IEICE TRANS. INF. & SYST., VOL.E92–D, NO.9 SEPTEMBER 2009

PAPER

An LVCSR Based Reading Miscue Detection System Using Knowledge of Reference and Error Patterns

Changliang LIU[†a)], **Fuping PAN**[†b)], **Fengpei GE**[†], **Bin DONG**[†], *Nonmembers*, **Hongbin SUO**[†], *Student Member*, *and* **Yonghong YAN**[†c)], *Nonmember*

SUMMARY This paper describes a reading miscue detection system based on the conventional Large Vocabulary Continuous Speech Recognition (LVCSR) framework [1]. In order to incorporate the knowledge of reference (what the reader ought to read) and some error patterns into the decoding process, two methods are proposed: Dynamic Multiple Pronunciation Incorporation (DMPI) and Dynamic Interpolation of Language Model (DILM). DMPI dynamically adds some pronunciation variations into the search space to predict reading substitutions and insertions. To resolve the conflict between the coverage of error predications and the perplexity of the search space, only the pronunciation variants related to the reference are added. DILM dynamically interpolates the general language model based on the analysis of the reference and so keeps the active paths of decoding relatively near the reference. It makes the recognition more accurate, which further improves the detection performance. At the final stage of detection, an improved dynamic program (DP) is used to align the confusion network (CN) from speech recognition and the reference to generate the detecting result. The experimental results show that the proposed two methods can decrease the Equal Error Rate (EER) by 14% relatively, from 46.4% to 39.8%.

key words: *CALL, reading tutor, reading miscues, LVCSR, multiple pronunciation*

1. Introduction

Computer Assisted Language Learning (CALL) has been proved very effective for language learners. Some CALL systems concentrate on pronunciation assessment or pronunciation training (CAPT) [2], [3] and some concentrate on improving reading proficiency and comprehension of learners (reading tutor) [4]–[7]. A reading tutor can listen to the learner's reading and provides help when the learner needs or it thinks the learner needs.

One of the reading tutor's main tasks is to detect reading miscues, which contain reading omission, insertion, substitution, etc. Most reading tutors use automatic speech recognition (ASR) techniques to achieve this goal. The speech recognizer in a reading tutor should not be considered as the same as a conventional one because what the reader ought to read (reference) is known in advance. Its goal is not to recognize the content of speech in an unconstrained search space but to calculate the similarity between the reference and speech and tell where the differences are. Compared with the conventional ASR, there are more knowledge sources available to the speech recognizer for the reading tutor, such as the reference and error predictions from learning history.

Many research groups have developed reading tutors with different architectures and algorithms to detect reading miscues. Mostow et al. [4] reconstruct two primary knowledge sources – the pronunciation dictionary and the language model (LM) of Sphinx-II [8] based on the reference. Its dictionary only contains the words in the reference and some word truncations derived from the original words which are used to predict some sub-word errors. Its LM is a very simple bi-gram model which only specifies the probability of the current word followed by the next correct word or any other words in the reference. Similarly, Wang et al. [7] construct a finite state grammar network from the reference and some predicted errors as the search space for the recognizer. And, a decision tree is used to balance the coverage of errors and the perplexity of the search space.

In general, reading miscues are very arbitrary, especially for beginners. Although some error patterns, such as word repetitions, can be pre-modeled, it is hard to capture all of them in the system. The insertion and substitution miscues are especially difficult to predicte because they can be of any word, even non-word. It is very hard for the systems with a small vocabulary dictionary [4] or a finite state grammar network [7] to handle these situations, because their limited search space may not cover such errors.

Bolanos et al. [5] and Duchateau et al. [6] both use a two-stage architecture. In the first stage, a sub-word decoder generates a dense sub-word lattice. The sub-word may be a syllable or a phone. In the second stage, the extended reference which models explicitly the expected, frequent reading miscues is aligned with the lattice to locate the reading miscues. They have not considered the prior miscue distribution information in the first stage. Moreover, the sub-word decoder uses rarely high level linguistic information. Therefore, the accuracy of the sub-word lattice is limited, especially for the non-fluent speech of beginners. The alignment of the reference and the recognition result may be disordered by the recognizing errors and consequently impairs the detection performance. Bolanos et al. used a domain-specific syllable language model which is only trained on the reading material (usually a passage) to improve the recognition performance [5]. However, it is highly task-dependent. If the reading material changes, the LM has to be retrained,

Manuscript received November 4, 2008.
Manuscript revised April 25, 2009.
[†]The authors are with the ThinkIT Speech Lab., China.
a) E-mail: chliu@hccl.ioa.ac.cn
b) E-mail: fpan@hccl.ioa.ac.cn
c) E-mail: yyan@hccl.ioa.ac.cn
DOI: 10.1587/transinf.E92.D.1716

