情報処理学会論文誌 Vol. 49 No. 8 2762–2772 (Aug. 2008)

音声対話システムにおけるラピッドプロトタイピングを指向したWFSTに基づく言語理解

福林 雄一朗[†1,*1] 駒谷 和範[†1] 中野 幹生[†2]
船越 孝太郎[†2] 辻野 広司[†2]
尾形 哲也[†1] 奥乃 博[†1]

音声対話システムの開発の初期段階において，言語理解部は，(i) 構築が容易，(ii) 様々な表現に対して頑健という 2 条件を満たす必要がある．本論文では，大量のコーパス収集や人手での詳細な言語理解ルールの記述を行うことなしに，簡単に言語理解部を構築する（ラピッドプロトタイピング）手法について述べる．本手法では，音声認識誤りを含む入力に対して，Weighted Finite State Transducer（WFST）により言語理解結果を出力する．この際の重みは複数種類を定義したうえで，学習データに基づき最適な重みづけを選択する．この重みづけは従来の WFST を利用した手法に比べて簡単であるため，少ない学習データで動作する．本手法を 2 つのドメインで評価した結果，本手法では 100 発話程度の学習で，ベースライン手法より高いコンセプト正解精度が得られた．開発の初期段階にある新たなドメインであっても，この程度の量の発話を集めることは容易であり，本手法は言語理解部のラピッドタイピングに適している．

WFST-based Language Understanding for Rapid Prototyping of Spoken Dialogue Systems

Yuichiro Fukubayashi,[†1,*1] Kazunori Komatani,[†1]
Mikio Nakano,[†2] Kotaro Funakoshi,[†2]
Hiroshi Tsujino,[†2] Tetsuya Ogata[†1]
and Hiroshi G. Okuno[†1]

Language understanding (LU) modules for spoken dialogue systems in an early phase of their development need to be (i) easy to construct, and (ii) robust against various expressions. In this paper, we describe a method for constructing LU modules easily without a large amount of corpus or complicated handcrafted rules. An LU result is selected with Weighted Finite State Transducer from an automatic speech recognition output that may contain speech recognition errors. We designed several weighting schemes. A weighting scheme is determined by using training data. Since these weighting schemes are simpler than conventional methods, our method does not need a large amount of data for determining an optimal scheme. We evaluated our method in two different domains. The results revealed that our method outperformed baseline methods with less than one hundred utterances as training data, which can be reasonably prepared for new domains. This shows that our method is appropriate for a rapid prototyping of LU modules.

1. はじめに

近年，携帯電話の普及もあり電話を利用した音声対話でのチケット予約などのサービスが多くなっている．また，いろいろなタイプのロボットが開発され，人間とのインタラクションのために音声認識機能を持つものも少なくない．音声対話システムが産業界で広く使われるためには，以下の 2 点が必要である．

(1) 音声認識誤りに対する頑健性
(2) 構築にかかるコストの低さ

一般に，音声対話システムにおいては，ユーザ発話の多様性や音声認識誤りなど，音声メディア特有の問題に起因する性能低下が避けられない．そのため，近年の音声対話システムの研究では，システムを実際に構築し収集した対話データを利用した統計的手法[1)–4)] がさかんに用いられる．統計的手法を用いることで，その対話データを収集したドメインでの高性能なアプリケーションが開発できる．しかし，大量のコーパスの収集にはコストがかかり，試作システムの構築は容易ではないので，新たなドメインでの商用システムの開発には現実的ではない．したがって，音声対話システムが産業界で広く使われるためには，システムを簡単に構築する技術，すなわち，ラピッドプロトタイピング技術の開発が重要である．

†1 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
†2 株式会社ホンダ・リサーチ・インスティチュート・ジャパン
Honda Research Institute Japan Co., Ltd.
*1 現在，日本電気株式会社
Presently with NEC Corporation

本論文では，開発初期段階の学習データの少ない状況においても，音声認識誤りに対して頑健に動作する言語理解部の構築について述べる．

これまで音声対話システムにおけるラピッドプロトタイピングとして，いくつかの方法が提案されてきた．対話管理部のラピッドプロトタイピングとしては，VoiceXML[5),6)] や XISL[7)] などの対話記述言語が開発され，プロトタイプシステムの作成を容易にしている．また，音声認識部では，音声認識誤りを抑制するためにドメインに合わせた言語モデルを少ない労力で構築する手法[8),9)] が提案されている．意味理解・検索においては，音声対話システムをドメイン・タスク依存な部分と非依存な部分に切り分け，タスク依存な部分のみを記述するだけでシステムを構築できる枠組みが提案されている[10)]．言語理解部においては，音声認識器に文法ベースのものを利用し，認識文法に言語理解結果を対応させておくといった単純な方法があげられる．これらの文法の記述は大規模なコーパスの収集に比べて少ない労力で可能で，プロトタイプシステムを作りやすいが，一方で音声認識誤りによる性能低下が問題になる．VoiceXML や文献 10) においても，音声認識誤りへの対処は大きな課題となっている．

この問題に対処するために，ユーザの発話をキーフレーズスポッティングやヒューリスティックなルールで分類する手法[11)] が提案された．これらの手法では，ルールを 1 度用意すれば大きな修正を加えることなく音声認識結果からコンセプトを抽出できる．また，コーパスを利用してコンセプトの出現確率を学習する手法[2)] や Weighted Finite State Transducer（WFST）を利用した手法[3),4)] が提案されてきた．しかし，複雑なルールの準備や，数千，数万発話の大量のコーパスの収集，正解の付与が必要で，時間の面でも費用の面でも大きなコストがかかり，新たなドメインの言語理解部を構築するのには適さない．また，ポータビリティや耐雑音性を指向した手法として，音声認識器と言語理解を統合した手法[12)–14)] も提案されているが，言語理解には専用のデコーダを準備する必要があり，音声対話システムに詳しくない開発者には構築が容易ではない．

本手法では WFST を導入し重みづけを適切に選択することで，音声認識誤りに頑健な言語理解部を少ない労力で実現する．**図 1** に本手法と従来の手法との関係を示す．単純ルールや文法に基づく手法では，大量の学習データを必要としないので，統計的手法や複雑なルールを必要とする手法と比べて少ない労力で言語理解部を構築できる．一方で，統計的手法や複雑なルールを必要とする手法は，大量のコーパスやルールの準備に大きな労力を必要とするが，音声認識誤りに頑健な言語理解部を実現できる．本手法はこれらの中間に位置する．つまり，本手法は単純なルールや文法に基づく手法より音声認識誤りに対して頑健であり，統計的手法や複雑なルールに基づく手法より少ない労力で言語理解部を実現するものである．したがって，本手法は大量のコーパスを持たない新たなドメインの開発の初期段階のラピッドプロトタイピングに適する．さらに，本手法に基づくシステムを初期システムとして十分な量のデータを収集し，それに基づき統計的手法を適用することで音声認識誤りに対する頑健性を高める，といったシステム開発サイクルも可能となる．このような実際の開発を意識した音声対話システムの研究は，産業界の視点に立った議論に基づいて初めて得られるものである．本研究の着眼点は，大学単独の視点からは得られにくく，産学連携の共同研究により得られたものであるといえる．

