

単一ドメインシステムの統合による 複数ドメイン音声対話システム

安田宜仁* 堂坂浩二* 相川清明* 上野晋一**

* 日本電信電話(株), NTT コミュニケーション科学基礎研究所

** 京都大学 情報学研究科

〒243-0198 神奈川県厚木市森の里若宮 3-1

yasuda@atom.brl.ntt.co.jp

あらまし

単一のドメインを扱う音声対話システムを情報検索技術で用いられるベクトル空間モデルに基づくシステム選択部を用いて, 少ない労力で複数のドメインを取り扱うことができる音声対話システムを構築する方法を提案する. さらに, 統合された対話システムが一貫性をもった文脈を保持し, 照応の解消などの文脈処理を行えるようにするために, 対話システム間の自然言語による通信によって意味表現の交換を行う方法を提案する. 音声対話システム作成のためのツールキット SpeechBuilder で作られた3つのシステムをこれらの手法を用いて統合することを試みた.

Developing a Multi-Domain Dialogue System by the Integration of Single-Domain Systems

Norihito YASUDA* Kohji DOHSAKA* Kiyooki AIKAWA* Shinichi UENO**

*NTT Communication Science Laboratories, NTT Corp.

**Graduate School of Informatics, Kyoto University

3-1 Morinosato-wakamiya, Atsugi, Kanagawa, 243-0198 Japan

yasuda@atom.brl.ntt.co.jp

Abstract

In this paper, we propose a method for developing a multi-domain spoken dialogue system by integrating single-domain systems with low effort. In this method, an appropriate single-domain system is selected by a system decision module that is based on a vector-space model, which is used in information retrieval technology. Furthermore, to enable the integrated system to perform context-dependent processing like anaphora resolution within a consistent context, we propose a method for exchanging system's understanding states among individual systems by natural language. We have attempted to integrate three single-domain systems developed using spoken dialogue system toolkit "SpeechBuilder" into a multi-domain system by the proposed methods.

1 はじめに

音声対話システムとは人間とコンピュータが音声による会話を通じて特定の仕事をを行うようなシステムである。近年の音声認識技術、音声言語処理の研究の発展により、音声対話システムは商用サービスとしても実現されてきている [1]。

また、音声対話システムのためのツールキットを使うことにより音声対話システムを作成することは身近になってきている。たとえば [2] は、web ブラウザを通して GUI で操作可能であり、音声言語処理の専門家でも比較的容易に音声対話システムを作成することを可能にしている。

しかし、現状では音声対話システムが世間で広く利用されているとは言い難い。音声対話システムの研究者でさえも日常的に音声対話システムを利用している人は稀である。この原因の 1 つとして音声対話システムが扱える話題の範囲が極めて狭いために、日常生活において音声対話システムを利用できる状況が少ないことが挙げられる。現在の技術では単一のシステムで広い話題の範囲を取り扱えるようにするためには大変な労力を伴う。ドメインの規模に応じて爆発的に増していく理解状態の組み合わせすべてに関してシステムが何らかの動作を行えるようにしなければならず、また、システムの一部への変更が及ぶ範囲が全体にわたることがあり、変更の影響を予測できない。大規模ドメインに対応した音声対話システムのためのツールキットも今のところ存在していない。

一方、広い話題の範囲を扱うための方法として、複数の単一ドメインシステムの切り替えをユーザからのコマンドによって行うシステム [1] も存在する。このようなシステムは構築は比較的容易であるが、ユーザにコマンドを覚えさせる必要もある。広範なドメインを扱えるようにした場合、話題切り替えのために覚えなければならないコマンドはドメインの大きさに応じて増大し、ユーザにそれらを記憶させるのは現実的な解とは言えない。

ユーザにシステムの切り替えを意識させないような、複数システムの組み合わせ方があれば、広範な話題を扱うシステムを小さい労力で作成することができる。

そこで、本稿では、複数の単一ドメインシステムと、対話の各局面においてどのシステムが発話すべきかどうかを決定するシステム選択部を組み合わせる方法を

提案する。システム選択部には、情報検索で用いられるベクトル空間モデル [3] を採用する。システムはユーザの発話に類似したコーパス中の発話を探すことにより適切な単一ドメインシステムを選択するので、ユーザはシステム切り替えのための特別なコマンドを覚える必要はない。さらに、システム選択の基準となるのはコーパスとユーザ発話のみなので各単一ドメインシステムの内容にかかわらずコーパスさえ収集すれば統合された複数のドメインを扱える対話システムを構成することが実現される。つまり、統合されたシステムを作る場合であっても各システムの作成にあたっては単一のドメインを扱う場合と同様に作成して良いので、既存の対話システムを複数ドメイン用に再利用することも可能になる。

