方言対訳コーパスを用いた日本語方言音声認識システム

平山 直樹[1,a)]　森 信介[1,b)]　奥乃 博[1,c)]

概要：本稿では，日本語方言音声認識のための言語モデルの統計的構築法を開発する．方言言語モデル構築においては，その方言の言語コーパスの不足が大きな課題である．その解決のため，大規模共通語言語コーパスの単語単位での方言への変換を行う．共通語・方言間の対訳コーパスを用いて統計的に変換ルールを学習し，重み付き有限状態トランスデューサ (WFST) で変換器を実装する．この WFST に共通語文章を入力することで，対応する方言文章が自動的に出力される．本手法で構築した方言言語モデルを用いて方言音声認識を行うことで，共通語言語コーパスで学習した言語モデルと比較し高い認識精度が得られた．

キーワード：方言，音声認識，言語モデル，重み付き有限状態トランスデューサ (WFST)

1. はじめに

異なる地域の人々が会話によりコミュニケーションを図るときに，方言は避けて通れない要素である．同じ方言を話す人々であれば円滑なコミュニケーションがとれるが，方言が異なると，互いの方言をすぐには理解できず，たどたどしいやりとりになることがある．また，駅や空港，観光地など，各地から人の往来のある場所では，音声による情報案内システム [1] は方言発話に対応する必要が生じる．本稿では，コミュニケーションや言語理解の補助を目的とし，方言音声認識システムを構築する．

方言とは，ある言語の中で地理的要因により異なる特徴を持つ言葉を指す [2]．方言間の差異は，大きく分けて (1) 発音変化，(2) 語彙変化，(3) 語順変化の 3 種類に分類できる．(1) 発音変化は，単語そのものは同じであるが，発音が部分的に変化するものである．日本語では，近畿地方などで「しつこい」を「ひつこい」と発音される例が挙げられる．(2) 語彙変化は，同じ対象を別の単語で表現するものである．例えば，「私」など一人称の代名詞は地域により「わて」など様々に変化する．(3) 語順変化は，文における単語の順番が変更されるものである．日本語には例が少ないが，英語では地域により next Tuesday を Tuesday next という場合がある [3]．本稿においては，日本語において特に多い (1) および (2) をターゲットとした音声認識システムの構築を行う．語順変化がないと仮定することで，単語ごとに方言での表現に差し替えれば方言変換が実現できる．但し，ことばの多様性は地域のみならず，話者の年齢，性別，集団などの属性にも依存する [4] ので，システム構築時には留意する必要がある．

方言音声認識システムには以下の 3 条件が要請される．
(1) 様々な方言に対する汎用性
(2) 少ない方言言語資源で動作
(3) 方言変換，言語理解との接続容易性
条件 1 は，データの差し替えのみで様々な方言に対応するシステム構築ができることを意味する．人手による方言間の変換ルール作成は，コスト面はもちろん，方言ごとに作業者の判断が必要になるという観点でも適さない．本稿では，変換ルールの自動学習による統一的手法を用いてこれを解決する．条件 2 は，音声認識用言語モデル学習に用いる言語コーパスの課題である．方言は話し言葉という性質上，大規模な言語コーパスの入手が困難である．本稿では，共通語であれば大規模な言語コーパスが利用できることに着目し，大規模な共通語言語コーパスを用いて大方言言語コーパスをシミュレートする．条件 3 は，音声認識結果を用いた後段の処理に関係する．方言音声を認識し，得られた文章を別の方言に変換する場合，方言間の変換ルールは対象とする方言の種類の 2 乗だけ必要となり，扱う方言の種類が多くなると変換ルールの種類も膨大になる．また，音声対話における言語理解モジュールも，入力される方言にかかわらず共通化できると望ましい．本稿では，音声認識結果を共通語で出力する設計とし，方言変換や言語理解のモジュールが共通語の入力を仮定できるようにする．

我々は，重み付き有限状態トランスデューサ (Weighted

[1] 京都大学
Kyoto University, Sakyo, Kyoto 606–8501, Japan
[a)] hirayama@kuis.kyoto-u.ac.jp
[b)] forest@i.kyoto-u.ac.jp
[c)] okuno@i.kyoto-u.ac.jp