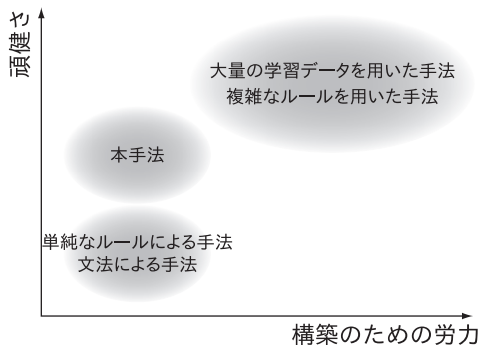


図 1　音声言語理解における本研究と従来の手法との関係
Fig. 1　Relationship between our method and conventional methods in spoken language understanding.

2. WFST に基づく音声言語理解と文法記述

我々が開発した WFST に基づく言語理解部は以下の 2 つの特長を持つ．

(1)　必要となるドメイン文法記述は，VoiceXML などと同程度の労力で記述できる．これから WFST を構築することで，より頑健な言語理解部が実現できる．

(2)　WFST に対する重みづけは従来と比較して単純に設計しているため，必要とする学習データの量が少なくて済む．

したがって，本手法では音声認識誤りに頑健な言語理解部を少ない労力で構築できるので，新たなドメインの言語理解部の開発が容易である．

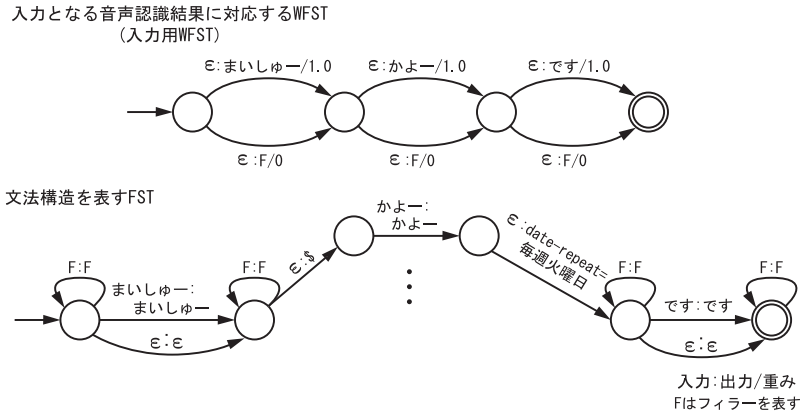


図 2　入力用 WFST と文法構造を表す FST の例
Fig. 2　Example of WFST for input and FST for parsing grammar.

表 1　言語理解に不要な単語を含む発話の例
Table 1　Example utterances including FILLERs.

音声認識結果
えーと ひにち わ にがつ にじゅーに にち です
ひにち わ **らいげつ の** にがつ にじゅーに にち です
ひにち わ **に，** にがつ にじゅーに にち **かよーび** です
ひにち わ にがつ **の** にじゅーに にち です **が**
言語理解結果 = [month=2, day=22]

表 2　「まいしゅー かよー です」に対する言語理解結果の例
Table 2　Language understanding results for input “Every Tuesday please.”

入力			言語理解結果	w
まいしゅー	かよー	です	date-repeat=毎週火曜日	3.0
FILLER	かよー	です	date-repeat=毎週火曜日	2.0
FILLER	かよー	FILLER	date-repeat=毎週火曜日	1.0
FILLER	FILLER	FILLER	n/a	0

2.1　WFST に基づく音声言語理解

本研究では文法構造を表す FST と入力となる音声認識結果に対応する WFST を合成することで，音声認識結果に対して，文法構造に基づく累積重みを計算し，言語理解結果を得る．つまり，入力となる音声認識結果に対応する WFST の出力が文法構造を表す FST に入力できるようにそれぞれを設計する．以下では，入力となる音声認識結果に対応する WFST を入力用 WFST と呼ぶ．ここで，言語理解結果はコンセプトの集合とする．コンセプトは，スロットと対応する値の組である．たとえば，[month=2, day=22] という言語理解結果は，スロット month に値 2 が設定され，スロット day に値 22 が設定されたコンセプトにより構成される．

図 2 に入力用 WFST と文法構造を表す FST の例を示す．この図の各遷移のラベルは，「入力単語：出力単語/重み」を表している．ただし，入力単語における ε は入力なしでの遷移，つまり ε 遷移，出力単語における ε は出力がないことを意味する．たとえば「です：ε/1.0」では，「です」が入力されると出力なしで遷移し，累積重みに 1.0 が加算される．文法構造を表す FST では遷移に重みを与えないので重みは省略されている．

文法構造を表す FST では，受理状態まで遷移できれば，入力された単語列が文法に合う単語列であったことが分かる．また，本研究では，文法構造を表す FST に FILLER 遷移を用意することで，**表 1** のように言語理解に不要な単語を無視する解釈を許容できる．この例では，「FILLER かよー です」が入力されると「FILLER $ かよー date-repeat=毎週火曜日 です」[⋆1]が出力される．

入力用 WFST は音声認識結果から，図 2 のように各単語の遷移と FILLER 遷移が並列となるように生成する．これにより，それぞれの認識単語を言語理解に必要な単語として扱うかフィラーとして扱うかを 1 つの WFST で表す．また，入力用 WFST に重みを付与することで，遷移のたびに累積重みに重みを加算する．たとえば，図 2 の入力用 WFST は「まいしゅー かよー です」という音声認識結果に対して生成されるもので，「まいしゅー かよー です」や「まいしゅー FILLER FILLER」や「FILLER かよー FILLER」など 2^3 通りの入力列を表す．それぞれの場合で累積重みは 3.0，1.0，1.0 である．

入力用 WFST と文法構造を表す FST の合成により，入力用 WFST が表すあらゆる入力が，文法構造を表す FST に入力されたときの，すべての可能な出力列とそのときの累積重みが得られる．これにより，あらゆる語がフィラーとして扱われる可能性を考慮した言語理解を行う．さらに，累積重みを計算することで，複数の出力列から累積重み w が最大の言語理解結果を採用し，最適な言語理解を行う．**表 2** の例では，累積重み w が 3.0 と最も高い [date-repeat=毎週火曜日] が言語理解結果として採用される．本研究で定義する重みづ

⋆1 $はコンセプトに対応する単語の範囲を特定するための記号である．

```
...
<keyphrase-class name="month">
...
  <keyphrase>
    <orth>にがつ</orth>
    <sem>2</sem>
  </keyphrase>
...
</keyphrase-class>
...
<action type="specify-attribute">
  <sentence> {ひにち わ} [*month] *day [です]
  </sentence>
</action>
```

図 3　文法記述の例
Fig. 3　Example of grammar description.

