社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－NL－179 (8)
2007－SLP－ 66 (8)
2007／5／25

音声認識結果とコンセプトへの重みづけによるWFSTに基づく音声言語理解の高精度化

福林雄一朗† 駒谷和範† 中野幹生‡ 船越孝太郎‡ 辻野広司‡ 尾形哲也† 奥乃博†

†京都大学大学院 情報学研究科 知能情報学専攻
fukubaya@kuis.kyoto-u.ac.jp {komatani,ogata,okuno}@i.kyoto-u.ac.jp
‡(株)ホンダ・リサーチ・インスティチュート・ジャパン
{nakano,funakoshi,tsujino}@jp.honda-ri.com

Weighted Finite State Transducer (WFST) を用いた言語理解では，入力となる音声認識結果の単語列に対して，各単語に適切な重みを与えることで頑健な言語理解を実現する．しかし一般にその学習には大量のデータが必要であるため，新たなドメインで構築した音声対話システムにおいてWFSTを用いた言語理解は困難であった．そこで我々は，音声認識結果をフィラーや単語，コンセプトなどとして抽象化し，これらに対して音素数や音声認識の信頼度を利用した重みを割当てる方法を開発した．これにより，大量の学習データが用意できない状況でも頑健な言語理解部を容易に構築できる．評価実験では，発話の音声認識率に応じて重みを適切に設定することで，言語理解精度が向上することを確認した．この結果は，音声認識率やユーザなどの状況に合わせて重みづけを選択することで言語理解精度が向上する可能性を示した．

Improving WFST-based Language Understanding Accuracy by Weighting for ASR Results and Concepts

YUICHIRO FUKUBAYASHI†, KAZUNORI KOMATANI†, MIKIO NAKANO‡,
KOTARO FUNAKOSHI‡, HIROSHI TSUJINO‡, TETSUYA OGATA† and HIROSHI G. OKUNO†

† *Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University*
‡ *Honda Research Institute Japan Co., Ltd.*

Weighted Finite State Transducers (WFST) have become common as language understanding modules in spoken dialogue systems. WFSTs promote robust language understanding by assigning appropriate weights to a sequence of recognized words. However, it is difficult to make a language understanding module for a new spoken dialogue system using a WFST because a lot of training data is required to learn the weights. To create a robust language understanding module with less training data, we developed a model in which ASR results are classified into two classes, fillers and accepted words and concepts are formed by the latter. We then assign appropriate weights to these simplified results and concepts. The weights are designed by considering the number of phonemes and their ASR confidence. Experimental results showed that the language understanding accuracy was improved when the optimal setting from these parameters was selected based on the ASR accuracy of the utterance. This result shows that a language understanding accuracy can be improved if the optimal setting is selected according to the environment such as users and ASR accuracy of the utterance.

1. はじめに

音声対話システムにおける言語理解として音声認識誤りに頑健なものが求められている．また，そうした言語理解部は少量の学習データで構築できることが望ましい．学習データの収集には，大量の時間と手間がかかるので，必要な学習データが少ない方が新たな対話システムの言語理解部を作りやすくなるからである．これまで，音声対話システムにおける言語理解部の実装手法としていくつかの方法が提案されてきた．音声

認識器に文法ベースのものを利用する方法が最も単純な方法である．この方法では，音声認識結果からシステムの内部表現であるコンセプトへの変換が容易である．しかし，ユーザの様々な表現を受け入れるためには複雑な文法を用意する必要があり，システム制作者への負担が大きい．

また，他の方法としては，ユーザの発話をキーワード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がある[1]．この方法では，ルールに大きな修正を加えることなくユーザの発話をコンセプトへ変換できる．しかし，複雑なルールの準備には時間や手間がかかり，文法を利用した手法と同様にシステム制作者への負担が大きい．

この問題に対処するために，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，これらの手法は大量の学習データを必要とし，新たなドメイン向けの言語理解部を構築するのは容易ではない．また，学習した結果は利用したコーパスのドメインに依存したものである．しかも，重みは固定なので発話の状況やユーザの変化には対応できない．