Finite-State Transducer, WFST) [5] による音素列変換器を導入し，文の変換を行う [6]．音素列ベースの変換とする理由としては，方言が話し言葉であるために方言研究資料の大半は方言をカナ表記している（漢字かな交じりでは書かれない）こと，カナ表記を音素表記することで扱うトークンの種類を削減できることが挙げられる．WFST により，小規模な対訳コーパスから抽出された確率的変換ルールをモデル化する．変換ルールが 1 対 1 ではなく確率的に与えられるとすることで，方言の多様性を表現する．また，WFST では n-gram ベースの変換ルールを表現でき [7]，前後の文脈依存性を扱うことも可能となる．

本稿の構成は以下の通りである．2 章で，方言音声認識に関連する研究を挙げる．3 章で，構築するシステムの要素を挙げた上で，それぞれの作成法を述べる．4 章で，評価実験を行い，手法の有効性を確認する．5 章で，残された課題および今後の展開について述べる．

2. 関連研究

音声認識を方言という観点で研究する際には，1 章で述べた方言間の差異に関係し，様々な方向性が考えられる．また，音声認識を音韻的・音響的特徴から捉えるか，言語的特徴から捉えるかという選択肢もある．

これまでの方言音声認識研究の多くは，音韻的・音響的特徴に注目している．Ching [8] は，中国語の方言である広東語の音韻的・音響的特徴をまとめている．Miller [9] は，米国の南北で話される方言の音韻的特徴を研究し，特徴量による 2 方言の分類を行っている．Lyu [10] は，中国語の 2 方言（普通話: Mandarin，台湾語: Taiwanese）に対応する音声認識システムを開発している．2 方言が混合した発話に対して，2 方言における文字と発音のマッピングを混合し，音声認識を行っている．しかし，これらの音韻的・音響的特徴に着目したシステムには 2 つの問題がある．

(1) 音声コーパスの収集
方言の音韻的・音響的特徴を捉えるには，大量の方言発話が必要となる．すなわち，対象となる方言は，話者が多く，発話の収集が容易なものに限られる．実際，前述の研究はすべて数千万人以上の話者を抱える大規模な方言を対象にしている．

(2) 方言に特有の語彙
音韻的・音響的特徴による方法は，方言の特徴において音素や発音の差異が支配的な場合には効果的である．しかし，日本語のように，音韻的・音響的特徴よりむしろ言語的特徴による差異が大きい場合には適用が難しい．方言を識別して言語モデルを選択する戦略も有り得るが，音響的特徴の差が小さいと方言識別に失敗する可能性が高い．

Zhang [11] は中国語方言の機械翻訳を扱っている．翻訳はピンイン (pinyin)[*1] ベースで行われており，本稿における音素列ベースの変換と方針は類似する．しかし，翻訳辞書を人手で作成しており，多大な時間を要すとともに，他の方言への対応にも同様の作業が必要となるという課題がある．

3. システム構築

ここでは，我々のシステム構築法を，手法の要となる方言言語コーパスのシミュレーションを中心に述べる．はじめに，システム構築に必要な要素を挙げる．続いて，WFST に基づく音素列変換器の構成法について述べる．最後に，例を用いてコーパス処理の流れを説明する．

本稿では，1 章で述べたように，共通語言語コーパスから方言言語コーパスをシミュレートする．以下の 3 点の前提のもとに議論を進める．

(1) 共通語と方言の間で語順の変化は起こらない．
(2) 入力発話の方言は既知とし，その方言と共通語との対訳コーパスは利用可能である．
(3) 共通語文章と方言文章は 1 対多対応している．すなわち，ある共通語文章は複数の方言文章に変換されうるが，ある方言文章に対する共通語文章は 1 つに決まる．

3.1 日本語方言音声認識のキーアイデア

本稿では，大規模方言言語コーパスをシミュレートすることで，統計的に信頼できる方言言語モデルを構築する．ここで，シミュレートするコーパスには，方言発音とともに，元の共通語単語を含めることにする．これには 2 つの理由があり，発音だけでは同音異義語の問題で前後の文脈が利用しづらいため，それに音声認識結果を共通語として出力するためである．方言言語コーパスのシミュレーションに際し，音素列で表現された共通語文章を，単語単位で方言発音の音素列に変換する変換器を構築する．本稿では，これ以降この変換器を音素列変換器と称する．

