

# ドメイン限定質問応答システムの対話制御

デネッケ・マティアス 安田宜仁

日本電信電話(株), NTT コミュニケーション科学基礎研究所  
〒619-0237 「けいはんな学研都市」精華町光台 2-4  
{denecke,n-yasuda}@cslab.kecl.ntt.co.jp

## 概要

タスク指向型の対話システムに比べ、対話型ドメイン限定質問応答システムでの一番の問題は、タスクの構造が利用できないために、タスク指向型対話システムの場合のような単純な仮定が成り立たない点である。この問題に対し、ユーザ発話から抽出されたキーワードによる表現方法と、対話制御を実現する機械学習を組み合わせることによって解決する方法を提案する。

## Towards Dialogue Management for Restricted Domain Question Answering Systems

Matthias Denecke Norihito Yasuda

NTT Communication Science Laboratories, NTT Corp.  
2-4 Hikaridai, Seika-cho, "Keihanna Science City" Kyoto, 619-0237, Japan  
{denecke,n-yasuda}@cslab.kecl.ntt.co.jp

## Abstract

The main problem when going from task-oriented dialogue systems to interactive restricted-domain question answering systems is that the lack of task structure prohibits making simplifying assumptions as in task-oriented dialogue systems. In order to address this issue, we propose a solution that combines representations based on keywords extracted from the user utterances with machine learning to learn the dialogue management function.

## 1 はじめに

質問応答とは、情報検索技術を用いてユーザからの自然言語による質問に自然言語での回答を提供するタスクである。質問応答では、問題の制約がないことから、表層表現に対する統計的な手法が用いられる。ユーザが質問応答システムに対して適切なキーワードを含んだ質問をすればうまく動作する一方、質問があいまいな場合には正常な動作を期待できないことがある。

一方、音声対話システムに関する研究は、音声によって時刻表のような構造化されたデータベースへアクセスしたり、機器を操作する上での問題を対象としてきた。音声対話システムでは、バックエンドによって定義されたアプリケーションの範囲によってシステムのドメインは明確に与えられる。そのため、ユーザからの入力があいまいであったり、誤りを含んでいる場合でも、アプリケーションの範囲によって得られた制限を用いることにより、システムがユーザの意図を把握できるまで対話的に繰返し情報を要求するような対話制御を行うことができる。本稿で

は、対話型質問応答システムにおいてシステムからのユーザへの情報要求を生成する方法について述べる。

### 1.1 システム概要

我々は、質問応答システムと対話システムの2つの性質を併せ持つようなシステムとして、制限ドメイン質問応答システムを実装した。ここでの2つの性質とは、まず1つ目の性質は質問応答システムのように、構造化されていないテキストからなるデータベースを情報源とする点である。2つ目の性質は音声対話システムのように、ユーザからの要求内容があいまいであったり、きちんと内容が指定されていない場合に、システムからユーザに対して対話的に情報を要求する点である。

本稿では特に、制限ドメインにおけるシステム、つまり、旅行や投資情報といった、システムの取り扱い範囲が単一の大きなドメインに制限されている場合について述べる。語彙の大きさに注目すれば、このようなシステムはQAシステムと対話システムの間

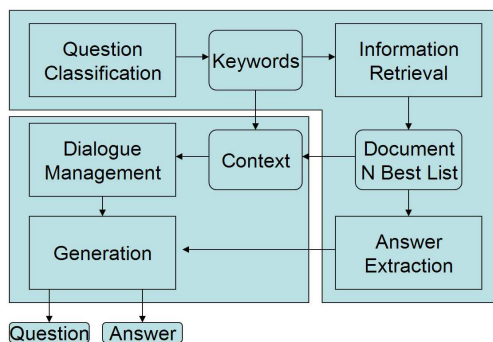


図 1: システム概観

的なものと位置付けられる。つまり、ドメインのモデル化を行って対話システムを構築するには現実的でないほど大きい。一方で、あらゆる種類の質問に対応できるほどは大きくはない。