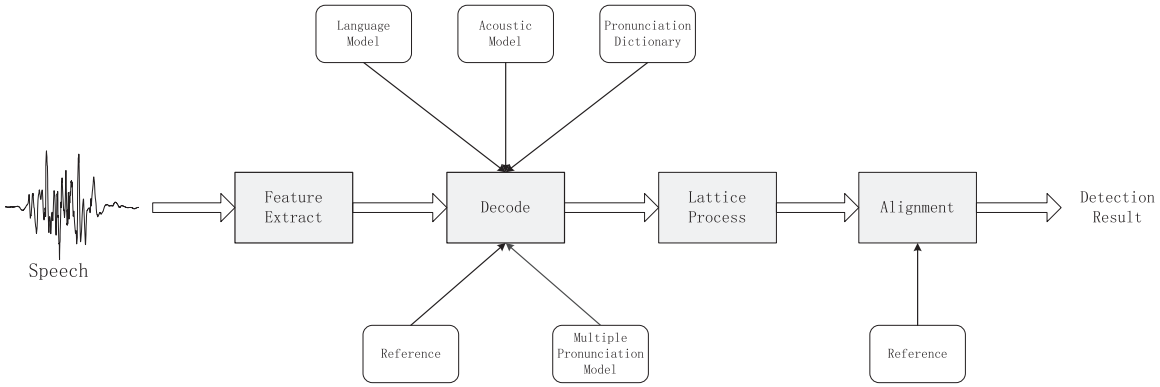


Fig. 1 The architecture of the reading miscue detection system.

which is very inconvenient.

This paper proposes a novel architecture for detecting reading miscues based on our LVCSR system [1]. LVCSR has no constraints for the speech content and can get more accurate recognizing result than the syllable or phone decoder because of the incorporation of high level linguistic information such as word language model. In order to adapt the LVCSR system to the new application, three methods are proposed. Firstly, in order to model some error patterns during the decoding process, we adopt the pronunciation variation model method which is usually used in conversational speech recognition [9] and propose an algorithm of Dynamic Multiple Pronunciation Incorporation (DMPI) for the new situation. Secondly, in order to improve the recognizing accuracy, we propose an algorithm of Dynamic Interpolation of Language Model (DILM) to dynamically interpolate the LM probability of reference in the original LM during the decoding process to constrain the search space. Finally, to further compensate for the inaccuracy of recognition, a confusion network (CN) is used to represent the result of the recognizer. Using multiple candidate hypotheses in the CN can help the alignment of the recognizing result and the reference more exactly. The detection result will be generated from the aligned path.

The rest of this paper is structured as follows: Sect. 2 describes the architecture of the system; details of DMPI and DILM are presented in Sects. 3 and 4 respectively; Sect. 5 presents the algorithm of aligning the reference and CN; the performance of this system is demonstrated in Sect. 6; the conclusion is given in Sect. 7.

2. System Architecture

Our reading miscue detection system is designed to help the Hongkong students to learn Chinese Mandarin. Its architecture is shown in Fig. 1. A decoder incorporates the knowledge sources, including language model (LM), acoustic model (AM), pronunciation dictionary (PD), reference and multiple pronunciation model (MPM), to transform the input speech into a hypotheses lattice. Then, the lattice is converted into a confusion network by a lattice processing module. Finally, the confusion network is aligned with the reference by an improved DP to locate the reading miscues. Compared with the conventional LVCSR [1], this system incorporates two additional knowledge sources: the reference and MPM. They are very important prior knowledge in CALL systems. The focus of this paper is to show how to apply them to constructing a reading miscue detection system.

The pronunciation dictionary in LVCSR provides correct pronunciations of each word for the decoder. When recognizing the correct reading, the decoder works well. However, when recognizing the speech with wrong pronunciations, the decoder will make mistakes because there are no appropriate entries in the dictionary. To resolve this problem, an MPM is used to model these wrong pronunciations. It is similar to what is used in conversational speech recognition [9]. However, we do not simply add all pronunciation variants in the MPM to the original dictionary, but use the algorithm DMPI. Considering that the reference is known in the CALL system, only pronunciation variants related to the current reference are added. It is implemented by modifying the search space online.

A domain-specific LM such as that in [5] can improve the recognition performance on non-fluent speech with miscues or even accents. Instead of training a complete task-dependent LM by the reading materials, we will interpolate the original LM with the reference LM in the decoding process dynamically. The reference LM can constrain the active decoding path near the reference. It is implemented by the algorithm DILM. It is also an online process and does not need to modify the original LM.

3. Dynamic Multiple Pronunciations Incorporation

Error predictions are very effective for the detection of reading miscues. We can place models of common errors parallel to the reference, and use decoder to see which one is preferred by the utterance. The error models can be derived from linguistic analysis [4], from real data or their combination [7].

In real situations, most reading miscues are substitutions. From the analysis of a corpus of about 600 Hongkong college students' reading of Chinese Mandarin, we found

```
长  ch ang2 0.59
长  zh ang3 0.41
长城  ch ang2 ch eng2 0.85
长城  zh ang3 ch eng2 0.14
长城  zh ang3 ch en2 0.01
长江  ch ang2 j iang1 0.088
长江  ch ang4 j iang4 0.012
.....
```

Fig. 2 An example of multiple pronunciation model.

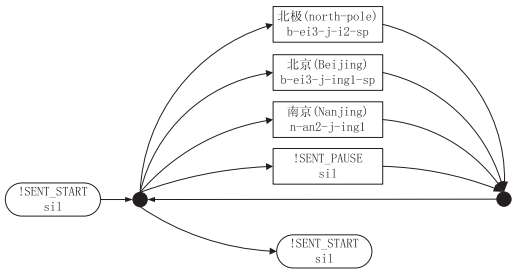


Fig. 3 An example of linear lexicon tree.

that the substitutions account for about 78% of all miscues. The substitutions are generally pronunciation variances caused by accent or misreading. We develop an MPM to predict the pronunciation variances of readers and propose an algorithm DMPI to dynamically incorporate it into the decoder as another knowledge source.