我々は，WFST に基づく言語理解の新しい手法を開発した．WFST への入力は，統計的言語モデルに基づく音声認識器による音声認識結果である．我々の手法では，WFST に対する重みづけを，認識された単語と言語理解結果であるコンセプトの 2 つのレベルで行う．この重みづけは，従来手法に比べ単純であり，少ないデータで言語理解部の構築が可能である．また，重みづけに利用する特徴量はドメイン非依存であり，一般的な音声対話システムに適用できる．評価実験では，対象とするドメインで適切なパラメータを選択することで言語理解精度が向上することを確認した．さらなる調査の結果，このパラメータは，音声認識率に依存して変化するため，我々の手法では当該ドメインで予測される音声認識率に応じて適切なパラメータを選択することで言語理解精度が向上する．この結果は，音声認識率やユーザなどの状況に応じて適切にパラメータを選択することで，適応的に言語理解精度が向上する可能性を示している．

2. 関連研究と WFST に基づく言語理解

音声対話システムの言語理解として，タグ付けされたコーパスを利用した学習による方法が提案されている[2]．この方法では，コーパスから音声認識結果とそれに対応するコンセプトの組の出現確率を学習する．したがって，言語理解部の構築には大量のタグ付けされたコーパスが必要であり，新たなドメインの言語理解部を構築するのは容易ではない．

言語理解の手法として，WFST を利用した方法も提案されている[3),4)]．ここでまず，FST について簡単に説明する．一般に，FST は入力列に対して，状態を遷移しながら入力に応じた列を出力するオートマトンで，一種の変換器とみなせる．WFST では，各状態遷移に対して重みを設定でき，最終的な出力列の他に累積重みが得られる．図 1 に WFST の例を示す．この図では，“a:y/0.7” は “a” が入力されたら “y” を出力し，0.7 を累積重みに足して遷移することを示している．この例では，入力 “abbd” に対して “yzz” が出力される．その時の累積重みは 2.5 である．

FST に基づく言語理解部では，音声認識結果を入力し，出力として言語理解結果を得る．図 2 はビデオ予約システムの言語理解部の FST の例である．入力の ε は，入力なしでの遷移が可能であることを表す．この例では，「開始時間 は 10 時 30 分 です」という入力列に対して，“開始時間 は $ 10 時 hour=10 $ 30 分 minute=30 です” という出力列が得られる☆．最終的に言語理解結果として，[hour=10, minute=30] を得る．しかし，この方法では「えーと 開始時間 は 10 時 30 分 です」という入力に対しては，「えーと」に対して遷移先がなく，言語理解結果が得られない．そこで，我々は任意の入力を受け入れる FILLER 遷移を導入した．0 回以上の FILLER 遷移 (図 2 の’F’) を各フレーズ間に挿入することで，フィラーの影響を受けることなく正しい言語理解結果が得られる．

しかしながら，FILLER 遷移を導入すると，ひとつの入力列に対して何通りもの出力列が結果として得られることになる．ひとつの入力列に対して，WFST 上での遷移は何通りもあるからである．WFST に基づく言語理解では，何通りもある出力列から累積重み w が最も大きいものを言語理解結果として採用する．表 2 では，累積重み w が 2.0 と最も高い [hour=10, minute=30] が言語理解結果として採用される．

WFST を利用した従来の手法では，各遷移の重みを大量のコーパスから学習していた[3),4)]．しかし，コーパスの収集には大きな労力が必要で，新たなドメインの言語理解部の構築は難しかった．また，重みは固定であるので，発話の状況やユーザの違いにより言語理解精度が大きく変わる可能性がある．特に，言語理解は音声認識の精度に強く依存したものであり，その精度に応じたモデル化が必要である．我々の手法では，重みづけを音声認識結果の単語の長さや信頼度などのド

