Statistical Method of Building Dialect Language Models for ASR Systems

Naoki HIRAYAMA[1] *Shinsuke MORI*[1,2] *Hiroshi G. OKUNO*[1]
(1) Graduate School of Informatics, Kyoto University
(2) Academic Center for Computing and Media Studies, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, Japan
hirayama@kuis.kyoto-u.ac.jp, forest@i.kyoto-u.ac.jp, okuno@i.kyoto-u.ac.jp

ABSTRACT
This paper develops a new statistical method of building language models (LMs) of Japanese dialects for automatic speech recognition (ASR). One possible application is to recognize a variety of utterances in our daily lives. The most crucial problem in training language models for dialects is the shortage of linguistic corpora in dialects. Our solution is to transform linguistic corpora into dialects at a level of pronunciations of words. We develop phoneme-sequence transducers based on weighted finite-state transducers (WFSTs). Each word in common language (CL) corpora is automatically labelled as dialect word pronunciations. For example, *anta* (Kansai dialect) is labelled *anata* (the most common representation of 'you' in Japanese). Phoneme-sequence transducers are trained from parallel corpora of a dialect and CL. We evaluate the word recognition accuracy of our ASR system. Our method outperforms the ASR system with LMs trained from untransformed corpora in written language by 9.9 points.

KEYWORDS: spoken language, dialect, language model, weighted finite-state transducer (WFST).

Proceedings of COLING 2012: Technical Papers, pages 1179–1194,
COLING 2012, Mumbai, December 2012.

1 Introduction

Automatic speech recognition (ASR) systems for spoken language are yet far from practical use. ASR systems for written sentences have been widely studied, and recognition accuracy has rapidly improved. In contrast, recognition accuracy is drastically lower for spontaneous speech (Anusuya and Katti, 2009, p. 194). People in their daily lives do not actually speak in a stable way like written sentences. Their speeches include casual expressions, fillers, and even vocabulary specific to dialects.

This paper especially handles improving Japanese dialect ASR. Most previous application systems with speech interface have assumed well-formed sentences in the common language (CL), although they have assumed non-expert speakers. Non-expert speakers will obviously utter informal expressions other than those in written language, and even words specific to their own dialect; dialect ASR systems have difficulty in recognition accuracy or scalability.

Dialects in the world have various kinds of differences (Benincà, 1989). The major differences between dialects and the CL are categorized into the following types: 1) pronunciation, 2) vocabulary, and 3) word order. The first type belongs to the difference in acoustic features, while the second and third belong to that in linguistic ones. Canadian English contains all of three types; 1) /tu/ is pronounced as /tju/, 2) 'high school' is called 'collegiate institute', and 3) 'next Tuesday' is changed into 'Tuesday next' (Woods, 1979). Many North American varieties of French have a tendency to take SVO (subject-verb-object) word order (Gadet and Jones, 2008). In Japanese dialects, the difference of vocabulary is characteristic, e.g., *tabe n* is used instead of *tabe nai* (do not eat) (Gottlieb, 2005).

Our method in this paper focuses on differences in pronunciation of vocabulary between dialects, which correspond to the first and second types. Vocabulary is a set of word entries used in a language or a dialect. We process pronunciation as the corresponding phoneme sequence to reduce the problem to text processing.

The main difficulty with dialect ASR lies in the shortage of linguistic corpora on dialects because they are spoken rather than written. This prevents us from building statistically reliable language models (LMs) including characteristic vocabulary for dialect ASR.

In this paper, we overcome the shortage of dialect corpora by training a vocabulary transformation system that gives labels of dialect expressions to each word in large CL linguistic corpora. (The LMs for ASR is trained based on the output of the above vocabulary transformation system.) The vocabulary transformation system is implemented as a weighted finite-state transducer (WFST) (Allauzen et al., 2007; Neubig et al., 2009). WFSTs model probabilistic transformation rules extracted from dialect-CL parallel corpora.

The three main advantages of our strategy are as follows. First, our system improves the recognition of dialect utterances even with a limited amount of dialect corpora. Second, our method dispenses with the manual enumeration of dialect transformation rules. Therefore, it enables us to build ASR systems for various dialects in the principled manner. Third, statistical corpus transformation gives a solution to how to choose one of multiple candidates for output by taking the contexts of parallel corpora into account.

This paper is organized as follows. Section 2 reviews related work on dialect ASR. Section 3 states major elements of our system, and describes our method of recognition of dialect utterances. Section 4 discusses our evaluation of the system in terms of word recognition accuracy, and finally conclusions summarize this paper and describe future work.

2 Related work

Most studies have focused on acoustic aspects in developing ASR systems for dialects, Ching et al. (1994) described the phonological and acoustic properties of Cantonese, one of the major dialects of Chinese language, based on energy profiles, pitch, and duration. Miller and Trischitta (1996) studied the phonetic features of Northern and Southern US dialects in linearly classifying each dialect. Their experiment achieved error rates of 8% in distinguishing Northern US dialect from those in the the South. Lyu et al. (2006) developed an ASR system for two Chinese dialects, Mandarin and Taiwanese. Dialect-mixed utterances could be recognized with combined character-to-pronunciation mapping in their system.

