# 方言音声認識のための話し言葉言語モデル構築

平山 直樹 † 森 信介 ‡ 奥乃 博 †

† 京都大学 大学院情報学研究科 ‡ 京都大学 学術情報メディアセンター

## 1 はじめに

近年，音声認識システムの認識精度向上のための研究が盛んに行われてきた．現在では，ニュース記事などのテキスト読み上げにおいては，認識精度は 95%を超えている．それに対して，自発的な発話では大きく認識精度が低下する [1, p. 194]．その原因として，話し言葉における個人性の大きさが挙げられる．

本稿では，話し言葉に影響する個人性のうち，特に地域依存性，すなわち方言に焦点を当てて，音声認識精度の向上に取り組む．とりわけ，日本語の方言における語彙の変化に着目する．外国語における多くの関連研究では，発音の変化，すなわち訛りに着目していた．対して日本語方言では，発音よりも語彙の方がより多様で特徴的な要素といえる．アクセントパターンは大きく分けると東京式，京阪式など数種類程度に収まるが，語彙に関しては各地にその土地でしか通じないものが多い．これら特有の語彙を捉えることが，日本語の方言音声認識において重要である．このとき，言語モデルがシステムの性能を大きく左右するため，適切な学習コーパスをどのように構築するかが最重要課題となる．

我々は，日本語方言は語順に影響を与えないと仮定し，言語モデル学習コーパスの各単語に方言発音を付与することで単語の変換を行い，方言音声認識を実現する．方言は話し言葉であって文として記述されることが少ないため，方言の言語コーパスを大量に収集するのは困難である．一方で，言語学の研究を通して得られた方言-共通語間対訳コーパスが，小規模ながら利用可能である．そこで本稿では，大規模な共通語コーパスを，小規模な対訳コーパスを用いて処理することで，大規模な方言コーパスをシミュレートする．

我々は，重み付き有限状態トランスデューサ (Weighted Finite-State Transducer, WFST) による音素列変換器を導入し，文の変換を行う [2, 3]．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．この確率的変換ルールが，方言における単語の多様性を表現することになる．

本稿におけるこれらの戦略の利点は以下である．

1. 大規模な方言言語コーパスがなくても，方言音声認識が実現できる．
2. 方言間の変換ルールを明示的に列挙する必要がなく，多くの方言への対応も同様の方法で行える．
3. 複数の出力候補がある際に，対訳コーパスの文脈を参照して実際の出力を決定できる．

本稿の構成は以下である．2 章で，方言音声認識の関連研究を挙げる．3 章で，システムの構成要素について述べ，我々の方言音声認識の手法を説明する．4 章で，単語認識精度の観点からシステムの評価を行う．6 章で，システム改良の展望を述べる．

## 2 関連研究

これまでの方言音声認識研究には，音響面に焦点を当てるものが多かった．Ching [4] は，広東語の音韻的・音響的特徴をまとめている．Miller [5] 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 方言の分類において 8%の誤り率を実現している．Lyu [6] 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て認識を行っている．しかし，これらのシステムには以下の 2 つの問題がある．

1. 音声コーパスの収集
   方言の音響特徴を捉えるには方言発話を大量に収集する必要がある．話者が多く，大規模なコーパスが利用できる方言にしか適用できない．
2. 方言に特有の語彙
   音響的特徴による方法は，方言の特徴で音素や発音の違いが支配的な場合には効果的である．しかし，語彙そのものの違いは扱えないため，日本語方言のように語彙の違いが重要な場合には効果的でない．

Zhang [7] は中国語方言の機械翻訳を扱っている．通常，方言は音で表され，文として書かれることは少ないので，翻訳はピンイン表現ベースで行っている．これは我々の手法と類似する．しかし，この研究では翻訳辞書を人手で作成している．人手での作成は多大な時間を要し，他の方言に対応する際にも同様の作業が必要となる．

