

## 複数の車内機器操作と雑談を扱える マルチタスク音声対話システムの ユーザビリティの向上

尾崎 健太郎<sup>†1</sup> 小暮 悟<sup>†1</sup> 甲斐 充彦<sup>†1</sup>  
小西 達裕<sup>†1</sup> 伊東 幸宏<sup>†1</sup>

本研究では、複数の新規タスクを容易に追加可能で、どのタスクも選択されなかったときには雑談として対応可能なマルチタスク音声対話システムの開発を行った。雑談では、あるタスクに関連する単語が発話された場合にそのタスクを励起するタスク誘発、テンプレートに沿った応答を返すテンプレート応答、ユーザ発話をそのまま返すオウム返しそれぞれ可能である。実際に、目的地検索、楽曲検索、ニュース検索の各タスクを適用した。410 発話を用いた評価実験において、タスク判定率は 80.2%、認識率はタスク全体で 66.0%（タスク判定率が 100%と仮定した場合は 73.4%）となった。また、システムのユーザビリティを向上させるためのシステム提案手法について検討・実装したので報告する。

### Usability Improvement of Multitask Spoken Dialogue System

KENTARO OZAKI,<sup>†1</sup> SATORU KOGURE,<sup>†1</sup> ATSUHIKO KAI,<sup>†1</sup>  
TATSUHIRO KONISHI<sup>†1</sup> and YUKIHIRO ITOH<sup>†1</sup>

In this paper, we have developed multitask spoken dialogue system that can add new task easily later and deal with a small talk. A small talk implemented by us have three functions; template response, task lead, parrot fashion. We implement car navigation, audio control and news retrieval. So, we evaluate the performance of this system, task decision rate is 80.2%, and speech recognition rate is 66.0% (if task decision is all correct, speech recognition rate is 73.4%). Moreover, we proposed and implemented some mechanisms that can offer favorite topic to users from this system for improving usability of this system.

### 1. はじめに

カーナビゲーションシステムやオーディオ機器操作などの車内機器を音声で操作できる環境がすでに実用化されている。それらのシステムを同時に扱うことはすでに可能であるが、片方のシステムの機器操作状況に応じて、もう片方のシステムの操作に影響を与えるようなシステム、つまり、あるタスクの達成までに入力した条件が他のタスクの達成に役立つシステムは、現時点においては実用化されていない。そこで、我々は車内機器操作を対象とした、複数の目的指向タスクを同時に扱えるマルチタスク音声対話システムの開発を進めてきた。タスクを単一もしくは複数扱う場合に、特定のタスクに限定されたシステム開発は好ましくなく、タスクごとに用意されるタスク情報に従ってそれぞれのタスクの対話が可能になることが望ましい。本研究では、システムのコア部はタスクに依存せず、タスクに依存する部分はタスク固有の情報を別途用意することで新規タスクも扱える環境を構築した。また、複数のタスクを扱う場合は、入力された発話がどのタスクに対するものなのかを判定する必要がある。タスク判定については、目黒ら<sup>1)</sup>は、シングルタスクにおいて、ある発話はそのタスクに関する発話かそうでないかを 98%の確率で判断できるタスク内発話判定手法を提案している。また、ユーザ発話からトピックの推定をし、対話履歴と併用してタスク判定を行う研究<sup>2)</sup>や、初回発話と 2 回目以降の発話に分けてタスク判定率を評価する研究<sup>3)</sup>などがある。本研究のシステムでは、タスクごとに語彙、文法を用意し、音声認識結果をタスクごとに得た後で、1 位の文の文尤度比（その認識結果がどの程度確からしいかを示す指標）の大小、その他、前のタスク判定結果等を利用して、現発話がどのタスクに対する発話なのかを判定する。また、人間同士の対話を考えた場合、複数の目的指向タスクを直接扱うだけでなく、雑談をしつつ、雑談の中の話から、いずれかの目的指向タスクが励起する場合が考えられる。上記の考えに基づき、情報が得られる雑談を情報探索雑談として利用し、トピック遷移の検出法を調査している研究<sup>4)</sup>もある。本研究の雑談としては、雑談の発話があるタスクに関係がある場合に、そのタスクを誘発するタスク誘発、用意したテンプレートに沿った発話がされた場合に、そのテンプレートに沿って応答を返すテンプレート応答、ユーザ発話を聞き返すオウム返しの 3 つの応答方法を考案した。現在、カーナビゲーション（以下、目的地検索）、楽曲検索、ニュース検索、雑談が可能な車内機器操作音声対話システム