These systems had two main problems:

1. difficulty of collecting acoustic corpora of dialects
2. incapability of incorporating vocabulary difference

The first problem means that many dialect speakers are necessary for reliable analyses. These systems would work well for major dialects whose corpora were abundant, whereas it was not realistic to collect large corpora even for relatively minor dialects. The second problem prevented these systems from being in general use. Phonological methods are effective for the situation that variation of dialects mainly stems from differences of their phonemes, while these systems do not cover difference of vocabulary. If target dialects have large difference of vocabulary, these systems are less effective. The strategy of classifying dialects and selecting LMs is effective only if the vocabulary of target dialects are almost the same, but actually, dialect vocabulary is rather likely to differ between dialects (Wolfram, 2009, p. 144). The strategy of classifying dialects and next selecting LMs is possible, of course, but effective case is limited; classification would not work well if vocabulary dominates difference of target dialects and these dialects have similar phonological characteristics.

Instead of studying acoustic aspects of dialects containing the problems above, some studies focused linguistic aspects. Zhang (1998) described dialect machine translation (MT) between dialects in the Chinese language. Since dialect sentences were only represented in sound and had not been written down, his translations were between *pinyin* representations of two dialects, which is similar to those in our study. Munteanu et al. (2009), related to the correction of ASR results, tried to correct ASR results in the lecture domain by using a transformation model trained from correct sentences and the corresponding outputs. The scoring for each rule was based on how much the word error rate (WER) could be reduced by applying the rules. These studies still had problems. Zhang (1998) created translation dictionaries manually, and dealing with various dialects required the same process for each dialect. Munteanu et al. (2009) assumed that ASR results were correct; if much vocabulary specific to a target domain were not covered, e.g., for dialects, these methods would not work well. These problems indicates that the key to successful ASR systems is automatic building of LMs in dialects.

This paper deals with dialect ASR as follows. We develop a dialect ASR system by building LMs instead of analyzing acoustic features, because vocabulary is more characteristic in Japanese dialects, as mentioned in Section 1. This enables an ASR system to recognize vocabulary specific to each dialect. Translation dictionaries, transformation rules in other words, are automatically extracted from dialect-CL parallel corpora. The extracted rules are probabilistic based on the statistical analysis of parallel corpora; using large CL corpora, we can simulate dialect linguistic corpora including variations in word choices. This strategy is applied to transformations between spoken and written language (Akita and Kawahara, 2010; Neubig et al., 2012). Since our transformation targeted dialects, it is more advanced than that for mere spoken language. Our transformation model is simpler than those in these studies, due to our assumption that the word order does not change.

3 ASR for Japanese Dialects

This section describes our method in detail. First, we enumerate elements of our system. Next, we explain how to develop the vocabulary transformation system based on WFST. Finally, we introduce examples of corpora to transform.

The inputs are utterances in dialects and the outputs are recognized word sequences in the CL. We make the following three assumptions behind the problem setting. First, dialects would have no effect on the word order; in other words, it would be only necessary to merely transform pronunciation. Second, dialects of input utterances are known and parallel corpora corresponding to the dialects are available. Third, one-to-many sentence correspondence for CL and dialect sentences, i.e., one CL sentence may be transformed into various dialect sentences by dialect speakers, while these dialect sentences have only one corresponding CL sentence. This problem setting has advantages that 1) we prefer that ASR systems output a CL sentence as its meaning instead of simple dialect transcription given a dialect utterance, and because 2) CL sentences are easy to handle as a canonical representation for applications such as speech dialogue systems.

3.1 Main idea underlying ASR for Japanese dialects

We simulate large dialect corpora to build a statistically reliable LM by transforming large CL corpora. The main problem in building a dialect ASR system is the shortage of large linguistic corpora in dialects due to rare transcriptions of sentences. The transformation produces large dialect corpora even if few actual dialect corpora are available. Each word in the new corpora contains the corresponding pronunciation in the dialects and the original word itself so that the dialect utterances can be recognized as CL sentences. This eliminates the cost of having to transform the ASR results again; we only need CL-to-dialect transformation for linguistic corpora and do not need reverse transformation. We can assume that only phoneme sequences are different because 1) as mentioned above, we handle dialects in structure of text processing, and because 2) Additionally, as we describe at the end of Section 2, we assume that the word order does not change. The vocabulary transformation system focuses on transformation at a level of phoneme sequences; it is called 'phoneme-sequence transducer' afterward.

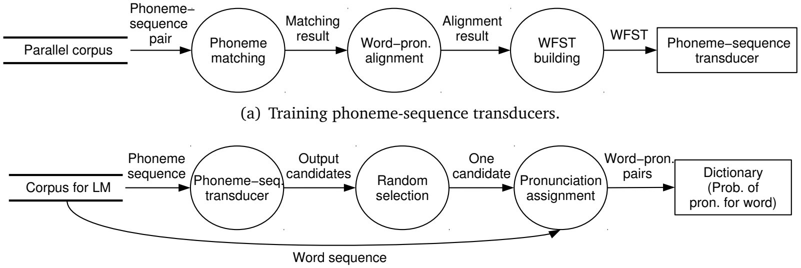
Our solution is composed of two steps:

1. training phoneme-sequence transducers
2. training LMs from corpora transformed with phoneme-sequence transducers

Figure 1 outlines data flow for our method.