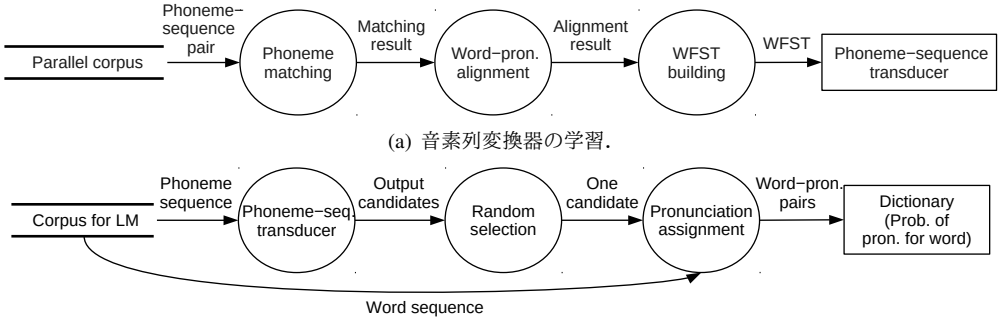
本稿における対処法を以下に述べる．日本語方言は語彙が特徴的であるため，音声コーパスの収集による音響的特徴の解析ではなく，言語モデルの構築により方言音声認識を実現する．これにより，方言に特有な語彙を扱うことができる．また，翻訳辞書は人手で作成することなく，方言-共通語間対訳コーパスを用いて確率的ルールを自動抽出する．この確率的ルールを用いれば，大規模な共通語コーパスで方言コーパスをシミュレートでき，方言言語モデル学習を進めることが可能となる．

## 3 日本語方言音声認識

本章では手法の詳細を述べる．まずシステムの構成要素を述べ，次に WFST による音素列変換器の構成方法を述べる．最後に，話し言葉に適した変換元コーパスについて述べる．

本稿における問題設定として，入力は方言発話，出力は共通語単語列とする．ここで，以下の事項を仮定する．

1. 方言による語順変化は起こらない．言い換えれば，各単語の発音のみが変化するとする．
2. 入力方言は既知とし，対応する方言と共通語間の対訳コーパスも存在するとする．