<sup>†1</sup> 静岡大学  
Shizuoka University

ムの試作が完了し、システムの評価をしたところ、システムのタスク判定率は 80.2%、認識率はタスク全体で 66.0% (タスク判定が正しくできると仮定した場合は 73.4%) になった。

また、本研究では、既存システムのようにユーザ発話を一方的に待つのではなく、システム側から話題を提案することでユーザが必要としている情報を提供できると考える。発話情報以外のシステム情報やユーザの情報を使い、システム理解と応答文に反映させることでより円滑にタスク達成が可能なシステム提案を考案した。

## 2. マルチタスク音声対話システム

### 2.1 マルチタスク音声対話システムの概要

我々の開発したマルチタスク音声対話システムは、タスクに依存しないように一般化されており、データを入れ替えることでタスクの追加、削除が可能なシステムである。図 1 はマルチタスク音声対話システム全体と、各モジュール間で送信する情報と値を示している。本研究のシステムは、対話モジュール、音声認識部、言語理解部、DB 検索部、BlackBoard の、5 つのモジュールから構成されている。BlackBoard は、データの保存をしており、それ自体が個別で動作することはない。図 1 の 1. - 8. は、システムの流れを示しているので、順に説明していく。

まず、ユーザからの音声入力を受け付け、対話モジュールから音声認識部に音声認識コマンドが送られる (図 1 の 1)。その後、音声認識部で音声認識が行われる (図 1 の 2)。Julius の出力である音声認識結果の出力をもとに、信頼度 (2.3 音声認識部にて後述) 生成を行う (図 1 の 3)。その後、音声認識部から音声認識完了コマンドと音声認識結果及び信頼度生成結果のデータが送信される。送信された音声認識結果と信頼度生成結果は、対話モジュールにて BlackBoard に送信される。次に、対話モジュールから言語理解部に言語理解コマンドが送られ、言語理解部にて、BlackBoard から、先程送信した音声認識結果・信頼度生成結果と各タスクのカテゴリ情報を受け取り、各タスクの単語のスコア (2.5 言語理解部にて後述) 生成が行われる (図 1 の 4)。スコア生成後、単語のスコアが最大となった単語の情報が BlackBoard に、言語理解完了コマンドが対話モジュールにそれぞれ送信される。次に、対話モジュールにおいて、タスク判定 (2.4 タスク判定にて後述) が行われ、どのタスクについて発話されたか判断される (図 1 の 5)。タスク判定後、BlackBoard に保存されている、判定されたタスクの応答文生成までの処理が書かれたルール群を読み込む。ルール群はタスクごとに用意されており、1 つのルールは ruleID: 適用条件, true 時の処理, true 時に次にチェックする ruleID, false 時に次にチェックする ruleID という情報を

持っている。タスクの初回起動時には、あらかじめ決めてある start ruleID からルールのチェックが行われる。ルール処理では次にチェックする ruleID が終了になるまで、ルールのチェック作業を継続する (図 1 の 6)。応答文生成までのルール処理の間で、DB 検索処理が必要になった場合は、対話モジュールから DB 検索部へ DB 検索コマンドが送信される。DB 検索部では、BlackBoard から言語理解結果を取得し、そのデータをもとに DB 検索を行う (図 1 の 7)。DB 検索終了後、DB 検索結果を BlackBoard、DB 検索終了コマンドを対話モジュールに送信する。対話モジュールでは、ルール処理終了後に、生成された応答文をユーザに提示する (図 1 の 8)。以上が、ユーザ発話から応答文提示までの流れである。

対話モジュール、音声認識部、言語理解部、DB 検索部、BlackBoard の各モジュール詳細については、以下で説明する。

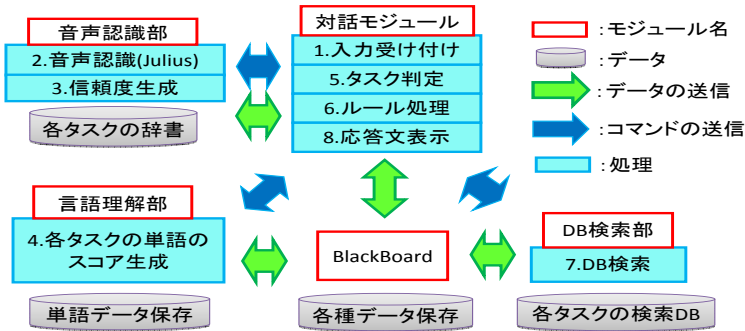


図 1 システム概要  
Fig. 1 Outline of system.