方言言語モデルの構築は，以下の 3 段階で行う．

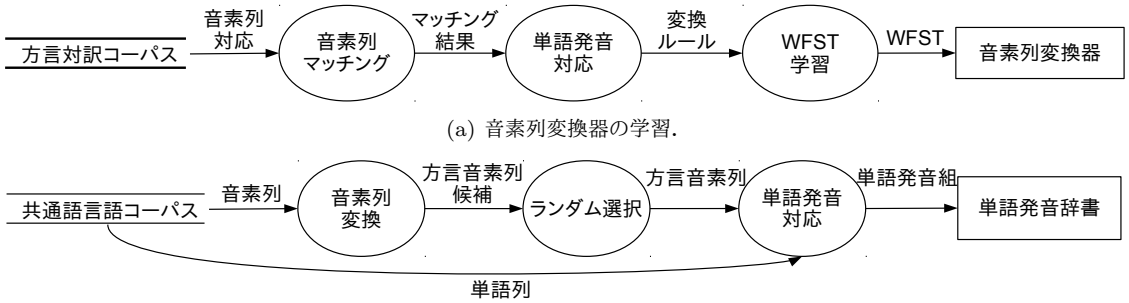
(1) 音素列変換器の学習
(2) 方言言語コーパスのシミュレーション
(3) 言語モデルの学習

図 1 に，各処理におけるデータフローを示す．これ以降，各段階における処理について述べる．

3.2 音素列変換器の構築

音素列変換器の構築には，共通語・方言間の変換ルールが必要である．変換ルールは，方言対訳コーパスを用いて学習する（**図 1(a)**）．方言対訳コーパスでは，共通語と方言の対応する文が音素列で表現されており，かつ共通語については単語境界が明示されているとする．本稿では，

[*1] 中国語において，発音をラテン文字で書き表す方法．



(a) 音素列変換器の学習.

(b) 言語モデルの学習.

図 1 本手法におけるデータフロー.

```
i  1 2 3 4 5 6 7 ...
x  a n a t a w a d o k o n i s u N d e i r u n o
y  a N   t a     d o k o     s u N d e   r u N
z  C S D C C D D C C C C D D C C C C C D C C S D
```

(a) 方言対訳コーパスにおける音素単位マッチング例．x は共通語，y は関西弁を表す．z はマッチング結果であり，C は一致，S は置換，D は削除を表す．

```
a+a n_a+N t+t a+a w_a+NULL d+d o+o k+k o+o n_i+NULL
s+s u+u N+N d+d e+e i+NULL r+r u+u n_o+N
```

(b) 音素対応の表現．共通語と方言で対応する音素列 （NULL は空列）を + で区切ったものを並べている．

図 2 音素変換ルールの基本的なアイデア.

ベースとなる方言対訳コーパスとして，国立国語研究所が編集した『日本のふるさとことば集成』 [12] を用いる．この文献には，各都道府県における地元の人々の談話の方言と，その共通語訳が収められており，共通語は漢字かな交じり表記，方言はカナ表記となっている．共通語は KyTea[*2] [13] を用いて単語境界を明示したカナ表記に変換する．共通語と方言を共に音素列表記に変換して，上記の前提を満たす方言対訳コーパスを作成する．以下の手法では，方言対訳コーパスと共通語言語コーパスの存在を前提とする．

3.2.1 音素列のマッチング

共通語文章と方言文章の各組に対し，最小編集距離 (minimum edit distance) により音素単位で動的計画法に基づくマッチング（DP マッチング）を行う（**図 2(a)**）．$\boldsymbol{x}$ を共通語音素列，$\boldsymbol{y}$ を方言音素列，DP マッチングの結果を $\boldsymbol{z}$ とする．$\boldsymbol{x}, \boldsymbol{y}$ の各要素は，1 個以下の音素である．$\boldsymbol{z}$ の各要素は，対応する $\boldsymbol{x}, \boldsymbol{y}$ の要素の関係であり，C（一致），S（置換），D（削除），I（挿入）のいずれかとなる．この $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ をもとに，**図 2(b)** に示す音素対応列を生成する．

本稿では，音素列変換器の実装として WFST を用いる．音素列変換器は，3 つの WFST T_1, T_2, L を用いて $T = T_1 \circ L \circ T_2$ と表せる[*3]（ここで演算 $\circ$ は WFST の合成を表す．より詳細な定義は [5] を参照されたい）．**図 3** に各 T_1, T_2, L の役割を示す．T_1 は，共通語音素列を入力すると，考えられる方言音素列との対応を図 2(b) の形式で列挙する．言い換えれば，あらゆる音素対応列のうち，+ の前の部分をすべて連結すると入力音素列になるものを列

