International Journal of Computer Processing of Oriental Languages
Vol. 18, No. 4 (2005) 243–264

World Scientific
www.worldscientific.com

Thai Spelling Recognition Using a Continuous Speech Corpus*

Chutima Pisarn[†,§], Thanaruk Theeramunkong[†,¶],
Nick Cercone[‡] and Junalux Chalidabhongse[†,//]

[†]Sirindhorn International Institute of Technology,
131 Moo 5 Tiwanont Rd., Bangkadi, Muang, Pathumthani 12000, Thailand
[‡]Faculty of Computer Science, Dalhousie University,
6050 University Avenue, Halifax, NS, Canada
[§]*chutimap@siit.tu.ac.th*
[¶]*thanaruk@siit.tu.ac.th*
[‡]*nick@cs.dal.ca*
[//]*junalux@siit.tu.ac.th*

Spelling recognition provides alternative input method for computer systems as well as enhances a speech recognizer to cope with incorrectly recognized words and out-of-vocabulary words. This paper presents a general framework of Thai speech recognition enhanced with spelling recognition. Towards the implementation of Thai spelling recognition, Thai alphabets and their spelling methods are analyzed. A method based on hidden Markov models is proposed for constructing a Thai spelling recognition system from an existing continuous speech corpus. To compensate speed difference between spelling utterances and continuous speech utterances, the adjustment of utterance speed is taken into account. Two alternative language models, bigram and trigram, are used to investigate the performance of spelling recognition under three different environments: close-type, open-type and mix-type language models. Using the 1.25-times-stretched training utterances under the mix-type language model, the system achieves 87.37% correctness and 87.18% accuracy for bigram, and up to 91.12% correctness and 90.80% accuracy for trigram.

Keywords: Thai spelling recognition; Thai speech recognition framework; hidden Markov model; utterance speed compensation.

1. Introduction

Recently automatic speech recognition (ASR) research for continuous speech has been made in the context of either systems that rely on dictionaries or those that

*Paper presented at the Int. Conf. on Intelligence in Communication Systems (IntellComm 2004), Bangkok, Thailand, 23–26 Nov 2004.

can recognize out-of-vocabulary circumstances. In the situation of misrecognition and out-of-vocabulary words, a practical and efficient solution to assist the ASR is to equip a system with a spelling recognition subsystem, in which users can spell out a word, letter by letter. Spelling recognition is a challenging task with much interest for directory assistance services, or other applications where a large number of proper names or addresses must be recognized. Many works on spelling recognition were widely developed in several languages, for instance, English, Spanish, Portuguese and German. In [1], hypothesis-verification Spanish continuous spelt proper name recognition over the telephone was proposed. Several feature sets were investigated in models of neural networks. As their succeeding work [2], three different recognition architectures, including the two-level architecture, integrated architecture and hypothesis-verification architecture, were analyzed and compared. In [3], a Portuguese subject-independent system for recognizing an isolated letter was introduced. The system simulated the recognition of speech utterances over a telephone line using hidden Markov models (HMMs). A number of experiments were made over four different perplexity language models. In [4], Mitchell and Setlur proposed a fast list matcher to select a name from the name list that was created from an *n*-best letter recognizer on spelling over the telephone line recognition task. In [5], an integration approach was proposed to combine word recognition with spelling recognition in a user-friendly manner as a fallback strategy. As a German city name recognizer, the system was applied to directory assistance services.

With regards to speech recognition in Thai, there are few works on large vocabulary continuous speech recognition (LVCSR). In [6], a Thai continuous speech recognition system is developed with the vocabulary of approximately 5,000 words. This work showed a method to improve the system by incorporating tone acoustic features to the classical Perceptual Linear Prediction (PLP) feature vector. This system could yield 81.70% accuracy in a closed environment, where the training and test sets were identical. As a commercial product on Thai automatic speech recognition, Tellvoice [7] claimed to gain up to 95% accuracy in both isolated and continuous speeches. This performance is achieved when the system is applied to certain optimized applications and services. Even with quite high recognition accuracy, the system is limited to only some specific domain. However, the performance drops drastically when it is applied to an open environment, where the test data are unseen beforehand. In our early investigation on Thai continuous speech corpus, four environments [8], which are the combinations of closed vs. open environments and acoustic vs. language models, were investigated. The system achieved high recognition rate of up to 97.91% accuracy in the closed environment for both acoustic and language models while it gained only 28.03% accuracy in the open environment of both acoustic

and language models. The result implies that the language model seems to play a main contribution in achieving high recognition rate. Recognizing unknown words or even word sequences outside the training corpus is still a hard problem. To solve this problem, it is possible to equip a speech recognition system with a mechanism to allow a user to spell misrecognized words.