けの具体的な実装法は，3.5 節で詳述する．

従来の WFST による言語理解では，階層的なコンセプトの n-gram [3] や単語とスロットの組をコンセプトとして扱いその n-gram [4] を利用していた．これらの手法では，n-gram を数千発話のタグ付けされたコーパスから学習していた．しかしながら，コーパスの収集や正解の付与には大きな労力が必要であるため，新たなドメインの言語理解部の構築に用いるのは困難である．

2.2　ドメイン文法記述

我々の開発したシステムでは，ドメイン文法とコンセプトの定義を人手で記述すれば，文法構造を表す FST を自動的に生成できる．**図 3** はドメイン文法記述の例である．この文法記述に必要な労力は，従来の VoiceXML などの文法の記述とほぼ同等である．本手法では，この文法記述と少量の学習データのみで音声認識誤りに頑健な言語理解を実現する．以下では，人手で用意すべき文法記述の詳細について説明する．

図 3 に文法とコンセプトの定義の例を示す．スロットの定義は，`keyphrase-class` タグ内で行い，コンセプトを表すキーフレーズと値の関係は `keyphrase` タグ内で定義する．図 3 の例では，スロット `month` に対して，「にがつ」という表記のキーフレーズに対して値 `2` が定義されている．文法は `action` タグ内の `sentence` タグにおいて，終端記号と非終端記号の列で記述する．“`*`” で始まる部分は非終端記号で，`keyphrase-class` タグ内で定義したコンセプトに対応するキーフレーズのうちのいずれかが入力できる．また，`[]` により一部の単語の省略も指定できる．2.1 節で説明した FILLER 遷移は，各終端・非終端記号間に自動的に挿入される．ただし，`[]` や`{}`で囲まれた区間には挿入されない．`{}`は FILLER 遷移の自動的な挿入を避けるために使用する．図 3 の例では，「ひにち わ にがつ にじゅーにち です が」や「ひにち わ にじゅーに にち」が受理可能であるが「ひにち わ にがつ」や「ひにち えと は にがつ にじゅーに にち」は受理されない．

本研究では，定義した文法で指定される部分にフィラーを挿入した例文を大量に生成することで，当該ドメインの統計的言語モデルを自動構築するツールも開発した．この統計的言語モデルを利用することで，文法記述に含まれない新たな表現を認識することはできないものの，フィラーや未知語に対しては比較的頑健な音声認識を期待できる．本論文では，フィラーとして，「あー」や「えっと」などフィラーとして現れやすい 6 単語を使用し，クラス内生起確率はこの 6 単語で等確率になるように指定した．

3.　音声認識結果とコンセプトに対する重みづけ

我々は入力用 WFST に対する重みづけとして，**音声認識器の N-best 出力**，**受理単語**，**コンセプト**の 3 種類の重みづけを定義する．さらに，それぞれにおいて数種類の重みづけ手法を用意した．本手法では，学習データを利用して最適な重みづけ手法を選択した後，文法構造を表す FST との合成により言語理解を行う．以下では順に，用意した重みづけ手法について説明した後，重みの計算例を示す．

3.1　音声認識器の N-best 出力に対する重みづけ

入力用 WFST の生成には，音声認識結果の N-best 出力を利用する．この N-best 文それぞれに対して重みを割り当てる．重みは N-best 文内の順位が高い，より信頼できる文に対してより大きくなるよう与える．具体的には以下のように設計した．

$$w_s^i = \frac{e^{\beta \cdot score_i}}{\sum_j^N e^{\beta \cdot score_j}}$$

ここで w_s^i は音声認識結果の N-best の i 番目の文に対する重み，β はスムージング係数，$score_i$ は i 番目の文の対数尤度である．この重みづけは音声認識結果の信頼度を反映している．本論文では，予備実験により β を 0.025 とした．

3.2　受理単語に対する重みづけ

文法構造を表す FST により受理された単語に対して重みづけを行う．この重みづけでは，音声認識結果の単語レベルで信頼できる単語に対してより大きな重みを与える．基本的に，

フィラー以外の単語が多くなるように，音声認識結果が信頼できる入力を優先するように設定する．この重みづけ w_w は以下のように設計した．

(1)　**word(const.)**: $w_w = 1.0$

(2)　**word(#phone)**: $w_w = l(W)$

(3)　**word(CM)**: $w_w = CM(W) - \theta_w$

word(const.) は受理されたすべての単語に対して一定の重みを加える．この重みづけは，受理単語の数が多い出力を優先する．**word(#phone)** は，各受理単語の長さを考慮に入れた重みづけである．各単語の長さは，それぞれの音素数で計算し，システムの語彙中で最も長い単語の長さで正規化する．単語 W に対してこの正規化された値を $l(W)$ $(0 < l(W) \leq 1)$ とする．**word(#phone)** は，入力列の長さを，**word(const.)** よりも詳細に表現したものといえる．さらに，受理単語の信頼度を考慮に入れた **word(CM)** も用意する．この重みづけは，音声認識結果中の単語 W に対する信頼度[15] $CM(W)$ を利用している．θ_w は W を受理するかどうかの閾値である．信頼できない W に対しては，$CM(W)$ が θ_w より小さくなり w_w は負の値をとる．この重みづけは，W に対する音声認識結果がどれだけ信頼できるかを反映しており，長くかつ信頼できる出力列を優先するための設計である．

3.3　コンセプトに対する重みづけ

音声認識結果に対する重みに加えて，コンセプトレベルにおける重みも用意した．コンセプトは，複数の単語からなり，音声認識結果を文法構造を表す FST に入力することで得られる．コンセプトに対する重みは，対応するキーフレーズに含まれる単語の信頼度などを用いて計算する．具体的には以下のように重みづけ w_c を 3 種類用意した．

(1)　**cpt(const.)**: $w_c = 1.0$

(2)　**cpt(CM)**:

$$w_c = \frac{\sum_{\boldsymbol{W}} (CM(W) - \theta_c)}{\#\boldsymbol{W}}$$

(3)　**cpt(#p·CM)**:

$$w_c = \frac{\sum_{\boldsymbol{W}} (CM(W) \cdot l(W) - \theta_c)}{\#\boldsymbol{W}}$$

$\boldsymbol{W}$ は当該コンセプトに含まれる単語の集合で，W は $\boldsymbol{W}$ に含まれる単語である．また，$\#\boldsymbol{W}$ は $\boldsymbol{W}$ に含まれる単語の数である．

cpt(const.) は，1 発話から得られるコンセプトを多くするための重みづけである．また，**cpt(CM)** はコンセプトを構成する単語の認識結果が信頼できないものを棄却するための設定である．**cpt(#p·CM)** は，コンセプトに含まれる単語の信頼度のほかに，それらの長さも考慮に入れている．θ_c はコンセプトを受理するかどうかの閾値である．

表 3　パラメータとして **word(CM)**，**cpt(#p·CM)** が選択されているときの重みづけの例
Table 3　Example of weighting for a parameter setting: **word(CM)** and **cpt(#p·CM)**.