(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) 対訳コーパスの音素列ペアのアライメント例．1 行目は共通語，2 行目は近畿方言の発音を示す．3 行目はマッチング結果で，C は一致，S は置換誤り，D は削除誤りである．

```
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) 音素列ペアの列による変換ルールの表現．対応する共通語と方言の音素列を + 記号で区切ってある．NULL は空列．

図 2: 音素列変換ルール構築のアイデア．

### 3.1 手法の概略

我々は，大規模な共通語言語コーパスの変換により，大規模な方言言語コーパスをシミュレートし，統計的に信頼できる方言言語モデルを構築する．方言音声認識システムの実現においては，方言言語コーパスの不足が課題となる．これは，文章が方言で書かれることが少ないことによる．変換後のコーパスは，元々の共通語単語を保持しつつ，方言における発音も保持する．これにより，認識結果を共通語文章として提示することが可能となる．

本手法は，大きく以下の 2 段階に分けられる．

1. 音素列変換器の学習
2. コーパスの方言変換と言語モデル学習

図 1 に本手法におけるデータフローを示す．第 1 段階 (図 1(a)) では， 方言-共通語間対訳コーパスからの音素列変換器学習を行う．まず，対応する文同士の音素単位でのマッチングを行う．続いて，その結果を用いて単語単位でのマッチングを行う．最後に，マッチング結果を音素列ペアの列とみなして，音素列変換に用いる *n*-gram モデルを学習する．第 2 段階 (図 1(b)) では，共通語コーパスの各文を音素列として音素列変換器に入力し，各単語に対する方言発音を推定する．全ての文に対する処理が終了した後，各共通語単語エントリに対して付与された方言発音を集計し，単語に対する読みの候補とその確率を計算する．以下，さらに詳しい手順を示す．

1. 音素列変換器
   各文の発音を方言におけるものに変換する．この変換は確率的に行われる．すなわち，複数の変換結果候補が対応する確率とともに出力される．
2. ランダム選択
   最大確率の候補のみが出力されるのを防ぐため，変換結果候補から一つを確率に従ってランダムに選択して変換結果とする．もし最大確率の候補のみを出力してしまうと，それぞれの共通語単語がただ一つの発音でしか認識されなくなってしまう．
3. 単語への発音割り当て
   音素列変換器は文そのものではなく，発音の変換を扱う．ここでは変換された発音を元の共通語単語に割り当てる．これにより，共通語コーパスに方言発音が付与される．
4. 単語辞書
   音声認識システムにおける単語辞書では，各単語エントリに対する発音が定義される．ここで，一つの共通語単語に対し確率付きで複数の発音を定義すると，その単語を複数の発音で認識できるようになる (ここで言語モデルは各共通語単語をクラスとするクラス *n*-gram と捉えられる)．単語エントリは共通語のままであるため，方言発音は共通語単語として認識されるようになる．付与する確率は前段階で付与された発音の頻度により決定する．

我々の手法では，対訳コーパスと言語モデル学習コーパスが必要となる．対訳コーパスは共通語と方言の音素列対応の組からなる．本システムでは，国立国語研究所 [8] の資料を対訳コーパスとして用いる．この資料には，日本の各都道府県での談話が収録されており，方言書き起こしとともに共通語訳が付与されている．言語モデル

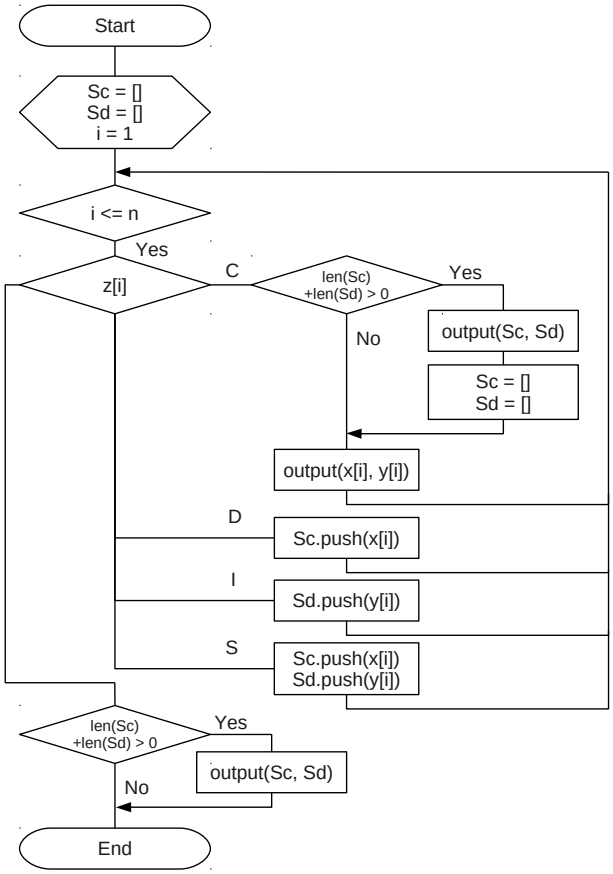



図 3: DP マッチング結果からの音素列ペアの生成.

学習コーパスは方言文の集合である．ここで，3.1 章で方言発音を付与したコーパスを用いる．

### 3.2 音素列変換器の開発

音素列変換器構築のルールは，方言-共通語対訳コーパスから生成する．大まかな流れとしては，*1)* 発音対応のマッチング，*2)* 共通語単語に対する方言発音の推定の 2 段階を行う．