Unlike other languages, spelling in Thai has several styles. One of them is similar to spelling in the English language, i.e., */h-@@4//m-@@0//z-aa0/* of "หมา" corresponding to */d-ii0//z-oo0// g-ii0/* for "dog". There are three additional methods in Thai spelling, where some syllables are inserted to make it clearer for the hearer to grasp the correctly spelt letter. The most common method is to spell out a letter followed by uttering its representative word. Another method is the mixture among the former two types. The third method is to spell out a set of letters that form a syllable followed by its corresponding pronunciation. Thus far spelling recognition for the Thai language has not been explored. One reason is that there is no public corpus for this purpose. However, creating a corpus of spelling utterances is a time-consuming task. A promising solution to this problem is to reuse some existing speech corpora. Based on the above background, this work has three objectives as follows. The first objective is to systematically analyze what are commonly used spelling methods in Thai. The second is to examine the possibility to apply an existing Thai continuous speech corpus in spelling recognition; even continuous speech is somehow different from spelling speech. The last is to examine the affect of using a higher gram language model (i.e. trigram) on recognition performance.

This paper is organized as follows. In Section 2, the recognition framework is presented. Thai language characteristics are introduced in Section 3. The implementation is discussed in Section 4. The experimental results and their analysis are shown in Section 5. Section 6 illustrates an analysis of the errors occurred in the experiments. Finally, our accomplishments are summarized and next steps are given in Section 7.

2. The Recognition Framework

It is well known that a speech recognition system achieves high performance when it analyzes utterances that are seen by the system during the training (later called a closed environment). However, the recognition performance drops dramatically when the system tries to recognize words or phrases that do not exist in the system, especially in the language model (so-called an open environment). In one of our preliminary experiments with Thai continuous speech recognition, we found that the system achieved 97.91% for the closed environment but only 28.03% for the open environment. Towards real applications, one possible way

to solve such misrecognition is to allow a user to spell those words and then to employ a spelling recognizer to recognize them. In this work, we propose a framework of complementing continuous speech recognition with spelling recognition. To achieve this, there are three main tasks corresponding to the following questions: (1) how to recognize continuous speech, (2) how to detect and handle misrecognition, and (3) how to recognize spelling speech. These tasks are embedded in our framework shown in Figure 1. With respect to the tasks, the framework consists of three modules; speech recognition module, validation module and spelling recognition module. The speech recognition module recognizes continuous speech utterances while the spelling recognition module performs recognizing spelling utterances. Each module is supported by two main models; an acoustic model and a language model. The former model can be trained by speech utterances in the speech corpus together with a word-level pronunciation dictionary. The latter is trained by a set of transcriptions in a text corpus. There are two main differences between speech recognition and spelling recognition. They are (1) the language model for the former is the connection likelihood between contiguous words while the language model for the later indicates the connection likelihood of contiguous letters, (2) the pronunciation dictionary for speech recognition indicates how to pronounce a word while that for spelling recognition displays the way to pronounce a letter in spelling. Moreover, compared to spelling recognition, the continuous speech recognition needs an extremely larger training corpus in order to achieve high accuracy. In the validation module, two possible approaches are: (1) to set a threshold and allow the system to automatically detect misrecognized words, e.g., triggered by low probability or (2) to allow a user to send signals, in the form of uttering a clue word or pressing a button to switch to perform spelling recognition. As the first stage, this paper focuses on only spelling recognition that is the last task in this framework.

3. Thai Language Characteristics

3.1. Thai alphabet

Theoretically, the Thai language has 69 letters, which can be grouped into three classes of phone expression: consonant, vowel and tone. There are 44, 21, and 4 letters for consonants, vowels, and tones, respectively. Some Thai consonant symbols share the same phonetic sound. Because of this, there are only 21 phones for Thai consonants. On the other hand, some vowels can be combined with other vowels, resulting in 32 possible phones. However, in practice, only 18 letters in the vowel class are currently used in Thai. There are 4 tone symbols to express 5 Thai tones. Thus, there are 66 practical letters as shown in Table 1.