音声認識結果	いーえ	ひにち	わ	にがつ	にじゅーに	にち
言語理解結果	FILLER	ひにち	わ	にがつ	にじゅーに	にち
$CM(W)$	0.3	0.7	0.6	0.9	1.0	0.9
$l(W)$	0.3	0.9	0.3	0.9	0.9	0.6
Concept	—	—	—	month=2	day=22	
word	—	$0.7 - \theta_w$	$0.6 - \theta_w$	$0.9 - \theta_w$	$1.0 - \theta_w$	$0.9 - \theta_w$
cpt	—	—	—	$(0.9 \cdot 0.9 - \theta_c)/1$	$(1.0 \cdot 0.9 - \theta_c + 0.9 \cdot 0.6 - \theta_c)/2$	

3.4　累積重みの計算と学習

音声認識結果の N-best 候補の i 番目（$1 \leq i \leq N$）の文それぞれに対して，以上で示した 3 種類の重み w_s^i，w_w，w_c の重みつき和 w^i を計算する．この際，この i 番目の文に対する WFST によるあらゆる解析結果の中で，この重みつき和が最大となるものを i 番目の文の言語理解結果とし，そのときの重みつき和を w^i とする．その後，N-best 候補の中で w^i が最も大きい出力列を選ぶことで，言語理解結果を得る．

$$\text{言語理解結果} = LU(\operatorname*{argmax}_i w^i) \tag{1}$$

$$w^i = \max_{p_i} w(p_i) \tag{2}$$

$$w(p_i) = w_s^i + \alpha_w \sum_{W \in \boldsymbol{A}_{p_i}} w_w + \alpha_c \sum_{W \in \boldsymbol{C}_{p_i}} w_c \tag{3}$$

ここで，$LU(i)$ は N-best の i 番目の文から得られる言語理解結果，p_i は，i 番目の文に対する WFST での解析結果，$\boldsymbol{A}_{p_i}$ は p_i で受理されている単語の集合，$\boldsymbol{C}_{p_i}$ は p_i で受理されているコンセプトに対応する単語の集合である．

累積重み w^i の計算方法を**表 3** を用いて説明する．この例では，パラメータとして **word(CM)**，**cpt (#p·CM)** が選択されているとする．入力が「いーえ ひにち わ にがつ にじゅーに にち」である場合，この表では受理単語に対する重みの総和は $\alpha_w(4.1 - 5\theta_w)$ である．また，コンセプト “month=2” に対する重み $\alpha_c(0.81 - \theta_c)$ とコンセプト “day=22” に対する重み $\alpha_c(0.72 - \theta_c)$ により，コンセプトに対する重みの総和は $\alpha_c(1.53 - 2\theta_c)$ である．したがって，この入力列に対する累積重み w^i は $w_s^i + \alpha_w(4.1 - 5\theta_w) + \alpha_c(1.53 - 2\theta_c)$ となる．

書き起こし	ろくがつ	みっか			から	です	
音声認識結果	ろくがつ	みっか	あー	ふぃっと	から	です	
$CM(W)$	0.978	0.757	0.152	0.525	0.541	0.521	言語理解結果
$l(W)$	0.73	0.46	0.09	0.46	0.36	0.36	
正解	ろくがつ	みっか	F	F	から	です	month:6, day:3
最適時	ろくがつ	みっか	F	F	から	です	month:6, day:3
ベースライン	ろくがつ	みっか	F	ふぃっと	F	F	month:6, day:3, car:FIT

（「ふぃっと」は車の種類，'F' はフィラーを表す）

図 4　重みづけが有効に働く例（レンタカー予約ドメイン）
Fig. 4　Example of language understanding with WFST (rent-a-car domain).

学習においては，様々なパラメータの組合せを試し，コンセプト誤り率（concept error rate，CER）が最小となる組合せを探す．$CER = (S + D + I)/N_{\mathrm{ref}}$ で計算する．ここで，N_{ref} は正解データに含まれるコンセプトの数で S，D，I はそれぞれ，置換誤り，削除誤り，挿入誤りの数である．誤りの数の計算では，スロットと値の両方が正解データと一致した場合，正解とする．また，スロットか値の一方が誤りであった場合は置換誤りとして扱う．したがって，スロットも値も一致しないコンセプトが含まれる場合は，正解データでは削除誤り，仮説データでは挿入誤りとして扱う．つまり，それぞれの誤りの数の計算ではまず，一致したコンセプトの数 C と，置換誤りの数 S を計算することで，削除誤りは $D = N_{\mathrm{ref}} - C - S$，挿入誤りは $I = N_{\mathrm{hyp}} - C - S$ で計算できる．ここで N_{hyp} は仮説データに含まれるコンセプト数を表す．パラメータは，音声認識結果やコンセプトの重みづけとしてどれを用いるか，N（1 または 10），$\alpha_{w,c}$（1 または 0），$\theta_{w,c}$（0 から 0.9 まで 0.1 ごと）とする．係数 $\alpha_{w,c} = 0$ が設定されたときは，対応する重みは利用しない．**図 4** に本手法の動作例を示す．この例では，キーフレーズスポッティングでは言語理解結果として，[month:6]，[day:3]，[car:FIT] のすべてが得られるが，文法構造を表す FST では「ろくがつ みっか から です」か「ふぃっと です」の 2 つの文法でしか受理できないので，[month:6]，[day:3] と [car:FIT] が同時に得られることはない．また，パラメータとして $\mathbf{word(CM)}-0.6$，$\mathbf{cpt(\#p{\cdot}CM)}-0.4$ が選択されているとすると，「ろくがつ みっか から です」の場合 $w = w_s^i + 0.660$ で，「ふぃっとです」の場合 $w = w_s^i - 0.216$ であるので，累積重みが大きい [month:6]，[day:3] が言語理解結果としている．つまり本手法では，文法構造と音声認識結果の信頼度を考慮して，音声認識誤りにより湧き出したコンセプト [car:フィット] を棄却している．