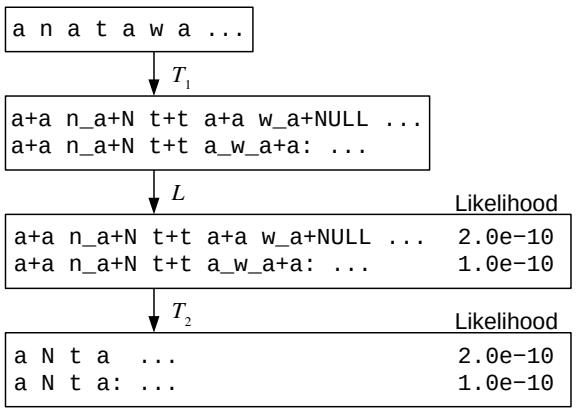
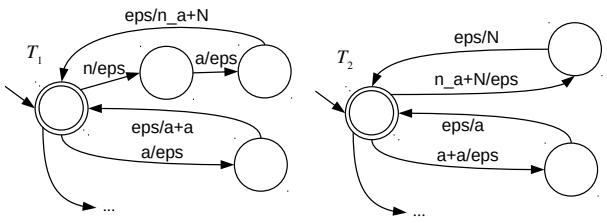


図 3 WFST T_1, T_2, L の役割.



(a) T_1 の構成. (b) T_2 の構成.

図 4 WFST T_1, T_2 の構成. 各遷移に対して入出力記号組を / で区切って示した．eps は記号なしでの遷移を示す.

挙する（**図 4(a)**）．T_2 は，音素対応列を入力すると，各対応の + の後の部分だけを取り出した方言音素列を出力する（**図 4(b)**）．L は，Kneser-Ney スムージングを施して学習した 3-gram モデルを用いて，音素対応列の尤度を計算する WFST [14] であり，音素対応の前後の文脈への依存を表現する重要な要素である．本稿では，WFST の実装

[*2] http://www.phontron.com/kytea/index-ja.html
[*3] 重みを持たない FST も WFST に包含されるので，本稿ではすべて WFST と呼ぶことにする.

a n a t a	w a	d o k o	n i	s u	N	d e	i	r u	n o
a N t a		d o k o		s u	N	d e		r u	N

(a) 図 2(b) の単語単位での対応付け.

a n a t a	w a	d o k o	n i	s u	N	d e	i	r u	n o
a N t a		d o k o		s u	N	d e		r u	N

(b) 生成された方言変換ルール.

図 5 単語対応に基づく方言変換ルールの生成．記号 | は単語境界を示す.

に OpenFst[*4] [5] を用い，L の学習には併せて Kylm[*5] を用いた．この 3 つの合成により，共通語音素列を尤度付き方言音素列に変換する WFST T が生成される.

次に，入力音素列に対応する出力音素列と尤度の扱いについて述べる．入力音素列 $\boldsymbol{x}$ に対し，出力音素列 $\boldsymbol{y}_1, \boldsymbol{y}_2, ...$ および対応する尤度 $L(\boldsymbol{y}_1|\boldsymbol{x}), L(\boldsymbol{y}_2|\boldsymbol{x}), ...$ が計算されたとする（出力音素列は尤度の大きい順，すなわち $i < j$ ならば $L(\boldsymbol{y}_i|\boldsymbol{x}) \geq L(\boldsymbol{y}_j|\boldsymbol{x})$ となるようにする）．尤度の非常に小さい出力音素列もあるため，すべての $L(\boldsymbol{y}_i|\boldsymbol{x})$ を計算するのは非効率である．そこで，尤度の大きい方から n 個の候補 $\boldsymbol{y}_1, ..., \boldsymbol{y}_n$ だけを選び，残りの結果は捨てる．$L(\boldsymbol{y}_i|\boldsymbol{x})$ は，すべての出力音素列のうち実際に $\boldsymbol{y}_i$ が実現する確率 $P(\boldsymbol{y}_i|\boldsymbol{x})$ に比例する．そこでこの確率を

$$P(\boldsymbol{y}_i|\boldsymbol{x}) = \frac{L(\boldsymbol{y}_i|\boldsymbol{x})}{\sum_{j=1}^{n} L(\boldsymbol{y}_j|\boldsymbol{x})} \qquad (1)$$

で求める．そして，$\boldsymbol{y}_1, ..., \boldsymbol{y}_n$ のうち 1 つを $P(\boldsymbol{y}_i|\boldsymbol{x})$ の確率でランダムに選択する．ここで選択した音素列が方言言語コーパスに取り入れられることになる．ランダムに選択することで，尤度の大きい音素列だけが方言言語コーパスに採用されるのを防ぐ．なお，後の実験では $n = 5$ とした.