図 1 に提案するシステムの質問応答システムと対話制御部との関係に着目した概観を示す。ユーザからの入力は、質問応答システムによって、キーワードの抽出と質問種類の分類が行われる。その情報はコンテキスト情報と統合された上で、情報検索エンジンによってその質問に合う新聞記事の  $n$ -ベストリストを取り出す。対話制御部ではこれらの結果を用いて、ユーザに対する行動を、上位の文書から取り出された回答の提示とするか、あるいは追加の情報を要求とするのかを決定する。

## 1.2 本稿で取り扱う課題

制限ドメイン質問応答システムはソフトウェアのトラブルシューティング [Kiyota et al.2002] のような問題解決の一種と捉えることができる。このような状況では対話的操作は必須のものとなる。なぜなら、ユーザからの質問内容において、システムが適切な回答を引出すために必要な情報がすべて含まれていることは期待できないからである。たとえば、[Kiyota et al.2002] におけるソフトウェアのトラブルシューティングでは、システムからユーザに対してソフトウェアのバージョンを尋ねる質問が頻繁に行われる。つまり、対話の履歴と現在解決しようとしている問題に応じて、ユーザからの追加の情報を得る必要があるのである。

本稿では、制限ドメイン質問応答システムでの対話制御関数の学習について述べる。このような関数はタスク指向型の対話システムにおける対話の管理の観点で意味がある。なぜなら、このような関数によって、認識器からの信頼性スコアのような、必ずしも正確とは言えないような情報を活用できる可能性があるからである。ドメイン限定質問応答システムにおいては、対話制御の学習はさらに重要な意味がある。なぜなら、対話システムにおいては比較的単純に決定できるような行動でさえ、対話型質問応答システムでは困難な場合が多いからである。たとえば、どのタイミングで対話を終えるかという行動について

は、タスク指向型対話システムでは、すべてのスロットが埋まっているかどうかで決定することができる。しかし、ドメイン限定質問応答システムでは、コンテキストと質問応答システムから得られた結果のみを用いて対話制御を行わなければならない。対話システムにおけるスロットのようなタスクの構造を用いることはできない。

対話制御によって決定される行動は、対話コンテキストと質問応答システムから得られた結果に基づいているので、学習のアルゴリズムはこのどちらも考慮したものである必要がある。より具体的には、対話コンテキスト  $c$  と検索結果文書の  $n$ -ベストリスト  $l$  を入力として適切な行動  $a$  を返すような対話制御のための関数

$$a = f(c, l)$$

を学習することが必要である。特に、我々が学習の対象としたいのは、次の 2 つの質問に答えられるような関数である:

1. ユーザからの追加の情報を得るべきか?
2. その場合どのように情報を要求するべきか?

我々はこの問題を分類問題、つまり、入力の対  $\langle c, l \rangle$  を適切な行動に対応付けられたクラスに分類する問題として捉える。この分類のために、サポートベクトルマシン [Vapnik1995] を使い、複数のカーネル関数の効率を比較する。

## 2 背景

### 2.1 関連研究

本稿での目的とは異なるが、質問と回答とのマッチングについては [Brill et al.2002] で述べられている。[Brill et al.2002] では質問の分類を避けるために、ユーザからの質問を宣言型の文に再構成し、検索エンジンによって検索された各文書が、ユーザからの質問文と似た構造を含むかどうか (もし含むならどの程度含まれているか) によって順位付けすることが提案されている。[Brill et al.2002] と我々が提案する手法とは 2 つの点で大きく異なる。まず、[Brill et al.2002] においてはユーザからの質問の再構成が質問内容の分類の代わりに用いられているのに対し、我々の手法では検索エンジンによる検索結果は、各文書がどの程度ユーザの質問に対して回答できているかを考慮した上でマッチングされる。次に、上記の違いから [Brill et al.2002] では、マッチングを行うために再構成を行うための手順が必要であるが、我々の手法では質問応答システムはブラックボックスとして扱うため、マッチングすべき質問と回答は直接得ることができ、再構成のための手順を必要としない。再構成の手順自体が正確に動作するというわけではなく、システムのエラーを引き起す原因となる可能性もあるので、再構成の手順の必要の有無は重要な違いである。