このようにシステムの作成方針や作成者が異なるという意味で由来の異なるシステムを単純に統合した場合、各システムは他のシステムから文脈情報を受け継ぐことができないので、照応の解消や対話の履歴の参照などといった文脈に応じた処理を行うことができない。たとえば、カレンダー検索システムとスケジュール管理システムが統合された状況で、システム選択部が理想的に機能したと仮定した以下のような対話を考える。

U1: 「来週の木曜日は何日だっけ?」

S2(カレンダー): 「18日です。」

U3: 「その日の予定を教えてください」

S4(スケジュール): 「午後から山田部長との打ち合わせです。」

この対話を実現するには、発話 U3 における「その日」が「来週の木曜日」つまり「18日」であることがスケジュール管理システムに理解できる必要がある。

このような文脈に応じた処理を行うために少なくとも必要なことは、システム間で内部の理解状態を交換しあうことが可能となっていることである。

由来の異なる対話システム間では、ひとつの概念をどのように表現するかどうかはまちまちである。たとえば、あるシステムは日付を単一の属性を用いて「1999年4月3日」といったように表現するかもしれない。またあるシステムは同じ日付を複数の属性を用いて「1999年」「4月」「3日」のように表現するかもしれない。このため、単純に理解状態の内部表現を受渡ししようとしてもうまくいかない。そのため、従来、複数のシステムが情報を交換しあうためには、概念に対応する表現

形式を事前に定めておいたり、専用のプロトコルを使用する必要があった[4].

この点を解決するために本稿では、言語理解機構の理解規則を使った言い換えを用い、自然言語によって理解状態の交換を実現する方法を提案する。この方法は、多様な言い回しに対応するように作られているはずの言語理解機構を逆向きに利用し、システムが現在の理解状態を得るようなすべての可能なユーザ発話を再現することによって実現する。システムの理解状態を多様な言い回しで表現すれば、その中には別のシステムにとっても理解できる場合があることに期待した手法である。この手法は、理解状態の交換のために専用の機構を用意するのではなく、音声対話システムが備えている言語理解機構を用いるので、情報交換や言い換えのための新たな機構を用意する必要がない。

本稿では、以上の 1) ベクトル空間モデルによるシステム選択部と、2) 自然言語による理解状態の交換の提案に加え、これらの手法を用いて SpeechBuilder[2] の日本語版を用いて作成された3つの音声対話システムを統合することを試みた。

2 ベクトル空間モデルによるシステム選択

2.1 システム選択方法

既存の単一ドメイン対話システムへの最小限にとどめたまま複数ドメイン対話システムを実現するために、各対話システムの発話権を決定するシステム選択部との組み合わせを考える。各単一ドメイン対話システムは、システム選択部によって発話が認められている場合のみ発話を行うという点以外は、通常通りの動作を行う。

システム選択部の動作方法については、ユーザの発話内容に応じた適切な単一ドメインシステムを選択する過程を、検索要求に類似した文書を検索する情報検索に見立て、情報検索技術で広く用いられているベクトル空間モデルを採用する。このために各単一ドメインシステムに対するユーザの発話を収集する。

ただし通常の文書検索が適用される場面とは異なる点がある。主な違いとして以下のような点が挙げられる:

- 1). 大量のコーパスを収集することが困難である
- 2). ユーザ発話の認識誤りがある

これらの違いに関して以下のような対策を提案する:

- A). 検索単位(文書検索の文書に相当)を対話システムが設定した発話行為タイプとする
- B). 書き起しではなく認識結果を学習データとする

A) は 1) に対応するためである。発話単位で検索を行った場合に対して少ないデータで実効的な検索が行えることを狙っている。一方でシステムを単位として文書に見立てることも考えられる。しかし、たとえば承諾と否定など、まったく言い回しが異なる発話を単一のシステム内で受け付けることができる場合が多い。そのためシステムを単位として文書に見立てた場合、文書ベクトル同士が似通ってしまい、適切なシステムを選択できなくなってしまう恐れがある。

B) は 2) に対応するためである。実際の動作時と同様に認識誤りを含んでいる可能性があるものを学習データとすることにより、認識の誤りの傾向が同様ならば適切な選択が行えるものとする。また、コーパス収集の手間を減らすこともできる。

ベクトルの重みづけには情報検索において広く用いられている、索引語頻度と出現文書頻度を用いた TF-IDF 重みづけを用いる。

2.2 SpeechBuilder で作成されたシステムとの統合

SpeechBuilder 日本語版は、NTT と MIT によって共同開発された音声対話システム作成のためのツールキットである。システムの作成者は web ブラウザを通じて知識表現と言語的制約を入力することによって対話システムを作成する。