Phoneme-sequence transducers are trained from CL-dialect parallel corpora in the first stage (Figure 1(a)). Units of *phonemes* are matched to corresponding sentences in parallel corpora. Next, pronunciations are aligned in units of *words*. Finally, *n*-gram models for phoneme-sequence transducers are trained from the results of alignment as sequences of phoneme-sequence pairs.

The second stage (Figure 1(b)) takes inputs from pronunciations of sentences in the CL corpora to phoneme-sequence transducers to obtain the corresponding pronunciations in the dialects.



(a) Training phoneme-sequence transducers.

(b) Training language models.

Figure 1: Data flow of our method.

After all sentences have been processed, the pronunciations that have been obtained are counted for each CL word entry, and the probabilities for each pronunciation are calculated.

The process involves four main steps:

1. phoneme-sequence transducer
2. random selection
3. pronunciation assignment
4. pronunciation dictionary

In the first step, the pronunciation of all sentences is transformed into dialects. This transformation is probabilistic; phoneme-sequence transducers output multiple candidates together with their probability. These probabilities are determined by the frequencies of transformation patterns in parallel corpora. If a transformation pattern frequently appear in parallel corpora, the phoneme-sequence transducers assign high probability to the pattern.

In the second step, to avoid only candidates with the maximum probability from being output, output is decided randomly from one of the candidates based on their probability. This is a kind of simulation of randomness of word choice. If only candidates with the maximum probability are output, only one pronunciation can be recognized for each CL word.

In the third step, phoneme-sequence transducers process pronunciations, not sentences themselves. This process deals with matching the output phoneme sequence to the original CL sentences (word sequences). After this process, each word in the original CL corpora will have its dialect pronunciation.

In the fourth step, pronunciation dictionaries in an ASR system contain each word entry and corresponding pronunciation as a phoneme sequence. Pronunciation dictionaries can contain multiple pronunciations together with their probabilities (LMs are treated as class n-gram models, in which each CL word entry corresponds to a class). Pronunciations and the corresponding probability is decided by the frequency of each pronunciation in the output of the previous process. These pronunciations make it possible to recognize dialect pronunciation as a word in the CL.

Our solution requires parallel corpora and linguistic corpora to train LMs. Parallel corpora are composed of pairs of phoneme sequences in a dialect and the CL. Our system uses the parallel

```
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) Example of alignment of phoneme-sequence pairs in parallel corpora. The first line is in the CL and the second line is in a dialect in the Kansai area (including Osaka). The third line is the matching result; `C` is a correct phoneme, `S` is a substitution error, and `D` is a deletion error.

```
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) Representation of transformation rules by using sequences of phoneme-sequence pairs. Symbol `+` separates two phoneme sequences in the CL and a dialect. `NULL` represents empty sequences.

Figure 2: Main idea in building rules for phoneme-sequence transducers.

corpus (National Institute for Japanese Language and Linguistics, 2008) composed of dialect sentences and the corresponding CL translations. They contain spoken sentences in various areas (prefectures) of Japan. The corpora for training LMs are sets of dialect sentences. They are created as explained in Section 3.1 using phoneme-sequence transducers.

3.2 Developing phoneme-sequence transducers

The rules for developing phoneme-sequence transducers are created from the parallel corpora previously mentioned. Briefly, the rules are created in two steps:

1. match of each pair of pronunciations,
2. obtain pronunciations in a dialect for each word in the CL.

First, each pair of pronunciations in parallel corpora is processed by matching based on the method of dynamic programming (DP-matching) using the minimum Levenshtein distance to create phoneme-pair sequences (Figure 2(a)), which describe what part of each pair of sequences corresponds to each other. Figure 3 outlines how to create sequences of pairs of the phoneme-sequences described in Figure 2(b) from the two phoneme sequences in Figure 2(a). Let `x[i]` be a phoneme sequence in the CL, and `y[i]` be that in a dialect. We have assumed that they have already been obtained by DP-matching together with the matching result, `z[i]`. Each element of `x` and `y` is a phoneme or empty. Each element of `z` is one of the following: `C` (correct phoneme), `S` (substitution error), `D` (deletion error) or `I` (insertion error).