3.1 Multiple Pronunciation Model

The MPM contains all possible pronunciations of words (including the normative pronunciation and pronunciation variances). And it quantifies each pronunciation v of word w with the probability $P_{MP}(v|w)$. An example of the multiple pronunciation model is shown in Fig. 2. It is very similar to the pronunciation variation model in conversational ASR [9]. There, it is used to model the pronunciation variations, while here it is used to model the reading errors.

The multiple pronunciation model is trained by data-driven method. Plenty of second language learners' reading speech is collected. Then, their actual pronunciations are transcribed by human experts. After aligning the transcriptions with the references through DP, the pronunciation probabilities are calculated by maximum likelihood estimation (MLE) as shown in Eq. (1):

$$P_{MP}(v|w) = \frac{N(w,v)}{N(w)} \tag{1}$$

where, $N(w)$ is the count of word w and $N(w,v)$ is the count of w with pronunciation v.

3.2 Dynamical Incorporation of Multiple Pronunciation Model

In ASR, the most usual method of using MPM is to add all pronunciation variants to the original pronunciation dictionary which provides correct pronunciations for all words [9], [10]. However, adding pronunciation variants to the dictionary usually also introduces new errors because the acoustic confusability within the lexicon increases. Many studies are carried out to determine which set of pronunciation variants can balance between solving old errors and introducing new ones [11]. In our LVCSR system, the size of the pronunciation dictionary is about 40,000 words. If all the entries in the MPM are added, the dictionary will grow to about 60,000 words, which is very large. It will introduce too many new confusions and reduce the accuracy of the recognizer, which further impairs the performance of miscue detection.

A significant difference between ASR and CALL is that we know the reference in advance. Therefore, based on this information we develop an algorithm DMPI. In this algorithm, we do not add pronunciation variants in the MPM to the dictionary altogether, but only add pronunciation variants of those words which pertain to the current reference. Before decoding, they are extracted from the MPM and then added to the dictionary. Other words in the dictionary keep unchanged. Thus, the size of the dictionary is increased as minimally as possible and the increased confusability within the dictionary is very slight.

In a practical effective speech recognizer, the pronunciation dictionary with other knowledge sources is often built to a refined search space. For example, in HDecode of HTK, it is a tripone-model-based network [12] and in our TDecode, it is a memory-efficient state network [1]. In LVCSR system, the search space is built by putting all the entries in the pronunciation dictionary together in parallel, and expanding the mono-phones into tri-phones, and operating the cross-word extension, etc. This is a time-consuming process, so it is often done offline beforehand. The change of the pronunciation dictionary will cause the change of search space. Therefore, dynamically changing the dictionary requires a re-compilation of the search space, which is very inconvenient and inefficient. The proposed DMPI algorithm directly adds some additional paths to the search space to represent the pronunciation variants and it does not need re-compiling the original search space.

In our LVCSR system as described in [1], the search space is a refined state network. It is built from a linear lexicon tree and phonetic decision trees. Each edge of the lexicon tree represents an entry of the pronunciation dictionary. An example of the linear lexicon tree is shown in Fig. 3. The word nodes in the lexicon tree are then substituted by phone nodes, and after tri-phone expansion, cross-word extension, the word-based lexicon tree are transferred into a tri-phone network. The corresponding HMM of each tri-phone can be found from the phonetic decision tree. After replacing all the tri-phones by HMM states, the prototype state network is got. The prototype state network is then optimized by forward and backward node-merging process and finally generates the final state network, which is shown in Fig. 4. The details about the state network can be found in [13].

In DMPI, the pronunciation variants are added to the

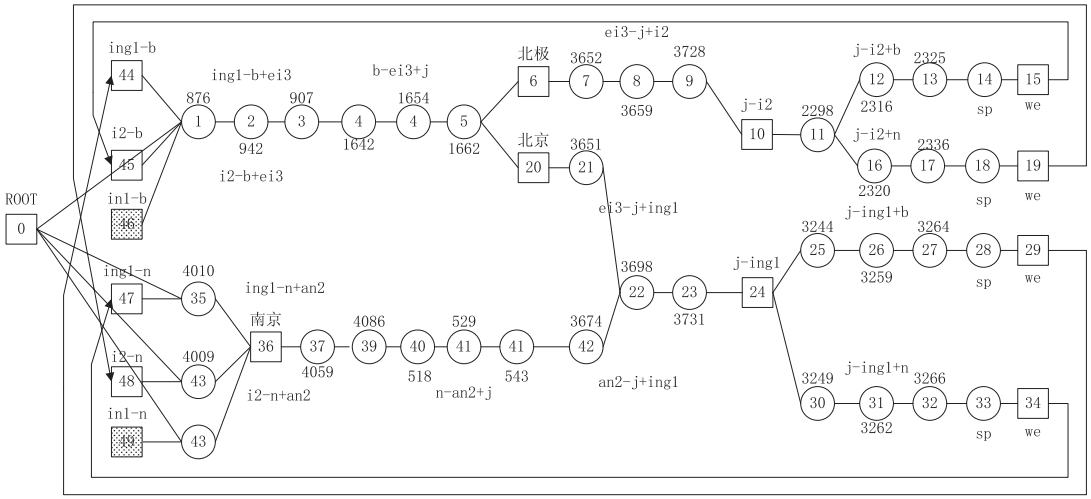


Fig. 4 The state network for the linear lexicon tree in Fig. 3. The circles denote real HMM states, while the rectangles denote dummy nodes for optimizing the network or needed in the decoding process. The numbers around the circles are state indexes in the state set of the acoustic model and the ones in the rectangles or circles are node indexes only for description. The FI and FO nodes are used for cross-word extension. Each of them represents a phone-pair and connects the tri-phones with this phone-pair at their tails or heads.