3.5　重みづけを実現するための実装の詳細

以上で説明した重みづけを実現するための，入力用 WFST と文法構造を表す FST の実装について説明する．

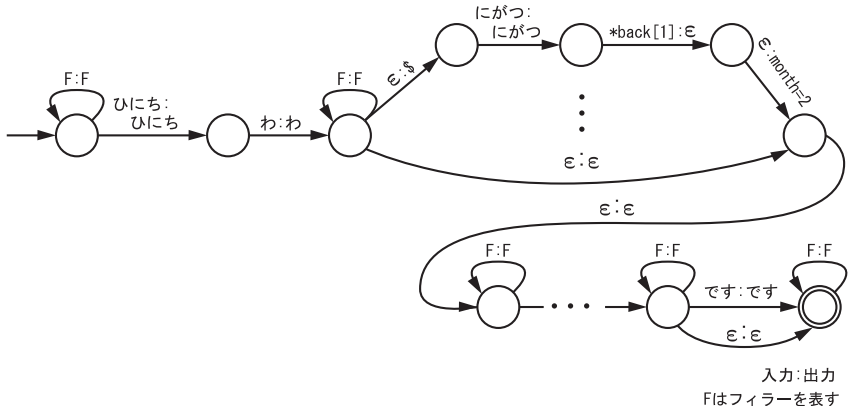


図 5　文法構造を表す FST の例
Fig. 5　Example of FST of a domain grammar.

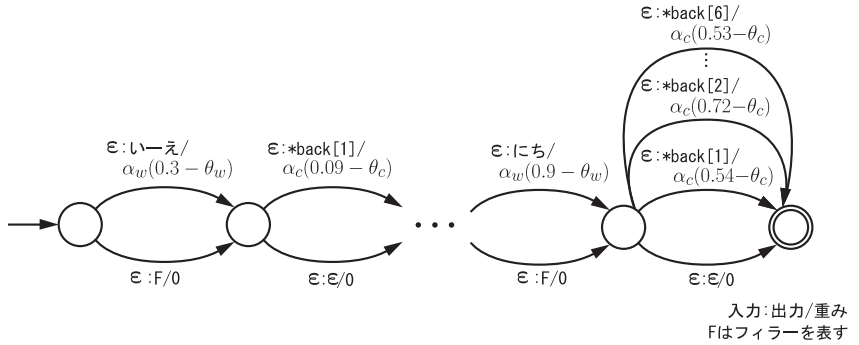


図 6　入力用 WFST の例
Fig. 6　Example of WFST for an input.

図 5 は文法構造を表す FST の例，**図 6** は入力用 WFST の例である．文法構造を表す FST はドメイン文法記述から自動的に生成する．構造は図 2 の文法構造を表す FST とほぼ同様であるが，コンセプトに対する重みを計算するための遷移「*back[j]：ε」をコンセプトを出力する遷移の直前に挿入する．この遷移での入力「*back[j]」は，コンセプトに対

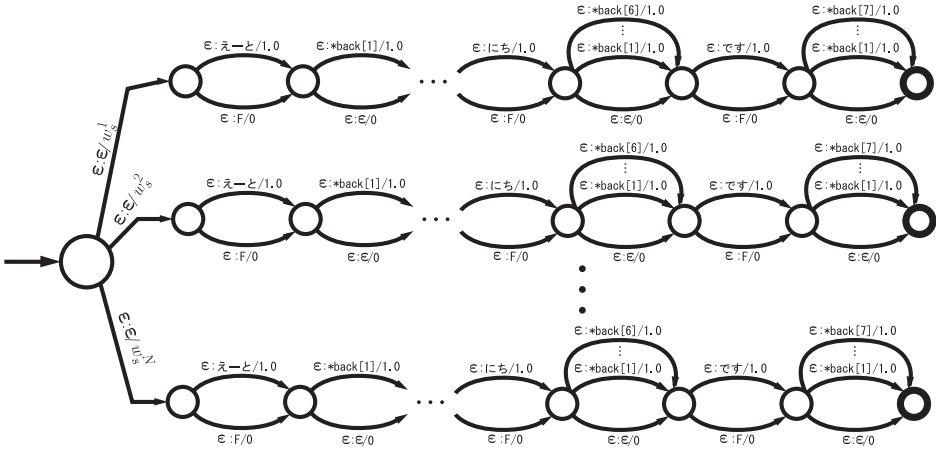


図 7　N-best 出力に対する重みづけを含めた入力用 WFST の例
Fig. 7　Example of WFST for an input with a weight for ASR N-bset output.

応するキーフレーズが直前 j 単語により成り立つことを示し，この FST では認識単語以外に「*back[j]」を入力しなければ遷移できない．図 5 の例では，「にがつ：にがつ」の遷移の後に「*back[1]：ε」が挿入されている．

入力用 WFST では，音声認識結果中の各単語の遷移の重み $\alpha_w w_w$ を，受理単語に対する重みづけ手法に基づいて計算し割り当てる．さらに，コンセプトに対する重みを計算するための遷移「ε：*back[j] / $\alpha_c w_c$」を各認識単語の間に挿入する．この遷移での重み $\alpha_c w_c$ は，直前 j 単語の認識結果を用いて，コンセプトに対する重みづけ手法に基づいて計算し割り当てる．入力用 WFST を生成した時点では，それぞれの単語で直前何単語までがコンセプトに対応するキーフレーズか（もしくはキーフレーズではないか）を知ることはできないので，認識結果の j 番目の単語の後では「*back[1]」から「*back[j]」までの遷移すべて（と ε 遷移）を挿入する．

図 6 は，表 3 の例に対する入力用 WFST である．選択されたパラメータは表 3 と同じである．この例では，「いーえ」の重みとして $\alpha_w w_w = \alpha_w(0.3 - \theta_w)$ が割り当てられている．また，キーフレーズ「にじゅーに にち」に対応する遷移として，「ε：*back[2] / $\alpha_c(0.72 - \theta_c)$」が挿入されている．音声認識結果の N-best 出力に対する重みづけは，**図 7** のように N 個の入力用 WFST のそれぞれに，w_s^i を加算する ε 遷移を挿入することで実現する．

