スロットフィリング型対話の制御一般化の試み

前田 陽平　　島津 明
北陸先端科学技術大学院大学 情報科学研究科

1 はじめに

近年、音声認識技術の向上によって、音声対話システムにも期待が高まっている。しかし、人間と対話するのと全く変わりのない対話を行うシステムが理想であるが、現状は、限定された話題で単純な会話を行うというレベルであると言える。現在の対話システムは会議室予約や、チケット予約などのタスクを対象に作られている。対話制御はスロットフィリングが一般的に用いられている。しかし、スロットフィリングには対話システムでのグランディングの扱い、対話の遷移の扱いなどいくつかの問題が指摘されている [1]。

人間同士の対話を分析し、モデルを示す対話理解の研究がある。[2] は、説明、修正などの行為をするためには、その行為を認識する必要があり、そのために部分対話の認識が必要であることを示している。[3] は、基盤化（共有信念の形成）および談話義務（相手の発話に対して負う義務）に基づく対話管理の一般的なアルゴリズムを示している。

本報告では、会議室予約、チケット予約などのスロットフィリング型の対話システムの対話制御に、このような人間の対話分析に基づいたモデルを導入し、一般化する試みを述べる。自然に近い対話を可能にする制御を行うため、制御を対話管理部とフレームによるタスクごとの処理とに分けて、それらを組み合わせることによってさまざまなタスクに対応できるようにするモデルを考える。具体例として、列車チケット予約タスクを対象に修正対話が可能な実験システムを示す。

2 対話制御モデル

(1) スロットフィリングモデル

スロットフィリングは、ユーザの発話を解析し、結果をフレーム（テンプレート）に当てはめることによって対話を進行させる。チケット予約を例にすると、日付、出発駅、到着駅、出発時刻がスロットにあたる。

対話では、グランディング行為が必要である [3][4]。グランディングとは相互信念を形成することであり、例えば、ユーザがチケット予約を依頼する発話をした場合、システムがユーザのその要求を理解したと示すことである。スロットフィリング型の制御では、これは確認処理の手続きとして扱われる。

ユーザがシステムの質問に対して直接的な答えを返すのではなく、別のリクエストをする場合がある。例えば、「何時発のチケットを予約しますか？」というシステムの質問に対して「やっぱり到着駅を変えたいんですけど」というように、新たな要求を提示してシステムにその要求が優先事項だと認識させるという場合がある。この場合、各スロットの状態からシステムがどうやって動作するかあらかじめ筋道を立てておく必要がある。スロットフィリング制御は問題解決をベースにして対話管理を行っているが、柔軟な対話が行えるシステムにするには、対話理解と問題解決とを適切にまとめる必要がある。

(2) グランディングに基づくモデル [3]

グランディングに基づくモデルの対話管理は、システムの内部状態を示す対話状態テーブルを変更しながら対話を行う。対話状態テーブルに基づいて、談話義務の処理、システムがターンを持っているかどうかの判断、システムの発話する言語行為 (intented speech act) への対処、グランディングへの対処などを行う。対話状態テーブルには以下の情報が保持される。

談話義務: ユーザの発話に対してシステムが負う義務でユーザの発話の後に何を行うかを示す。例えば、ユーザが質問を行ったなら、その質問に対して答える談話義務が生じる。その結果、談話義務が対話状態テーブルに書き込まれる。

対話ターンの保持者: ユーザとシステムのどちらが対話を行う番であるか示す。ユーザが発話ターンを保持しているときはシステムはユーザの発話を待ち、ユーザが発話を終えたときにシステムが発話ターンを持つ。

システムの発話する言語行為: システムが発話すべき内容を表す。例えば、未了解の言語行為をグランディングするためにシステムは確認する発話をしなければならない。

未了解の言語行為: ユーザ発話により了解が取れていない行為は、未了解の言語行為となる。

対話ゴール: これが達成されるまではシステムは対話を続けようとする。例えば、チケット予約システムではチケット予約が完了することが対話のゴールとなる。

3 提案モデル

グランディングに基づいたモデルをベースに対話制御を行い、システムが発話する言語行為をタスクテー

スロット	システムが発話する言語行為
day	request(sys user inform(user sys ?day))
start-loc	request(sys user inform(user sys ?start-loc))
arrive-loc	request(sys user inform(user sys ?arrive-loc))
time	request(sys user inform(user sys ?time))