☆ $は何も出力されなかった場合を考慮したダミー記号である．

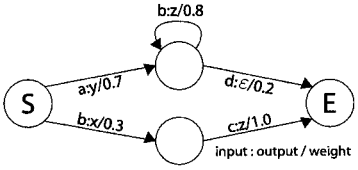

図 1　WFST の例

表 1　「開始時間 は 10 時 30 分 です」に対する言語理解結果の例

出力					言語理解結果	w
開始時間	は	10 時	30 分	です	hour=10, minute=30	2.0
F	F	10 時	30 分	です	hour=10, minute=30	2.0
F	F	10 時	F	です	hour=10	1.0
F	F	F	30 分	F	minute=30	1.0
F	F	F	F	F	n/a	0

(F は FILLER を表す)

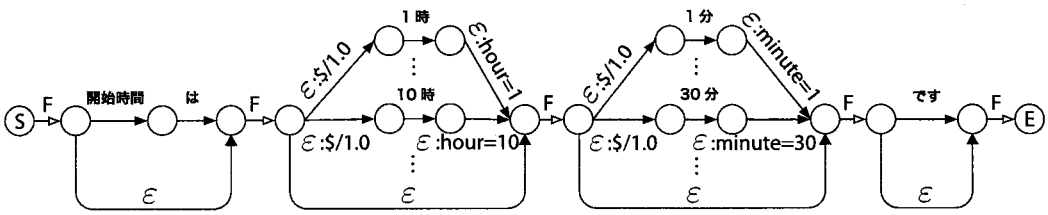

図 2　WFST による言語理解の例

メインに非依存な特徴量を利用して行う．したがって，大量のコーパスがなくても容易に言語理解部を構築できる．さらに，評価実験では音声認識率ごとに最適な重みづけの組み合わせを調べ，すべての発話に対して同じ重みづけをした場合よりも言語理解精度が向上することを確認した．

3. 音声認識結果とコンセプトに対する重みづけ

我々は WFST に対する重みづけを 2 つのレベルで定義する．ひとつは，**音声認識結果に対する重みづけ**で単語レベルで信頼できる出力結果を選択するために設定する．もうひとつは，**コンセプトに対する重みづけ**でコンセプトレベルで信頼できる出力結果を選択するために設定する．コンセプトに対する重みづけは，認識された単語よりも抽象的なレベルでの正しさを反映する．また，音声認識結果に対する重みづけは**受理単語に対する重みづけ**と *FILLER* **に対する重みづけ**の 2 つに分けられる．本節では，これらの重みづけを説明する．

3.1 受理単語に対する重みづけ

WFST に入力し受理された単語，つまり FILLER 以外の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．通常は，フィラー以外の単語が出力列に多くなるように，音声認識結果が信頼できる入力が優先されるように設定する．我々は，この重みづけ w_w を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)**: $w_w = CM(W)$

word (const.) は受理された全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ための設計である．**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ていると言える．さらに，受理単語の信頼度を考慮に入れた **word (CM)** も提案する．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[5] $CM(W)$ を利用している．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と言える．

3.2 FILLER に対する重みづけ

我々はフィラーに対する重みも設計した．すべての入力単語をフィラーとして扱えるので，フィラーに対する重みはペナルティとして考え，負の値を設定した．一般的には，入力となる音声認識結果が信頼でき，正しい理解結果が含まれているならば，フィラーが少なくかつ短かい出力列を優先するように設定する．

我々は，受理単語に対する重みづけと同様にして以下のように重みづけ w_f を 2 種類設計した．

(1)　**FILLER (const.)**: $w_f = -1.0$

(2)　**FILLER (#phone)**: $w_f = -l(W)$

FILLER (const.) はフィラーの数に対するペナルティであり，**FILLER (#phone)** はフィラーとされた単語の長さも考慮したペナルティである．