We adopt WFSTs to build phoneme-sequence transducers. Phoneme-sequence transducers are represented as WFST $T = T_1 \circ L \circ T_2$ (The operation $\circ$ denotes the composition of (W)FSTs. See Allauzen et al. (2007) for more details), T takes phoneme sequences in the CL as input and the corresponding phoneme sequences in dialects together with their likelihoods as output. Figure 4 lists the roles of each (W)FST T_1, T_2, and L. T_1 is the FST for transforming a phoneme sequence in the CL into a sequence of phoneme-sequence pairs, in other words, enumerating sequences of phoneme-sequence pairs whose concatenation at the left is equal to the original phoneme sequence (see Figure 5(a)). T_2 is the FST for transforming a sequence of phoneme-sequence pairs into a phoneme sequence in a dialect, in other words, cutting down the left of each

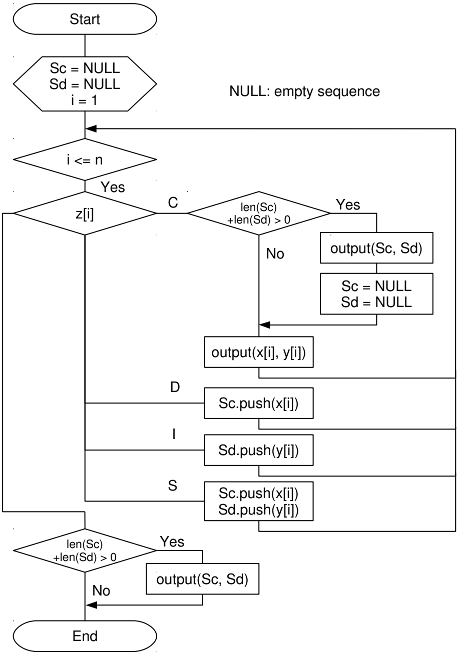


Figure 3: Creating sequences of phoneme-sequence pairs based on DP-matching results. Here ‘Sc.push’ append the symbol specified in the parameter to the end of sequence ‘Sc’.

phoneme-sequence pair (see Figure 5(b)). L is the WFST of a 3-gram model of phoneme-sequence pairs (Chen, 2003) with the method of Kneser-Ney smoothing. L gives a likelihood value to each candidate and it can model phoneme transformation depending on the context. In this paper, OpenFst (Allauzen et al., 2007) is used for creating these (W)FSTs, and additionally Kylm is used for creating L.

We input phoneme sequence $\boldsymbol{x}$ in the CL to WFST T to obtain phoneme sequences in dialect $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ together with their likelihoods $L(\boldsymbol{y}_1|\boldsymbol{x}), L(\boldsymbol{y}_2|\boldsymbol{x}), \ldots$ (If $i < j$, $L(\boldsymbol{y}_i|\boldsymbol{x}) \geq L(\boldsymbol{y}_j|\boldsymbol{x})$). It is not efficient to calculate $L(\boldsymbol{y}_i|\boldsymbol{x})$ for all possible $\boldsymbol{y}_i$ since some of $\boldsymbol{y}_i$ have very small likelihoods and the number of candidates is sometimes very large. We only consider the n-best results $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ for the possible candidates, and cut off candidates from $\boldsymbol{y}_{n+1}$. Likelihoods $L(\boldsymbol{y}_i|\boldsymbol{x})$ for the possible candidates determine the probability of choosing $\boldsymbol{y}_i$; these probabilities $P(\boldsymbol{y}_i|\boldsymbol{x})$ are regularized likelihoods whose sum is equal to one:

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

Next, we obtain pronunciation in a dialect for each word. One problem occurs here. The way pronunciation is transformed depends on its context, e.g., whether a given phoneme sequence

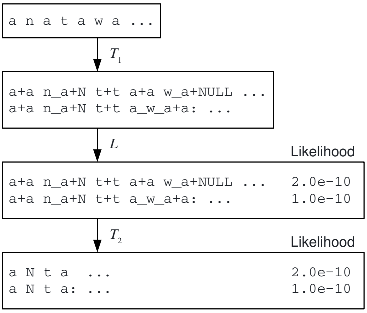
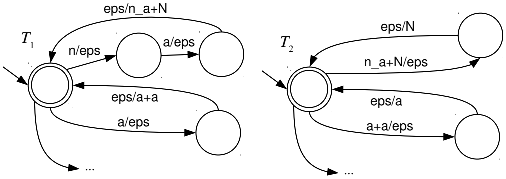

a n a t a w a ...
T_1
a+a n_a+N t+t a+a w_a+NULL ...
a+a n_a+N t+t a_w_a+a: ...
L
Likelihood
a+a n_a+N t+t a+a w_a+NULL ... 2.0e-10
a+a n_a+N t+t a_w_a+a: ... 1.0e-10
T_2
Likelihood
a N t a ... 2.0e-10
a N t a: ... 1.0e-10

Figure 4: Roles of each (W)FST T_1, T_2, and L.



(a) Structure of T_1.

(b) Structure of T_2.

Figure 5: Structure of FST T_1 and T_2. Each transition has a pair of input and output symbols delimited by symbol '/'. Symbol 'eps' represents a transition with no input or output symbols.

is itself a word or part of a word; viz., only given a pronunciation in the CL, some of the outputs of the phoneme-sequence transducers may be not suitable as pronunciation of the original word. We introduce word boundaries to the phoneme-sequence transducers. The modified phoneme-sequence transducers take phoneme sequence $\boldsymbol{x}$ containing some word boundaries in the CL and output at most n candidates of phoneme sequences containing word boundaries in a dialect. The modified phoneme-sequence is trained in three steps (see Figure 6).