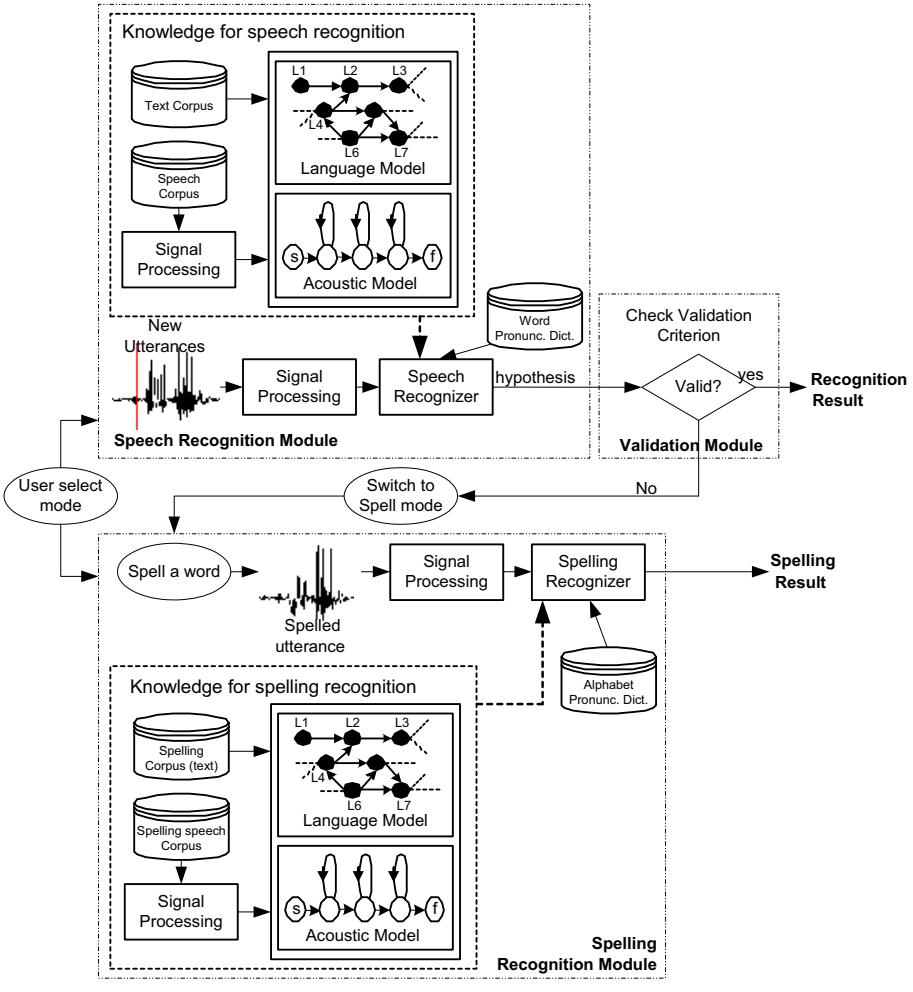

Figure 1. The recognition framework.

Table 1:. Three classes of Thai alphabet: consonants, vowels and tones.

Basic Classes	Letters in each class
Consonant (44)	ก,ข,ฃ,ค,ฅ,ฆ,ง,จ,ฉ,ช,ซ,ฌ,ญ,ฎ,ฏ,ฐ,ฑ,ฒ,ณ,ด,ต,ถ,ท,ธ,น,บ,ป,ผ,ฝ,พ,ฟ,ภ,ม,ย,ร,ล,ว,ศ,ษ,ส,ห,ฬ,อ,ฮ
Vowel (18)	อั, อะ, อิ, อา, อิ, อี, อึ, อื, อุ, อู, เอ, แอ, โอ, อำ, ไอ, ใอ, ฤ, อ์
Tone (4)	อ่, อ้, อ๊, อ๋

Table 2. Phonetic symbols grouped into initial consonants, vowels, final consonants and tones.

Initial Consonant (C_i)		**Vowel** (V)	**Final Consonant** (C_f)	**Tone** (T)
Base	**Cluster**			
p,t,c,k,z,ph, *th,ch,k,h,b,* *br,bl,d,dr,* *m,n,ng,r,f,* *fr,fl,s,h,w,j*	*pr,phr,pl,* *phl,tr,thr,* *kr,khr,kl,* *khl,kw,* *khw*	*a,aa,i,ii,v,vv,u,* *uu,e,ee,x,xx,o,* *oo,@,@@,q,* *qq,ia,iia,va,* *vva, ua, uua*	*p^,t^,k^,n^,m^,n^,* *g^,j^,w^,f^,l^,s^,* *ch^, jf^,ts^*	0 (Mid) 1 (Low) 2 (Falling) 3 (High) 4 (Rising)

3.2. Thai syllable characteristics and phonetic representation

Like most languages, a Thai syllable can be separated into three parts; (1) initial consonant, (2) vowel and (3) final consonant. The phonetic representation of one syllable can be expressed in the form of $/C_i\text{-}V^T\text{-}C_f/$, where C_i is an initial consonant, V is a vowel, C_f is a final consonant and T is a tone which is phonetically attached to the vowel part. Some initial consonants are cluster consonants. Each of them has a phone similar to that of a corresponding base consonant. For example, *pr*, and *pl* are similar to a base consonant *p*. In the vowel part, there are 18 vowel phones and 6 dipthongs. Following the concept presented in [6], there are totally 76 phonetic symbols and 5 tone symbols in Thai, as shown in Table 2.