3.2.2 単語境界情報の導入

発音変化は前後の文脈に依存するが，前後の音素だけでなく，注目する部分が単語全体か単語の一部かにも依存する．そこで，音素列変換器に単語境界の情報が入力できるようにする．入力音素列には，単語境界（記号 |）を含めることができ，単語境界の個数分だけ出力音素列にも単語境界が含まれるように設計する．これは，音素列変換器の入出力の単語境界の対応付けを容易にするためである．図 2(b) に示す音素対応列を，元の共通語文章における単語境界で対応付ける．元の共通語文章の単語境界は

あなた | は | どこ | に | 住 | ん | で | い | る | の

となる[*6]ため，**図 5(a)** の対応が得られる．最後に，図 2(b) と同様の形式で，単語対応をトークンとして音素列変換器の変換ルールを記述する（**図 5(b)**）．但し，実際には図 2(b) に示す音素対応が，単語境界をまたぐ場合がある．この場合は，共通語 m 単語 ($m \geq 2$) にまとめて方言音素列を対応付ける．方言音素列には，単語境界をまたいだことを示す記号を $m - 1$ 個付加しておく.

このとき，方言対訳コーパスに含まれない単語が音素列

a n a t a | n o | ...	a N t a | n o | ...
a n a t a | w a | ...	a N t a: | w a | ...
a n a t a | t o | ...	a N t a | t o | ...
a n a t a | k a r a | ...	a: t a | k a r a | ...
... | w a | a n a t a | ...	... | w a | a N t a | ...

P(a N t a | a n a t a) = 3/5, *P*(a N t a: | a n a t a) = 1/5,
P(a: t a | a n a t a) = 1/5.

図 6 音素列変換結果により，共通語音素列（左列）a n a t a に対する方言音素列（右列）の確率を求める例.

変換器に入力されると方言音素列が存在しなくなるため，その対策を行う．図 5(b) の変換ルールをすべての文に対して生成した後，すべての音素および単語境界 | に対して，各記号に対する恒等変換ルールを 1 回ずつ追加する．これで，必ず方言音素列が存在することが保証される.

3.3 方言言語コーパスのシミュレーション

まず，前節で作成した音素列変換器を用いて，共通語言語コーパスにより方言言語コーパスをシミュレートする．前処理として，共通語言語コーパスに単語境界や読みが付与されていなければ，KyTea を用いて付与する．読みは音素列に変換し，単語境界には記号 |を付加して，各文を音素と | の列として表現する．この列を音素列変換器に入力すると，方言音素列が音素，記号 |，および単語境界をまたぐ記号の列として得られる．複数得られた方言音素列のうち，尤度に従ってランダムに 1 つを選択し（3.2.1 節を参照），選択した方言音素列と元の共通語単語列の対応を方言言語コーパスに追加する.

次に，この方言言語コーパスを用いて言語モデルを学習する．共通語言語コーパスと同様の方法で学習すると，語彙サイズが共通語単語と方言音素列の組の種類になり，n-gram の出現頻度がスパースになると同時に，語彙サイズを制限すると未知語が多くなる．そこで，共通語単語をクラスとするクラス n-gram モデルで言語モデルを学習し，各クラスには付与された方言音素列を含めるようにする．これにより，語彙サイズを増大させずに，共通語単語に対応する複数の方言音素列を認識できる．コーパスの変換が終われば，各方言音素列のクラス内確率を求める．共通語単語 $\boldsymbol{x}$ と方言音素列 $\boldsymbol{y}$ の組の出現回数 $\#(\boldsymbol{x}, \boldsymbol{y})$ を，$\boldsymbol{x}$ の出現回数 $\#(\boldsymbol{x})$（方言音素列は問わない）で除した

$$P_c(\boldsymbol{y}|\boldsymbol{x}) = \frac{\#(\boldsymbol{x}, \boldsymbol{y})}{\#(\boldsymbol{x})} = \frac{\#(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}} \#(\boldsymbol{x}, \boldsymbol{y})} \qquad (2)$$

を，クラス内確率と定める．**図 6** の例では，a n a t a が

[*4] http://www.openfst.org/
[*5] http://www.phontron.com/kylm/
[*6] 変換ルール数削減のため，用言の語幹と活用語尾は分割している.