1. extract what parts of given sequences correspond from phoneme-sequence pairs of pronunciations (Figure 6(a)) *without* word boundary information (Figure 6(b)).
2. align phoneme sequences to each word based on the extracted information (Figure 6(c)).
3. train word-based transformation rules from corresponding sequences *including* word boundary information (Figure 6(d)).

In the second step, we regard phoneme-sequence pairs crossing word boundaries as alignment of multiple ($m \geq 2$) CL words to a single dialect word, and insert $m - 1$ symbol(s) representing boundary crossing before the next word boundary. The word-based transformation rules include identity transformations of a single phoneme such as a in the CL to a in dialects, so that the transducers can accept sequences containing a word that does not appear in parallel corpora. The transducers only output the same phoneme sequences for such words in input sequences.

Now, we are ready to transform the corpora. We segment words and estimate pronunciations for each sentence in large linguistic corpora in the CL to create input data for the modified

```
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) Example pairs in parallel corpora. The first line is in the CL and the second line is in a dialect of the Kansai area. Symbol | represents word boundaries automatically decided by a morphological analysis tool.

```
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) Align two sentences at the phoneme level without word boundaries and express them with pairs of phoneme sequences. (Same as figure 2(b))

```
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) Align two sentences at the word level. This represents how each word is pronounced in a dialect.

```
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) Transformation rules of phoneme sequences based on word-level alignment. Symbol | represents word boundaries.

Figure 6: Way in which word-level transformation rules were developed.

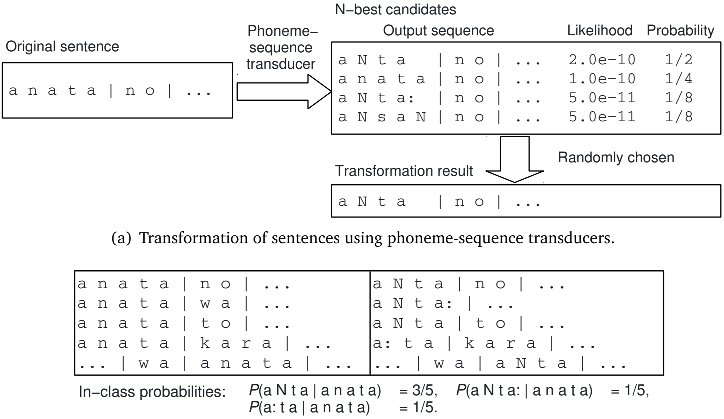
phoneme-sequence transducers. The modified phoneme-sequence transducers output phoneme sequences including word boundary information in a dialect. The transformed sequences are randomly chosen from the corresponding *n*-best results.

Figure 7 lists the process of building class *n*-gram LMs, in which each word entry is a class, from the transformed sentences. Class *n*-gram LMs allow many kinds of pronunciations to be manipulated without increasing the number of word entries of LMs. After all sentences have been transformed, the frequencies of pronunciations for each word entry are counted. We define the frequencies divided by the frequency of the word as the in-class probability of the pronunciation. Let $\#(\boldsymbol{x})$ be the number of CL word $\boldsymbol{x}$ that appears in the original sentences and $\#(\boldsymbol{y}|\boldsymbol{x})$ be the number of pronunciations $\boldsymbol{y}$ given to word $\boldsymbol{x}$; then the in-class probability, $P_c(\boldsymbol{y}|\boldsymbol{x})$, is written as

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

3.3 Linguistic corpora to transform

The transformation method previously mentioned, of course, requires large linguistic corpora to transform. A former study (Lee et al., 2002) adopted 75 months of newspaper articles for a corpus, which is typical in studies on language models. Newspaper articles are relatively formal and in consistent style; therefore, they are suitable for recognizing speech in written articles, while not for spoken sentences including expressions that are characteristic of spoken language. One candidate for corpora in spoken language is the academic presentation speech corpora included in the *Corpus of Spontaneous Japanese* (CSJ) (Maekawa, 2003). This corpus consists of



Output sequence	Likelihood	Probability
a N t a | n o | ...	2.0e−10	1/2
a n a t a | n o | ...	1.0e−10	1/4
a N t a: | n o | ...	5.0e−11	1/8
a N s a N | n o | ...	5.0e−11	1/8

(a) Transformation of sentences using phoneme-sequence transducers.

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) Way in which in-class probabilities are determined. Phoneme-sequence transducers transform sentences including word a n a t a in the CL (at left) into sentences in a dialect (at right).

Figure 7: Way in which a corpus is transformed.

sentences including many technical terms, which are also not very suitable for spoken language.

This paper adopts corpora available on the Web, especially the Yahoo! *Chiebukuro* (Q&A) corpus. This corpus is presented by Yahoo! Japan Corporation and National Institute of Informatics (NII). Since corpora on the Web are created by various users, they contain various and some informal expressions like spoken language. The Yahoo! Q&A corpus contains sentences together with categories and subcategories to which each sentence belongs. It is possible to build LMs from sentences belonging to some specific categories near the target topics.