まず，対訳コーパスの対応する発音ペアそれぞれに対し，最小編集距離規準により動的計画法に基づくマッチング (DP マッチング) を行い，音素列の対応を求める (図 2(a))．図 3 は，図 2(a) のマッチング結果から図 2(b) で示される音素列ペアを作成する方法を示す．`x[i]` を共通語音素列，`y[i]` を方言音素列とする．また，`z[i]` を DP マッチングの結果とする．各 `x[i]`, `y[i]` は高々1 音素を含む．各 `z[i]` は次のうち 1 つの値を持つ: `C` (一致)， `S` (置換誤り)，`D` (削除誤り)， `I` (挿入誤り)．

本稿では，WFST を用いて音素列変換器を構築する．音素列変換器は，WFST $T = T_1 \circ L \circ T_2$ で表される (演算 $\circ$ は (W)FST の合成演算である．詳しくは [2] を参照されたい)．これは共通語音素列を入力にとり，方言音素列を尤度とともに出力するものである．図 4 に (W)FST $T_1, T_2, L$ の役割を示す．$T_1$ は共通語音素列を音素列ペアの列に変換する FST である．言い換えれば，音素列ペアの列のうち，各ペアの左辺を結合すると元の入力音素列になるようなものを列挙することに当たる (図 5(a))．$T_2$ は音素列ペアの列を方言音素列に変換する FST である．これは各音素列ペアの左辺を捨てる動作に当たる (図 5(b))．$L$ は音素列ペアの 3-gram モデルを表現する WFST

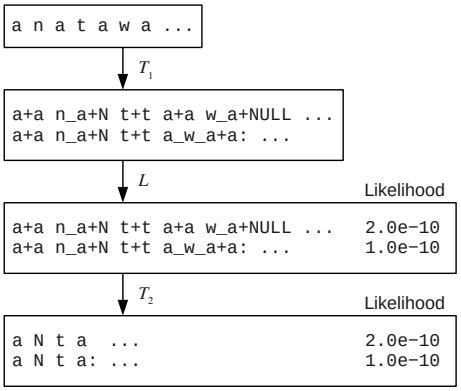
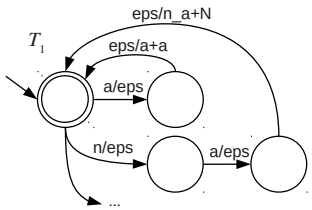
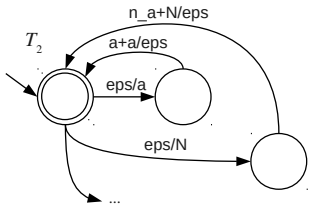



図 4: (W)FST $T_1, T_2$ and $L$ の役割.




(a) $T_1$ の構造.




(b) $T_2$ の構造.

図 5: FST $T_1$ and $T_2$ の構造．各状態遷移枝に対して，入出力記号のペアを記号 / で区切って示す．eps は入力記号または出力記号なしで遷移することを示す.

であり，学習時に Kneser-Ney スムージングを行う．$L$ は各音素列ペアの列に対して尤度を付与する．3-gram モデルにより，文脈に依存する発音の変化をモデル化できる．本稿では，WFST の構築に OpenFst を使用し，$L$ の学習には併せて Kylm[1] を使用した．

共通語音素列 $x$ を WFST $T$ により方言音素列候補 $y_1, y_2, \ldots$ とその尤度 $L(y_1|x), L(y_2|x), \ldots$ に変換することを考える．($i < j$ ならば $L(y_i|x) \geq L(y_j|x)$ となるように順序を付ける). 候補 $y_i$ は場合により組合せ爆発を起こす可能性があるが，十分下位の候補であれば 1 位候補に比べて非常に小さい尤度を持つと考えられる．そのため，すべての $y_i$ に対して尤度を計算することはせずに，$n$-best 候補 $y_1, \ldots, y_n$ のみを残して $y_{n+1}$ 以降は捨てる. $n$-best 候補から一つを実際の出力として選択するが，尤度 $L(y_i|x)$ を和が 1 になるよう正規化したものをそれぞれの候補が選択される確率 $P(y_i|x)$ とする．すなわち以下が成り立つ.

$$P(y_i|x) = \frac{L(y_i|x)}{\sum_{j=1}^{n} L(y_j|x)}. \quad (1)$$

続いて，各単語に対する方言発音を得る．ここで，一つの問題が発生する．発音の変化の仕方は文脈に依存し，

[1] http://www.phontron.com/kylm/