Naturally phones, especially those in the vowel class, are various in their durations. In Thai language, most vowels have their pairs. For example, the vowel pair *a* and *aa* have a similar phone but different durations. The other vowel pairs are *i-ii, v-vv, u-uu, e-ee, x-xx, o-oo, @-@@, q-qq, ia-iia, va-vva,* and *ua-uua*. Intuitively, these pairs are easily confused in the recognition process.

3.3. Basic pronunciation of Thai alphabet

Thai alphabets of different classes have different styles of pronunciation. The consonantal letters can be uttered in either of the following two styles. The first style is simply pronouncing the core sound of a consonant. For example, the letter 'ก', its core sound can be represented as the phonetic sound */k-@@0/*. Normally, some consonants share a same core sound. For example, 'ค', 'ต', 'ฆ' have the same phonetic sound */kh-@@0/*. In such case, the hearer may encounter with letter ambiguity. To solve this issue, the second style is generally applied by uttering a core sound of the consonant followed by the representative word of that consonant. Every consonant has its representative word. For example,

the representative word of the letter ‘ก’ is “ไก่” (meaning: “chicken”, sound: */k-a1-j^/*), and that of the letter ‘ข’ is “ไข่” (meaning: “egg”, sound: */kh-a1-j^/*). To express the letter ‘ก’ using this style, the syllable sequence */k-@@0/+/k-a1-j^/* is uttered.

Expressing letters in the vowel class is quite different from that of the consonant class. There are two types of vowels. The first-type vowels can be pronounced in two ways. One is to pronounce the word “สระ” (meaning: “vowel”, sound: */s-a1//r-a1/*), followed by the core sound of the vowel. The other is to simply pronounce the core sound of the vowel. On the other hand, for the second-type vowels, they are uttered by calling their names. The vowel letters of each type are listed in Table 3. As the last class, tone symbols are always pronounced by calling their names. Table 4 concludes how to pronounce a letter in each alphabet class.

3.4. Thai word spelling methods

Spelling a word is done by pronouncing letters in the word one by one in order. We can refer to spelling as a combination of the pronunciation of each letter in the word. Only four Thai commonly used spelling methods are addressed. For all methods, the second-type vowels and tones are pronounced by calling their names. The difference among the four methods is in spelling consonants and the first-type vowels. An example of these methods (M1–M4) in spelling the word “เท่ห์” is depicted in Figure 2.

Table 3. Two types of vowels.

First-type vowels	ะ, า, ิ, ี, ึ, ื, ุ, ู, เ, แ, โ, ำ, ไ, ใ
Second-type vowels	ั, ็, ์, ฤ

Table 4. Pronounciation methods for each alphabet class.

Alphabet class	Pronounciation methods
Consonant	1. consonant core sound + representative word of consonant
	2. consonant core sound
First-type vowel	1. */s-a1//r-a1/* + vowel core sound
	2. vowel core sound
Second-type vowel	1. the vowel name
Tone	1. the tone name

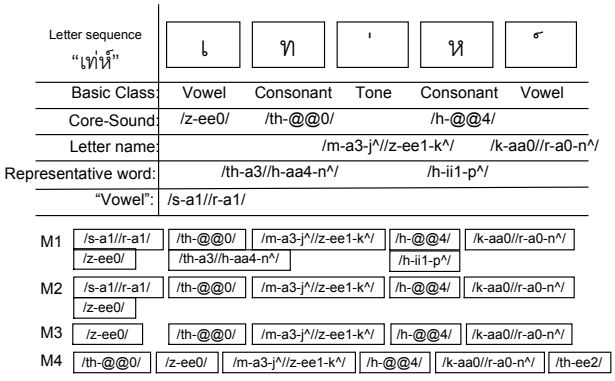

Figure 2. Four spelling methods for the word “เท่ห์”.

