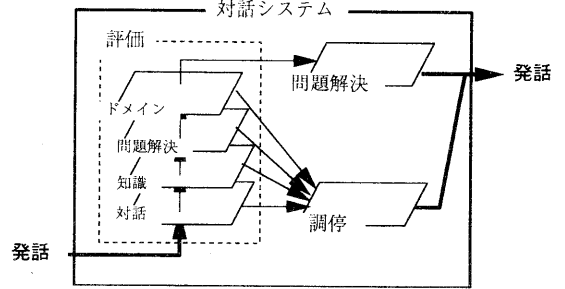


図 2: 対話システム

- ドメイン — 対話の目的であり、この目的に競合が生じた場合は、対話の継続が不可能となる。

例、スケジュールの削除、登録

- 問題解決 — 目的達成のために選択される行為のプランニングにおいて、そのプランの正しさを検証する。このプランについての競合が生じた場合には、対話参加者でプランの調整が必要となる。

例、スケジュールの登録のために、スケジュール検索を行う

- 知識 — 対話参加者間での知識の相違をみつかる。知識の競合が生じたとき、対話相手の知識を修正する必要がある。
- 対話 — 各発話が正しく行われたか、正しく認識されたかを評価する。このレベルでの競合は、対話相手が必要な情報を伝えていないときに生じる。

対話システムは、上記の 4 つのレベルに対してユーザの発話の評価を行う。

また、システムの処理モデルを、図 2 の評価、調停、問題解決、という 3 つの過程にモジュール化した。ユーザの発話は、まず評価部において 4 つのレベルについて、ボトムアップに評価を行う。このユーザの発話は、ある命題について提案を行っていると見なせ、システムがあるレベルにおいてユーザの提案を受諾するか、拒否するかを決定する。そして、受諾する場合、システムはユーザに対して問題解決を提供する。また、拒否するとき、これは、システムとユーザとの間で競合が生じていると見なし、調停部においてその問題を解決する。これはユーザに対してユーザの知識、問題解決を修正する行為を生成する。

発話行為は、大きく分けて提案、受諾、拒否に分類される (表 1 参照)。基本的に、ユーザ、システムとは、ある提案に対して、受諾、拒否の発話を行う。だが、人間—人間の対話では、明確な提案—受諾、提案—拒否の系列ではなく、提案—提案という対話が行われている。これ

表 1: 発話行為例

発話行為		発話例
offer	assign_schedule	(スケジュール)を登録してください。
accept		登録できます。はい。
reject		登録できません。いいえ。
offer	change_schedule	(スケジュール)を変更してください。
accept		変更できます。はい。
reject		変更できません。いいえ。

は、対話相手に対して暗示的に受諾、拒否の意図をしめしているものとみなせる。たとえば、本研究室の人間—人間対話コーパスでは、

System: 月曜日の午前 10 時から午後 2 時まで、東京で講演会に出席します。

User: 講演会の行なわれる場所は、どこですか。

のような発話がある。これはシステムの提案である、*offer(schedule, [Koenkai, From, To, Tokyo])* に対してユーザはシステムの提案を評価し、この内容を受諾した。そして、問題解決により、*offer(specify_place, [Koenkai, []])* という提案を行っている。システムは、ユーザの発話 *offer(specify_place, [Koenkai, []])* を自分の以前の発話と行為、内容を比較することにより、自分の発話を受諾されたことが分かる。

このように、提案—受諾、提案—拒否、提案—提案の系列を評価するには何らかの比較基準が必要である。本研究では、スケジューリング問題において登録されているスケジュールを知識として持たせ、その評価により対話システムを実現した。スケジューリング問題において、各発話の内容は、あるユーザがどれだけ時間のスロットを示しているかで示すことができる。

3.2 対話例

上記の対話モデルに基づいて、システム—システム対話によるシミュレーションを行った。システム—システム対話を行うことにより、対話システムの評価を行う手法が提案されており [4]、本研究において参考にした。

対象とするタスクは、スケジューリング問題であり、ユーザは、ある制約内で複数人によるミーティングを登録するものである。今回は、知識の競合について扱い、スケジューリング問題では、知識の評価として、スケジュール・データベースを用いた。表2、シミュレーションの対話例を示す。このシミュレーションでは、知識の競合がどのように扱われるかを示したものであり、発話には意味表現を用いた。また、ユーザのスケジュールは、11月18日から11月20日の3時までスケジュールを埋めておき、また、システムは、11月19日の12時までスケジュールを埋めておいた(図3参照)。この例では、システムとユーザとの知識の相違があり、これを対話により解消し、スケジュールの決定を行った。