3.3 コンセプトに対する重みづけ

我々は，単語レベルでの重みに加えて，コンセプトレベルにおける重みも設計した．コンセプトは，複数

の単語から成り，音声認識結果をWFSTに入力することで得られる．コンセプトに対する重みは，それぞれのコンセプトに含まれる単語の信頼度などを用いて計算する．

我々は，以下のように重みづけw_cを5種類設計した．

(1) **cpt (const.)**: $w_c = 1.0$

(2) **cpt (avg)**: $w_c = \frac{\sum_{\boldsymbol{W}} CM(W)}{\#\boldsymbol{W}}$

(3) **cpt (min)**: $w_c = \min_{\boldsymbol{W}}(CM(W))$

(4) **cpt (lenCM(avg))**: $w_c = \frac{\sum_{\boldsymbol{W}}(CM(W) \cdot l(W))}{\#\boldsymbol{W}}$

(5) **cpt (lenCM(min))**: $w_c = \min_{\boldsymbol{W}}(CM(W) \cdot l(W))$

$\boldsymbol{W}$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が多くなるようにするための重みづけである．また，**cpt (avg)**や**cpt (min)**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 (lenCM(avg))**や**cpt (lenCM(min))**は，コンセプトに含まれる単語の信頼度の他にそれらの長さも考慮に入れた重みづけである．どちらもコンセプト部分が長くかつ信頼できる発話を優先するための設定である．**cpt (avg)**や**cpt (lenCM(avg))**で平均を計算しているのは，コンセプトを構成するすべての単語の影響を反映するためである．また，**cpt (min)**や**cpt (lenCM(min))**で最小値を選ぶのは，不当に信頼度が高い単語による湧き出し誤りを防ぐためである．

3.4 累積重みの計算

言語理解結果は，以上で示した3種類の重みw_w, w_f, w_cの重みつき和である累積重みwによって選ばれる．言語理解部は，累積重みwが最も大きい出力列を選ぶ．

$$w = \sum \alpha_w w_w + \sum \alpha_f w_f + \sum \alpha_c w_c$$

累積重みwの計算方法を表2を用いて説明する．この例では，パラメータとして**word (CM)**, **FILLER (const.)**, **cpt (lenCM(avg))**を選択している．入力が「いいえ2月22日です」がである場合，この表では受理単語に対する重みはの総和は$3.5\alpha_w$で，FILLERに対する重みの総和は$-1.0\alpha_f$である．また，コンセプト“month=2”に対する重み$\alpha_c(0.9 \cdot 0.9)/1 = 0.81\alpha_c$とコンセプト“day=22”に対する重み$\alpha_c(1.0 \cdot 0.9 + 0.9 \cdot 0.6)/2 = 0.72\alpha_c$により，コンセプトに対する重みの総和は$1.53\alpha_c$である．したがって，この入力列に対する累積重み$w$は$3.5\alpha_w - 1.0\alpha_f + 1.53\alpha_c$となる．

4. 評価実験

4.1 実験条件

3章で定義した重みづけを実験的に評価する．実験ではまず，ユーザ発話の音声認識結果をWFSTに入力し，累積重みwが最も高い出力列を言語理解結果として採用する．この言語理解結果を正解データと比較して言語理解精度を計算する．なお，言語理解が得られない「なし」が正解であることもあるので，音声認識率が0%でも言語理解精度が100%になることはありうる．実験では，重みづけや各重みの係数$\alpha_{w,f,c}$をさまざまな組合せで変化させ言語理解精度を比べた．係数α_wは1.0に固定し，他の係数α_fとα_cを0, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0と変化させた．$\alpha_{f,c} = 0$は，対応する重みが利用されないことを表している．

実験では，ビデオ予約ドメインの4186発話とレンタカー予約ドメインの3281発話を用いた．ビデオ予約ドメインは25人の被験者の8対話から，レンタカー予約ドメインは23人の被験者の8対話から発話を収集した．音声認識器はJulius☆を用いた．言語モデルは，各ドメインの認識文法から生成した例文10000文から作成した統計的言語モデルである．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の言語モデルの語彙サイズは226であった．平均の音声認識率☆☆はビデオ予約ドメインで80.3%，レンタカー予約ドメインで52.8%であった．それぞれのドメインの言語理解の正解は，書き起こしをWFSTに入力して作っている．

