

音声対話フレームワーク EUROPA の 新幹線チケット予約タスクへの適用

谷口雄一郎, 河野恭之, 木戸出正継
奈良先端科学技術大学院大学情報科学研究科
笹島宗彦, 屋野武秀, 上原龍也
(株) 東芝 研究開発センター

新幹線チケット予約に代表されるスロットフィル型対話では、ユーザは希望の解を得るために何らかの方略にしたがって条件を書き換える。この方略はユーザ毎に異なり、予め対話状態の遷移を設計することが難しい。ユーザとの円滑な対話のためには、システムがその方略を解釈しユーザの意図を理解しなければならない。本報告では対話を通してユーザの方略を判断し対話制御を行う手法について述べる。さらにこの手法を、我々が開発した音声対話システム構築フレームワーク EUROPA を用いて音声対話システムとして実装し、提案手法とフレームワークの汎用性を評価した。

SARPEDON: A spoken dialog system for ticket reservation based on the framework EUROPA.

Yuichiro Taniguchi, Yasuyuki Kono and Masatsugu Kidode
{yuichi-t,kono,kidode}@is.aist-nara.ac.jp
Graduate School of Information Science, Nara Institute of Science and Technology
Munehiko Sasajima, Takehide Yano and Tatsuya Uehara
{munehiko.sasajima,takehide.yano,tatsuya.uehara}@toshiba.co.jp
R&D Center, Toshiba Corporation

A user has some strategies to make a reservation in the goal-oriented dialog. It is difficult for a designer of a spoken dialog system to predefine a certain state transition model, because the strategies depend on each user. We propose a method to gain the user's strategy through a dialog and to control the dialog. This method enables the system to solve ellipsis and references. We have implemented this method to speech dialog system based on EUROPA, a framework for speech dialog systems.

1 はじめに

近年、音声認識技術が向上し実用的な音声認識ソフトウェアが多数登場した。それに伴い実用的な音声対話システムにも期待が高まっている。しかし、実用的な音声対話システムの構築のためには、音声認識以外にも対話制御手法やユーザモデル、分かりやすい応答文生成などの様々な問題が存在する。本報告では、その中でもスロットフィル型対話に見られるユーザの試行に注目し、それに柔軟に対応できる対話制御手法について提案する。

音声対話システムを実用化する場合のもう一つの問題として、タスク依存の知識の拡張、移植にかかるコストが挙げられる。実用的な音声対話システムには、知識の拡張、移植が低コストであること、再利用が可能であることなどが要求

される。このことから我々は音声対話システム構築フレームワーク EUROPA を開発し、これを用いたカーナビ音声対話システム MINOS[3] を既に試作している。本研究では提案手法を新幹線チケット予約タスクに適用し、EUROPA フレームワークを用いて音声対話システムを構築した。本報告では提案する対話制御手法を評価するとともに、フレームワークの汎用性についても検証する。

本報告では、まず次章で EUROPA フレームワークの概要について述べた後、3章でスロットフィル型対話の対話制御アーキテクチャについて詳しく述べる。4章で新幹線チケット予約タスクへ適用するにあたり行った知識の移植、拡張と実験結果についての考察を述べ、最後にこれからの課題、展望について述べる。

2 EUropa の概要

EUropa フレームワークは音声対話システムの構築を目的とした汎用フレームワークである。図1にその構成を示す。

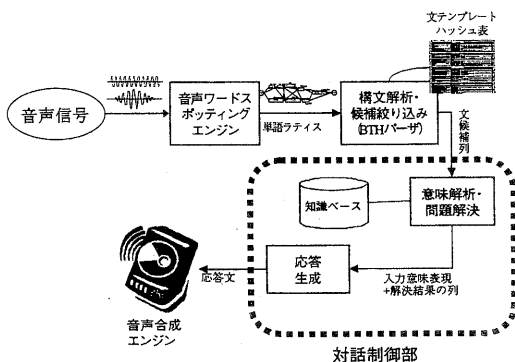


図1: EUropa フレームワークの構成

音声認識部 EUropa では音声認識にキーワードスポッティング法を用いる。対象タスクで使用される語彙を、単語、発音、品詞インスタンスの三つ組で表現し、その集合を語彙セットとして与える。認識結果はキーワードラティスの形で出力される。