まず、ユーザはシステムに対してスケジューリングを依頼し(User(1))、システムは自分のスケジュールデータベースを検索した結果、「19日の2時から」という提案を行っている(System(2))。ユーザはシステムの提案を評価し、知識レベルにおいて、ユーザの提案した内容(「19日の2時から」)は、自分のスケジュールと競合を起こしていることが分かる(図3参照)。従って、調停部では、システムの知識を修正する発話を生成する必要がある。このために、ユーザは明示的な拒否

reject(assign_schedule, [[1996, 11, 19, [12, 0]], [3, 0], lab2, meeting])

を選択するか、あるいは他のスケジュールを提案することにより拒否し、また新しく自分の知識を伝えることができる(User(3))。ユーザの発話(User(3))をシステムの前の発話(System(2))とを比較すると、提案の内容が全く異なったものである(図3参照)。つまり、システムの発話System(2)は拒否されており、「19日の2時から」は何らかのスケジュールがユーザにある、と推測できる。また、システムはユーザの発話自体を評価し、この発話は自分の知識と競合がないことが確かめられる。従って、発話(4)では、スケジュール登録の確認を行っている。こうして、これはもっとも単純な例ではあるがシステムとユーザとの知識の相違が解消されている。

4 認識誤りによる誤解

上記の対話システムでは、認識誤りによる競合を扱うことができない。音声言語はメディアとして信頼度が特に高くはなく、対話システムを作成する上では、ill-formed

表 2: 知識の競合、対話例

User(1)	18日から1週間以内に3時間の、二研のミーティングはできますか。 <i>offer(planning_schedule, [[1996, 11, 18], [1996, 11, 25], [3, 0], lab2, meeting])</i>
System(2)	19日の12時から登録できます。 <i>offer(assign_schedule, [[1996, 11, 19, [12, 0]], [3, 0], lab2, meeting])</i>
User(3)	20日の3時からならいいです。 <i>offer(assign_schedule, [[1996, 11, 20, [15, 0]], [3, 0], lab2, meeting])</i>
System(4)	11月20日の3時から二研のミーティングでいいですか。 <i>offer(confirm_schedule, [[1996, 11, 20, [15, 0]], [3, 0], lab2, meeting])</i>
User(5)	はい。 <i>offer(accept)</i>
System(6)	わかりました。 <i>offer(accept)</i>

(注) システム—システム対話、実際には意味表現を用いた

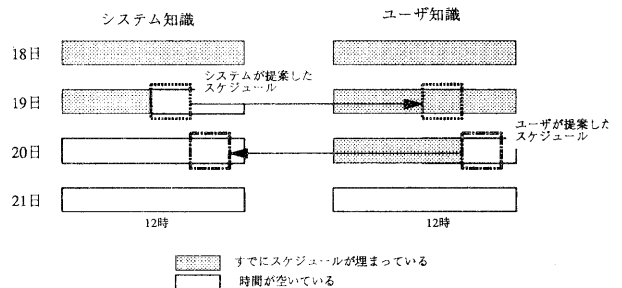


図 3: スケジュール例

な入力に対処する必要がある。

4.1 誤認識の分析

誤認識を分析するために、上記の対話システムにおけるシステム—システムの対話を、一方をシステム側とし、発話入力にノイズを挿入した(図4参照)。ノイズは、単語の 카테고리内 で単語を入れ替えることにより行い、同時に脱落の現象を導入した。今回は、数詞に関するもの(日付、時間等)についてノイズを挿入し、単語認識率を80%に設定して行った。

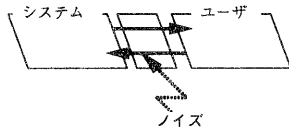


図4: 実験システム

この結果、以下のような結果を得た。

- 発話の脱落 — 発話の一部が全く意味をなさないものとなり、発話全体では、理解不可能な場合(表3参照)
- 完全な誤解 — 発話全体では、意味をなすが、ユーザーの意図したものとは異なる内容となる場合(表4参照)。

表3: 対話例1

User	18日の2時から二時間二研のミーティングを登録してください。
User	18日の2時、9時、二研のミーティングを登録してください。
System	何時間ですか。

(注) システム—システム対話、ノイズを挿入

User'—システムが認識した発話

4.2 誤解の解消

発話の脱落は、解析不可能な部分が脱落したものであり、ユーザーに質問することにより解消可能である。これは、図2の対話システムでは、対話レベルの評価を行い、脱落があれば、ユーザーに質問する。対話例2の場合、時間に関するものが二カ所に現れた。今回用いたバーザは、後者を優先的に解析するので、システムは「18日の9時か