4. 評価実験

4.1 実験条件

実験では，ビデオ予約ドメインの 4,186 発話とレンタカー予約ドメインの 3,281 発話を用いた．ビデオ予約ドメインでは 25 人の被験者から 1 人につき 8 対話のデータを収集した．レンタカー予約ドメインでは 23 人の被験者から 1 人につき 8 対話のデータを収集した．対話データの収集では，被験者に簡単な予約タスクを課し，実際に構築したシステムと対話をしてもらった．被験者への指示の中に，システムで利用できる内容語（チャネル名や営業所名）は含まれるが，具体的にどのような文法を使用できるかは含まれていない．音声認識器は Julius [⋆1] を用いた．ビデオ予約ドメインの言語モデルの語彙サイズは 209 で，レンタカー予約ドメインでは 891 であった．音声認識に用いる統計的言語モデルは，ドメイン文法記述から生成した例文から作成した．本論文ではビデオ予約ドメインでは 10,000 文，レンタカー予約ドメインでは 40,000 文の例文から作成した言語モデルを実験に使用した．レンタカー予約ドメインの例文数をビデオ予約ドメインの 4 倍にしたのは，それぞれのドメインの語彙サイズの比を考慮したからである．平均の音声認識率はビデオ予約ドメインで 83.9%，レンタカー予約ドメインで 65.7%であった．ビデオ予約ドメインの文法は，日づけ，時間，チャンネル，コマンドを指定するキーフレーズからなる．レンタカー予約ドメインの文法は，日づけ，時間，営業所，車のクラス，オプション，コマンドを指定するキーフレーズからなる．WFST の作成と合成には，MIT FST toolkit [16] を利用した．

4.2 WFST による言語理解精度

本手法による言語理解をビデオ予約ドメインとレンタカー予約ドメインの発話データに対して行った．学習データに対して，CER が最も低くなる重みづけの組合せを学習し，テストデータでの CER を比べた．評価は 4-fold cross validation で行い，それぞれ全データの 4 分の 3 を学習に，残りを評価に用いた．学習・評価セットでは話者による区別は行っていない．

比較対象となるベースライン手法は，単純なキーフレーズスポッティングと，音声認識結果の信頼度を利用したキーフレーズスポッティングとする．これは，大量の学習データが利用できない状況を想定したためである．単純なキーフレーズスポッティングは，音声認識結

⋆1 http://julius.sourceforge.jp/

表 4　それぞれのドメインでの concept error rate（CER）
Table 4　Concept error rates (CERs) in each domain.

音声認識	言語理解	信頼度	ビデオ予約ドメイン	レンタカー予約ドメイン
文法	キーワードスポッティング	無	22.1	51.1
文法	キーワードスポッティング	有	22.0	43.7
統計的言語モデル	キーワードスポッティング	無	16.9	28.9
統計的言語モデル	キーワードスポッティング	有	17.0	26.4
統計的言語モデル	WFST	—	13.5	22.0

果に含まれるコンセプトを単純に取り出し，音声認識誤りや文法的制約は考慮しない．信頼度を利用するキーフレーズスポッティングでは，単語の信頼度が閾値より低いコンセプトは棄却する．信頼度の閾値は学習データでの CER が最小となるよう選んだ．音声認識器は，ドメイン文法記述による文法に基づくものと，それから自動的に生成される統計的言語モデルに基づくものの両方を用いた場合を調べた．

使用する文法は，ラピッドプロトタイピングにおいて使用することを想定し，収集した対話データに基づく文法の追加や修正は行っていない．また，この文法では文法構造を表す FST と同様の位置でフィラーによる遷移を許容する．フィラーとして使用する単語は，統計的言語モデルの作成時と同じく，「あー」や「えっと」などフィラーとして現れやすい 6 単語を使用した．文法による音声認識の認識率は，ビデオ予約ドメインで 66.3%，レンタカー予約ドメインで 43.2%であった．これら文法による音声認識率が低いのは，被験者への文法についての教示がなく，文法に完全に一致しない発話が含まれていたことが原因として考えられる．また，本研究では言語理解部のラピッドプロトタイピングを指向しているため，作成した文法は追加や修正などを行っておらず，ユーザの多様な発話に対応できないことも原因としてあげられる．

表 4 に実験の結果を示す．単純なキーフレーズスポッティングを行う場合，統計的言語モデルを利用することで文法に基づき音声認識を行う場合に比べて，CER がビデオ予約ドメインで 5.2 ポイント，レンタカー予約ドメインで 22.2 ポイント改善された．信頼度を利用した場合は，それぞれ 5.0 ポイント，17.3 ポイントずつ改善されている．この結果は，音声認識器には文法モデルよりも，それから自動生成した統計的言語モデルが適していたことを示している．これは，フィラーや未知語を含む発話に対して，文法による音声認識では発話全体を文法にあてはめようとして誤認識してしまうのに対し，統計的言語モデルでは，フィラーや未知語以外の部分を正しく認識できたからだと考えられる．

さらに，同じ統計的言語モデルによる認識結果に対して WFST の重みづけを学習することで，信頼度を利用したキーフレーズスポッティングよりそれぞれ 3.5 ポイント，4.4 ポイント CER が改善した．これは，WFST に基づく言語理解では文法構造（たとえば，月と日の間には時間は入らないなど）を考慮して重みづけを行うからである．

表 5　それぞれのドメインでの最適なパラメータの組合せ
Table 5　The optimal parameters in each domain.

ドメイン	N	α_w	w_w	α_c	w_c
ビデオ予約	1	1.0	**word(const.)**	0	—
レンタカー予約	10	1.0	**word(CM)**-0.0	1.0	**cpt(#p·CM)**-0.8

表 6　最適なパラメータを入れ替えたときの CER
Table 6　CER when the optimal parameters are exchanged.

パラメータ					ドメイン/CER	
N	α_w	w_w	α_c	w_c	ビデオ予約	レンタカー予約
1	1.0	**word(const.)**	0	—	13.5	25.4
10	1.0	**word(CM)**-0.0	1.0	**cpt(#p·CM)**-0.8	18.7	22.0

表 5 にそれぞれのドメインでの最適なパラメータの組合せを示す．ここで，ビデオ予約ドメインでの $\alpha_c = 0$ はコンセプトに対する重みが利用されなかったことを意味する．ビデオ予約ドメインでは，音声認識率が高く語彙サイズが小さいので，音声認識器の N-best 出力に対する重みづけを利用せず，受理単語数が最大となるような重みづけが選択された．一方レンタカー予約ドメインでは，音声認識率が低く語彙サイズが大きいので，音声認識器の N-best 出力に対する重みづけや，音声認識の信頼度を利用した重みづけが選択された．この違いは，それぞれのドメインでのタスクや語彙の違いや音声認識精度の違いが反映されていると考えられる．