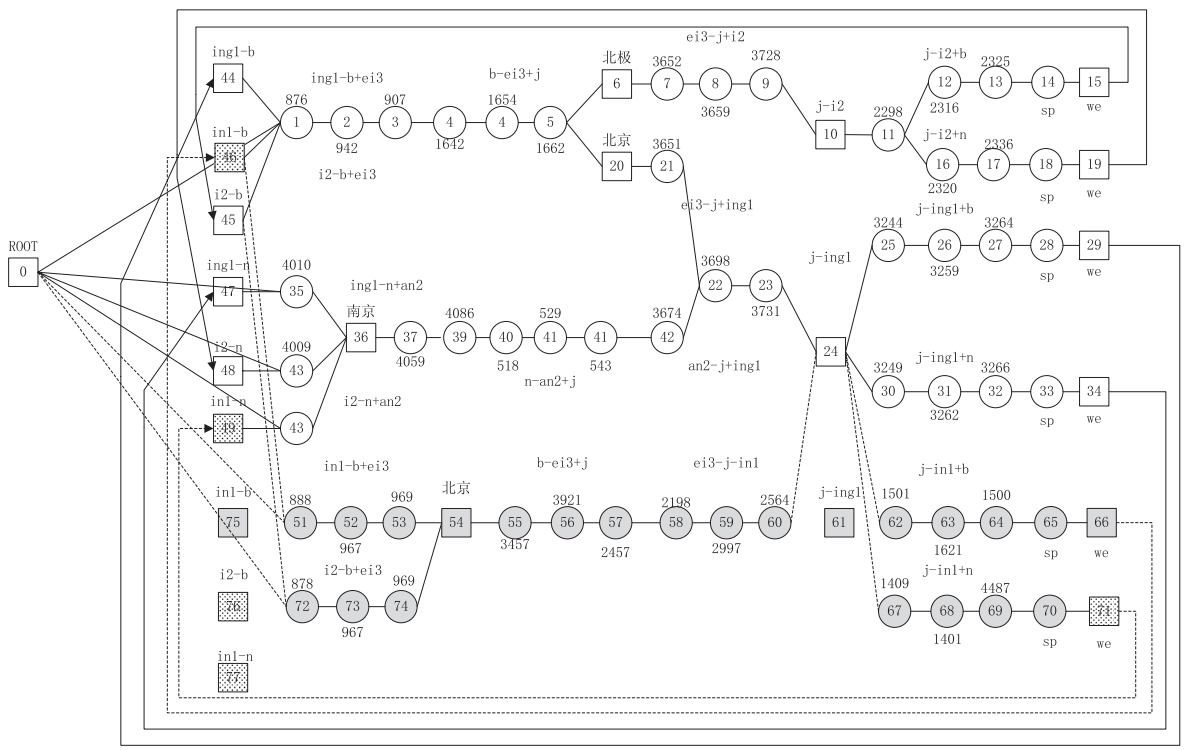


Fig. 5 The state network after merging an additional pronunciation path. The grey nodes denote a new pronunciation of the word 北京: Bei3 jin1.

state network as additional paths. They are firstly converted to HMM state paths, and then merged into the original state network one by one. The most difficult of this is to deal with the cross-word extension. We make some change on the state network described in [1]. In that network, only the essential fan-in (FI) and fan-out (FO) nodes are added. Each FI or FO node denotes a connection pair of two phones in the cross-word extension, such as "b-a3" or "e4-d". When we merge a new word to the network, if some corresponding FI or FO nodes do not exist, it will be very difficult to merge it in. Therefore, we reserve the whole set of FI and FO nodes when building the state network, no matter whether it exists

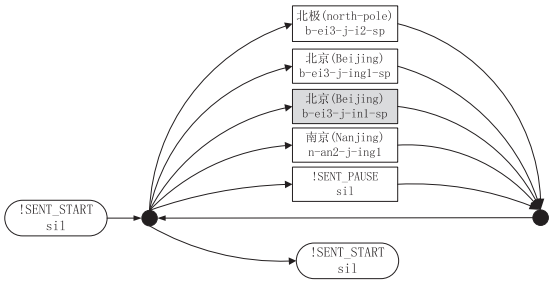


Fig. 6 The lexicon tree with an additional pronunciation.

in the lexicon tree or not. In Fig. 4, not all FI and FO nodes are listed due to space reason. The whole set of FI and FO nodes makes it easy to merge a new pronunciation variant state path to the state network. We only need to copy the links at the FI and FO nodes of the pronunciation variant state path to the corresponding FI and FO nodes of the state network.

Figure 5 shows the state network after merging a pronunciation variant of 北京: Bei3 jin1. The white nodes belong to the original state network and the gray ones belong to the merged state path of the pronunciation variant. For reference, the corresponding liner lexicon tree is also shown in Fig. 6, though the final state network is not derived from it. The gray rectangle denotes the pronunciation variant of 北京. It allows the reader to read 北京 as two pronunciations: Bei3 jing1 or Bei3 jin1.