We adopt a corpus filtering method (Misu and Kawahara, 2006) in the Yahoo! Q&A corpus to build LMs. The major disadvantage of Web corpora is that some sentences are too inconsistent or not even in the form of sentences, e.g., Internet slang and ASCII arts. Speech recognition does not require these sentences and they are need to be excluded from corpora for training LMs. Corpus filtering is based on perplexity; we choose sentences with small perplexity on an LM from a set of sentence examples. These sentence examples were blog articles in the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa, 2008) core data. Words were segmented and pronunciations were estimated in BCCWJ core data these were manually checked by humans. Blog articles in the BCCWJ core data contained sentences that were close to those in spoken language including informal words and expressions.

4 Experiment

Our experiment evaluated the recognition accuracy of ASR. Dialect utterances were recognized as CL sentences and compared to referential CL sentences. We collected utterances in the CL and Kansai dialects. People from the Kansai area (Osaka, Hyogo, Nara and Shiga Prefectures in this experiment) read these sentences in their own dialects. The LMs for the CL were simply trained from the Yahoo! Q&A corpus. The LMs for the Kansai dialect were trained from the

	Data set	# of persons	# of sentences	# of words
Parallel corpora	Total		619	24,597*
	CL-Osaka		249	8,730*
	CL-Kyoto		226	6,980*
	CL-Hyogo		144	8,887*
Training LMs	BCCWJ Core		53,899	1,163,426
	Yahoo! Q&A		26,300**	1,164,317*
Evaluation	Kansai	4	100	1,682*
	CL	3		

*: Estimated by automatic word segmentation with KyTea.

**: Number of questions, because of difficulty of count sentences due to informal expressions.

Table 1: Size of corpora. The number of words in parallel corpora have been counted with reference to CL sentences.

pronunciation-transformed corpus based on the Yahoo! Q&A corpus mentioned in Section 3.2.

4.1 Conditions

Here we describe the training data for the phoneme-sequence transducers and LMs. Table 1 summarizes the size of corpora. Each LM had a common vocabulary size of 10,000.

This experiment adopted the parallel corpus (National Institute for Japanese Language and Linguistics, 2008) of the Kansai area (Osaka, Kyoto and Hyogo Prefectures). Each dialect sentence in this corpus was represented as pronunciation while each CL sentence was in plain text. We segmented CL sentences into words and estimated pronunciations of the words with KyTea (Neubig and Mori, 2010) so that the two kinds of sentences would have the same format in pronunciation.

We transformed the pronunciation of the Yahoo! Q&A corpus into that of the Kansai dialect to train the LMs. We chose 23,600 out of 335,685 questions in the category of daily life with the filtering method mentioned in Section 3.3, which has approximately the same number of words as the BCCWJ core data. One of at most the five-best dialect pronunciations was randomly chosen in the transformation, with the probability of their normalized likelihoods, and determined the probability of each pronunciation by using Equation 1.

Spoken sentences were translated into Kansai dialect by each speaker so that speakers would utter sentences clearly, since each speaker's dialect was slightly different. Each speaker read 100 sentences from blog articles in the BCCWJ. The spoken sentences and sentence examples for the filtering method did not overlap. We adopted Julius (Lee et al., 2001) as the ASR engine in this experiment, and the acoustic model was a phonetic tied-mixture (PTM) trigram model for Japanese language available at the Julius website.

4.2 Evaluation

This experiment evaluated ASR by recognition accuracy, *Acc*, calculated as

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

Language model		#1	#2	#3	#4	Average
Y	(Untransformed)	53.6	47.0	57.0	45.4	50.8
Y	(Dialect-transformed)	60.5	51.8	64.4	52.6	**57.3**
B	(Untransformed)	53.5	43.4	54.8	43.3	48.8
B	(Dialect-transformed)	60.1	49.4	63.9	49.4	55.7

Table 2: Word recognition accuracy of Kansai dialect [%]. Y and B stand for Yahoo! Q&A and BCCWJ, respectively.

Language model		#1	#2	#3	Average
Y	(Untransformed)	71.8	64.0	71.5	69.1
Y	(Dialect-transformed)	62.4	55.1	62.3	59.9
B	(Untransformed)	72.1	64.5	72.5	**69.7**
B	(Dialect-transformed)	62.0	56.3	58.7	59.0

Table 3: Word recognition accuracy of the CL [%]. Y and B stand for Yahoo! Q&A and BCCWJ, respectively.

where N, S, I, and D correspond to the sum of the lengths of referential word sequences, substitution errors, insertion errors, and deletion errors.

Tables 2 and 3 summarize word recognition accuracy for the Kansai dialect and CL, for LMs trained from transformed and untransformed corpora of Yahoo! Q&A and BCCWJ core data. The LMs from the Yahoo! Q&A corpus had better recognition accuracy than the LMs from the BCCWJ. As explained in Section 3.3, the Yahoo! Q&A corpus contained more sentences that had the characteristics of spoken sentences, which matched the blog articles of spoken sentences. The Yahoo! Q&A corpus made it easy to match specific kinds of topics by category filtering. These characteristics of the Yahoo! Q&A corpus made a difference despite the same size of the two corpora. Additionally, the LMs from transformed corpora resulted in better word recognition accuracy for the Kansai dialect than those from untransformed corpora (the opposite characteristics appeared for the CL). This means that the ASR system actually recognized some dialect-specific expressions like spoken language with LMs from transformed corpora. This demonstrated our method's effectiveness.