```
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) 対訳コーパスの発音ペアの例．1 行目は共通語発音，2 行目は近畿方言発音である．記号 | は形態素解析器により自動推定された単語境界である．

```
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) 2 つの音素列を，単語境界を無視して音素単位でマッチングし，音素列ペアの列として表現する (図 2(b) に同じ).

```
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
```

(c) 2 つの音素列を単語単位でアライメントする．これは各単語に対する方言発音を決めることになる．

```
a_n_a_t_a_|+a_N_t_a_|  w_a_|+|  d_o_k_o_|+d_o_k_o_|  n_i_|+|
s_u_|+s_u_|  N_|+N_|  d_e_|+d_e_|  i_|+|  r_u_|+r_u_|  n_o_|+N_|
```

(d) 単語単位アライメントに基づく変換ルール．記号 Symbol | で単語境界を示す.

図 6: 単語レベルの変換ルールの構築.

例えば同じ発音をする部分であっても，それ自体が単語なのか，あるいは単語の一部なのかによって異なる．つまり，発音を入力するだけでは共通語単語から考えて不適切な変換結果が得られやすい．そこで，我々は音素列変換器に単語境界を導入する．この改良型音素列変換器は，入力音素列 $x$ に単語境界情報を含むことを許す．また，出力音素列においても対応する箇所に単語境界を挿入する．改良型音素列変換器の学習方法を示す (図 6).

1. 与えられた音素列ペア (図 6(a)) において，単語境界を無視して音素単位でマッチングを行う (図 6(b)).
2. 得られたマッチング結果に基づき，共通語単語に発音を対応付ける (図 6(c)).
3. 共通語単語と発音の対応から，単語ベースの変換ルールを学習する (図 6(d)).

ここで，対訳コーパスに含まれない単語が入力されても動作するように，単語ベースの変換ルールには各音素に対する恒等変換を追加しておく．このとき，対訳コーパスに含まれない単語の入力に対しては同じ発音を出力する．なお，以降で単に音素列変換器といえば，改良型音素列変換器を指すものとする．

ここまでで，コーパスの方言変換の準備が整った．言語モデル学習用の大規模な共通語コーパスの各文に対して，単語分割と発音推定を行い，音素列変換器への入力データ (発音と単語境界) を作成する．変換結果は単語境界を含む方言発音の $n$-best 候補であり，ここから尤度に従って一つの候補が選択される．

コーパスの変換結果に基づく，各単語エントリをクラスとするクラス $n$-gram 言語モデルの作成方法を図 7 に示す．クラス $n$-gram 言語モデルにより，単語エントリ数を増加させずに多くの種類の発音を扱える．また，方言発音のバリエーションを，認識結果の上では共通語単語として吸収できる．すべての文に対して変換が終われば，各単語エントリごとに付与された発音の出現回数を集計する．これを単語エントリ全体における総和で割って正規化したものを，その発音に対するクラス内出現確率とする．コーパス全体における共通語単語 $x$ の出現回数を $\#(x)$，$x$ に対して方言発音 $y$ が割り当てられた回数を $\#(y|x)$ とすると，$y$ のクラス内確率 $P_c(y|x)$ は

$$P_c(y|x) = \frac{\#(y|x)}{\#(x)} = \frac{\#(y|x)}{\sum_y \#(y|x)}. \tag{2}$$

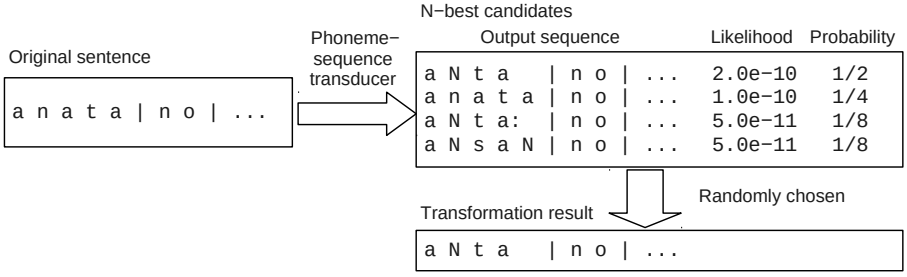
で表される．

## 4 実験

本稿では，音声認識システムの性能を単語認識精度により評価する．方言発話が共通語単語列として認識されるため，あらかじめ共通語で用意した正解データと比較すれば認識精度が得られる．共通語言語モデルは，単に共通語コーパスから学習した単語 $n$-gram モデルとする．一方で，近畿方言言語モデルは，共通語コーパスを音素列変換器で方言コーパスにしたものから学習する．