テキストデータベースに対して自然言語検索を行うような初期のシステムの 1 つとして、Wilensky ら

のシステム [Wilensky et al.1984] を挙げるができる。対象のテキストは Unix のマニュアルページからなり、システムの対話の能力については限定的であったものの、文脈情報を取り扱うことができた。近年では、Kiyota ら [Kiyota et al.2002] がコンピュータシステムのトラブルシューティングの手助けを行う対話的な質問応答システムを提案している。ユーザからの質問内容が、ヘルプ集のテキスト等を決定するための十分な情報を含んでいない場合には、対話制御部はあいまいな入力を検知し、追加の情報を必要に応じてユーザに要求する。情報の追加の要求は、簡略化対話スクリプトのような対話カードに基づいて行われる。もしユーザの質問が対話カード中の質問のリストのどれかにマッチした場合、対話カード中で、事前に定められた情報の追加の要求がユーザに対して行われる。しかしながら、この対話カードを用いるアプローチは対話カード作成のコストを考慮すると、オープンメインの対話型質問応答システムへ適用することは容易ではないと考えられる。

Hori ら [Hori et al.2003] はユーザからの入力がある度にユーザへの情報要求の質問の仮説を作成するような対話型音声質問応答システムを提案している。ユーザに対する情報要求の質問は、テンプレートをを用いており、ユーザからの質問の断片テンプレートに挿入することによって生成する。このテンプレートは質問応答システムによって決定されたユーザ質問の質問種別に応じて事前に準備されている。各情報要求の質問の仮説に対し、あいまいさ度合のスコアが計算される。このスコアはテンプレートに挿入されたフレーズから計算される。もし閾値よりもスコアが高いようなあいまいな情報要求の質問がある場合、その情報要求の質問を行い、そうでない場合には回答を生成する。

質問応答システムが返す文書の構造を増やすために、Small ら [Small and Strzalkowski2004] は文書に対してフレーム構造のようなアノテーションを行い、音声対話システムの対話制御の手法を用いて対話を進行する。

上記の対話的質問応答システムおよび我々の提案手法とは異なるアプローチとして、DeBoni と Manandhar はユーザからの質問が、前の質問内容に対する追加の質問なのかそうでないのかを区別する手法を提案している [Boni and Manandhar2003]。このことは我々の手法が、システム主導の確認する必要があるかどうかと確認すべき内容を決定しようとしていることと対照的である。対話的なシステムとして見た場合には、これらのアプローチの違いはシステムの主導権の強さの違いに過ぎないかもしれないが、そこで使われる根底の技術は異なっている。

## 2.2 サポートベクトルマシン

サポートベクトルマシン (SVM) は数々のパターン認識のアプリケーションで成功を納めてきた [Vapnik1995]。SVM は、明示的に特徴マップを計算するようなことをせず低い計算コストで高次元の特徴空間に適用できることがこの背景になっている。

SVM は教師あり学習の一種である。教師あり学習では、決定のための関数  $f: \mathcal{X} \rightarrow \mathcal{Y}$  をラベル付きデータ  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  を用いて学習する。本稿の場合、2 値分類すなわち  $\mathcal{Y} = \{-1, +1\}$  となる。

SVM は、ラベル付き訓練データから、クラス間のマージンが最大となるような正例と負例の分離超平面を決定する。分類の性能を上げるため、訓練データは元の空間ではなく、別の高次元の特徴空間において分離される。特徴空間での訓練データの写像  $\phi(x_i)$  を決定し、特徴空間での内積  $\langle \phi(x_i), \phi(x_j) \rangle$  を計算する代わりに、カーネル関数  $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$  によって、距離が暗黙のうちに計算されるため、特徴空間が高次元であるにもかかわらず効率的に計算することができる。