表 1: タスクテーブル

ブルに基づいて決定する。タスクテーブルと対話管理部を分離することにより、さまざまなタスクに対応できるようにする。システムが発話する言語行為が生成されると、システム発話が生成され対話が行われる。ユーザの要求に修正要求などがあれば、対話ゴール、対話状態テーブルをスタックして、部分対話を扱う。

タスクに応じた処理はタスクテーブルで定義することによって、対話管理部からタスクに依存する部分を切り離す。タスクテーブルは、対話のゴールを達成するためにスロットの状態からシステムの発話する言語行為を生成するのに使われる。例えば、表1で表すdayスロットが空ならば、システムが発話する言語行為としてrequest(sys user inform(user sys ?day))が対話状態テーブルに書き込まれる。

提案モデルは、以下の各処理を対話状態テーブルに基づいて繰り返し行う。

- ユーザの発話から意味解析結果が出れば、ユーザの発話を解釈する。

ユーザの発話にrequestが含まれる場合、requestに答えるという談話義務が対話状態テーブルに書き込まれる。requestは、修正や説明などの要求である。了解の取れていないユーザの発話は未了解の言語行為とされ対話状態テーブルに書き込まれる。

- システムが談話義務を持っているなら、談話義務に対処する。

対話状態テーブルに、requestに答えるという談話義務がある場合、談話義務を削除し、システムの発話する言語行為を対話状態テーブルに書き込む。修正要求などの部分対話がある場合は、部分対話に入ることになるので、対話状態テーブルをスタックし、修正のサブゴールを対話状態テーブルに書き込む。

本報告では、修正などのユーザの要求が意味解析結果で得られたなら、それに答える談話義務が生成され、その談話義務に対処することによって部分対話に対処する。

- システムが発話ターンを持っていて、システムが発話する言語行為があるなら、システムが発話する言語行為によって発話を生成する。

対話ターンの保持者がシステムにあるなら、システムが発話する言語行為があるか確かめ、あれば発話を生成する。

- グランディングが達成されていないなら、グランディングに対処する。

グランディングは未了解の言語行為が存在し、かつシステムが発話する言語行為が生成されていないときに行われる。グランディングによって、未了解の言語行為が対話状態テーブルで参照され、システムが発話する言語行為生成される。

- 高いレベルのゴールが達成されていないなら、対処する。ゴールが達成されていれば、対話を終了する。

サブゴールまたはゴールが達成されていないか確かめ、ゴールが存在するなら、それに対処する。部分対話に対するサブゴールが達成された場合には対話状態テーブルをポップする。

4 実験システム

システムの概要を図1で示す。上述のモデルにより対話管理部は対話状態テーブルを参照しながら、タスクに応じた発話生成をタスクテーブルによって決定する。

音声認識部と構文解析部: チケット予約に関する文パターンと、音声認識されたユーザ発話を比較し、解析する。

意図理解部: 入力発話のタイプがassertionあるいはrequestであるか判断する。requestがあった場合、requestに答える談話義務が生成される。システムの確認発話後ならば、確認の答えが肯定であるか否定であるか判断する。

(a) request ユーザからの修正および説明等の要求をrequestとする。「AをBに修正したい」「AをBにして」「Bにしたい」「Aを変えて」等の形の文が修

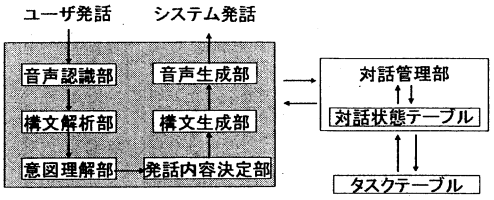


図 1: システム概要

正の request であると判断される。request であると判断されると、その要求に対して答えるための談話義務が生じる。また、request の内容が対話状態テーブルの未了解の言語行為に書き込まれる。