### 4.1 実験条件

以下に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，各コーパスのサイズをまとめた．この実験では対訳コーパスを国立国語研究所 [8] の 3 府県 (大阪府，兵庫県，京都府) 分のデータとした．このコーパスでは，元々は方言発音と，対応する共通語文 (漢字かな交じり) とが対応付いている．KyTea [9] により共通語コーパスの単語境界と発音を推定して，対訳文を音素列表記に揃え，改良型音素列変換器の学習に必要な単語境界情報を加える．言語モデル学習においては，現代日本語書き言葉均衡コーパス (BCCWJ) [10] のコアデータ (単語境界および発音が人手でチェックされている) を用いる．コーパス変換に当たっては，方言発音の候補を最大 5 個挙げて，式 (1) により候補をランダムに選択する．我々は，実験のために共通語と近畿方言の発話を収集した．共通語発話は，話し言葉調の原稿をそのまま読み上げた音声とした．近畿方言発話は，同じ原稿を近畿地方 (大阪府，兵庫県，奈良県，滋賀県) 出身の話者が各自の方言になおした上で読み上げた音声とし




(a) 音素列変換器を用いたコーパス変換.

| | |
|---|---|
| a n a t a \| n o \| ... | a N t a \| n o \| ... |
| a n a t a \| w a \| ... | a N t 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 \| ... |

In−class probabilities: $P$(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.

(b) クラス内確率の決定方法．左図に出現する共通語単語 'あなた' (a n a t a) が，右図で様々な発音に変換される例を示している．

図 7: コーパス変換方法.

表 1: コーパスの規模．対訳コーパスの単語数は，共通語を基準にカウントしている．

| | データ | 話者数 | 文章数 | 単語数 |
|---|---|---|---|---|
| 対訳 | 合計 | | 619 | 24,597* |
| | 大阪府 | | 249 | 8,730* |
| | 京都府 | | 226 | 6,980* |
| | 兵庫県 | | 144 | 8,887* |
| LM | BCCWJ | | 53,899 | 1,163,426 |
| 評価 | 近畿方言 | 4 | 100 | 1,682* |
| | 共通語 | 3 | | |

*: KyTea の自動単語分割による．

表 2: 近畿方言の単語認識精度 [%]. 番号は話者に対応.

| 言語モデル | #1 | #2 | #3 | #4 | 平均 |
|---|---|---|---|---|---|
| コーパス変換なし | 53.5 | 43.4 | 54.8 | 43.3 | 48.8 |
| コーパス変換あり | 60.1 | 49.4 | 63.9 | 49.4 | **55.7** |

た．本実験では，音声認識エンジンに Julius [11] を利用し，音響モデルは Julius の Web サイト[2]からダウンロードできる PTM モデルとした．

### 4.2 評価

本実験では音声認識を単語認識精度

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

により評価する．但し，$N,S,I,D$ は，それぞれ正解データの合計単語数，置換誤り単語数，挿入誤り単語数，削除誤り単語数を示す．

表 2, 3 に，近畿方言と共通語における単語認識精度の結果を示す．近畿方言の場合には，コーパスを変換してから学習した言語モデルにより，変換せずにそのまま学習した言語モデルより平均で 6.9 ポイントの精度向上を実現した (逆に共通語の場合には，コーパス変換をしない方が精度が高かった)．話者により認識精度にばらつきがあり，認識のしやすさに差が見受けられるが，コーパス変換による精度の向上はどの話者でも起こった．ここから，本手法の有効性が確認できた．

表 3: 共通語の単語認識精度 [%]．番号は話者に対応 (共通語の場合とは異なる).

| 言語モデル | #1 | #2 | #3 | 平均 |
|---|---|---|---|---|
| コーパス変換なし | 72.1 | 64.5 | 72.5 | **69.7** |
| コーパス変換あり | 62.0 | 56.3 | 58.7 | 59.0 |

表 4: コーパス混合時の近畿方言の単語認識精度 [%]. 話者と番号の対応は表 2 と同様.

| 言語モデル | #1 | #2 | #3 | #4 | 平均 |
|---|---|---|---|---|---|
| 0% (変換なし) | 53.5 | 43.4 | 54.8 | 43.3 | 48.8 |
| 25% | 60.6 | 50.1 | 63.6 | 46.7 | 55.2 |
| 50% | 61.4 | 50.6 | 63.4 | 47.6 | 55.8 |
| 75% | 61.2 | 52.2 | 63.7 | 47.6 | **56.2** |
| 100% (完全に変換) | 60.1 | 49.4 | 63.9 | 49.4 | 55.7 |