The first spelling method is to pronounce the representative word of each consonant, after its core sound, and to pronounce a first-type vowel by uttering the word “สระ” (sound: */s-a1//r-a1/*) and then its core sound. In the second method, consonants are spelt by using only their core sounds, and first-type vowels are pronounced by their core sound without the word “สระ” (sound: */s-a1//r-a1/*). This spelling method is similar to the spelling approach in English. However, normally this method is slightly modified in order to cope with letter ambiguity. As mentioned above, some consonantal letters may share a same core sound. However, there will be exactly one letter, which is the most frequently used letter for each core sound, later called a representative letter. We will call the other letters with the same core sound subordinate letters. Table 5 indicates a set of core sounds with their representative letters and subordinate letters.

In the second spelling method, a representative letter is pronounced by its core sound while a subordinate letter is pronounced by its core sound followed by its representative word in order to differentiate which letter it is. In the third method, the way to pronounce a consonant and a vowel is varied. For instance, the word can be spelt out by spelling a consonant using only its core sound but spelling a vowel by pronouncing “สระ” (sound: */s-a1//r-a1/*) and then the vowel’s core sound. The last method is used to spell a set of letters that form a syllable and then follow with the corresponding pronunciation of that syllable. The spelling sequence of letters in each syllable starts with the initial consonant letter and is followed by the vowel letter, the final consonant letter (if any) and the tone symbol (if any), and then the sound of that syllable is inserted at the end of this sequence. As the initial stage, this paper concentrates on the first method, which is the prevalent spelling method.

Table 5. A set of core sounds with their representative letter and subordinate letters.

Core sound	Representative letter	Subordinate letters
/*kh-@@4*/	ข	ฃ
/*kh-@@0*/	ค	ฅ, ฆ
/*ch-@@0*/	ช	ฌ
/*j-@@0*/	ย	ญ
/*d-@@0*/	ด	ฎ
/*t-@@0*/	ต	ฏ
/*th-@@4*/	ถ	ฐ
/*th-@@0*/	ท	ฑ, ฒ, ธ
/*n-@@0*/	น	ณ
/*ph-@@0*/	พ	ภ
/*r-@@0*/	ร	ฤ
/*l-@@0*/	ล	ฬ
/*s-@@4*/	ส	ศ, ษ

4. The Implementation

As previously stated, this paper mainly contributes to Thai spelling recognition. In the initial stage, due to lack of a spelling corpus for training the system, a continuous speech corpus named the NECTEC-ATR [9], is used instead for training the acoustic part of the system. To evaluate the system, a set of spelling utterances of the first spelling style are collected to form the test set. The system is constructed using phone-based HMMs. Based on the current settings, there are two issues needed to be concerned. First, it is necessary to consider the difference between the set of phones in the NECTEC-ATR corpus (the training set) and the set of phones in spelling utterances (the test set). Fortunately, the former is a superset of the latter, making it possible to use the NECTEC-ATR in recognizing spelling utterances. The second issue is speed difference between training utterances (continuous speech) and test utterances (spelling speech). Normally, people spell a word with lower speed than normal conversation since they would like to make clear to the listener in what they want to say. To reveal this fact, a preliminary exploration can be done to measure the approximate speeds of training and test utterances in the form of the number of phones per second. To obtain these measurements, all utterances are automatically aligned, yielding the information of the phones and their durations. Based on this alignment information, the average speed of an utterance is calculated by subtracting silence and short-pause durations from the total utterance duration,

and then dividing the result by the number of phones except short-pause and silence phones in that utterance. To compensate for speed difference between the training utterances and the test utterances, a time-stretching method [10–12] is applied to stretch a speech signal with the preservation of pitch and auditory features of the original signal in our signal preprocessing. The basic concept in time-stretching is described by $s'(t) = s(\alpha t)$, where an original signal s at a time t can be transformed to time-stretching signal (s') with a scaling factor α. Here, $\alpha > 1.0$ means stretching and $\alpha < 1.0$ means compressing utterances. Among time-domain, frequency-domain and time-frequency techniques, we select the time-domain technique since it conserves the waveform for small stretching coefficient and easier adaptation to real-time (see details in [11]). This work explores two alternative approaches; stretching training utterances and compressing test utterances.