さらに，それぞれのドメインで最適であったパラメータを入れ替えた場合の CER を調べた．この実験は，他のドメインで最適化されたパラメータを用いることで，パラメータが当該ドメインで最適化されていない場合の性能劣化を検証している．結果を**表 6** に示す．ビデオ予約ドメインでは CER が 13.5 ポイントから 18.7 ポイントへ，レンタカー予約ドメインでは 22.0 ポイントから 25.4 ポイントへとそれぞれ劣化した．以上の結果は，最適なパラメータの組合せはドメインごとに違い，それぞれのドメインで学習が必要であることを示している．

4.3　学習データ量と言語理解の性能

我々はさらに，重みづけのための学習量と言語理解の性能の関係を調べた．実験では，

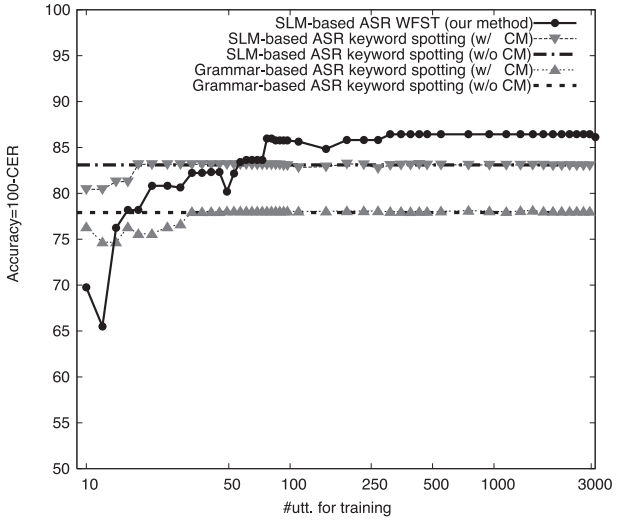


図 8　学習データ量によるコンセプト正解精度の変化（ビデオ予約ドメイン）
Fig. 8　Accuracy when training data increased (video domain).

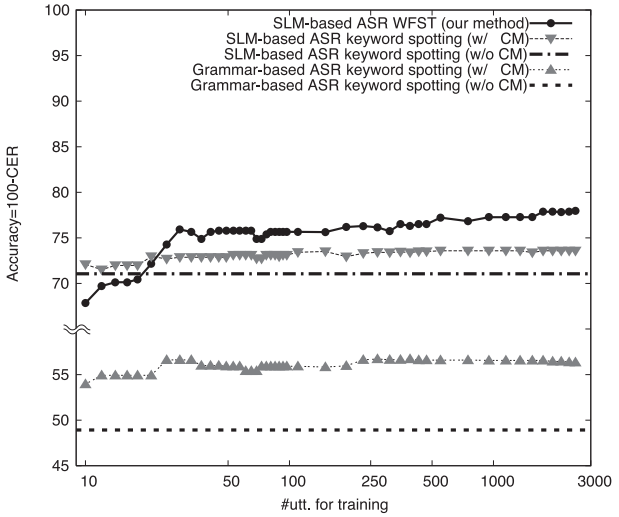


図 9　学習データ量によるコンセプト正解精度の変化（レンタカー予約ドメイン）
Fig. 9　Accuracy when training data increased (rent-a-car domain).

学習に使用する発話の数を変化させたときの，テストデータでのコンセプト正解精度（$Accuracy = 100 - CER$）を比べた．評価は同じく 4-fold cross validation で行った．**図 8**，**図 9** に実験の結果を示す．信頼度を利用したキーフレーズスポッティングでは，学習量を増やせばある程度コンセプト正解精度が向上するが，提案手法より改善幅は小さい．これは，信頼度だけでは音声認識誤りを正しく棄却ができず，文法構造などを考慮する必要があることを示している．

これらの図から本手法はビデオ予約ドメインで約 80 発話，レンタカー予約ドメインで約 30 発話でキーフレーズスポッティングより高いコンセプト正解精度に達している．つまり，およそ 100 発話もあれば，従来の信頼度のみに基づく言語理解よりも精度良く本手法は動作する．従来の WFST を利用した手法では学習に数千発話のデータを利用していた．したがって，この結果は本手法は従来手法より少ない学習データで言語理解部を構築でき，ラピッドプロトタイピングに効果的であることを示している．

5. ま と め

我々は，音声対話システムにおける言語理解部のラピッドプロトタイピング手法を開発した．評価実験では，100 発話程度の学習データがあれば，ベースラインより高いコンセプト正解精度を達成することを確認した．

本研究の意義を以下に述べる．

(1)　音声対話システムにおける言語理解部のラピッドプロトタイピング技術の必要性を指摘し，少ない労力で音声認識誤りに頑健な言語理解部を構築する手法を開発した．これまでは，対話のフローや音声認識用言語モデルに対するラピッドプロトタイピング技術の研究は行われていたが，言語理解部に対する手法はなかった．本手法は，WFST に対する単純な重みづけの設計により，言語理解部の構築に必要なデータが少なくて済み，ラピッドプロトタイピングに適している．

(2)　音声対話システム開発のライフサイクルにおいて，開発開始直後と，統計的手法の適用に十分なデータが得られた時点との間を埋める技術の必要性を指摘した．本手法に基づく比較的頑健なシステムでデータ収集を進め，十分な量のデータが用意できれ

ば，統計的手法の適用が可能となり，より高精度な処理が行える．開発初期の，データ量が少ない段階において，より頑健な言語理解を実現することは，様々なドメインで新たなシステムを開発するうえで必須の技術である．

WFST を用いた言語理解部をラピッドプロトタイピングに生かせるというアイディアは産学連携研究から得られた．「学」の立場で研究される音声対話システムは，大量のデータを利用して統計的に言語理解を行うシステムが多い．一方産業界では，顧客の要求などに応じて，当該ドメインのデータがない状態から，新たなドメインで音声対話システムを構築するというニーズがある．本研究のアイディアは，そのような産業界の視点に立った議論に基づいて得られたものである．

今後，さらに学習データが用意できた場合の，従来の統計的手法との比較が必要である．この比較結果に基づき，本手法から統計的手法へ移行可能なデータ量の自動的決定が，さらなる課題としてあげられる．

謝辞　レンタカー予約システムの作成にご協力いただいた，北海道大学情報学研究科伊藤敏彦氏，永野由佳氏に感謝する．本研究の一部は，科学研究費補助金（基盤（S），特定領域「情報爆発」，若手（B）），グローバル COE プログラム「知識循環社会のための情報学教育研究拠点」，SCAT 研究助成の支援を受けた．