表 4, 5 には，方言変換したコーパスとしていないコーパスから構築した言語モデルに対して，発音のクラス内確率を線形補間したうえでの単語認識精度を示している．単語認識精度の平均値は，方言発話では方言の比率が 75%，共通語発話では 0% の場合に最も高くなった．同一言語モデルに対して，方言話者 4 名と共通語話者 3 名分の認識精度の平均値は，25%の場合に最も高くなった (60.8%)．方言変換された言語モデルは方言発音を含むが，逆に共通語の発音が含まれにくい．そこでモデルを混合して，方言と共通語双方の発音を持つようにすることで，より高精度の認識が可能となったことがいえる．

音声認識精度に影響する要素を以下に挙げる．

1. 音素列変換器と対訳コーパス
   どの発音を方言表現として認めるかは対訳コーパスの量や使用する素性により決定される．本稿では，音素列の他に単語境界情報を導入した．しかし，依

[2] http://julius.sourceforge.jp/en_index.php?q=index-en.html

表 5: コーパス混合時の共通語の単語認識精度 [%]. 話者と番号の対応は表 2 と同様.

| 言語モデル | #1 | #2 | #3 | 平均 |
|---|---|---|---|---|
| 0% (変換なし) | 72.1 | 64.5 | 72.5 | **69.7** |
| 25% | 70.3 | 66.0 | 68.6 | 68.3 |
| 50% | 68.7 | 64.0 | 67.2 | 66.6 |
| 75% | 66.8 | 62.8 | 66.5 | 65.4 |
| 100% (完全に変換) | 62.0 | 56.3 | 58.7 | 59.0 |

表 6: 表記ゆれチェック後の方言単語認識精度 [%].

| 言語モデル | #1 | #2 | #3 | #4 | 平均 |
|---|---|---|---|---|---|
| 25%変換 | 63.3 | 52.5 | 65.1 | 48.6 | 57.4 |

然として同音異義語の区別という課題が残っている．すなわち，対訳コーパスにおいては同音であればすべて同じ単語として学習が行われるので，ある用法では単語が変化するが，別の用法では変化しない，という状況は表現できない．この課題の解決方法として，前後の単語の品詞を図 6(d) の音素列ペアに追加することが考えられる．

2. コーパス
   実際の発話においては，しばしば地域特有の固有名詞が出現することがある．今回の実験に用いたデータはそのような単語が無いように設定したが，長期的には固有名詞を含むコーパスを収集し，固有名詞を含む発話を正しく認識できるようにする必要がある．コーパス候補として，毎日新聞社等が配布している新聞記事データの地域版が挙げられる．地域版には自治体名や行事名などの地域特有の固有名詞が多く含まれると予想されるためである．

3. 音響特徴
   今回の実験の音響モデルは，方言発話を考慮したものではない．大量の方言発話から音響モデルを学習して用いれば，さらなる認識精度向上が期待される．

4. 話し言葉に起因する表現ゆれ
   特に話し言葉では，同じ意味や役割を持つ単語や表現が存在し，それらを互いに区別する必要のない場合がある．実験では「... しているのだろう」が「... してるんだろう」と,「... すれば」が「... したら」と認識された例があった．音声認識システムは一字一句誤りのない認識を行うより，意味伝達が正しく行われることが重要である．そこで，これらの組をいかにして同一視するかが課題となる．また，日本語においては漢字かな交じり文に起因する表記ゆれをチェックする必要もある．

先ほど全体として最も認識精度が高かった，方言比率が 25%の場合について，表記ゆれを考慮した場合の単語認識精度を表 6 に示す．実質的にはこちらの値がシステムの性能を示しているといってよい．前述した表記ゆれの問題に対し，人手で認識結果のチェックを行い，誤り箇所のうち表記ゆれと認められる部分を正解とみなして認識精度を再計算している．表 2 の 25%変換の場合と比較すると，平均単語認識精度で 2.2 ポイントの差が出た．

## 5 今後の課題

本稿では方言による語順変化はないと仮定して議論を進めた．ここでは，語順変化を認める場合を考える．例えば,「7 月 1 日」を表す “July first” と “the first of July” のようなケースは，語順変化によらない現状の音素列変換器では適切にモデル化できない．一つの解決策としては，IBM モデル 3 [12] のように，変換後単語の湧き出し確率 $p_0$ を導入し，共通語と直接対応しない方言発音が生じることを許す方法が考えられる．このとき，以下のような対応付けができることになる．但し NULL は対応する単語がないことを示す.