All pronunciation variants of the same word share the same language model probability during the decoding process because they stand for the same word identity and the pronunciation probability $P_{MP}(v|w)$ is incorporated into decoding as shown in the following equation:

$$P(w|o) \triangleq P_{AM}(o|v) \cdot P_{MP}(v|w) \cdot P_{LM}(w) \tag{2}$$

where $P(w|o)$ is the probability of word w given the observation vector o, $P_{AM}(o|v)$ is the AM probability of the observation vector o given the pronunciation v, and $P_{LM}(w)$ is the LM probability of word w.

4. Dynamic Interpolation of Language Model

When using the reading tutor, the reader is supposed to read the given reference. Though the reader's real utterance may not match the reference exactly due to reading miscues, it is reasonable to assume that most words match. Therefore, the words in the reference should have higher probabilities to be read. These words should also have higher probabilities in the decoding process. This can be considered as a highly domain-specific speech recognition problem. In [5], a domain-specific LM trained only on the reading materials is used to constrain the search space. However, we do not want to replace the whole original LM because it is very inconvenient when the reading materials change and the over-constrained search space by the small LM may impede error recovery [4].

Language model adaption is widely used in domain-specific speech recognition [14]. Here, this scheme is used in our application. The original LM of LVCSR is considered as a background LM and it will be adapted by the reference for each utterance. The model merging is selected as the adaption method. Before detecting the current sentence, the reference LM is calculated and then interpolated in the background LM. However, we do not modify the values of the original LM actually. An on-line process is used. The final LM probabilities are calculated in the decoding process dynamically. Only the current sentence is used to adapt the background LM, hoping to keep the active decoding path near the reference. This method is called Dynamic Interpolation of Language Model (DILM). Using this method, it does not need to re-train the LM when the reading materials change and the background LM can provide more opportunities for the recovery of the arbitrary reading miscues.

In DILM, the original language model probability $P_{LM}(w)$ in Eq. (2) is changed to $\hat{P}_{LM}(w)$ as in Eq. (3) in the decoding process,

$$\hat{P}_{LM}(w) = (1-\alpha)P_{LM}(w) + \alpha P_{ref}(w) \tag{3}$$

where $P_{ref}(w)$ is the probability of w in current reference to be decoded and α is a coefficient between 0 and 1, used to tune the weight of the reference LM. $P_{ref}(w)$ is calculated using Eq. (4),

$$P_{ref}(w) = \frac{N(w)}{\sum_{i=1}^{n} N(w_i)} \tag{4}$$

where $N(w)$ is the count of word w in the current reference, and n is the number of individual words in the current reference.

The calculation of $P_{ref}(w)$ is very similar to the training of an n-gram language model. If considering the context of word w, the unit of the reference model can be bi-gram or tri-gram, etc. Different n-gram lengths have different characteristics in improving recognition and detection performance. The three types of n-grams, tri-gram, bi-gram and uni-gram, are compared in our experiments below.

5. Alignment of the Reference and the Confusion Network

The final stage of the detecting system is to align the recognizer output and the reference to locate the reading miscues. Some researches just use the 1-best hypothesis as the recognizer output [4] and some others use more complex output, such as lattice [5], [6] to compensate for the errors caused by the recognizer itself. We select the confusion network (CN) as the object to be aligned with the reference. It is not only because the confusion network is simpler than the lattice, but also because it is built based on Minimum Word Error (MWE) criterion [15], which is more appropriate for the word miscue detection application.

CN is built from the lattice by merging the equivalent word (edge) classes using intra-word and inter-word clustering algorithm based on the time similarity, phonetic similarity and word posterior probabilities [15]. The word posterior

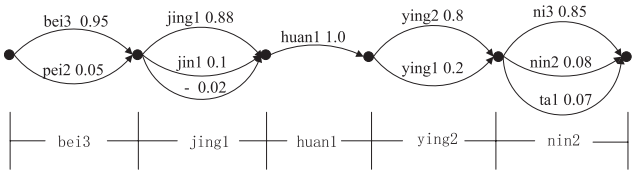


Fig. 7 An example of confusion network and its alignment with the reference "bei3 jing1 huan1 ying2 nin2."

probability is the sum of the posterior probabilities of all lattice paths of which the word is a part. CN can be thought as a highly compacted representation of the original lattice with the property that all word hypotheses are totally ordered. It has been widely used in many applications, such as minimization of word error rate, lattice compression, word spotting, confidence annotation, etc. An example of CN is shown in Fig. 7. It has many candidates in each alignment column and each candidate has a posterior probability with it.

Generally, the minimum miscue detection unit of Chinese Mandarin is syllable (or character). Therefore, the word lattice from the decoder will be split into a syllable lattice firstly and the final CN is also a syllable-based one. A syllable-based CN is shown in Fig. 7, in which each edge stands for a syllable y with a posterior probability $P(y)$ and "-" denotes a deletion edge.

The alignment of the reference and CN is implemented by an improved DP. Because each alignment column in CN has one or more edges, the cost function of DP should be re-defined, as in Eq. (5).

$$F_{cost} = \begin{cases} 1 & \text{insertion,} \\ 1 - \hat{P}(\text{-}) & \text{deletion,} \\ 1 - \hat{P}(r) & \text{substitutioan/correction.} \end{cases} \quad (5)$$

Where,

$$\hat{P}(*) = \begin{cases} P(*), & * \in \text{current slice} \\ 0. & * \notin \text{current slice} \end{cases}$$

"$*$" denotes "-" or the current reference syllable r.