## 3 コーパス

ウィザードとユーザによる対話コーパスを収集した。ウィザードは、我々が作ろうとしているシステムのように振舞うよう指示されている。つまり、ウィザードはさらなる情報をユーザに要求するか、ユーザが望んでいると思われる情報を提供するかのどちらかのみを行うことができる。

被験者へは旅行に関する質問をするように指示した。旅行に関する質問は具体的には以下のいずれかである：可能な宿泊施設の種類の、現地特産の食べ物があるか、有名な観光地やイベント、目的地が歴史的に有名なところであるか、もし歴史的に有名なところならばどのような歴史があるのか。質問の種類を制限したのは、ある種の質問は対話的ドメイン限定質問応答システムの質問としては向いていないためである。特に、「ホテル 3 泊分の料金はいくらですか?」といった単純な事実に関する質問はタスク指向型の対話システムの方がうまく取り扱うことができるため、本稿での対象とはしない。

被験者 20 名であり、どの被験者もコンピュータの専門家ではなく、また、この種の実験の未経験である。1 被験者あたり 8 ないし 15 対話、全体で 201 対話を収集した。各対話の長さは 11 ターンから 84 ターンの間で、中央値は 34 ターンだった。総計 6785 ターンのうち 3299 はユーザのターン、3486 はウィザードのターンであった。前述の通り、データ収集時の対話の進行に関して強い制限を設けたため、ユーザのターン数とウィザードのターン数が同一 (ユーザが対話を終えた場合、14 対話) か、ウィザードのターン数が 1 ターンだけ多い (ウィザードが対話を終えた場合、187 対話) かのいずれかであった。対話の開始部分の例を図 2 に示す。

### 3.1 コーパスへのアノテーション

1.1 節で述べた通り、システムが質問を生成する目的は、ユーザからの追加の情報を検索のための情報として追加して活用することだった。収集したコーパスにおけるウィザードからユーザへの質問は、質問のあいまいさという点で程度の異なるものが含ま

W: こちらは旅行ガイドです  
 U: 京、京都で1日ゆったりとお寺巡りをしようと思っているんですが、京都駅から、お寺を、お寺巡りをする、効率的というか、こうしたらいんじゃないかっていうコースはありますか  
 W: はい、それでは京都駅周辺でお調べ致します。京都駅周辺をゆったりと回る、でよろしいですか  
 U: はい  
 W: それではJR 京都駅から東寺、梅小路蒸気機関車館、門屋もてなしの文化美術館、西本願寺、コウセイジ、京都タワー、そしてJR 京都駅に戻ってくるというコースはいかがでしょう

図 2: 対話コーパスの一例. U, W はそれぞれ、ユーザの発話、ウィザードの発話を示す.

れている。適切な質問の生成を行うためには、このようなあいまい度合をコントロールする必要がある。我々はあいまいの程度を定義し、このアノテーションを行った。各対話をユーザ発話とそれに続くウィザード発話の対に分け、各発話対に対して人手によって、下記に定める8つのクラスのうちのいずれのかをアノテートした。(ここで、 $ut$ ,  $wt$  はそれぞれユーザ発話、ウィザード発話を示す):

```

if ( $wt$  は質問ではない)
  if ( $ut$  が yes/no 質問) 発話対のクラスは 1
  else if ( $ut$  が 列挙質問) 発話対のクラスは 2
  else if ( $ut$  が WH 質問) 発話対のクラスは 3
  else 発話対のクラスは 4
else
  if ( $wt$  が yes/no 質問) 発話対のクラスは 5
  else if ( $wt$  が 列挙質問) 発話対のクラスは 6
  else if ( $wt$  が WH 質問) 発話対のクラスは 7
  else 発話対のクラスは 8
  
```

アノテーションの結果、ラベルの分布が非常に偏っていたため、クラス2とクラス3、およびクラス6とクラス7をそれぞれ1つのクラスとしてまとめた。さらに、クラス4とクラス8の事例をコーパスから取り除いた。この結果、4つのクラスが得られた元データとこの処理後の数は以下の通りである。