Dialect transformation was proved to reduce the effect of pronunciation-estimation errors, seeing the created dialect corpora. Automatic pronunciation estimation causes some errors, and these errors affect the recognition accuracy. Dialect transformation was proved to output correct pronunciations for inputs of mistakenly-estimated pronunciations, by training this "mistaken" transformation rules. Since dialect pronunciation in parallel corpora is correct, words with mistaken pronunciation in corpora for LMs are transformed into the correct dialect pronunciation if errors occur in a consistent way.

We interpolated the in-class probabilities of pronunciations of a dialect and the CL to recognize both pronunciations. The interpolation of probabilities is defined as follows; let P_c of Equation (2) for dialect d be rewritten as $P_{c,d}$, then

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

Transformation ratio			#1	#2	#3	#4	Average
Y	0%	(Untransformed)	53.6	47.0	57.0	45.4	50.8
Y	25%		60.8	52.6	66.1	52.2	57.9
Y	50%		61.3	51.9	65.9	51.1	57.6
Y	75%		62.0	53.8	66.0	52.9	**58.7**
Y	100%	(Completely transformed)	60.5	51.8	64.4	52.6	57.3

Table 4: Word recognition accuracy of Kansai dialect [%] with interpolated pronunciation dictionaries.

Language model		#1	#2	#3	#4	Average
Y	(75% transformed)	66.0	56.6	68.9	55.6	**61.8**

Table 5: Word recognition accuracy [%] after ignoring variation of expressions in spoken language.

gives the interpolated in-class probabilities, $P_{c,mix}$. Table 4.2 lists the recognition accuracy with interpolated pronunciation dictionaries.

Word recognition accuracy with a transformation ratio of 75% (i.e., $\alpha_{\text{dialect}} = 0.75$, and $\alpha_{\text{CL}} = 0.25$) scored the best (58.7%), which is 9.9 points higher than the result for LMs trained from the BCCWJ, and 7.9 points higher than that for LMs trained from the untransformed Yahoo! Q&A corpus. Dialect-transformed LMs have dialect pronunciations of words, but fewer kinds of CL pronunciations. Not all words in spoken language are characteristic of dialects; spoken language is composed of both dialect and CL pronunciations. The result showed that interpolated dictionaries was able to improve recognition accuracy more.

Word recognition accuracy depends on four components.

1. phoneme-sequence transducers and their parallel corpora
2. corpora with dialect-specific words
3. acoustic models
4. variation of expressions in spoken language

The first component, *phoneme-sequence transducers and their parallel corpora*, was the main idea presented in this paper. Parallel corpora determine what pronunciation ASR systems can recognize as dialect expressions. Phoneme-sequence transducers in this paper had word boundary features as well as phoneme sequences themselves. Pronunciation in a dialect can actually differ from words having the same pronunciation in the CL, depending on the part-of-speech (POS) tags of each word. One of possible improvement is including the POS tags of each word and its previous and next words to phoneme-sequence pairs in Figure 6(d).

The second component, *corpora with dialect-specific words*, is necessary to recognize actual dialogues in specific areas. Spoken sentences in this experiment did not include dialect-specific proper nouns. In other words, the problem was how to collect sentences containing such proper nouns. Corpus candidates are local pages in newspapers. Among major newspaper companies in Japan, Mainichi newspapers Co., Ltd. distributes data of local pages as well as national press.

The third component, *acoustic models*, is required to deal with acoustic features specific to dialects (e.g., changes in phonemes). Since the acoustic model in this experiment did not assume dialect speech recognition, acoustic features specific to dialects may affect word recognition accuracy.

The fourth component, *variation of expressions in spoken language*, makes non-essential errors affect recognition accuracy. Some words and phrases, especially those in spoken language, have the same meaning and role in sentence and are unnecessary to distinguish. It is important for ASR systems to rather recognize the meaning of sentences than strictly recognize sentences word by word. If they also have similar pronunciation, the variations in recognition results are likely to increase recognition errors. For example, *want to* and *wanna* in English language, even though not in all cases, can be regarded as the same phrases. In Japanese language, some particles (e.g., *na* and *ne* on the end of sentences, corresponding to tag questions in English) and verbs (*...teiru* and *...teru*: progressive form) have similar variations.

We modified the results of the 75%-transformed Yahoo! Q&A corpus to correct errors related to the fourth component. Table 5 lists word recognition accuracy after modifications. We should regard these results to demonstrate the considerable accuracy of ASR systems. These were 3.1 points higher than that of the same LM in Table 2 (61.8%) on average.