After the alignment, the miscue detection results can be acquired from the aligned path. In each position, whether the actual reading word matches the reference is determined by a threshold T that balances the detection error and false alarm rate.

6. Experiment

6.1 Corpus and Evaluation Metrics

The corpus used to evaluate the proposed system is from Hongkong college students' reading speech of Chinese Mandarin. The reading materials are four passages from "Hongkong Putonghua Shuiping Kaoshi" (PSK) test [16]. Each passage has about 350 characters. There are 664 students totally, half of which are male and half are female. Each of them reads one of the four passages. The speech was recorded in quiet classrooms with head microphones and stored in the format of 16 K sample rate, 16 bit sample length, mono channel. All speech was transcribed with syllable sequences (Pinyin) by Chinese native human experts.

Table 1 Reading miscues' distribution in the development and test sets.

Data set	Insertion	Omission	Substitution	Total
DevSet	4.7%	0.41%	18.24%	23.35%
TestSet	5.14%	0.18%	15.9%	21.26%

The corpus is divided into two parts: 100 students' data as test set and the rest as development set. The development set is used to adjust the acoustic model and train the multiple pronunciation model. The test set is used to evaluate the performance of the system. The distribution of the three kinds of reading miscues (insertions, omissions and substitutions) in the development set (DevSet) and test set (TestSet) are shown in Table 1.

We use three metrics to measure the performance of the system: syllable recognition error rate (SER) is used to evaluate the accuracy of recognition; miscue detection error rate (DER) and false alarm rate (FAR) are used to evaluate the detecting performance. SER is computed by comparing the 1-best recognizing result with the transcription using DP. DER is defined as the number of miscues which have not been detected divided by the number of all miscues; and FAR is defined as the number of words erroneously detected as reading miscues divided by the number of all miscues. The real miscue information is obtained by aligning the reference with the transcription. By changing the threshold T referred to in Sect. 5, different points of DER and FAR can be obtained. They can be plotted as a detection error tradeoff (DET) curve. For the convenience of comparison, equal error rate (EER) is also used, which is the rate when the DER equals to FAR.

6.2 Experiment Setup

The front end of the system extracts 39-dimension MFCC features, including 12-dimension static cepstrum and 1-dimension energy, with their 1st and 2nd derivatives. The HMM acoustic model was trained on about three hundred hours of speech from Chinese native speakers, then was adapted by the development set of the Hongkong corpus referred to in Sect. 6.1 with a Maximum A Posteriori (MAP) algorithm [17]. There are about five thousand states in the final AM and there are 32 Gaussian mixtures in each state. The language model is a general tri-gram model which was trained on about 2 gigabytes text materials. There are about forty thousand words in the dictionary, including all characters and most words of Chinese Mandarin.

6.3 Experimental Result

6.3.1 Baseline Performance

In order to investigate the performance of the algorithms DMPI and DILM, we construct a baseline system using a

Table 2 Performance of the baseline system.

	SER	EER
baseline	31.0%	46.3%

Table 3 Performance of dynamic multiple pronunciations incorporation.

	SER	EER
baseline	31.0%	46.3%
AddDict	30.5%	49.1%
DMPI	27.8%	42%

conventional LVCSR decoder followed by a CN alignment module, without any manipulation on the LM or the pronunciation dictionary. Its performance is shown in Table 2 in terms of SER and EER.

6.3.2 Performance of Dynamic Multiple Pronunciation Incorporation

In this section, the performance of DMPI is investigated. The multiple pronunciation model is trained on the development set mentioned in Sect. 6.1.

For comparison, the simple method of incorporating the MPM which adds all pronunciation variants to the pronunciation dictionary ('AddDict' for short) was tested. Table 3 shows the SER and EER results of AddDict and DMPI compared with the baseline. Though the AddDict reduces the SER slightly, it makes the EER much worse. As described in Sect. 3.2, AddDict introduces too many new confusions to the original pronunciation dictionary and impairs the positive effect of MPM. DMPI significantly improves both the recognition and detection results. The SER and EER are reduced by relative 9.5% and 9.3% respectively. DMPI only adds the most useful pronunciation variants to the original pronunciation dictionary. It exploits the positive effect of the MPM efficiently and has the minimal new confusability, thus can achieve a better performance.

6.3.3 Performance of Dynamic Interpolation of Language Model

In this section, the performance of the algorithm DILM is investigated. It was implemented on the baseline and DMPI systems respectively. Table 4 shows the SER and EER results of the baseline, DILM and combination of DMPI and DILM.

When DILM is implemented on the baseline system, it improves the accuracy of the recognizer significantly. The SER is decreased 25.5% relatively, from 31% to 23.1%, which proves the effectiveness of using reference to constrain the active decoding paths near the reference. While the recognizing result is improved, the detection result EER is hence improved. The gain is 3.5% relatively. When DILM is implemented on DMPI system, the improvement is still significant. The SER and EER are reduced relative 22.7% and 5.2% respectively. Furthermore, because DMPI increase the capability of the decoder to recognize the erroneous reading words, the gain of DILM on DMPI system is larger than that on the baseline system.

Table 4 Performance of DILM on baseline and DMPI systems.