ラベル	1	2	3	4	5	6	7	8
元データ	530	2	2209	3	454	23	76	2
処理後	530		2211	0	454		99	0

## 4 表現方法と対話制御

SVM ベースの手法を用いる場合、適切なカーネルを用いることにより背景知識を取り込むことができるので、カーネル関数の選択は極めて重要である。カーネル関数の選択について述べる前に、我々が学習を行おうとしている関数  $f$  について詳しく述べる。前述の通り、我々は直前のユーザ発話までを含んだ対話コンテキスト  $c$  と文書の  $n$ -ベストリスト  $l$  を対話制御部によって行うべき行動を決定する分類器  $f(c, l)$  によって分類しようとしている。全体の目的は何の制限もないテキストコーパスから抽出した回答をユーザに提示することである。

以下では、コンテキストの表現  $c$  と回答リスト  $l$  からどのように特徴を取り出すかについて述べる。

### 4.1 コンテキストの表現方法

コンテキストの表現方法として、ユーザ発話およびそこから取り出される情報を保持するような単純な談話表現を用いる。具体的には、時刻  $T$  におけるユーザのターンを  $ut_T$ 、質問文解析モジュールによって取り出されたキーワード集合  $kw(ut_T)$ 、質問種別判定器によって決定された質問種別  $qt(ut_T)$ 、質問応答システムから返された新聞記事の  $n$ -ベストリスト  $A_T = \langle a_T^1, \dots, a_T^n \rangle$  を保持する。関数  $answer$  は、質問応答システムが、ユーザの質問に回答していると考えている文書中から文を選択するような関数である。

時刻  $T$  における対話状態は3つのリスト  $\langle ut_1, \dots, ut_T \rangle$ ,  $\langle kw(ut_1), \dots, kw(ut_T) \rangle$ ,  $\langle A_1, \dots, A_T \rangle$  によって与えられる。対話状態は、直接分類器が分類の対象とするには多くの情報を保持しているため、次節以降では分類に必要な特徴を選択するいくつかの方法について議論する。

音声対話システムにおけるユーザからの入力とコンテキストは、典型的には、スロット集合への値の格納によって表現される。このような表現を対話型ドメイン限定質問応答システムへ適用することは容易ではない、なぜなら、対話状態を保持するために必要なスロットを事前に決定することは困難だからである。我々は対話状態の表現を分類の目的に使用しようとしているので、カーネルベースの手法でよく用いられる *bag-of-word* 表現を用いることを提案する。

#### 4.1.1 コンテキストのみによる表現

タスク指向型対話システムにおいては、コンテキストの表現は、スロットの値を蓄積しておくことで実現できたが、しかし対話型ドメイン限定質問応答システムにおいては、そのような方法は使うことができない。なぜなら、事前に定められたスロットの集合は与えられていないからである。そこで、情報検索の過程と関連の深い情報を記憶するという、一般化されたスロット表現手法を提案する。文のを形態素解析した結果による *bag-of-word* 表現を用いる。すべての内容語は *bag-of-word* に追加される。しかし、その内容語は分類のためには詳細すぎたり、あいまいすぎたりすることも考えられる。そこで、内容語中の名詞については、オントロジより得られたその名詞の下位語をすべて *bag-of-word* に追加する。

まとめると、時刻  $T$  におけるコンテキストの表現

方法は以下のようになる

$$C_T = Cl \left( \bigcup_{t=1}^T kw(ut_t) \right) \quad (1)$$

ここで、 $Cl$  は単語集合から、上位語も含んだより大きな集合への対応付けのクロージャ関数である。直感的には、この集合は、これまでのユーザ発話から得られたすべてのキーワード、およびクロージャによって得られた各キーワードの上位語からなるような集合である。