構文解析部 構文解析部では、音声認識部から受け取ったキーワードラティスをもとに該当する文型を求める。必要なデータは、認識対象語彙を品詞クラスに分類した品詞データと、その品詞クラスの並びで表現した文型データである。これらのデータをもとに、文テンプレートハッシュ法 [4] を用いてキーワードラティスの中から、与えられた文型で受理できるキーワード系列の集合をもとめる。

意図変換部 意図変換部は構文解析によって得られた文型と認識単語系列から、意図表現と呼ばれるユーザの意図を表すフレームに変換する。意図変換部には、各文型に対応したテンプレートが知識として与えられる。各々のテンプレートには認識単語をどのスロットに代入するかを示す畳み込み知識が組み込まれており、意図変換部はこの畳み込み知識を解釈して、認識単語を意図フレームの適切なスロットに代入し、ユーザの意図を表すフレームを作成する。

問題解決部 問題解決にかかる手続きは USHI(Unification-based Script Han-

dling Instruction set) 言語によって記述する。USHI は Pascal に似た表記法をもつスクリプト言語で、If 文、For 文などの制御構造や各種演算子を持ち、フレーム表現のデータを直接操作する機能を備えている。また、手続きの定義と呼び出しが記述できるようになっており、USHI スクリプト中で定義された手続きと C++ プログラム中で定義された手続きの両方を呼び出すことが可能である。

応答生成部 文生成テンプレートを元に問題解決結果から応答文を生成する。テンプレートには特定のスロットの値の取得や範囲指定された複数のスロットの値を列挙するなどの機能が用意されており、これらを利用して応答文を生成する。応答文生成にどのテンプレートを利用するかは USHI 言語による手続きで記述する。問題解決結果に応じたテンプレートを選択することで、適切な応答文を生成できる。

EUropa フレームワークはすでに音声対話によるカーナビゲーションタスクに適用されている [3]。今回は、新たに新幹線チケット予約タスクへ適用することで、フレームワークの汎用性も合わせて評価する。

3 スロットフィル対話の制御

対象とする新幹線チケット予約タスク対話では、ユーザはできるだけ希望に近い予約をとるために、様々な条件を変更して試行した後、最終的な予約を行うことがある。最初に希望していた条件で予約が出来なかった場合は特にこのような試行が見られる。この試行を予約に関するユーザの方略と捉えることが出来る。対象タスクでは、禁煙席を予約するための方略として一本遅らせたり、早く目的地に到着するための方略として喫煙者が禁煙席を予約する、などの条件の妥協が考えられる。

3.1 対話の解析

対象タスクにおける対話を実際に収集し、それらの分析を行った結果、次のような特徴を見つけた。

予約に失敗したときの条件変更 ユーザが最初に希望していた条件での予約がとれなかった場合、幾つかの条件を変更して予約をとり直す。

過去に試行した条件の一部変更 過去に試行し放棄した方略から試行を再開する。

複数の予約候補の検索 予約可能な候補を見つけた後も、方略を変えて試行を繰り返す、最終的に一つの候補を予約する。

図2に条件変更の対話例を示す。ユーザは最初に希望した「ひかり号の喫煙席」が予約できなかったため、「のぞみ号の喫煙席」を予約しようとした。このユーザの方略は「喫煙席を予約する」ための方略と捉えることが出来る。そのため喫煙席以外の条件として「ひかり号」の条件を妥協した。このような妥協を伝えるユーザ発話には省略が多く見られ、ユーザは変更する条件に関して明示的に述べる一方で、それ以外の条件については触れないことが多い。先に特徴として挙げた試行の再開や、複数候補の中から一候補を特定する場合の発話についても同様に、一部の条件が省略されることが多い。したがってシステムは、ユーザが省略した条件を適切に補完した上でユーザの意図を理解し、対処する必要がある。本報告で述べる対話制御手法ではこの点に留意して、ユーザが持つ方略への適切な対処を目指す。