4.2 すべての発話に対して最適なパラメータの組み合わせ

本稿では，入力に対して単純に文法との最長一致をとる言語理解をベースラインとする．このベースラインは，重みづけをw_w =**word (const.)**に，α_fを0に(FILLERに対する重みづけは利用しない)，α_cを0に(コンセプトに対する重みづけは利用しない)する場合が相当する．

全発話に対して最適な重みと係数の組み合わせを求めた．ビデオ予約ドメインでは，w_w =**word (const.)**, $\alpha_f = 1.0$, w_f =**FILLER (#phone)**, $\alpha_c = 5.0$, w_c =**cpt (lenCM(avg))**の時で平均言語理解精度は87.3%，レンタカー予約ドメインでは，w_w =**word (CM)**, $\alpha_f = 0.5$, w_f =**FILLER (#phone)**, $\alpha_c = 0$の時で平均言語理解精度は65.0%であった．ビデオ予約ドメインのベースラインの平均言語理解精度は81.5%で，最適時とそれほ

☆ http://julius.sourceforge.jp/

☆☆ 本稿では音声認識率を挿入誤りまで考慮して計算したので，音声認識率は負になることもある．

表 2　パラメータとして word (CM), FILLER (const.), cpt (lenCM(avg)) を選択したときの重みづけの例

入力 (音声認識結果)	いいえ	2 月	22	日	です
出力	F	2 月	22	日	です
$CM(W)$	0.3	0.9	1.0	0.9	0.7
$l(W)$	0.3	0.9	0.9	0.6	0.6
コンセプト	-	month=2	day=22		-
word	-	$\alpha_w \cdot 0.9$	$\alpha_w \cdot 1.0$	$\alpha_w \cdot 0.9$	$\alpha_w \cdot 0.7$
FILLER	$\alpha_f \cdot (-1.0)$	-	-	-	-
cpt	-	$\alpha_c(0.9 \cdot 0.9)/1$	$\alpha_c(1.0 \cdot 0.9 + 0.9 \cdot 0.6)/2$		-

ど大きな差はなかった．これは，ビデオ予約ドメインの平均音声認識率が 80.3%と比較的高かったことが原因であると考えられる．つまり，音声認識に誤りがほとんどないならば，単純に最も長く一致するような出力結果を選択すればよいということである．一方で，レンタカー予約ドメインの平均音声認識率は 52.8%と低く，最適時の平均言語理解精度 65.0%と比べて，ベースラインでは 45.5%と大きな差ができている．つまり，音声認識率が低い場合，最適となる組み合わせはベースラインとして設定した組み合せとは異なる．

4.3 音声認識率に合わせたパラメータの組み合わせ

4.2 節の結果より，最適な重みづけの組み合わせは発話の音声認識率に合わせて決定すると改善することが分かる．そこで，音声認識率ごとに適切なパラメータの組み合わせを調べた．発話データを音声認識率ごとに分類し，それぞれの音声認識率ごとに言語理解精度を計算した．そして，各認識精度ごとにベースラインと言語理解精度を比べた．表 3, 4 はその結果である．表中のクラス 10-30 は，音声認識率が 10%以上 30%未満であることを表す．ただし，90-100 は 100%も含む．

この表からすべてのクラスでベースラインより言語理解精度が向上していることが分かる．特にレンタカー予約ドメインでは，言語理解精度がベースラインと比べて大きく向上している．

この結果から，どちらのドメインでも音声認識率が高い発話では受理単語に対する重みづけとして，**word (const.)** や **word (#phone)** が有効で，音声認識率が低い発話では **word(CM)** が有効であることが分かる．特にレンタカー予約ドメインではその傾向が強い．これは，発話が正しく認識されているときは，受理単語が最も多い出力を選択し，音声認識率があまりよくないときには，信頼できる部分だけを選択すると正しい言語理解が得られるからであると考えられる．