### 2.2 対話モジュール

対話モジュールでは、ユーザからの音声入力を受け付け、各モジュールへの命令コマンド送信、ユーザへの応答文提示を行っている。また、対話モジュールでは、BlackBoard から各タスクの情報を受け取り、タスク判定とタスクごとに定められた応答文生成までのルールの実行および、そこに記述されている、条件を満たした場合の処理であるプリミティブ処理を実行している。プリミティブ処理は、各目的指向タスクにおいては、タスクに依存されない共通された処理を、雑談に関しては固有の処理を実装している。

### 2.3 音声認識部

音声認識部では、Julius<sup>5)</sup> を使って認識した音声認識結果から、各タスクごとに、認識された単語の信頼度と1位の候補文の文尤度比(2.4タスク判定にて後述)を計算する。音声認識には各タスクごとに用意した文法制約の文法と、雑談用に用意した trigram 言語モデルのすべてを Julius に入力したうえで、全ての文法による認識結果を得ている。認識された単語の単語信頼度は、その単語が存在する候補文の文確率(各候補文の文尤度の和で各候補文の文尤度を正規化した値)を総和して疑似的に事後確率を推定することで計算している<sup>6)</sup>。

### 2.4 タスク判定

タスク判定は、音声認識部より得られる、タスクごとの音声認識結果のそれぞれのタスクの1位の文の文尤度比を用いて、図2のアルゴリズムで1ターンごとに行われる。まず、タスク  $t_i$  の認識結果1位候補文の文尤度比  $lhr_i$  は、(1)式で計算される。 $Lh_i$  はタスク  $t_i$  の認識結果1位候補文の文ログ尤度<sup>\*1</sup>、 $Lh_u$  は任意の音節列を認識可能な文法で認識(挿入ペナルティは0)した時の認識結果1位候補文の文ログ尤度である。

$$Lhr_i = Lh_i - Lh_u \quad (1)$$

本研究では、前回選択されたタスクは継続されやすいという経験則に基づき、タスク判定時に前回選択されたタスクが選ばれやすい仕組みを導入している。これを実現するため、前回判定されたタスクの認識結果1位候補文の文尤度比に、そのタスクが新規に選択された時のタスク判定の確からしさを加えている。その時のタスク判定の確からしさは、そのタスク判定時における、文尤度比が一番高いタスクの文尤度比と文尤度比が二番目に高いタスクの文尤度比の差に をかけた値を用いて求めている(実装時には  $= 0.5$  を用いた)。最後に、認識時の入力フレーム数の大小により選択される閾値よりも、選択されたタスクの文尤度費が高ければそのタスクが選択され、閾値以下であれば、どのタスクも選択されず雑談と判定される。ただし、前回の対話戦略で確認(「はい」「いいえ」を要求する)が行われ、実際にユーザが「はい」「いいえ」を発話したと考えられる場合には、前回のタスク判定結果をそのまま継続して利用する。

### 2.5 言語理解部

言語理解部は、目的指向タスクの発話がされたらタスク判定された場合の、対話開始からタスク達成の間に認識された単語を文脈として保持する。この文脈には、その単語がそれま

\*1 認識時に用いた挿入ペナルティを考慮し、元々の文尤度から、挿入ペナルティ  $\times$  (単語数-1)の値を引いている

```
int task_distinction(double lhr[])
{
    if (turn > 1) lhr[pid] = lhr[pid] + *diff;
    lhr[] を降順ソートし、結果を r[i] に保存
    if(pid != r[1]) diff = lhr[r[1]] - lhr[r[2]];
    if(lhr[r[1]] > th ) return r_i;
    else return ELIZA;
}
```

tid : タスク id  
lhr[tid] : タスク tid の文尤度比  
turn : ターン数  
r[i] : lhr[] が i 番目に高いタスクの tid  
pid : 前回判定された tid  
ELIZA : 雑談を表す id  
th : フレーム数が 190 以下の時 -80  
フレーム数が 190 より大きい時 -160

図2 タスク判定アルゴリズム  
Fig.2 Task decision Algorithm.