共通語音素列に 5 回出現し，a N t a に変換されたものが 3 回あるため，a N t a のクラス内確率は 3/5 となる．

3.4 使用する共通語言語コーパス

方言言語コーパスのもとになる共通語言語コーパスとしては，新聞記事 [15] や講演原稿 [16] など，様々な可能性が考えられる．ただ，話し言葉というドメインにおいては，話し言葉と文体の異なる新聞記事や，専門用語の多い講演原稿を用いるのは好ましくない．

本稿では，ヤフー株式会社と国立情報学研究所により提供されている『Yahoo! 知恵袋データ（第 2 版）』を用いる．Web の同名サイトにおける質問および回答文がまとめられたもので，一般ユーザが作成した文であるため，話し言葉調のくだけた表現も多く含まれている．カテゴリ情報が付与されているため，認識したい話題が限定されている場合には，それに近いカテゴリの文章だけを取り出すことも可能である．この Yahoo! 知恵袋データに対し，コーパスのフィルタリング [17] を行い，音声認識に必要のない Web 特有のスラングや，そもそも文になっていない表現（アスキーアートなど）を取り除く．このフィルタリング手法では，想定発話文集合から言語モデルを学習した上で，共通語文章をこの言語モデルにおけるパープレキシティの小さいものから順に選択する．想定発話として，日本語書き言葉均衡コーパス (BCCWJ) [18] コアデータ[*7]のうちブログドメインに属するもの（857 文，12,948 語）を使用した．

4. 評価実験

本稿では，方言の読み上げ発話の音声認識精度により，手法の有効性を確認する．実験に先立ち，BCCWJ ノンコアデータ[*8]から 100 文を抽出し，文体を常体に統一したものを読み上げ原稿とした．この原稿を，共通語話者（東京都，埼玉県出身者）と関西弁話者（大阪府，兵庫県出身者）各 5 名に提示し，共通語話者には原稿をそのまま読むように，関西弁話者には方言に訳して読むように指示した．本稿で述べた方言音声認識では認識結果が共通語文章で出力されるため，正解の文章は共通語・方言のいずれの場合も元の共通語の読み上げ原稿とし，単語認識精度を計算した．

4.1 実験条件

まず，音素列変換器および言語モデルの学習に用いたデータについて述べる．**表 1** に，データの規模をまとめた．音素列変換器の学習には，[12] の大阪府，京都府，兵庫県の 3 府県のデータを用いた．言語モデルの学習には，Yahoo! 知恵袋データ (第 2 版) の「暮らしと生活ガイド」カテゴリに属する質問の一部 335,685 件のうち，23,600 件のみを 3.4 節のようにフィルタリングしたものを用いた．

[*7] 単語境界および読みが人手で付与されたデータ．
[*8] アノテーションされていない生のテキストデータ．

表 1 データ規模．対訳コーパスの単語数は共通語のもの．

	データ	文数	単語数
音素列変換器（対訳コーパス）	合計	619	24,597*
	大阪府	249	8,730*
	京都府	226	6,980*
	兵庫県	144	8,887*
言語モデル	Yahoo! 知恵袋	26,300**	1,164,317*
評価用発話	関西弁 5 名	100	1,682*
	共通語 5 名		

*: KyTea による自動単語分割による推定値． **: 質問数．

表 2 関西弁および共通語単語認識精度 [%].
(a) 関西弁発話の認識．再計算は認識結果をチェックして表記ゆれによる誤りを除いたもの．

言語モデル	関西弁話者 #1	#2	#3	#4	#5	平均
共通語	47.1	43.0	52.7	46.7	45.1	46.9
関西弁	53.6	47.6	57.7	54.6	53.3	53.4
関西弁:再計算	64.2	55.2	67.6	64.5	60.8	**62.5**