5. Experimental Results and Analysis

5.1. Experimental environment

In the experiments, a continuous speech corpus, named the NECTEC-ATR Thai speech corpus, is utilized as the training set. The corpus was constructed by the National Electronics and Computer Technology Center (NECTEC) in cooperation with the Spoken Language Translation Research Laboratories at the Advanced Telecommunication Research International Institute (ATR). Containing utterances of 390 sentences, it was gathered by assigning 42 subjects (21 males and 21 females) to read all sentences for one trial, obtaining a total of 16,380 read utterances. For the sake of implementation, only utterances of ten subjects (five males and five females) are used. To gain deeper insight into the performance comparison, the experiments are performed under three different environments of language models, closed-type, open-type and mix-type models. The closed-type language model, later denoted by LM1, is constructed from the test transcription, i.e. the 136 proper names. The open-type model, later denoted by LM3, is trained by another corpus, which is not used as the test transcription. In this experiment, we use 5,971 location names including Thai provinces, districts and sub-districts. The mix-type model, denoted by LM2, is generated from a corpus that includes both the test transcription and some other material text. In this experiment, we use the intersection of 136 proper names and 5,971 location names. Here, LM1 is the most restricted model that implies the environment of spelling recognition for a telephone directory assistance services while LM3 is the most relaxed model that implies a general environment. Due to the limitation of the corpus we used, LM2 seems the most natural environment, simulating that the corpus

Table 6. Phonetic units found in the spelling corpus and the NECTEC-ATR corpus.

	Type of speech corpus	
Type	**Spelling (73 phones)**	**NECTEC-ATR (195 phones)**
Initial consonant	*b, c, ch, d, f, h, j, k, kh, khw, l, m, n, ng, p, ph, pl, r, s, t, th, tr, w, z*	*b, bl, br, c, ch, d, dr, f, fl, fr, h, j, k, kh, khl, khr, khw, kl, kr, kw, l, m, n, ng, p, ph, phl, phr, pl, pr, r, s, t, th, thr, tr, w, z*
Vowel	*@@(0,4), a(0-4), aa(0-1,3-4), e1, ee(0-1), i(0-1,4), ii(0,4), o(0,3), oo(0,2), qq0, u(1,4), uu(0,2-4), uua3, v(1-3), vv0, vva(0,4), xxx(0,4)*	*@(0-4), @@(0-4), a(0-4), aa(0-4), e(0-4), ee(0-4), i(0-4), ia1, ii(0-4), iia(0-4), o(0-4), oo(0-4), q(0-3), qq(0-4), u(0-4), uu(0-4), uua(0-4), v(0-4), vv(0-4), vva(0,4), x(0-4), xx(0-4)*
Final consonant	*ch^, j^, k^, m^, n^, ng^, p^, t^, w^*	*ch^, f^, j^, jf^, k^, l^ m^, n^, ng^, p^, s^, t^, ts^ w^*

is large enough. In this work, two types of *n*-grams as the language model are compared.

As the recognition engine, phone-based HMMs are occupied with context-independent basic in the sense that the recognition of a phone in an utterance is independent of its preceding and following phones. It was observed that the set of phonetic units in the spelling corpus and that in the NECTEC-ATR corpus are not exactly identical. The former has fewer phones than the latter due to the limited number of possibilities in spelling utterances compared to normal utterances. Table 6 illustrates the list of phonetic units in each corpus. In the case of vowels, the number in a parenthesis denotes the possible tone expansions of the vowel. For example, "*a*(0-4)" means the vowel '*a*' occupies all five possible tones, that is 0 (mid), 1 (low), 2 (falling), 3 (high) and 4 (rising).

Following the standard evaluation, the recognition performance is evaluated in terms of correctness and accuracy. Since the task concerned is spelling recognition, not conventional speech or word recognition, the original definitions of word correctness and word accuracy are modified to letter correctness and letter accuracy as follows. The letter correctness (COR) is defined as the ratio of the number of correct letters to the total number of letters, i.e. *H/N*. Slightly different from the correctness, the letter accuracy (ACC) is the ratio of the subtraction of the number of correct letters by the number of letter insertion errors, to the total number of letters, i.e. (*H-I*)/*N*. Here, *H* is the number of correctly recognized letters, *I* is the number of inserted letters, and *N* is the total number of actual letters. It is clear that the accuracy is always lower than the correctness. The low accuracy compared to the correctness indicates that there are a lot of letter insertion errors. The details of correctness and accuracy can be found in [13].

Table 7. Recognition performance when the weight between acoustic and language models is varied (bigram as the language model).