でに発話されたであろう可能性を示す単語スコアも一緒に保持されている。言語理解部の単語スコア生成には、すでにある文脈上の各単語と最新発話中に含まれる単語信頼度付きの単語の関係から、文脈上の各単語の単語スコアを増減させることで文脈の更新を行う。これらの更新は、前回の応答文中で使われた単語に対して、最新発話中の単語がどのような立場をとるかも考慮して行われる。また言語理解部では、前述の詳細化と訂正の度合いを使って、発話タイプを判定する。言語理解部は、単語スコアの値と、タスクごとに定められたカテゴリの関係、および発話タイプを考慮して、カテゴリ判定を行う。

カテゴリ判定では、それまでにどのカテゴリが入力されたかを判定する。各タスクにおいて定められたカテゴリ数から求められるすべてのカテゴリの組み合わせパターンに対して、カテゴリスコアを求め、カテゴリスコアが最大になるカテゴリ組み合わせをカテゴリ理解結果とする。そして、そのカテゴリ組み合わせのカテゴリスコアを計算するときに使った単語の組み合わせを言語理解結果として生成する<sup>6)</sup>。

### 2.6 DB 検索部

DB 検索部は、対話モジュールのルール判定において検索実行のプリミティブ処理が実行される際に呼び出される。DB 検索命令が送信された場合に、言語理解部で得られる言語理解結果の単語の組み合わせと BlackBoard から取得した SQL 文の作成情報を用いて、データベース検索用の SQL 文を作成する。その後、各タスクごとに用意されたデータベースから検索を行い、検索結果を BlackBoard に出力している。

### 2.7 BlackBoard

BlackBoard は単体で何かの動作をするわけではなく、各タスクの応答文生成ルール、言語理解結果、検索結果、タスク情報などのデータを保存している。保存されるデータは主に、以下のように大別される。対話が進むにつれて、変更されるデータに関しては、BlackBoard

データの送信が可能な各モジュールから随時 BlackBoard に送信され、データが更新される。

- タスク情報 (カテゴリ情報, DB 検索情報, 応答文生成規則情報)
- ルール情報 (ルール群情報, ルール判定に利用する変数)
- 認識情報 (肯定, 否定等のフラグ, 音声認識結果)
- 言語理解情報 (言語理解関係変数, 言語理解結果)
- 検索情報 (DB 検索時の SQL テンプレート, 検索結果)
- 雑談情報 (雑談応答用テンプレート (3.3 雑談にて後述), 雑談関係変数)

### 3. タスクの適用

#### 3.1 本研究のタスク

本研究では, 目的指向タスクとして, 目的地検索 (navi), 楽曲検索 (music), ニュース検索 (news) を, そして雑談として 3 つの機能 (タスク誘発, テンプレート応答, オウム返し) を実装した。以下では, 各タスクの辞書, データベースの構成と雑談について説明する。

#### 3.2 辞書, データベースの構成

表 1 に示す通り, 辞書, 言語制約は各タスクごとに, データベースは目的指向タスクのみ用意されている。各辞書は, 各データベースが検索できるように設定したので, 各タスクのデータベースのデータ収集方法について以下で説明する。まず, 目的地検索のデータベースであるが, 検索対象となる目的地は, 静岡県のインター, 駅, 飲食店, コンビニ, 遊園地などの 8108 施設について, 先行研究<sup>6)</sup> のものをそのまま利用した。次に, 楽曲検索とニュース検索のデータベースであるが, 楽曲検索は著者および共同研究者が任意に選んだ 19 アーティスト, 76 アルバム, 1010 曲について, 音楽検索サイト mora<sup>7)</sup> から収集し, ニュース検索のデータベースについても, NHK の Web ニュースサイト<sup>8)</sup> から, 2009 年 10 月 1 日から 2009 年 10 月 31 日までの間, 更新された全ニュース, 2862 記事をダウンロードした。最後に, 雑談の辞書であるが, Web から学習した約 6 万語彙の trigram を辞書として使用している。