	SER	EER
baseline	31.0%	46.3%
baseline+DILM	23.1%	44.7%
DMPI+DILM	21.5%	39.8%

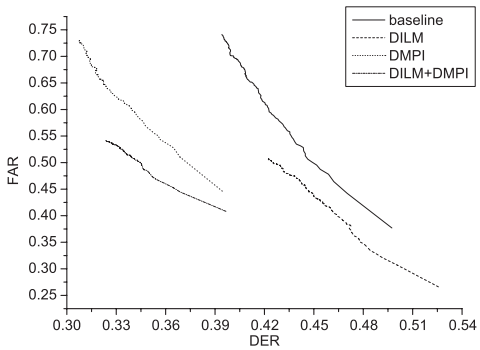


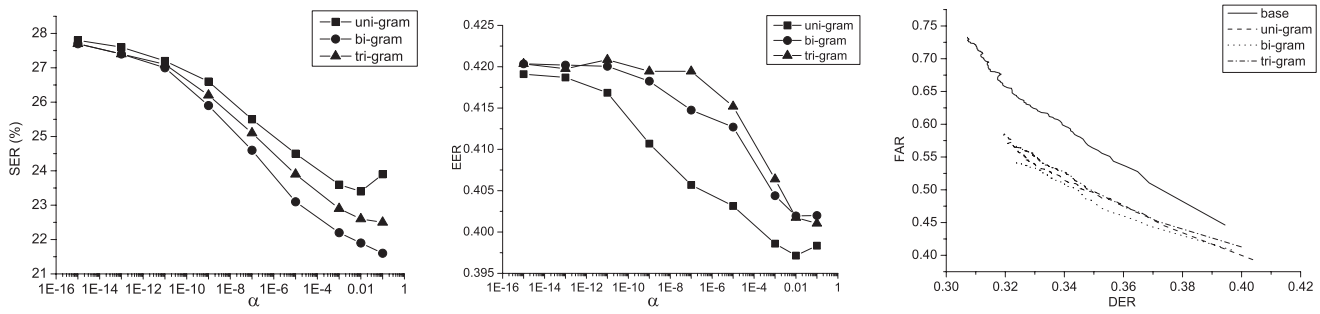
Fig. 8 DET curves of different methods compared with baseline.

Figure 8 shows the DET curves of the baseline, DMPI, DILM and combination of DMPI and DILM. It is shown that DMPI and DILM significantly improve the detection performance compared with the baseline. The final EER 39.8% is achieved by the combination of DMPI and DILM.

We also investigate the effect of different n-gram lengths of the reference LM (uni-gram, bi-gram and tri-gram) and different α value in Eq. (3) in this section.

The performance of different n-gram lengths and different α based on the DMPI system is illustrated in Fig. 9. Along with the increase of α, both the SER and EER decrease significantly. A larger α imposes tighter constraint on the original LM, which makes the decoding active path closer to the reference. Therefore, larger α results in better recognition accuracy, and thus result in better detection performance. However, when α is too large, the over-constrained search space will impair the capability of the recognizer to recover the reading miscues. It can be observed from Fig. 9 (b) that when α is larger than 1e-2, the EER begins to become worse. For an extreme example, when $\alpha = 1$ (where only the reference LM works actually), the EER is about 41%, which is lower than the best EER 39.8%.

The different n-gram lengths have a consistent effect on the detection of reading miscues, though there is a little difference among their performances. The longer the n-gram is, the tighter the constraint on the original LM is. Shorter n-gram allows more words to be recognized and can capture more reading miscues. However, it also results in more recognition errors which are adverse for the miscue detection. The two facets should be balanced. Figure 9 (c) shows the DET curves of different n-gram lengths. They are close to each other. Figure 9 (b) shows that the uni-gram is a little better than others on EER performance. A uni-gram reference LM is enough for the detection of reading miscues.



(a) SER (b) EER (c) DET

Fig. 9 Performance of dynamic interpolation of language model with dynamic multiple pronunciation incorporation.

6.4 Discussion

By using DMPI and DILM algorithms, the proposed LVCSR-based miscue detection system finally achieves the result EER 39.8%. It is compared with the system described in [4]. We tested the performance of that system on our corpus and got the EER of 42.8%, which is 7% lower relatively. The rich search space and background LM in the LVCSR-based system provide more opportunities for the reading miscues to be captured than the system in [4]. Therefore, a better performance can be obtained.

For a practical CALL system, the EER 39.8% is not very satisfactory. However, among all the detection errors, more than half of them are caused by tone errors. Chinese language is a tonal language. There are about 1300 tonal syllables in Chinese, but the base syllable number is only about 400. Our current LVCSR system has no special consideration for tone recognition. This is one of the reasons for the poor detection performance. However, our research provides a general framework for the reading miscue detection. Under this framework, many information sources can be integrated in. It will be a promising framework for the research of reading miscue detection.

7. Conclusion

In this paper, we described the building of a reading miscue detection system based on the conventional LVCSR framework. In order to compensate for the poor recognition performance of a conventional LVCSR on non-fluent speech, we proposed two algorithms: DMPI and DILM. DPMI trains an MPM on the history reading data to model the common reading errors and dynamically adds the pronunciation variants relative to the current reference to the search space. It provides additional paths in the original search space and makes the decoding more accurate. The dynamical incorporation ameliorates the conflict between the coverage of error predications and the perplexity of the search space. It decreases EER by 9.3% relatively. DILM dynamically interpolates the background LM using the online-building reference LM. By this method, a domain-specific LM obtained from the current reference is used for each utterance actually. The using of the background LM also supplies more opportunities for the reading miscues to be captured. It further improves the EER by 5.2% relatively. The experiments of different n-gram lengths of the reference LM shows that uni-gram is enough for the detection of reading miscues.