(b) 共通語発話の認識．

言語モデル	共通語話者 #1	#2	#3	#4	#5	平均
共通語	80.5	75.9	83.4	79.4	76.0	**79.0**
関西弁	69.3	65.7	73.8	69.3	64.2	68.4

表 3 発音確率の関西弁重みと関西弁音声認識精度 [%] の関係．

関西弁重み		関西弁話者 #1	#2	#3	#4	#5	平均
0	（共通語）	47.1	43.0	52.7	46.7	45.1	46.9
0.25		54.2	46.9	**59.8**	53.6	52.9	53.5
0.5		**55.4**	**48.1**	59.2	54.7	53.2	**54.1**
0.75		54.4	47.2	58.9	**54.8**	**54.0**	53.9
1	（関西弁）	53.6	47.6	57.7	54.6	53.3	53.4

言語モデルの語彙サイズは 10,000 に統一した．

続いて，音声認識エンジンについて述べる．本稿では Julius[*9] [19] を用い，音響モデルとして連続音声認識コンソーシアム (CSRC) 2002 年度版 [15] に含まれる ATR 高精度音響モデル（trigram, 5000 状態，32 混合）を用いた．

4.2 評価

本実験における単語認識精度 Acc は，

$$Acc = \frac{N - S - I - D}{N} \tag{3}$$

の式で計算される．但し，N, S, I, D はそれぞれ正解文章の単語数，置換単語数，挿入単語数，削除単語数を表す．

表 2 に，関西弁発話および共通語発話の音声認識精度を示す．関西弁発話の認識では，話者による翻訳のゆれや表記ゆれによる誤り[*10]を手動でチェックし，意味的な誤りでない箇所を正解扱いした場合の再計算精度も掲げた．関西

[*9] http://julius.sourceforge.jp/
[*10] 「... ている」と「... てる」，「... でしょう」と「... だろう」等．

弁言語モデルにより，単純計算値ベースで平均 6.5 ポイント，再計算値ベースで 15.6 ポイントの向上がみられた．逆に，共通語発話の場合には従来の共通語言語モデルによる認識精度が高くなった．すなわち，本手法で関西弁音声認識に特化した言語モデルを構築していることが示された．

次に，関西弁と共通語の発音確率（クラス内確率）の重み付き平均で単語発音辞書を作成した場合の音声認識精度を調べた．方言 d に対し，式 (2) で与えられる P_c を改めて $P_{c,d}$ と書くと，クラス内確率の重み付き平均 $P_{c,mix}$ は

$$P_{c,mix}(\boldsymbol{y}|\boldsymbol{x}) = \sum_d \alpha_d P_{c,d}(\boldsymbol{y}|\boldsymbol{x}), \qquad (4)$$
$$\text{s.t.} \quad \sum_d \alpha_d = 1,\ \alpha_d \geq 0$$

で計算される．本実験では，関西弁と共通語の重みをそれぞれ α_{K}, $\alpha_{\mathrm{CL}} = 1 - \alpha_{\mathrm{K}}$ とし，α_{K} の値を 0, 0.25, 0.5, 0.75, 1 と変化させた．**表 3** に結果を示す．平均的には $\alpha_{\mathrm{K}} = 0.5$ の場合に認識精度が最大となり，単純に関西弁単語発音辞書を用いた場合 ($\alpha_{\mathrm{K}} = 1$) を 0.7 ポイント上回った．関西弁単語辞書には，関西弁の発音は多く含まれるが，共通語の発音は少なくなる．関西弁に限らず，方言であっても共通語と同様に発音する単語は多いため，両方の発音を持つ単語発音辞書により認識精度が向上したと考えられる．また，話者ごとに認識精度を最大化する重みが異なり，話者の方言の「混合割合」の存在が示唆される．すなわち，同じ地域の話者であっても，共通語や他の方言の影響の程度には個人差があるということである．

5. 今後の課題

本手法による方言音声認識精度に影響を与える要素には，以下の 4 点がある．

(1) 音素列変換器と方言対訳コーパス
(2) 方言特有の単語を含むコーパス
(3) 音響モデル
(4) 話し言葉表現の多様性

1 点目の音素列変換器と方言対訳コーパスは，本手法の根幹をなす要素である．本手法で導入した単語境界情報の他に，品詞の異なりや同音異義語等，方言発音に影響する要素を導入する改良が考えられる．