#### 3.3 雑談

まず, 雑談機能にどのような機能が望まれるかを定義しておく。雑談対話には, 大きく分けて「雑談を継続する対話」と「別の目的指向タスクを誘発する対話」があると考えている。

前者は, 雑談を継続するために, 後者は, 雑談に含まれる目的指向タスクに関連した話題を抽出し, ユーザに目的指向タスクを起動するか尋ねるために必要である。雑談機能の

表 1 辞書, データベースの構成  
 Table 1 Composition of dictionary and data base.

タスク名	navi	music	news	ELIZA
言語制約	CFG	CFG	CFG	Trigram
語彙数	9574 語	1227 語	2618 語	60086 語
名詞数	9521 語	1207 語	2537 語	46147 語
DB の件数	8108 件	1010 件	2862 件	-

(発話テンプレート)	: (応答テンプレート)
(すいません, ごめん, ゴメン)	: (謝らないでください.)
(もし(.*))	: (あなたは(.*)だったらいいと考えているのですか?)
(夢)	: (その夢とは何ですか?, あなたはそのような夢をよく見るのですか?)
(たぶん)	: (はっきりしてください., なぜはっきりした口調で言えないんですか?)
(おはよう, こんにちは, こんばんは)	: ((挨拶のオウム返し). あなたが今思っていることを教えてください.)

図 3 テンプレートルールの例  
 Fig. 3 Example of template rule.

実現方法であるが, 前者の「雑談を継続する対話」には, 相手の発話に対して, リアルタイムで応答を返す研究<sup>9)</sup> 等でも, 使われている ELIZA<sup>10)</sup> の枠組みを利用することとした。つまり, 用意したテンプレートにマッチする場合には, そのルールに従った応答をし, マッチしなかった場合にはオウム返しで質問をする。これらをそれぞれ, テンプレート応答, オウム返しとした。テンプレート応答では, 図 3 の例のように, テンプレートルールとして, 発話テンプレートと応答テンプレートが用意されており, 発話テンプレートにマッチした発話があれば, 応答テンプレートのいずれかがランダムで応答文として生成される。テンプレート応答のテンプレートルールは, 17 個のルールと 54 個の応答文変換規則を用意した。

次に後者であるが, 雑談認識時に得られた音声認識結果から, 雑談タスクの辞書の単語の組み合わせで表現できる, 各目的指向タスクの辞書に含まれる単語 (タスク誘発単語) を抽出し, 各目的指向タスクが起動できるかどうかを信頼度を使って判定し, タスクが起動できると判定された場合には, ユーザに, その目的指向タスクを, タスク誘発単語を使って立ち上げていいかを尋ね, その質問にユーザが肯定的な発言をした場合に, 指定された目的指向タスクを起動する。これをタスク誘発とした。具体的なアルゴリズムは (1) - (3) のようになり, タスク誘発の発話例は図 4 のようになった。また, 複数のタスク誘発単語があった場合には, あるタスク誘発単語でタスクが誘発されるか, 別のタスクの発話がされるまで, タスク誘発単語を使い, ユーザに問いかけるようにした。

U1:今日は、文化施設までドライブに行くんだ。  
S1:文化施設までのルート検索をしますか？  
U2:お願い。  
S2:文化施設で検索した結果 件見つかりました。

図 4 タスク誘発の例  
Fig. 4 Example of task lead.

表 2 タスク判定率について  
Table 2 Task system decision rate.

タスク名	navi	music	news	ELIZA	全体
再現率 (%)	78.9	66.3	91.3	84.0	80.2
精度 (%)	65.2	95.6	87.5	80.2	80.2
F 値 (%)	71.4	78.3	89.4	82.0	80.2

- (1) 認識されたタスク誘発単語は複数の単語から構成されている場合が多いので、Julius から出力された信頼度の平均を取りタスク誘発単語の信頼度を算出
- (2) (1) で求めた信頼度がある程度信頼できる (0.5 以上) 場合に、タスク誘発単語に関係のあるタスクを起動するかの問いかけ
- (3) ユーザの発話に肯定表現が含まれると理解された場合には、タスク誘発単語の情報をそのタスクに渡し、タスクを誘発