- U1: 東京までのひかりの指定席をお願いします。
S1: 禁煙席ですか？喫煙席ですか？
U2: 喫煙席の窓側をください。
S2: 喫煙席が予約できるひかりは2本先になります。
U3: のぞみにしてください。
S3: のぞみ号も1本先まで窓側が満席です。
U4: じゃ禁煙席をおねがいします。
S4: 禁煙席の窓側がとれるひかりはもう1本先になります。
U5: のぞみなら空いていますか？
S5: のぞみの禁煙席ですと予約できます。

図2: 収集した対話例

3.2 対話制御手法の詳細

本手法はユーザの発話と対話履歴を用いた、1) 過去の検索条件からの検索再開、2) 検索条件の補完、を目的とする。前者には、過去の対話履歴を全て記憶すること、及びその対話履歴の中から特定の検索条件をもつ履歴を捜し出すことが要求される。このため本手法では木構造の形で対話履歴を保持管理する。後者は次に述べる条件レベルを用いて未指定条件の補完を行う。

はじめに本手法で用いる条件レベルについて述べる。条件レベルは、ユーザが各々の検索条件

に対して持つ重要度と考えることが出来る。レベルの高い条件とはユーザにとってより重要な(妥協したくない)条件である。この条件レベルをユーザとの対話から判断し、対話制御に利用する。

検索失敗をうけてユーザが条件を変更する場合、重要度の低い条件から妥協して行く傾向がある。対象タスクにおいて「禁煙席」の条件を重視する煙草を好まないユーザの場合、検索失敗の際に「禁煙席」条件は変更せず、発車時刻を延ばしたり、「のぞみ」や「グリーン車」に変更したりする。この条件の重要度の違いを反映するため、ユーザが妥協した順に条件レベルが高くなるよう各条件に動的にレベルを付与する。過去に条件レベルが付与された条件を再度変更した場合は、既に付与されている条件レベルを用いる。

次に対話履歴の管理方法について述べる。本手法では一つ一つの対話状態を予約条件の集まりと考え、各時点での予約条件の集合をフレーム形式で表現する。対象タスクでは図3に示すような13の条件スロットを持つフレーム表現を使用するが、本稿では簡単化のため、着駅、座席指定、列車種、煙草、座席位置の5つのスロットのみで構成されるフレームを用いて説明する。

条件を変更するユーザ発話が入力されると、それからフレーム $F_{t'}$ を作成し、発話によって明示的に変更された条件の条件レベル $L_{t'}$ を求める。一状態前の対話状態フレーム F_t の条件レベルを L_t としたとき、 $L_t, L_{t'}$ の関係から次に示す規則 R1, R2, R3 を用いて木構造を構築する。

R1: ($L_t < L_{t'}$ の場合) フレーム $F_{t'}$ をフレーム F_t の子ノードにし、省略された条件を親ノードのフレーム F_t からコピーする。

R2: ($L_t = L_{t'}$ の場合) フレーム $F_{t'}$ をフレーム F_t の兄弟ノード子にし、省略された条件を親ノードのフレームからコピーする。

R3: ($L_t > L_{t'}$ の場合) $L_{t'} = L_u$ となる条件レベルを持つフレーム F_u が見つかるまで木をバックトラックし、フレーム $F_{t'}$ をフレーム F_u の兄弟ノードとする。見つからずにルートノードまでバックトラックした場合は、フレーム $F_{t'}$ をルートノードの子ノードにする。どちらの場合も省略された条件は親ノードのフレームからコピーする。

図2の対話例で、最初に妥協した列車種条件(ひかり)に最も重要度の低い条件レベル1が付与され、次に妥協した座席指定条件(指定席)に条件レベル2が付与される。発話U5が完了した時点での条件レベルは図4のようになる。条