SpeechBuilder で作成された対話システムの構成を図 1 に示す¹。対話システムは、SpeechBuilder によって用意された音声認識、言語理解、音声合成を行う部分と、システム作成者によって作成された談話理解、対話制御、発話文生成を行うバックエンド CGI 部分の大きく 2 つの部分に分けることができる。この 2 つの部分の通信は、HTTP を用いて行われる。バックエンド

¹図は説明のために簡略化しているが、実際には GALAXY アーキテクチャ[5]で構成される

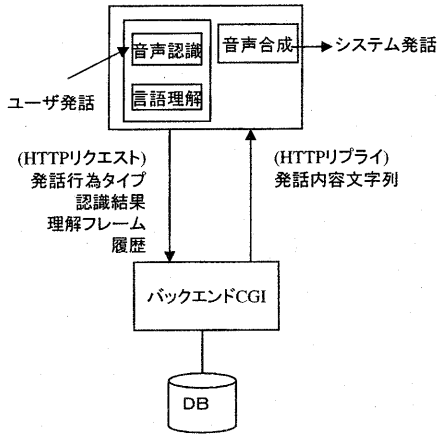


図 1: SpeechBuilder を用いて作成されたシステムの構成

CGI へはユーザー発話に休止がある度に発話行為タイプ、理解フレーム、認識結果および履歴が送られる。バックエンド CGI からは発話すべき文字列が返される。

SpeechBuilder を使って作られた 3 つの音声対話システムを対象に統合を試みた。これらのシステムは PDA として知られる携帯情報端末の各機能を実現するために作られたものである。以下は各システムの概要である。

- カレンダー検索システム
曜日や日付を尋ねる対話、祝祭日を尋ねる対話、ある日からある日までの日数を尋ねる対話
- スケジュール検索システム
ある日の予定の内容や場所を尋ねる対話、予定の内容から日時を尋ねる対話、面会相手から予定の内容や時刻を尋ねる対話
- アドレス帳検索システム
名前や部署から連絡先を尋ねる対話、名前から所属を尋ねる対話、指定した部署に所属する人の社内の部署名から所属する社員名を尋ねる対話

次にシステム全体の構成について述べる。先にも述べた通り言語理解部とバックエンド CGI との通信は HTTP を用いて行われ、CGI の応答として発話すべき文字列を返す。そこで、システム選択部を HTTP プロ

カレンダー	92.9%
スケジュール	73.9%
アドレス帳	82.8%
全体	83.2%

表 1: システム選択部の分類正解率 (10-fold CV)

キシとして実現し、選択されなかったシステムからの HTTP 応答の代わりに内容が空の HTTP 応答を音声合成部へ返すことで、既存のシステムには一切手を加えずに複数ドメイン対話システムを実現することができた (図 2)。

2.3 評価

上記の 3 つのシステムに対するユーザーの発話 1 システムにつき 100 ずつ集め、システム選択部によって再認識させた結果を用いて評価を行った結果を表 2.3 に示す。

短い発話に関して分類に失敗する例が多く見られた。短い発話には含まれる索引語が少なく、たとえ認識誤りがなかったとしても適切なシステムが選択できないことがある。たとえば、「明日は」という発話は、スケジュール検索システムへの発話にも、カレンダー検索システムへの発話にも見られるが、システム選択部は常にスケジュール検索システムを選択していた。

3 自然言語を利用した理解状態の交換

複数の対話システムが統合された場合において、照応の解消や履歴の参照といった文脈に応じた処理を行う場合、少なくとも複数のシステム間で理解状態を交換しあうことが必要である。

一般に、システムはユーザーの多様な言い回しに対してより頑健に動作するため、1 つの理解状態に対応するユーザー発話は複数存在する。たとえば、図 3 はスケジュール管理システムでの理解規則の一部である。これらは「○○の予定は何ですか」「○○の予定は」「○○の予定」「○○は」「○○はあいていますか」といった表現を受け付けるための規則であり、「○○」に相