#### 4. 現状のシステムのタスク判定率、認識率について

前述の 3 つの目的指向タスクと雑談において、現在考えうる全対話パターンから 80 対話、410 発話をランダムに発話を自動生成し、タスク判定率、タスクが正しく判定された場合と仮定した場合の認識率、システムのタスク判定を用いた場合の認識率を求め、雑談の機能の 1 つであるタスク誘発の評価をした。

##### 4.1 タスク判定率

現状のタスク判定率は表 2 のようになった。今回のタスク判定に用いた尤度比の閾値、フレーム数の閾値は全体のタスク判定の結果が良くなるように設定している。ただし、そのパラメータを算出時に用いたデータは辞書、データベースの更新の関係上、news の発話を含まない 318 発話だった。なので、全体のタスク判定率が高くなるように設定してあるが、結果を見ると news のタスク判定率が最大となっている。今後は、今回用いたデータを使い、タスク判定率に用いる尤度比の閾値、フレーム数の閾値の再設定をする必要がある。

表 3 タスク判定が正しいと仮定した場合の認識率  
Table 3 Recognition rate of task all correct.

タスク名	navi	music	news	ELIZA	全体
発話数 (個)	95	98	92	125	410
正解率 (%)	85.8	81.0	88.6	67.4	73.4
正解精度 (%)	80.9	75.0	87.6	57.1	65.2

表 4 システムのタスク判定を用いた場合の認識率  
Table 4 Recognition rate of task system decision.

タスク名	navi	music	news	ELIZA	全体
発話数 (個)	115	68	96	131	410
正解率 (%)	53.9	80.3	88.5	64.2	66.0
正解精度 (%)	48.2	72.8	87.5	45.1	51.9

#### 4.2 認識率

タスク判定が正しいと仮定した場合とシステムのタスク判定を用いた場合の音声認識率を、それぞれ表 3、表 4 に示す。タスク判定を正しいと仮定した場合は、全体の認識率は 73%程度だったが、目的指向タスクの認識率だけ見るとすべて 80%以上であった。これは、言語制約が CFG のものに限り、挿入ペナルティの値を調整しており、その成果だと言える。システムのタスク判定を用いた場合も、news、music の認識率は 80%を超えている。これらは、タスク判定の精度が高いものであり、逆にタスク判定の精度が低かった navi に関しては正解率は 53%程度に落ち込んでいる。また、全体を通してみると、タスク判定を正しいとした場合は、システムのタスク判定を用いた場合よりも正解率が 8%向上しているため、タスク判定率が上がれば認識率はより改善できるとわかった。今後は、タスク判定のパラメータ調整後の認識率の再評価をする必要がある。また、今回のタスク判定に用いた尤度比の閾値、フレーム数の閾値は全体のタスク判定の結果が良くなるように設定しているため、navi のタスク判定の精度が高くなるように設定した場合に、認識率がどう変わるのかも評価していきたい。

#### 4.3 タスク誘発の評価

410 発話中、タスク誘発をする発話が 11 個あり、それに対してシステムが実際に誘発した数は 15 個であった。15 個の内訳を見てみると、意図どおりに正しく誘発できた数が 6 個、目的指向タスクの発話を雑談と判定してタスク誘発した数が 2 個、雑談を誤認識してタスク誘発をした数が 7 個であった。正解のタスク誘発数 11 個中、6 個は正しく誘発できてい

るので、全体の半数は正しく誘発できた。目的指向タスクの発話を雑談と判定してタスク誘発をした数が2個であったが、これは目的指向タスクで認識できなかった情報を雑談で得られることができたと言える。最後に、雑談を誤認識してタスク誘発をした数が6個であったが、誤認識が多いのは、タスク誘発をするための閾値を低めに設定したことが理由だと考える。今後は、タスク誘発をするための閾値の調査をして、誤認識の数を減らしていこうと考えている。

## 5. ユーザビリティの向上

既存のカーナビゲーションなどの音声対話システムは、ユーザ発話という入力があって初めてタスクを開始する。しかし、我々は文脈情報やユーザ情報、その時の状況などからシステム側から話題を提案したり、それらを考慮した応答をすることで、ユーザが必要としている情報をより効率的に提供し、問題解決に役立てると考える。よって、本研究ではよりユーザに効率的にシステムを使ってもらうためのシステム提案手法を確立した。