発時刻(時)	発時刻(分)	発駅	着時刻(時)	着時刻(分)	着駅	列車種	号数	座席指定	煙草	座席位置
--------	--------	----	--------	--------	----	-----	----	------	----	------

図 3: 対象タスクのフレーム表現

件レベルの数値が大きい程、重要度が高いことを表す。

着駅	列車種	座席指定	煙草	座席位置
	1		2	

図 4: 条件レベル

条件レベルの付与と同時に、前述の規則を用いて対話履歴の木構造を構築する。この対話例で生成される履歴は図5のようになる。各フレームの左上のラベルは発話文に付与されているラベルに対応しており、フレームがどの発話によって生成されたものかを示している。左下の数値はユーザが変更した条件のレベルを表している。まず最初の検索条件が木構造のルートになる。次に発話 U3 で列車種条件が「のぞみ」に変更され、列車種条件に条件レベル1が付与される。発話 U3 のフレームは規則 R3 によってルートノードの子ノードになる。続く発話 U4 で煙草条件が「禁煙席」に変更されると、システムは条件レベルに2を煙草条件付与する。直前の対話状態の発話 U3 のフレームでは、条件レベル1が列車種条件に付与されている。両者の関係から規則 R3 を適用し、発話 U4 のフレームをルートノードの子ノードに設定する。その結果「ひかりの禁煙席」が次の検索条件となる。直前の検索条件は「のぞみの喫煙席」であることから、「のぞみ」を継承した「のぞみの禁煙席」で検索することも考えられる。しかし「のぞみ」はユーザの妥協によって書き換えられた条件であるため使用せず、ユーザが最初に希望した条件「ひかり」の方を補完する。発話 U5 で列車種条件を「のぞみ」に書き換えると条件レベルは1となる。直前の対話状態の U4 は、フレームの条件レベルが2であることから、規則 R1 が適用され、U5 のフレームは U4 の子になる。

本手法では、ユーザが各検索条件に対して持つ重要度を条件レベルとして定義した。この条件レベルを対話履歴の木構造を構築する際の基準とし、ユーザにとっての重要な条件を優先した検索条件の作成を実現した。対話履歴を木構造でもつことにより、過去の全ての検索条件を参照することが可能になった。更に、ルートから各検索条件までの深さが、その検索条件がユーザの最初の希望条件からいくつの条件を妥協したか

に比例することも、木構造による利点と言える。過去の検索条件を参照するユーザ発話（「やっぱりさっきののぞみにしてください。」など）を解決するために、木構造中の浅い位置にある検索条件を選択することで、ユーザの希望を出来る限り満たした検索条件が選択できる。

4 チケット予約タスクへの適用

ここでは、新幹線チケット予約音声対話システムを構築するために、EUROPA フレームワークに与えた知識について、フレームワークを構成するモジュール毎に述べる。

4.1 音声認識部

対象タスクにおいて使用される認識対象語彙として、新幹線の停車駅の名称や、指定席、自由席などの予約に必要な諸条件、時刻の表現など、計約 220 単語を用意した。

4.2 構文解析部

認識対象語彙を 38 の品詞クラスに分類した品詞データを作成した。さらに、品詞クラスを組み合わせて、タスクで発話されると想定される 116 個の文型を定義した。受理可能な文型の例を表 1 に示す。

文型	例文
着駅と煙草	東京までの禁煙席
座席指定	指定席をおねがいします
時刻と列車	10時20分のひかり
否定と訂正	いえ、禁煙席をお願いします

表 1: 受理可能な文型と対応する例文(例)