両ドメインのどのクラスでもフィラーに対するペナルティが必要であることが分かる．これは，フィラーによるコンセプトの湧き出し誤りを抑制する必要があるためと考えられる．また，どちらのドメインでもほとんどのクラスで **FILLER (#phone)** が最適である．確かに，フィラーは言語理解には意味のない情報であるから，単語数よりも音素数 (継続時間) の方がペナルティの基準としては適切であると考えられる．

コンセプトに対する重みは，どちらのドメインでも必要であり，単語レベルの重みに加えて，コンセプトレベルの重みも有効であることが示されている．レンタカー予約ドメインの $-\infty$-100 (全発話) クラスでは，コンセプトに対する重みはなしが最適となっているが，**cpt (lenCM(avg))** や **cpt (lenCM(min))** としても言語理解精度は 64.9%とほとんど変わらない．どちらのドメインでも，ほとんどのクラスで **cpt (avg)** や **cpt (lenCM(min))** など単語信頼度を利用したものが最適である．コンセプトに対する重みとして単語信頼度が有効に働いていると言える．

以上に示した結果は，重みづけのパラメータの組み合わせを音声認識率に応じて適切に選択することで，言語理解精度が向上することを示している．音声対話システムにおける音声認識率は，それほど大量の発話データがなくても計算できる．したがって，我々の手法は，大量のコーパスが用意できない新しいドメインにおいて言語理解部の構築する場合に効果的であると言える．また，今回の実験の結果は，事前に認識率を計算できなくても，ユーザや状況に合わせて重みづけの組み合せを変えることで，言語理解精度が向上する可能性を示していると考えられる．例えば，ユーザが音声対話システムに慣れていない初心者ならば低い音声認識率に合わせたパラメータを選択し，周囲が静かで雑音が少ない環境ならば高い音声認識率に合わせたパラメータを選択すれば，言語理解精度のさらなる向上が期待できる．

最後に本手法の動作例を図 3 に示す．この例では，ユーザの発話は「ろくがつ みっか おねがい します」であるが「ろくがつ みっか あー ふぃっと おねがい します」と誤って認識されている．ベースライン手法では，単純に受理単語が最も多くなるように「ふぃっと おねがい します」が受理され，「ろくがつ」「みっか」は誤って棄却されてしまう．一方で，我々の手法では，「ろくがつ」「みっか」の信頼度やフィラーの長さを考慮することで正しい言語理解結果が得られる．

表 3 音声認識率ごとの最適な重みづけの組み合わせとそのときの言語理解精度 (ビデオ予約ドメイン)

音声認識率	発話数	w_w	α_f	w_f	α_c	w_c	言語理解精度	(ベースライン)
$-\infty$-100	4186	**word (const.)**	1.0	**FILLER (#phone)**	5.0	**cpt (lenCM(avg))**	87.3	81.5
$-\infty$-10	343	**word (CM)**	0.5	**FILLER (#phone)**	5.0	**cpt (lenCM(avg))**	60.8	59.2
10-30	63	**word (#phone)**	0.5	**FILLER (const.)**	1.0-5.0	**cpt (lenCM(avg))**	23.9	8.2
30-50	113	**word (const.)**	1.0	**FILLER (#phone)**	5.0	**cpt (lenCM(min))**	55.4	43.1
50-70	389	**word (#phone)**	0.5	**FILLER (#phone)**	0.5-1.0	**cpt (lenCM(avg))**	64.5	48.4
70-90	283	**word (const.)**	0.5	**FILLER (#phone)**	4.0-5.0	**cpt (lenCM(avg))**	73.3	58.7
90-100	2995	**word (const.)**	0.5-5.0	**FILLER (const.)**	0.5-5.0	**cpt (const.)**	98.4	94.6
認識率ごとの言語理解精度の平均							88.2	81.5

表 4 音声認識率ごとの最適な重みづけの組み合わせとそのときの言語理解精度 (レンタカー予約ドメイン)