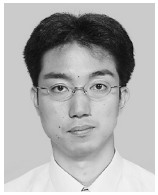
参　考　文　献

1) Schatzmann, J., Weilhammer, K., Stuttle, M.N. and Young, S.: A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies, *The Knowledge Engineering Review*, Vol.21, No.2, pp.97–126 (2006).
2) Sudoh, K. and Tsukada, H.: Tightly integrated spoken language understanding using word-to-concept translation, *Proc. EUROSPEECH*, pp.429–432 (2005).
3) Potamianos, A. and Kuo, H.-K.J.: Statistical recursive finite state machine parsing for speech understanding, *Proc. ICSLP*, Vol.3, pp.510–513 (2000).
4) Wutiwiwatchai, C. and Furui, S.: Hybrid statistical and structural semantic modeling for Thai multi-stage spoken language understanding, *Proc. HLT-NAACL Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing*, pp.2–9 (2004).
5) VoiceXML Forum: VoiceXML Forum. http://www.voicexml.org/
6) 荒木雅弘：ボイスウェブの可能性：VoiceXML 概説，情報処理，Vol.44，No.10，pp.1044–1051 (2003).
7) 小林　聡，中村有作，桂田浩一，山田博文，新田恒雄：マルチモーダル対話記述言語 XISL の提案，情報処理学会研究報告，2001-SLP-37-8 (2001).
8) Misu, T. and Kawahara, T.: A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts, *Proc. ICSLP*, pp.9–13 (2006).
9) Weilhammer, K., Stuttle, M.N. and Young, S.: Bootstrapping language models for dialogue systems, *Proc. ICSLP*, pp.17–20 (2006).
10) 小暮　悟，中川聖一：データベース検索用音声対話システムにおける移植性の高い意味理解部・検索部の構築と評価，情報処理学会論文誌，Vol.43, No.3, pp.714–733 (2002).
11) Seneff, S.: TINA: A natural language system for spoken language applications, *Computational Linguistics*, Vol.18, No.1, pp.61–86 (1992).
12) Seto, S., Kanazawa, H., Shinchi, H. and Takebayashi, Y.: Spontaneous speech dialogue system TOSBURG II and its evaluation, *Speech Communication*, Vol.15, No.3-4, pp.341–353 (1994).
13) Kawahara, T., Lee, C.-H. and Juang, B.-H.: Flexible speech understanding based on combined key-phrase detection and verification, *Speech and Audio Processing, IEEE Trans.*, Vol.6, No.6, pp.558–568 (1998).
14) 駒谷和範，鹿島博晶，田中克明，河原達也：複合的言語制約に基づくキーフレーズ検出を用いた汎用的なデータベース検索音声対話プラットフォーム，情報処理学会論文誌，Vol.44, No.5, pp.1333–1342 (2003).
15) Lee, A., Shikano, K. and Kawahara, T.: Real-time word confidence scoring using local posterior probabilities on tree trellis search, *Proc. ICASSP*, Vol.1, pp.793–796 (2004).
16) Hetherington, L.: The MIT finite-state transducer toolkit for speech and language processing, *Proc. ICSLP*, pp.2609–2612 (2004).

(平成 19 年 11 月 11 日受付)
(平成 20 年 5 月 8 日採録)

福林雄一朗（正会員）

2006 年京都大学工学部情報学科卒業．2008 年同大学院情報学研究科知能情報学専攻修士課程修了．在学中は音声対話システムの研究に従事．現在，日本電気株式会社勤務．本学会第 70 回全国大会学生奨励賞受賞．



駒谷　和範（正会員）

1998 年京都大学工学部情報工学科卒業．2000 年同大学院情報学研究科知能情報学専攻修士課程修了．2002 年同大学院博士後期課程修了．京都大学博士（情報学）．同年京都大学情報学研究科助手．2007 年より助教．音声対話システムの研究に従事．情報処理学会 2004 年度山下記念研究賞，FIT2002 ヤングリサーチャー賞等受賞．電子情報通信学会，言語処理学会，人工知能学会，ACL 各会員．

中野　幹生（正会員）

1988 年東京大学教養学部基礎科学科第一卒業．1990 年同大学院理学系研究科相関理化学専攻修士課程修了．1990〜2004 年日本電信電話（株）にて，自然言語処理，音声対話システムの研究に従事．この間 2000〜2002 年 MIT 計算機科学研究所客員研究員．博士（理学）．2004 年より（株）ホンダ・リサーチ・インスティチュート・ジャパンに勤務．現在，同社プリンシパル・リサーチャ．音声コミュニケーションの研究に従事．言語処理学会，人工知能学会，日本ロボット学会，電子情報通信学会，ACL，ACM，IEEE 各会員．

船越孝太郎（正会員）

2000 年東京工業大学工学部情報工学科卒業．2002 年同大学院情報理工学研究科計算工学専攻修士課程修了．2005 年同大学院博士課程修了．同年同大学院特別研究員．2006 年より（株）ホンダ・リサーチ・インスティチュート・ジャパン，リサーチャ．博士（工学）．自然言語理解，音声対話に関する研究に従事．2005 年度言語処理学会年次大会優秀発表賞受賞．人工知能学会，言語処理学会，AAAI 等会員．

辻野　広司

1984 年東京工業大学理学部情報科学科卒業．1986 年同大学院情報科学専攻修士課程修了．1987 年（株）本田技術研究所入社．2003 年より（株）ホンダ・リサーチ・インスティチュート・ジャパン，チーフ・リサーチャ．脳型コンピュータ，知能システム，ヒューマンロボットインターフェース，画像認識等の研究に従事．IEEE，SFN，INNS，日本ロボット学会，人工知能学会，日本ソフトウェア科学会各会員．

尾形　哲也（正会員）

1993 年早稲田大学理工学部機械工学科卒業．日本学術振興会特別研究員，早稲田大学理工学部助手，理化学研究所脳科学総合研究センター研究員，京都大学大学院情報学研究科講師を経て，2005 年より同助教授（現・准教授）．博士（工学）．この間，早稲田大学ヒューマノイド研究所客員助教授（現・准教授）．人間とロボットのインタラクションと協調，神経回路モデル等の研究に従事．2000 年度日本機械学会論文賞，IEA/AIE-2005 最優秀論文賞等を受賞．RSJ，JSME，JSAI，IEEE 等会員．

奥乃　　博（正会員）

1972 年東京大学教養学部基礎科学科卒業．日本電信電話公社，NTT，JST，東京理科大学を経て，2001 年より京都大学大学院情報学研究科知能情報学専攻教授．博士（工学）．この間，スタンフォード大学客員研究員，東京大学工学部客員助教授．人工知能，音環境理解，ロボット聴覚，音楽情報処理の研究に従事．1990 年度人工知能学会論文賞，IEA/AIE-2001，2005 最優秀論文賞，IEEE/RSJ IROS-2001，2006 Best Paper Nomination Finalist，第 2 回船井情報科学振興賞等受賞．JSAI，RSJ，ACM，IEEE 等会員．本学会英文図書出版委員．