Though the result is not very satisfactory now, it provides a promising framework for studying reading miscue detection. In the future, more complex MPM using decision tree, natural networks, etc. will be investigated and more confidence information will be also used to help to decrease the false alarms. For improving tone detection performance, some tone-based features and methods will also be considered.

Acknowledgement

This work is partially supported by The National High Technology Research and Development Program of China (863 program, 2006AA010102, 2006AA01Z195), MOST (973 program, 2004CB318106), National Natural Science Foundation of China (10574140, 60535030).

References

[1] J. Shao, T. Li, Q. Zhang, Q. Zhao, and Y. Yan, "A one-pass real-time decoder using memory-efficient state network," IEICE Trans. Inf. & Syst., vol.E91-D, no.3, pp.529–537, March 2008.
[2] S. Witt and S. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," Speech Commun., vol.30, no.2-3, pp.95–108, 2000.
[3] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic scoring of pronunciation quality," Speech Commun., vol.30, no.2-3, pp.83–93, 2000.
[4] J. Mostow, S. Roth, E. Hauptmann, and M. Kane, "A prototype reading coach that listens," Proc. Twelfth National Conference on Artificial Intelligence, 1994.
[5] D. Bolanos, W. Ward, S.V. Vuuren, and J. Garrido, "Syllable lattices as a basis for a children's speech reading tracker," Tenth European Conference on Speech Communication and Technology, ISCA, 2007.
[6] J. Duchateau, M. Wigham, K. Demuynck, and H. Hamme, "A flexible recogniser architecture in a reading tutor for children," Speech Recognition and Intrinsic Variation Workshop, ISCA, 2006.
[7] H. Wang and T. Kawahara, "Effective error prediction using decision tree for ASR grammar network in call system," IEEE International

Conference on Acoustics, Speech and Signal Processing, pp.5069–5072, 2008.
[8] X.D. Huang, F. Alleva, H.W. Hon, M.Y. Hwang, K.F. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: An overview," Comput. Speech Lang., vol.7, no.2, pp.137–148, 1993.
[9] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for ASR: A survey of the literature," Speech Commun., vol.29, no.2-4, pp.225–246, 1999.
[10] M. Adda-Decker and L. Lamel, "Pronunciation variants across systems, languages and speaking style," Proc. ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition, pp.1–6, 1998.
[11] N. Cremelie and J. Martens, "In search of better pronunciation models for speech recognition," Speech Commun., vol.29, no.2-4, pp.115–136, 1999.
[12] S. Young, G. Everman, M. Gales, et al., The HTK book(for HTK version 3.4), 2007.
[13] J. Shao, Chinese Spoken Term Detection towards Large-Scale Telephone Conversational Speech, Ph. D. Thesis of Chinese Academy of Science, 2008.
[14] J. Bellegarda, "Statistical language model adaptation: Review and perspectives," Speech Commun., vol.42, no.1, pp.93–108, 2004.
[15] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," Comput. Speech Lang., vol.14, pp.373–400, 2000.
[16] PSK, "http://www.psk.polyu.edu.hk/psk/"
[17] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixtureobservations of Markov chains," IEEE Trans. Speech Audio Process., vol.2, no.2, pp.291–298, 1994.

Changliang Liu received his B.S. degree from University of Science and Technology of China. Now he is a Ph.D. candidate of ThinkIT Speech Lab, Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS). He researches on pronunciation quality assessing, speech signal processing and speech recognition.

Fuping Pan received his Ph.D. in Information and Signal Processing from IOA, CAS, 2007. He is currently an Assistant Researcher in IOA. His research is focused on automatic pronunciation evaluation, speech signal processing and speech recognition.

Fengpei Ge received her B.E. degree from Tianjin University in 2005. Now she is a Ph.D. candidate of ThinkIT Speech Lab, Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS). Her research interests include pronunciation quality assessing, discriminative training algorithm for acoustic modeling, speech signal processing and speech recognition.

Bin Dong received his Ph.D. in Information and Signal Processing from IOA, CAS, 2006. He is currently an Assistant Researcher in IOA. His research is focused on computer assistant language learning, automatic pronunciation evaluation, speech signal processing and speech recognition.

Hongbin Suo received his B.E. degree in Mechanical Engineering from Zhejiang University in 2002. Now he is a doctor student of ThinkIT Speech Lab, Institute of Acoustics (IOA), Chinese Academy of Sciences (CAS). His research is focused on automatic spoken language recognition.

Yonghong Yan received his BE from Tsinghua University in 1990, and his PhD from Oregon Graduate Institute (OGI). He worked in OGI as Assistant Professor (1995), Associate Professor (1998) and Associate Director (1997) of Center for Spoken Language Understanding. He worked in Intel from 1998–2001, chaired Human Computer Interface Research Council, worked as Principal Engineer of Microprocessor Research Lab and Director of Intel China Research Center. Currently he is a professor and director of ThinkIT Lab. His research interests include speech processing and recognition, language/speaker recognition, and human computer interface. He has published more than 100 papers and holds 40 patents.