2 点目の方言特有の単語を含むコーパスは，地元の人々の会話の認識には不可欠である．方言特有の単語とは，方言による発音や語彙の差では捉えられない，地名等の固有名詞を指す．これらを語彙に加える方法として，新聞の地方版記事を言語コーパスに加えることが考えられる．

3 点目の音響モデルは，使用される音素集合や音素の特徴量分布を扱っている．今回の実験では，共通語発話により学習されたモデルを用いたが，方言発話のみからモデルを学習したり，既存モデルの方言発話への適応を行ったりすることで，認識精度向上が図れると考えられる．

4 点目の話し言葉表現の多様性は，表記ゆれ等の意味伝達に影響しない認識誤りの扱いに関係する．話し言葉における，表記ゆれの網羅的な判定基準の獲得が課題である．

謝辞 本研究の一部は，科研費 (S) (No. 24220006)，グローバル COE プログラムの援助を受けた．

参考文献

[1] 翠輝久ほか：質問応答・情報推薦機能を備えた音声による情報案内システム，情報処理学会論文誌， Vol. 48, No. 12, pp. 3602–3611 (2007).
[2] 真田信治（編）：日本語ライブラリー 方言学，朝倉書店 (2011).
[3] Woods, H.: A socio-dialectology survey of the English spoken in Ottawa: A study of sociological and stylistic variation in Canadian English, PhD Thesis, The University of British Columbia (1979).
[4] 小林隆，篠崎晃一（編）：ガイドブック方言研究，ひつじ書房 (2003).
[5] Allauzen, C. et al.: OpenFst: A general and efficient weighted finite-state transducer library, *Proc. of CIAA 2007, Lecture Notes in Computer Science*, Vol. 4783, Springer, pp. 11–23 (2007).
[6] Neubig, G. et al.: A WFST-based Log-linear Framework for Speaking-style Transformation, *Proc. of InterSpeech 2009*, pp. 1495–1498 (2009).
[7] 堀貴明，塚田元：重み付き有限状態トランスデューサによる音声認識, 〈特集〉音声情報処理技術の最先端，情報処理， Vol. 45, No. 10, pp. 1020–1026 (2004).
[8] Ching, P. et al.: From phonology and acoustic properties to automatic recognition of Cantonese, *Proc. of Speech, Image Processing and Neural Networks, 1994*, pp. 127–132 (1994).
[9] Miller, D. and Trischitta, J.: Statistical dialect classification based on mean phonetic features, *Proc. of ICSLP 1996*, Vol. 4, pp. 2025–2027 (1996).
[10] Lyu, D. et al.: Speech recognition on code-switching among the Chinese Dialects, *Proc. of ICASSP 2006*, Vol. 1, pp. 1105–1108 (2006).
[11] Zhang, X.: Dialect MT: a case study between Cantonese and Mandarin, *Proc. of ACL and COLING 1998*, Vol. 2, pp. 1460–1464 (1998).
[12] 国立国語研究所（編）：全国方言談話データベース 日本のふるさとことば集成 (全 20 巻)，国書刊行会 (2001–2008).
[13] Neubig, G. et al.: Pointwise prediction for robust, adaptable Japanese morphological analysis, *Proc. of ACL HLT 2011*, pp. 529–533 (2011).
[14] Chen, S.: Conditional and joint models for grapheme-to-phoneme conversion, *Proc. of EuroSpeech 2003*, pp. 2033–2036 (2003).
[15] 河原達也ほか：連続音声認識コンソーシアム 2002 年度版ソフトウエアの概要，情報処理学会研究報告. SLP, 音声言語情報処理，Vol. 2003, No. 104, pp. 1–6 (2003).
[16] Maekawa, K.: Corpus of Spontaneous Japanese: Its design and evaluation, *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition* (2003).
[17] 翠輝久，河原達也：ドメインとスタイルを考慮した Web テキストの選択による音声対話システム用言語モデルの構築，信学論 (D)， Vol. 90, pp. 3024–3032 (2007).
[18] Maekawa, K.: Balanced corpus of contemporary written Japanese, *Proc. of ALR6 2008*, pp. 101–102 (2008).
[19] 河原達也，李晃伸：連続音声認識ソフトウエア Julius，人工知能学会誌， Vol. 20, No. 1, pp. 41–49 (2005).