4.3 意図変換部

ユーザの予約に関する意図を表すフレーム“Reservation”を定義し、さらに各文型に対する Reservation フレームのテンプレートを定義

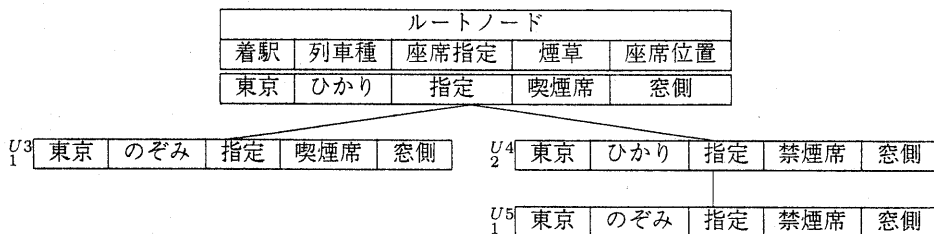


図 5: 履歴

した。このテンプレートに音声認識部で認識された単語をフレームのどのスロットに代入するかをあらわす畳み込み知識を記述した。

4.4 問題解決部

対象タスクにおける問題解決とは、ユーザから提示された条件に該当する新幹線を検索することである。新幹線の検索は、我々が開発した GUI による新幹線チケット予約システムの検索エンジンを、USHI 言語からの手続き呼び出しの形で利用できるように修正した。この作業は C++ で約 300 ステップの追加、修正を要した。これはステートメント数で数えて検索エンジンプログラム全体の約 16 検索エンジン呼び出す問題解決手続きの記述には USHI スクリプトで約 500 ステップを要した。

4.5 応答生成部

システムの応答文の生成では、補完した条件や認識結果を応答文に含める間接応答を用いる。システムはこの応答文によって検索結果をユーザに伝えるだけでなく、認識結果や現在の対話状態を知らせる。

4.6 実験

4.6.1 対話例

本システムを用いた対話の例を図 6 に示す。この対話例は 19 時の東京駅という状況を想定し、図 2 とおなじ方略をとった場合のシステムとの対話例である。

図 6 に示す発話 S1 から U3 まではシステムが対話を主導し、検索に必要な条件を質問する。提案手法では最初の検索条件が以降の検索に影響を与えるため、システムが確認発話 S3 を発話する。確認応答をユーザが肯定するまではユーザ

- S1: どちらまでご乗車ですか?
 U1: 京都までのひかり。
 S2: 禁煙席ですか? 喫煙席ですか?
 U2: 喫煙席。
 S3: 京都までひかりの喫煙席でよろしいですか?
 U3: はい。
 S4: 京都までひかりの喫煙席ですと、20 時 38 分発のひかり 273 号があります。
 U4: のぞみにしてください。
 S5: のぞみはみつかりません。
 U5: 禁煙席をお願いします。
 S6: 禁煙席ですと、19 時 30 分発のひかり 211 号があります。

図 6: 対話例

の訂正発話を受け付ける。その後、システムは検索を行い結果を応答 (S4) する。ユーザは U4 のようにのぞみの検索を要求したが、該当する結果は見つからなかった (S5)。そこでユーザは発話 U5 で煙草条件を禁煙席に変更する。この発話をうけて、システムは「ひかりの禁煙席」を検索する。これは煙草条件が書き換えられたことによって、最初の希望であった「ひかり」で予約できる可能性が生じるためである。

4.6.2 考察

提案手法では、ユーザにとっての各条件の重要度を条件レベルで定義し、条件レベルを使用して対話履歴を木構造の形で構築した。これによって省略を含むフレーム表現に対して、出来る限りユーザの希望に近い条件を補完することができた。一般に省略解決は文脈や文法の情報を用いて行われているが、提案手法では条件レベルを用いてフレームの関係を木構造にすることで、文脈や文法の情報を用いずに省略解決がおこなっている。また、木構造によって全ての対話履歴を持つことで、過去の検索条件の参照