#### 4.1.2 検索結果を用いて選別した表現

情報検索システムでは、情報源に関する補完的な情報として、大規模テキストコーパス中での単語間の共起に関する情報が用いられることがある。たとえば、2つの単語が同一の文書中に現れる回数が偶然よりも多いという事実を類義語に起因する問題の解決のために用いるといった具合である。分類を行うためのコンテキストの表現するにあたり、我々もこのような考え方の利点を取り入れる。ここでは、間接的な方法ではあるが、検索結果の文書から関連する単語を取り出し、それをコンテキスト表現に追加方法によって実現する。つまり、旅行対話ドメインにおけるこのような関連単語を取り出す際には、訓練コーパス中に出現するすべての名詞  $N$  と文書中の共通部分を考えるという近似的な手法である。つまり、以下の式を得る。

$$C_T = Cl \left( \bigcup_{t=1}^T kw(ut_t) \cap N \cap Cl(a_T^1) \right) \quad (2)$$

#### 4.2 回答の表現方法

本節では、コンテキストの表現に加えて、検索された文書がどの程度ユーザの質問に回答するかということを表示する。ここで、このような適合度は、コンテキストの表現と検索された文書との関係によって表すことができるものと仮定する。コンテキストに含まれる情報と、検索された文書との関係を表すための方法として、それぞれを bag-of-word として表現した際の共通部分として関係を定義する、つまり、

$$A_T = Cl \left( \bigcup_t kw(ut_t) \cap answer(a_T^1) \right) \quad (3)$$

ここで、 $answer$  は文書を入力として、ユーザの質問に対応するような文を返すような関数であり、質問応答システムによって決定される。

#### 4.3 対話制御

これまでに述べてきた表現方法に応じた対話制御の方法について述べる。ある表現形式を選択すると

いうことは、対話状態の2つ側面を捉えるという意図を示すものであった。つまり、(1) これまでに集まった情報はどの程度詳細か、(2) 現在の  $n$ -ベクトリストはどの程度ユーザの質問に対して良いものなのか、の2つの側面である。前者についてはコンテキストの表現を用い、後者については収集された情報と  $n$ -ベクトリスト中の上位の文書との関係で表現した。

4.1.1 節、4.1.2 節で提案した表現方法と4.2 節で応答の表現方法を組み合わせることによって、分類器の入力を得る。

#### 4.4 情報要求のための質問

ユーザからの質問  $q$  と、その質問に対する質問応答システムからの回答の  $n$ -ベクトリスト  $\langle a_1, \dots, a_n \rangle$  が与えられると、分類器は  $q$  と各  $a_i$  との対  $A(q, a_i)$  について、 $a_i$  が  $q$  に対する回答候補であるかどうかを判定する。もし、 $a_i$  が  $q$  に対する回答候補と判定された場合は、その回答は情報要求のための質問内容として追加する。

情報要求のための質問について、もう1つ重要な問題として、分類器によって分類された行動をどのようにして自然な発話表現にするかということが挙げられる。本稿ではこの問題については詳しく論じないが、現在、コーパス中からの適切な文を選択し必要に応じてその文をコンテキストに合うように変更し、ユーザに提供するような事例ベースの発話生成の方法を検討中である。この発話生成方法では、類似したコンテキストの対話を用例のコーパスから検索し、その類似したコンテキストの対話の中で発話された情報要求発話を選択する。その情報要求発話中の内容語を判定し、現在のコンテキストでの内容語によって置き換えた上で質問する。質問は3.1 節で述べた質問の種類も考慮した上で選択される。事例ベース発話生成について詳しくは [Denecke and Tsukada2005] にて論じる予定である。

### 5 評価

#### 5.1 質問応答システム

実験における提案システムの質問応答システム部分として、日本語質問応答システム SQA [Sasaki et al.2002] を用いた。質問応答システム構築のための元テキストとして、1991年から2002年までの朝日新聞を用いた。

#### 5.2 評価方法

2種類のSVMの訓練用に、151対話を用いた。3.1で述べた処理後のコーパスを用いたので、この問題は4クラスの分類問題と見なすことができる。各SVMを用いて訓練に用いていない対話をテストデータとして、対話のラベルを予測した。1種類のSVM毎に、4つのクラスの中から1つを分類するような (one-

	コンテキスト	
	式 (1)	式 (2)
精度	59.22%	60.22%
再現率	62.4%	65.4%

表 1: Results for the four classifiers. Shown are precision and recall.