表4: 対話例2

User	18日の2時から二時間二研のミーティングを登録してください。
User'	8日の2時から二時間二研のミーティングを登録してください。
System	8日の2時から二時間二研のミーティングでよろしいですか

(注) システム—システム対話、ノイズを挿入

User'—システムが認識した発話

ら」というように理解した。また、ミーティングの登録には期間の スロットがあり、ミーティングの登録実行には適さないとして対話レベルにおいて評価した。従って、調停部において「何時間ですか」という応答により競合の解消を行っている。

完全な誤解については、ユーザーにより解消してもらわねばならず、システムはユーザーの指摘を待つ必要がある。ここで、対話例2においてシステムは確認の発話行っているが、これはユーザーが伝えたものとは日付が異なっている。ユーザーは対話レベルにおいて自分の意図とシステムの発話(およびその意図)とを比較することにより、競合を検知することができる。そして、調停部においてシステムに対して誤りを指摘することができる。システムは、ユーザーの指摘に対して自分の状態の変更を行う必要がある。問題解決部では、ユーザーの提案した修正を実行し、また、次の行為を決定しなければいけない。

ここで問題となるのが、ユーザーが指摘した発話がどの誤りを参照しているかである。対話例2では、システムの現在の発話に対して行われており、容易に修正が可能である。だが、対話例1では、2ターン後にユーザーの修正の発話に来るかもしれない、また、さらに遅れて修正が行われる可能性がある。特に、長い対話になると、ユーザーが気付かない誤りが存在したとき、システムが目的達成と信じた時点でユーザーが誤りに気付くことが考えられる(図参照)。

現在考えられる手法として、システムは対話履歴を参照し、ユーザーの指摘した誤りを順次最新のものから比較することを検討している。そして、ユーザーにより指摘されたものが存在した場合、バックトラックによりそのときの状態に戻り、それまでの知識を破棄する。そして、対話を再開する。だが不可能な場合には、システムをリセットし、再度対話を開始する必要がある。

このようにシステム、ユーザー双方にとって認識誤りは負担のかかるものである。これらの誤理解消に加えて、誤解を最小限にする対話戦略についても考慮する必要がある。

4.3 対話戦略

対話戦略とは、相手の発話に対して、どのような応答をするかを決定する。特に、発話を生成する時にどのような内容を選択すべきかを決定する戦略である。単語認識率が低い環境のもとでは、ユーザの発話毎に毎回確認をとる戦略が有利と考えられる。また、認識率が高い環境では逆にシステムが行為を実際に行うときに、最後に確認を行うことができる。両者の間で、ある程度対話が進行した時点で確認を行う戦略も考えられる。このように、対話戦略は単語認識率と何らかの関連性があると考えられ、逆に対話戦略を単語認識誤りに応じて設定することにより、誤解をあらかじめ防ぐことができる。現在、この対話戦略を問題解決部において導入し、単語認識率との関連性を評価することを検討中である。

5 おわりに

対話において、各発話を提案とみなし、システムはその提案を評価、受諾できるならば、問題解決により行為の実行を行う。また拒否するとき、これは競合が起こっているものとみなし、ユーザに対して修正の発話を生成する。このモデルにおいて、知識の相違が解消されることをシステム—システム対話によるシミュレーションで例をあげた。また、音声認識器による音声認識誤りが生じた場合の誤解についてシミュレーションし、誤理解消の過程の概観を示した。

今後の課題として、音声認識誤りによる誤理解消のモデルを導入した対話システムを完成させることである。また、対話戦略を取り入れ、単語認識誤りによるシステム—システム対話のシミュレーションを行っていきたい。

参考文献

- [1] J. Chu-Carroll and S. Carberry. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proc. of AAAI*, pp. 799-805, 1994.
- [2] C. I. Guinn and A. W. Biermann. Conflict resolution in collaborative discourse. In *Proc. of IJCAI*, 1993.
- [3] 石崎雅人, 伝康晴. コミュニケーションによる誤解の解消. 音声言語情報処理, Vol. 96, No. 104, pp. 13-18, 1996.
- [4] Curry I. Guinn. The role of computer-computer dialogues in human-computer dialogue system development. In *Proc. of AAAI*. 1995.
- [5] Alain Cozannet and Jacques Siroux. Strategies for oral dialogue control. In *ICSLP*, pp. 963-966. 1994.