Conclusions

This paper described how to develop an ASR system that could recognize utterances in Japanese dialects. The main idea behind our system was how to create dialect corpora, few of which are actually available. We developed phoneme-sequence transducers trained from dialect-CL parallel corpora to statistically model transformations of pronunciations between dialects and the CL. Each word in the linguistic corpora was labelled as dialect word pronunciations to simulate dialect corpora. The experiment with measuring of word recognition accuracy confirmed the effectiveness of our system in recognizing dialect utterances. We were able to obtain higher recognition accuracy by adopting sentences like spoken language as corpora for training LMs. Furthermore, interpolation of in-class (pronunciation) probabilities of the CL and dialects improved recognition accuracy a little more. Our method does not depend of language and dialects; an ASR system for another language could be developed as long as parallel corpora were available.

Even though this paper assumed dialects would have no effect on the word order, a little more work should be necessary to handle slight changes in the word order as mentioned in Section 1. One possible solution is to introduce a parameter of extraneous word generation probability, p_0, like IBM model 3 (Brown et al., 1993). If WFST L (see Section 3.2) is represented by larger n-gram models ($n = 3$ in this paper) than the length of phrases within which the word order changes, WFST would model a few changes in the word order.

Our next project will be developing an ASR system to recognize various dialects alone. One possible solution to recognize utterances in multiple dialects is interpolation of in-class probabilities of pronunciations of the dialects in the same way as Section 4.2. This treatment may be too simple to work well because it assumes independence of dialects of each word. Adjacent words are intuitively likely to belong to the same dialects, and its modeling will be the main problem of recognition of multiple dialects. After that, it will be a further step to train acoustic models from dialect utterances and develop a method of switching dialects.

Acknowledgments

This study was partially supported by a Grant-in-Aid for Scientific Research (S) (No. 24220006) and the Global COE Program.

References

Akita, Y. and Kawahara, T. (2010). Statistical transformation of language and pronunciation models for spontaneous speech recognition. *Audio, Speech, and Language Processing*, 18(6):1539–1549.

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

Anusuya, M. and Katti, S. (2009). Speech recognition by machine: A review. *International Journal of Computer Science and Information Security*, 6(3):181–205.

Benincà, P., editor (1989). *Dialect variation and the theory of grammar*. Foris Publications.

Brown, P., Pietra, V., Pietra, S., and Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Chen, S. (2003). Conditional and joint models for grapheme-to-phoneme conversion. In *Proc. of EuroSpeech 2003*.

Ching, P., Lee, T., and Zee, E. (1994). From phonology and acoustic properties to automatic recognition of Cantonese. In *Proc. of Speech, Image Processing and Neural Networks, 1994*, pages 127–132.

Gadet, F. and Jones, M. (2008). Variation, contact and convergence in french spoken outside france. *Journal of language contact*, 2(1):238–248.

Gottlieb, N. (2005). *Language and society in Japan*. Cambridge University Press.

Lee, A., Kawahara, T., and Shikano, K. (2001). Julius—an open source real-time large vocabulary recognition engine. In *Proc. of EuroSpeech 2001*, pages 1691–1694.

Lee, A., Kawahara, T., Takeda, K., Mimura, M., Yamada, A., Ito, A., Itou, K., and Shikano, K. (2002). Continuous speech recognition consortium —an open repository for CSR tools and models—. In *Proc. of LREC 2002*, pages 1438–1441.

Lyu, D., Lyu, R., Chiang, Y., and Hsu, C. (2006). Speech recognition on code-switching among the Chinese dialects. In *Proc. of ICASSP 2006*, volume 1, pages 1105–1108.

Maekawa, K. (2003). Corpus of spontaneous japanese: Its design and evaluation. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*.

Maekawa, K. (2008). Balanced corpus of contemporary written Japanese. In *Proc. of ALR6 2008*, pages 101–102.

Miller, D. and Trischitta, J. (1996). Statistical dialect classification based on mean phonetic features. In *Proc. of ICSLP 1996*, volume 4, pages 2025–2027.

Misu, T. and Kawahara, T. (2006). A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts. In *Proc. of ICSLP 2006*, pages 9–12.

Munteanu, C., Penn, G., and Zhu, X. (2009). Improving automatic speech recognition for lectures through transformation-based rules learned from minimal data. In *Proc. of ACL and AFNLP*, pages 764–772. Association for Computational Linguistics.

National Institute for Japanese Language and Linguistics, editor (2001–2008). *Database of Spoken Dialects all over Japan: Collection of Japanese Dialects (In Japanese)*, volume 1–20. Kokushokankokai.

Neubig, G., Akita, Y., Mori, S., and Kawahara, T. (2012). A monotonic statistical machine translation approach to speaking style transformation. *Computer Speech & Language*, 26(5):349–370.

Neubig, G. and Mori, S. (2010). Word-based partial annotation for efficient corpus construction. In *Proc. of LREC 2010*, pages 2723–2727.

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

Wolfram, W. (2009). Dialect awareness, cultural literacy, and the public interest. In *Ethnolinguistic Diversity and Literacy Education*, chapter 6, pages 129–149. Routledge.

Woods, H. (1979). *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.

Zhang, X. (1998). Dialect MT: a case study between Cantonese and Mandarin. In *Proc. of ACL and COLING 1998*, volume 2, pages 1460–1464.