音声認識率	発話数	w_w	α_f	w_f	α_c	w_c	言語理解精度	(ベースライン)
$-\infty$-100	3281	**word (CM)**	0.5	**FILLER (#phone)**	0	n/a	65.0	45.5
$-\infty$-10	718	**word (CM)**	0.5	**FILLER (#phone)**	0	n/a	15.0	11.7
10-30	142	**word (CM)**	0.5	**FILLER (#phone)**	3.0-5.0	**cpt (min)**	35.1	9.3
30-50	148	**word (CM)**	0.5-5.0	**FILLER (#phone)**	0.5-5.0	**cpt (lenCM(avg))**	48.1	22.4
50-70	369	**word (#phone)**	0.5	**FILLER (#phone)**	0.5	**cpt (avg)**	55.0	26.3
70-90	202	**word (const.)**	0.5	**FILLER (#phone)**	4.0	**cpt (lenCM(avg))**	63.4	14.2
90-100	1702	**word (const.)**	0.5-5.0	**FILLER (const.)**	0.5-5.0	**cpt (avg)**	91.5	75.2
認識率ごとの言語理解精度の平均							64.5	45.5

書き起こし	ろくがつ	みっか			おねがい	します	
音声認識結果	ろくがつ	みっか	あー	ふぃっと	おねがい	します	言語理解結果
$CM(W)$	0.978	0.757	0.152	0.525	0.741	0.521	
正解	ろくがつ	みっか	*F*	*F*	*F*	*F*	type:refer-time, month:6, day:3
最適時	ろくがつ	みっか	*F*	*F*	*F*	*F*	type:refer-time, month:6, day:3
ベースライン	*F*	*F*	*F*	ふぃっと	おねがい	します	type:specify-class, car:フィット

図 3 重みづけが有効に働く例 (レンタカー予約ドメイン)

5. おわりに

我々は，音声対話システムにおける WFST を利用した言語理解部を開発した．利用する WFST では，2 レベルの重みづけを設計した．この 2 レベルの重みづけは，認識単語レベルの言語理解とコンセプトレベルの言語理解に対応している．これらの重みは，音声認識結果中の単語の音素数や信頼度を利用して計算される．したがって，重みづけが比較的単純であり，新たなドメイン向けの言語理解部の構築が容易である．

評価実験では，2 つの異なるドメインの発話データに対して，音声認識率ごとに最適なパラメータを求めた．音声認識率ごとにパラメータを選択することで，ベースラインと比べて言語理解精度が向上することを確認した．また，2 つのドメインの音声認識率と最適な重みづけの違いから，音声認識率と最適な重みづけの関係を考察した．この結果は，音声認識率やユーザなどの発話の状況に合わせた重みづけによる言語理解精度の向上の可能性を示したと言える．今後の課題としては，音声認識結果の N-Best 候補の利用が挙げられる．N-Best 候補を利用すれば，認識結果の第 1 候補に正しい認識結果が含まれていない場合でも，N-Best の候補中に正しい認識結果が含まれていれば，その結果を利用して言語理解精度のさらなる向上が期待できる．

謝辞 レンタカー予約のシステムの作成については，北海道大学情報学研究科，伊藤 敏彦氏，永野 由佳氏のご協力を得ました．

参考文献

1) Stephanie Seneff. TINA: A natural language system for spoken language applications. *Compurational Linguistics*, Vol.18, No.1, pp. 61–86, 1992.
2) Katsuhito Sudoh and Hajime Tsukada. Tightly integrated spoken language understanding using word-to-concept translation. In *Proc. EUROSPEECH*, pp. 429–432, 2005.
3) Alexandors Potamianos and Hong-KwangJ. Kuo. Statistical recursive finite state machine parsing for speech understanding. In *Proc. ICSLP*, pp. 510–513, 2000.
4) Chai Wutiwiwatchai and Sadaoki Furui. Hybrid statistical and structural semantic modeling for Thai multi-stage spoken language understanding. In *Proc. HLT-NAACL Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing*, pp. 2–9, 2004.
5) Akinobu Lee, Kiyohiro Shikano, and Tatsuya Kawahara. Real-time word confidence scoring using local posterior probabilities on tree trellis search. In *Proc. ICASSP*, Vol.1, pp. 793–796, 2004.