Weight		1.00	0.20	0.10	0.05
LM1	COR	83.47	90.70	**93.08**	90.04
	ACC	73.87	88.86	**92.89**	88.35
LM2	COR	83.38	**86.22**	85.71	74.38
	ACC	73.28	84.17	**85.26**	73.92
LM3	COR	83.32	**85.74**	84.03	73.63
	ACC	72.79	83.06	**83.33**	73.07

5.2. Setting a baseline

The first experiment is to investigate spelling recognition using the original sets of training and test speech utterances without any modification, in order to set a baseline through this work. Utilizing the NECTEC-ATR continuous speech corpus as a training set, a HMM is constructed for each phone. All experiments are performed under the consideration of context-independent basis. This means that the recognition of a phone, in the acoustic level, does not depend on preceding or following phones of that phone. In the recognition process, two components that affect the result are acoustic and language models. A weight can be given to set the importance ratio between these two components. In this experiment, the weight is varied from 0.05 to 1.0 in order to find the most effective one. The smaller the weight is, the less important role the acoustic model plays, compared to the language model. Table 7 shows the results of various ratio weights when a bigram is applied as the language model. The bold number indicates the best correctness and accuracy for each language model. It was observed that the weight of 0.1 tended to achieve the best result for most cases. Therefore, if not specified, the succeeding experiments will be done based on the weight of 0.1. Unsurprisingly, the closed-type language model (LM1) achieves higher performance than the others, i.e., 93.08% correctness and 92.89% accuracy. The mix-type model (LM2) gains 85.71% correctness and 85.26% accuracy. Even with the hardest problem, the recognition performance of the open-type model (LM3) is comparative. In this environment, the correctness and accuracy are 84.03% and 83.33%, respectively.

5.3. Duration adjustment

Even at first glance, we observe a dominant difference between utterance speeds of the NECTEC-ATR corpus (training) and the spelling corpus (testing). To clarify

Table 8. Recognition performance when training utterances are stretched with various scaling factors (bigram as the language model).

Language Model		1.00Train	1.25Train	1.43Train	1.67Train
LM1	COR	93.08	93.92	91.79	84.39
	ACC	92.89	93.76	91.65	84.01
LM2	COR	85.71	87.37	85.37	77.47
	ACC	85.26	87.18	85.19	76.94
LM3	COR	84.03	85.75	83.84	76.03
	ACC	83.33	85.41	83.38	75.37

this, we measure the utterance speeds of both corpora in terms of the number of phones per second as shown in Section 4. As a result, the spelling utterances are approximately 1.527 times slower than the NECTEC-ATR utterances. To compensate for this duration difference, the time-domain stretching method [11] is occupied. In the experiment, the original speech signals are stretched with three scaling factors; 1.25, 1.43 and 1.67 times. They are denoted by 1.25Train, 1.43Train and 1.67Train. These three sets of stretched speech signals are used for training the system to recognize the spelling utterances. The results are compared with the system using the original speech utterance (1.00Train), the baseline, as shown in Table 8.

For all scaling factors, the close-type language model (LM1) gains the highest recognition rate while the open-type model (LM3) obtains the lowest one. In principle, stretching training utterances causes the original utterances to be distorted. The more an utterance is stretched, the more distorted utterance we obtain. As a result, stretching training utterances to 1.25 times of the original one yields the highest recognition rate while stretching them with 1.43 and 1.67 scaling factors causes the recognition rate to drop. The results show that 1.25Train gains higher correctness and accuracy for every language model. By this training set, the mix-type language model (LM2) obtains 87.37% correctness and 87.18% accuracy. They are improvements of 1.66% and 1.92%, respectively, compared to the baseline (1.00Train).

5.4. Investigating test utterances of each subject

The recognition result in Table 7 is the average performance gained from recognizing spelling utterances of the six subjects (three females and three males). To grasp an insight into this result, we investigate recognition performance for utterances of each individual subject. Table 9 displays the recognition performance as well as the spelling speed of each subject when the original utterances of

Table 9. Recognition performance of each subject's utterances and spelling speeds (bigram as the language model).

Subject	LM1		LM2		LM3		Speed (Phones/Sec.)
	COR	ACC	COR	ACC	COR	ACC	
FS1	94.51	94.44	87.12	86.84	85.50	85.15	6.72
FS2	94.23	94.09	86.00	85.86	84.45	84.17	6.87
FS3	87.84	86.77	81.98	80.23	80.93	78.54	5.07
MS1	93.67	93.67	85.57	85.22	84.52	84.24	6.51
MS2	93.31	93.24	86.14	86.60	83.74	83.32	5.59
MS3	95.29	95.14	87.47	87.40	85.01	84.59	6.45

training and test sets are used. Here, FS1, FS2, FS3 are female utterances and MS1, MS2, MS3 are male utterances.