(b) assertion 「１２日」「富山駅」「１４時」などのシステムに対する応答を assertion とする。request に当てはまらず、かつ現在入力が許される値であるかタスクテーブルで確認して assertion が判断される。assertion であり、未了解であると判断されると、対話状態テーブルの未了解の言語行為に書き込まれる。

(c) 肯定 「はい」「そうです」等を肯定とする。システムがユーザに対して「確認」を行った後、ユーザの発話が「肯定」ならば、その確認した行為が確定される。

(d) 否定 「いいえ」「ちがいます」等を否定とする。システムがユーザに対して確認を行った後、ユーザの発話が否定ならばその確認した行為が否定され、確定されない。そして、否定された場合の動作を実行する。

発話内容決定部と構文生成部: システムが発話する言語行為より、システムの発話内容を決定する。システムが発話する言語行為に対応した構文がパターンマッチにより生成される。

音声生成部: 構文生成部で生成された構文が入力となる。あらかじめ用意してある wav ファイルを組み合わせてシステム発話とし、出力する。

5 対話実験

より自然な対話を目指し、どのような応答、言い回しがふさわしいか上述のシステムで対話実験を行った。

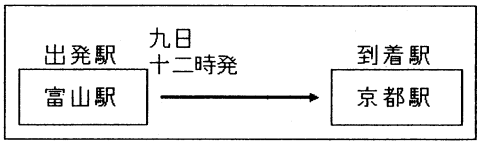
- 実験方法

音声認識ソフトは VIAVOICE を用い、１１人の被験者に、確認方法（２種類）と言い回し（３種類）を組み合わせた計６種類について、それぞれ日付、出発駅、到着駅、時間の修正を行ってもらった。タスクとしてチケット予約を用い、システム表現の正しさ、使いやすさ、円滑性、修正するのに費やしたターン数について、アンケートの結果より評価を行った。

ユーザ：	出発駅を変更したいんですが	
システム：	出発駅を変更しますか？	(a)
ユーザ：	はい	(a)
システム：	出発駅をどこに変更しますか？	(b)
ユーザ：	大阪に	
システム：	出発駅を大阪に変更しますか？	
ユーザ：	はい	

図 2: 確認方法

被験者には、実験の目的、方法が書かれたシートを配布した。チケット予約に関する情報は以下のように示した。



- 確認方法

図２の例は、出発駅がすでに入力され、出発駅を大阪駅に変更する場合の対話例である。確認方法１では図２の確認発話を行う。確認方法２では、(a) の発話は行わず、(b) の対話から確認を行う。

- 言い回し

ユーザの発話をシステム確認で繰り返すときの、単語の言い回しの違いを述べる。

図３は言い回し１の例である。ユーザが言った直接の駅名「富山駅」をそのまま「富山駅」と返し、スロット名である「出発駅」をそのまま「出発駅」と返す。

言い回し２では、(a) が「出発駅を変更しますか？」になる。ユーザが言った直接の駅名「富山駅」をスロット名「出発駅」と返し、スロット名である「出発駅」をそのまま「出発駅」と返す。

言い回し３では、(b) が「富山駅を変更しますか？」になる。ユーザが言った直接の駅名「富山駅」をそのまま「富山駅」と返し、スロット名である「出発駅」を直接の駅名「富山駅」と返す。

予約が完了すると、上記のシートの一部が変更されたシートを見てもらい、修正を行ってもらった。結果を以下に示す。

ユーザ	：富山駅を変更したいんですが	
システム	：富山駅を変更しますか？	(a)

ユーザ	：出発駅を変更したいんですが	
システム	：出発駅を変更しますか？	(b)

図 3: 言い回し 1 の例

- 結果、考察

表 2～表 5 に実験結果を示す。表 3 より、確認方法 2 のほうが使いやすい傾向といえる。ここで、t 検定を行った。"確認方法の違いで使いやすさの平均値に差が出るか" という検定では、有意な結果は得られなかった。使いやすさとターン数の相関は、-0.21 であった。

表 4 より、確認方法 2 のほうが円滑性が高いと言える。"確認方法の違いで円滑性の平均値に差が出るか" で t 検定を行ったが、確認方法が対話の円滑性に影響を及ぼすと言う結果が得られた ($t = |-2.12| > 2.00$, 0.05)。これは、表 5 に見るように、確認の回数が少ない確認方法 2 がターン数が少なくなり、円滑性が高いと感じるためであると推測される。円滑性とターン数の相関は、-0.30 であった。