まず、最初のシステム提案として、現状のシステムでは、検索結果が4件以上で、固有名詞（施設名称、曲名など）以外では絞り込みがこれ以上できない場合には、システム側からユーザにDB登録順に全検索結果を提示していたが、検索結果をタスクごとに決められた条件（目的地：現在地からの距離、ニュース：最新順、楽曲：DBの登録順）でソートし、ユーザに提示する固有条件提示を実装した。

次のシステム提案では、ユーザのお気に入りのお店などの個人情報を利用する。ある発話が行われ、検索を行い、追加条件を指定する応答を返す場合に、現在の検索に使った単語に加え、お気に入りの情報を使って検索が可能ならば、お気に入り検索が可能とする。ユーザに、本来の追加条件の応答に加え、お気に入り検索を実行して良いか問いかけ、ユーザが肯定的な反応を示した場合には、お気に入り検索を行うようにシステムを変更した。

最後に、提案段階であるが、ユーザのお気に入りなどの個人情報を使い、ユーザの気に入っている情報が上位に来るように、データベース検索結果をソートし、応答文として提示するような、固有条件提示とお気に入り検索の合わせたシステム提案を行えば、よりユーザビリティの向上に繋がると考えている。

## 6. まとめ

本稿では、マルチタスク音声対話システムを構築し、目的地検索、ニュース検索、楽曲検索の3つの目的指向タスクと雑談について実装を行った。また、システムの性能評価とし

てタスク判定率タスク判定率は80.24%、認識率はタスク全体で65.98%（タスク判定が正しくできると仮定した場合は73.40%）を得ることができた。さらに、システムのユーザビリティの向上のためのシステム提案として、固有条件提示とお気に入り検索の実装を行い、他のシステム提案の考案を行った。

今後の課題として、システム全体としては、タスク判定率に用いる閾値のパラメータの調整、認識率の再評価をし、システム提案としては、考案した方法について実装を行っていく予定である。

## 参考文献

- 1) 目黒豊美, 鈴木基之, 伊藤彰則, 牧野正三: 音声対話システムにおけるタスク外発話処理の高度化に関する研究, 情報処理学会音声言語情報処理研究会 SIG-SLP-69, pp. 283-287 (2007).
- 2) 池田智志, 駒谷和範, 尾形哲也, 奥野博: マルチドメイン音声対話システムにおけるトピック推定と対話履歴の統合によるドメイン選択手法, 情報処理学会論文誌, Vol.50, No.2, pp.488-500 (2009).
- 3) 栗野健太郎, 伊藤仁, 伊藤彰則, 牧野正三: マルチタスク型音声対話システムの対話制御のための発話識別, 情報処理学会音声言語情報処理研究会 SIG-SLP-79, pp.1-6 (2009).
- 4) 宮村祐一, 徳永健伸: 情報探索雑談におけるトピック遷移検出, 情報処理学会死産言語処理研究会 SIG-NL-90, pp.71-76 (2008).
- 5) 京都大学河原研究室, 名古屋工業大学 Julius 開発チーム: 汎用大語彙連続音声認識エンジン Julius, <http://julius.sourceforge.jp/> (参照 2009-12).
- 6) 細貝 隆行, 小暮 悟, 甲斐 充彦, 小西 達裕, 伊東 幸宏: 音声認識における認識単語の競合情報を利用した言語理解と対話制御の構築, 人工知能学会 第55回言語・音声理解と対話処理研究会 技術報告, pp.13-18 (2009).
- 7) 株式会社レーベルゲート: 音楽検索サイト mora, <http://mora.jp/> (参照 2009-02).
- 8) NHK: NHK ニュース, <http://www3.nhk.or.jp/news/> (参照 2009-10-31).
- 9) 竹内真司, 北岡教英, 中川聖一: 韻律・表層的言語情報を発話タイミング制御に用いた雑談対話システム, 情報処理学会音声言語情報処理研究会 SIG-SLP-50, pp.87-92 (2004).
- 10) Weizenbaum, J.: Eliza - a computer program for the study of natural language communication between man and machine, *Comm. ACM*, Vol.9, No.1, pp.36-45 (1966).