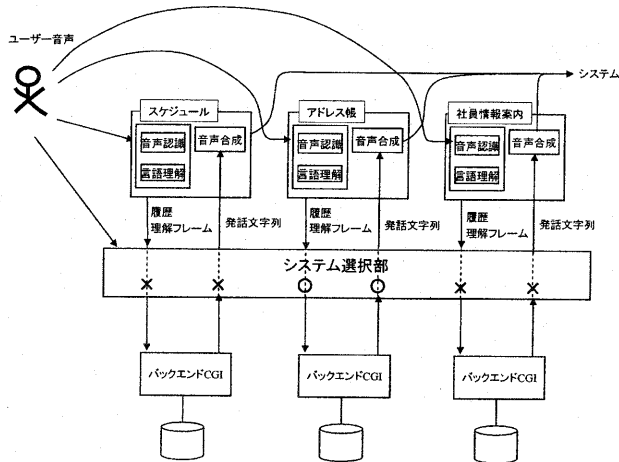


図 2: SpeechBuilder を用いて作成されたシステムとシステム選択部の統合

```

sentence full_parse @phrase list start== point_day $今日 今日
sentence full_parse @phrase list $の の
sentence full_parse @phrase list $予定 予定
sentence full_parse @phrase list $は は
sentence full_parse @phrase list $何 何
sentence full_parse @phrase list $です です
sentence full_parse @phrase list $か か

sentence full_parse @phrase list start== point_day $今日 今日
sentence full_parse @phrase list $の の
sentence full_parse @phrase list $予定 予定
sentence full_parse @phrase list $は は

sentence full_parse @phrase list start== point_day $今日 今日
sentence full_parse @phrase list $の の
sentence full_parse @phrase list $予定 予定

sentence full_parse @phrase list start== point_day $今日 今日
sentence full_parse @phrase list $は は

sentence full_parse @phrase list start== point_day $今日 今日
sentence full_parse @phrase list $は は
sentence full_parse @phrase list $あいて あいて
sentence full_parse @phrase list $います います
sentence full_parse @phrase list $か か

```

図 3: スケジュール管理システムでの理解規則の一部

当する部分が同じユーザ発話に対しては、発話行為タイプ (指定曜日の予定の問合せ) も含めてすべての理解状態をもたらす。

現在の理解状態をもたらしたユーザ発話の言い換えは言い換えのために特別な機構を用意するのではなく、言語理解の規則を逆向きを用いることによって行う。言語理解部がパターンマッチベースであれば、現在の理解状態中の発話行為タイプと同様の発話行為タイプをもつ理解規則を求め、規則中の変数を現在の理解状態に

対応する値によって置き換えることで言い換えを行うことができる。

SpeechBuilder で作られた言語規則に関して、部分解析を除いた部分に関して言い換えを行うモジュールを作成した。このモジュールにより、たとえば理解状態が

```
action=指定曜日の予定の問合せ&frame=(曜日=月曜日)
```

であり、理解状態中の「月曜日」という表現は、「月曜」と「月曜日」である場合、図 3 の規則たちによって

- 「月曜日の予定は何ですか」
- 「月曜日の予定は」
- 「月曜日の予定」
- 「月曜日は」
- 「月曜日はあいていますか」
- 「月曜の予定は何ですか」
- 「月曜の予定は」
- 「月曜の予定」
- 「月曜は」
- 「月曜はあいていますか」

が生成される。

これら言い換えされた文字列を受けとった単一ドメインシステムでは、システムの言語理解部を用いて理解フレームを作成する。これにより理解状態が交換されることが期待される。

4 おわりに

本稿では、複数の既存の単一ドメイン対話システムを統合して複数ドメイン対話システムを構成する方法について述べた。ベクトル空間モデルを用いたシステム選択部と組み合わせることで、単一ドメイン対話システムへの変更を最小限にとどめることができるという利点がある。SpeechBuilder で作られたシステムとの統合は、システム選択部を HTTP プロキシとすることで、既存のシステムへの変更を加えずに行うことができた。

さらに、各単一ドメインシステム間での理解状態の交換のために自然言語を用いる手法を提案した。この手法によって、由来の異なる対話システムの場合においても、理解状態の交換が可能になる場合がある。

今後の課題として、システム選択部への文脈の反映方法や、システム選択部に用いる言語モデルの検討が必要である。

今回 SpeechBuilder で作成された対話システムの統合を行ったが、異なる方法で作られたシステムの場合であっても、発話の可否が行えて理解規則を取り出せるような状況ならば比較的用意に複数ドメインシステムへ組み込むことが可能であると考えられる。

謝辞 日頃よりご指導いただく当研究所メディア情報研究部 村瀬洋部長、有益な示唆をいただくマルチモーダル対話研究グループの諸氏に感謝します。

参考文献

- [1] Vポータル <http://www.ntt.com/v-portal/>.
- [2] Glass, J. and Weinstein, E.: SPEECHBUILDER: Facilitating Spoken Dialogue System Development, in *Proc. Eurospeech*, pp. 1335-1338 (2001).
- [3] Salton, G. and Buckley, C.: Term-weighting approaches in automatic text retrieval, *Information*

Processing & Management, Vol. 24, No. 5, pp. 513-523 (1998).

- [4] Lin, B., Wang, H. and Lee, L.: A Distributed Architecture for Cooperative Spoken Dialogue Agents with Coherent Dialogue State and History, in *Proc. ASRU99* (1999).
- [5] Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P. and Zue, V.: GALAXY-II: A Reference Architecture for Conversational System Development, in *Proc. ICSLP*, pp. 931-934 (1998).