```
July  |  first  | NULL | NULL |
the   |  first  | of   | July |
```

WFST $L$ (3.2 章を参照) が語順変更が行われるフレーズの長さ (上の例であれば 4) 以上の $n$-gram モデルで学習されている場合には，文脈によりその範囲内での語順変更を捉えられると考えられる．幸いにしてこの変更は WFST と相性がよく，比較的単純な変更で実現できそうである．新たに空の単語を挿入する WFST $E$ を導入し，音素列変換器として WFST $T' = E \circ T = E \circ T_1 \circ L \circ T_2$ を用いるようにする．語順が大きく異なる言語・方言間ではこの方法による対応は難しいが，それでも当初の語順変化なしという仮定を緩めることはできる．

方言によるモデル切り替えも検討すべき課題である．うまく切り替えが行えれば，事前に入力方言が分からなくても方言発話が認識できるようになる．駅や空港など，人々が様々な地域から訪れる可能性のある場所でシステムを稼働させるときには，できる限り多くの方言に対応できることが望ましい．前述の実験では，発音に対するクラス内確率を線形補間した．すなわち方言 $d$ に対応する，式 (2) の $P_c$ を $P_{c,d}$ と置きなおせば，線形補間により得られる新たなクラス内確率 $P_{c,mix}$ は

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

と書ける．このモデルでは，それぞれの単語の方言は互いに独立となっている．しかし，多くの単語を自身の方言で話すなど，実際には単語間の方言の依存性はあると考えられる．依存性をどのようにモデル化するかは，今後解くべき課題であるといえよう．

## 6 結論

本稿では，日本語方言発話をモデル化した言語モデルにより，方言発話を認識する音声認識システムの開発について述べた．手法の鍵となるのは方言コーパスの構築である．我々は対訳コーパスから音素列変換器を構築し，共通語コーパスの各単語に方言発音を付与した．実験により，方言変換コーパスから学習した言語モデルによる方言発話の認識精度が高くなり，手法の有効性が示された．この方法は，対訳コーパスさえあれば，他の言語に対しても適用できると考えられる．

今後の課題としては，多くの方言の認識への対応が挙げられる．本稿では語順変化しないとしたが，音素列変換器に変更を加えると，文脈として語順変化を扱うことができそうである．さらに精度向上を目指すには，方言切り替えモデルの構築および方言発話に対する音響モデルの学習も行うことが考えられる．

## 謝辞

本研究の一部は，科研費 (S)，GCOE の援助を受けた．

## 参考文献

[1] M.A. Anusuya and S.K. Katti. Speech recognition by machine: A review. *International Journal of Computer Science and Information Security*, Vol. 6, No. 3, pp. 181–205, 2009.

[2] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. OpenFst: A general and efficient weighted finite-state transducer library. In *Proc. of CIAA 2007, Lecture Notes in Computer Science*, Vol. 4783, pp. 11–23. Springer, 2007.

[3] G. Neubig, S. Mori, and T. Kawahara. A WFST-based log-linear framework for speaking-style transformation. In *Proc. of InterSpeech 2009*, pp. 1495–1498, 2009.

[4] P.C. Ching, T. Lee, and E. Zee. From phonology and acoustic properties to automatic recognition of Cantonese. In *Proc. of Speech, Image Processing and Neural Networks, 1994*, pp. 127–132, 1994.

[5] D.R. Miller and J. Trischitta. Statistical dialect classification based on mean phonetic features. In *Proc. of ICSLP 1996*, Vol. 4, pp. 2025–2027, 1996.

[6] D. Lyu, R. Lyu, Y. Chiang, and C. Hsu. Speech recognition on code-switching among the Chinese dialects. In *Proc. of ICASSP 2006*, Vol. 1, pp. 1105–1108, 2006.

[7] X. Zhang. Dialect MT: a case study between Cantonese and Mandarin. In *Proc. of ACL and COLING 1998*, Vol. 2, pp. 1460–1464, 1998.

[8] 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.

[9] G. Neubig and S. Mori. Word-based partial annotation for efficient corpus construction. In *Proc. of LREC 2010*, pp. 2723–2727, 2010.

[10] K. Maekawa. Balanced corpus of contemporary written Japanese. In *Proc. of ALR6 2008*, pp. 101–102, 2008.

[11] 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.

[12] P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.