使いやすさ、円滑性の面で確認方法 2、言い回し 3 の組み合わせが最もよいとわかる。修正対話では言い回しについてはあまり違いが感じられなかった。ターン数に直接かかわるような確認方法の違いが円滑性においてユーザにより影響を与えると言える。表 5 のように、音声認識誤りのためにターン数が多いため、円滑性に影響していると考えられる。

6 終わりに

本報告では、スロットフィリング型の対話制御法に、人間の対話に基づいたモデルを導入し、一般化を行った。さらに修正対話の実験により、動作の確認を行った。対話制御を一般的部分とタスク依存部とで構成する初期モデルは [6] で研究され、OAA[7] により実装された。本稿のモデルはこれを発展させ部分対話が扱えるようにしたものである。

[8] の会議室予約の実験では、ターン数が減る間接確認の評価が低かった。[8] でのターン数が減らすための確認応答は、日常的に用いる機会が少ないために、違和感を感じて評価が低くなったと推測された。今回は間接確認は用いなかったが、チケット予約システムで修正を行う場合は [8] での対話よりもシステムに伝える情報量が多いため、音声認識誤りが増え、ユーザは発話する回数が少なくて済む確認方法を好んだのではないかと推測される。

表 2: システム表現の正しさのアンケート結果

確認方法	言い回し	平均値	標準偏差	最高	最低	中央値	最頻値
1	1	3.182	0.575	4	2	3	3
1	2	3.091	0.668	4	2	3	3
1	3	3.091	0.900	4	1	3	4
2	1	3.000	0.426	4	2	3	3
2	2	3.091	0.793	4	1	3	3
2	3	3.455	0.782	5	2	3	3

表 3: 使いやすさのアンケート結果

確認方法	言い回し	平均値	標準偏差	最高	最低	中央値	最頻値
1	1	2.727	0.862	4	1	3	3
1	2	2.909	0.514	4	2	3	3
1	3	2.818	1.113	4	1	3	4
2	1	2.909	0.900	4	1	3	3
2	2	3.091	0.900	4	2	3	2
2	3	3.273	0.962	5	2	3	4

表 4: 円滑性のアンケート結果

確認方法	言い回し	平均値	標準偏差	最高	最低	中央値	最頻値
1	1	2.545	0.782	4	1	3	3
1	2	2.727	1.052	4	1	3	3
1	3	2.182	1.029	4	1	2	3
2	1	3.000	0.953	4	1	3	3
2	2	3.000	0.953	4	2	3	2
2	3	3.000	0.953	4	2	3	3

表 5: 修正対話ターン数

確	言	平均値	標準偏差	最高	最低	中央値	最頻値	認識誤り率
1	1	7.657	7.391	49	4	8	4	67%
1	2	6.406	2.177	10	4	7	4	63%
1	3	6.406	1.902	9	4	7.5	8	55%
2	1	5.969	2.778	17	4	6	4	61%
2	2	5.469	1.677	13	4	6	6	52%
2	3	4.875	0.992	7	4	4	4	57%

参考文献

[1] D.Jurafsky, J.H.Martin. Speech and Langage Processing, Computational Linguistics and Speech Recognition, Prentice Hall, 2000.

[2] D.J.Litman, J.F.Allen. A plan recognition model for subdialogues in conversations, Computational Science.

[3] D.R.Traum, J.F.Allen. Discourse obligations in dialogue processing, Proceedings ACL, 1994.

[4] 石崎, 伝. 対話と談話, 東京大学出版会, 2001.

[5] D.Jurafsky, J.H.Martin. Speech and Language Processing,pp719-762.

[6] 竹内. 対話実験システムの構築法に関する研究, 北陸先端科学技術大学院大学, 修士論文, 2002.

[7] SRI International. Open Agent Architecture (OAA).

[8] 市野, 島津. 対話システムの確認応答がユーザに与える効果の分析, 言語処理学会 8 回年次大会, 2002.