解決も可能になる。

類似する研究として、電話による列車の情報検索、予約を目的とした音声対話システム ARISE[1]がある。ARISEもフレーム表現によって条件を表現し、対話履歴から省略解決を行っているが、条件の重要度やユーザの方略に対する考慮はなされていない。航空機の情報検索、予約を行う音声対話システム GALAXY[2]もフレームを用いて対話状態を表現している。このシステムはスタックを用いた対話履歴管理をおこない、直前の検索条件から条件を補完することで省略解決している。更にユーザ発話に特定の語が含まれる場合には継承しない条件(例：フライト番号が発話された場合、発着空港は継承しない)、という規則が予め与えられている。これはシステムに埋め込まれたタスク依存の方略であって、ユーザの方略によって変化するものではない。

本手法には音声の誤認識に対する対処が不十分な面がある。誤認識による条件書き換えが妥協と受け取られてしまうため、ユーザの意図に反する条件レベルが付与されることになり、以降の対話制御に支障を来す。システムの応答に対するユーザの反応や、ユーザの妥協のパターン等を分析し、誤認識に対処する必要がある。

対話中にユーザの希望条件が変更された場合に、それを的確に処理できないことも問題として挙げられる。「喫煙席がとれるなら列車種は問わないが、禁煙席しかとれないならのぞみが良い」という方略の場合、喫煙席に空席がある時より満席である時の方が、列車種条件の重要度が高くなる。ユーザがこのような方略で対話する場合には、一貫した重要度で対話する場合に比べ、対話ターン数が増加する。

4.7 EUROPA フレームワークの検証

EUROPA フレームワークを用いて、提案手法を音声対話システムとして実装した。その作業内容は4章のとおりである。本節ではこの作業を通して明らかになったフレームワークの利点、問題点について述べる。

USHI 言語の使用により、タスク依存の問題解決手続きを容易に変更することができた。新幹線検索エンジンを DLL の形で外部モジュールとして供給し、USHI 言語のもつ手続き呼び出しによって問題解決に利用したことでシステム構築にかかるコストを大幅に軽減できた。また、本システムでは、対話制御部を検索エンジンと同様に DLL 形式で追加し、USHI 言語により手続きを記述した。MINOS は一問一答の対話を行うシステムであったが、これによって複数ターンから構成される対話を扱うシステムを実現できた。一方、意図変換部にタスク依存の部分がある

ことが明らかになった。意図を表現するフレームは対象タスクに応じて定義しなおす必要があるが、今回の対象タスクへの適用に当たり C++ コード部分の修正が必要となった。USHI 言語におけるデータ型も問題となった。問題解決に用いる知識は整数、文字列、フレームとそのリスト型のデータで記述される。しかし USHI 言語にはそれらのデータ型間の変換能力が無いため、USHI 言語で記述した手続きと外部モジュールの手続きを併用し型変換を行う必要があった。

5 おわりに

本報告では、スロットフィル型対話におけるユーザの方略に注目し、条件レベルを用いてユーザの妥協を考慮した対話制御を行う手法について述べた。さらに提案手法を新幹線チケット予約に適用した。また、音声対話システム構築フレームワーク EUROPA を用いて提案手法を音声対話システムとして実装し、フレームワークの汎用性を検証した。

今後の課題として、対話の途中で変化する重要度への対応と、音声の誤認識の検出と復旧が挙げられる。またシステム実装の過程で明らかになったフレームワークの問題点について改良を行い、その汎用性を高める。

参考文献

- [1] L.Lamel, S.Rosset, J.L.Gauvain, S.Bennacef, M.Garnier-Rizet, and B.Prouts. The LIMSI ARISE system. *Speech Communication*, Vol. 31, pp. 339-353, 2000.
- [2] S Seneff, L.Hirshman, and V.Zue. Interactive problem solving and dialogue in the ATIS domain. In *Proc. of 4th DARPA Speech and NL Workshop*, 1991.
- [3] 河野恭之, 屋野武秀, 笹島宗彦. カーナビ音声対話システム MINOS の試作. 人工知能研資, SIG-SLUD-9901-4, 1999.
- [4] 屋野武秀, 笹島宗彦, 河野恭之. テンプレートハッシュ方式によるラティス解析方式と質問-応答システム. 情処学研報, 98-SLP-23-13, 1998.