against-all)4 つの分類器を作成した。どの実験でもパラメータ  $C$  は 1 とした。

なお、この実験の目的は、利用者研究の際に用いる対話制御のための適切な分類器を決定することにある。

### 5.3 結果

2 つの分類器の結果を表 1 に示す。式 (2) によるコンテキストから得た特徴と、式 (3) による回答から得た特徴との組み合わせの結果は良好であった。

SVM で用いるカーネルとして多項式カーネル以外に、ラジアルカーネルとニューラルカーネルを使うことを検討してみたが、いずれも問題があった。まず、ラジアルカーネルの場合は、精度の点では優れていたが、観測数の少ないクラスについての再現率は 30% 以下であった。ニューラルカーネルを用いたサポートベクトルマシンの訓練は収束しなかった。これは、 $C$  を非常に大きな値とした場合でも同様であった。

## 6 議論と今後の課題

本稿では、対話的ドメイン限定質問応答システムにおける対話制御に関する技術について述べた。対話のコンテキストについての表現と質問応答システムが検索した文書についての表現を用いて対話制御のための分類器を学習する方法を提案した。

タスク指向型の対話システムと比較した場合の対話型ドメイン限定質問応答システムでの一番の問題は、タスクの構造が利用できないために、タスク指向型対話システムの場合のような単純な仮定が成り立たない点であった。この問題に対し我々は 2 つの要素からなる解決法を提案した。1 つは、ユーザ発話から取り出したキーワードと、オントロジーから取り出した概念からなる表現方法を採用したことである。このような表現方法は、スロット表現を一般化したものと見なすことができる。もう 1 つは、対話制御の関数の学習のために機械学習を取り入れたことである。今回、既存のコーパスを用いてシステム行動を予測するという評価を行なったが、今後の課題として、被験者との対話実験を通して提案法の有効性を検証したい。

## 謝辞

事例ベースの発話生成システムについてご協力くださった鈴木琢也氏に感謝します。また、日頃ご議論くださる知識処理グループの磯崎秀樹氏およびグループ員のみなさまに感謝します。

## 参考文献

- [Boni and Manandhar2003] M. De Boni and S. Manandhar. 2003. An analysis of clarification dialogue for question answering. In *Proceedings of HLT-NAACL 2003, Edmonton, Canada*.
- [Brill et al.2002] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. 2002. Data-Intensive Question Answering.
- [Denecke and Tsukada2005] M. Denecke and H. Tsukada. 2005. Instance-Based Generation for Interactive Restricted-Domain Question-Answering Systems. In *Submitted*.
- [Hori et al.2003] C. Hori, T. Hori, H. Tsukada, H. Isozaki, Y. Sasaki, and E. Maeda. 2003. Spoken interactive odqa system: Spiga. In *Proc. of the 41th Annual Meeting of Association for Computational Linguistics (ACL-2003), Sapporo, Japan*.
- [Kiyota et al.2002] K. Kiyota, S. Kurohashi, and F. Kido. 2002. "dialog navigator": A question answering system based on large text knowledge base. In *Proceedings of The 19th International Conference on Computational Linguistics, Coling 2002, Taipei, Taiwan*.
- [Sasaki et al.2002] Yutaka Sasaki, Hideki Isozaki, Koji Kokuryou, Tsutomu Hirao, and Eisaku Maeda. 2002. Ntt's qa systems for ntcir qac-1. In *Working Notes of the Third NTCIR Workshop Meeting*, pages 63–70.
- [Small and Strzalkowski2004] S. Small and T. Strzalkowski. 2004. Hitiqa: Towards analytical question answering. In *Proceedings of The 20th International Conference on Computational Linguistics, Coling 2004, Geneva Switzerland*.
- [Vapnik1995] V. N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.
- [Wilensky et al.1984] Robert Wilensky, Yigal Arens, and David Chin. 1984. Talking to unix in english: an overview of uc. *Communications of the ACM*, 27(6):574–593.