It was observed that the six subjects spelled words with different speeds. The spelling utterances made by the subject FS3 are the slowest ones with 5.07 phones per second on average. Note that the utterances in the NECTEC-ATR continuous speech corpus, which we applied as the training set, are 9.47 phones per second on average. Reflecting this figure, we obtain the lowest correctness and accuracy for the recognition of utterances made by FS3. As further investigation, two additional experiments are performed. One is to stretch the training speech (NECTEC-ATR utterances) with a higher scaling factor. The other is to compress the test speech (spelling utterances) to examine the improvement of recognition performance. The recognition result of using stretched speech as the training set is shown in Table 10.

Unlike the previous experiment, instead of 1.25Train, the 1.43Train achieves the highest recognition rate while stretching with the scaling factor of 1.67 causes the recognition to drop down. The result shows that 1.43Train with the mix-type language model (LM2) gain up to 87.33% correctness and 86.70% accuracy, which results in the improvement of 5.35% and 6.47% over the baseline, respectively, compared to the baseline. After adjusting the speed of FS3 utterances, we can expect that the speed of these test utterances becomes compatible with the speed of the training utterances from the NECTEC-ATR corpus. Then the recognition can be improved. Since the speed utterance of FS3 is slower than the other subjects, the suitable scaling factor for FS3 is larger than those for the other subjects.

Table 11 shows the recognition result of the time-compressed spelling speech when the scaling factor is varied. Here, the original training utterances are used for training. Similar to the case of stretching the training utterances, compressing the FS3's utterances with factors of 0.6, 0.7 and 0.8, yields better

Table 10. Recognition performance of FS3's spelling utterances when the training utterances are stretched with various scaling factors (bigram as the language model).

Language Model		1.00Train	1.25Train	1.43Train	1.67Train
LM1	COR	87.84	91.77	92.75	90.29
	ACC	86.77	91.34	92.19	89.23
LM2	COR	81.98	86.14	87.33	84.80
	ACC	80.23	85.43	86.70	83.81
LM3	COR	80.93	84.80	85.86	82.97
	ACC	78.54	83.81	84.24	81.28

Table 11. Recognition performance of FS3's compressed spelling utterances when the original training utterances are applied (bigram as the language model).

Language Model		1.00Test	0.80Test	0.70Test	0.60Test
LM1	COR	87.84	92.12	92.89	92.26
	ACC	86.77	91.77	92.61	92.05
LM2	COR	81.98	85.38	86.42	84.38
	ACC	80.23	84.38	85.86	84.10
LM3	COR	80.93	83.95	83.81	83.53
	ACC	78.54	83.11	83.32	83.11

recognition rates. The 0.70Test achieves higher correctness and accuracy than the 0.80Test and 0.60Test for most cases. For the mix-type language model, the improvement using 0.70Test is around 4.44% for correctness and 5.63% for accuracy, compared to the original test utterances (1.00Test).

5.5. Exploiting trigram language model

In this experiment, first we calculate the perplexity of a language model against unseen test data in order to evaluate how predictive the model is. Related to entropy, perplexity indicates the level of ambiguity. It is defined as the exponential form of entropy, i.e. $PP = 2^H$, where H is the entropy defined by a limit of the summation of $p(w_1w_2\ldots w_m) \times \log p(w_1w_2\ldots w_m)$, and $w_1w_2\ldots w_m$ is a letter sequence. More details can be found in [13]. Low perplexity of a language model means that the model is more predictive. In speech recognition, a language model with low perplexity on the test data tends to achieve better recognition performance, even not guaranteed [14]. In our corpus settings, the perplexity of the bigram model is calculated with results of 25.41 and 23.96 for the mix-type and the open-type environment, respectively. They are 12.80 and 18.71 for the trigram

Table 12. Recognition performance when the trigram model is applied (comparing the original corpus to the 1.25-times stretched corpus).

Model		Type of training speech corpus	
		1.00Train	1.25Train
Tri-LM1	COR	93.15	93.19
	ACC	92.82	93.08
Tri-LM2	COR	90.50	91.12
	ACC	89.75	90.80
Tri-LM3	COR	84.10	84.65
	ACC	82.77	83.92

model. This implies that the trigram model is more predictive than the bigram model and should gain better performance. Moreover, normally the perplexity of the closed-type environment cannot be calculated since the models cover all the test data. We also investigate how the trigram model performs in spelling recognition and compare it to the bigram model. The models are explored in the three environments; closed-type (Tri-LM1), mix-type (Tri-LM2), and open-type (Tri-LM3). Table 12 shows the recognition rates of two different training corpora; the original NECTEC-ATR (1.00Train) and the 1.25-times stretched NECTEC-ATR (1.25Train).

The table indicates the results obtained from two different training corpora; the original NECTEC-ATR corpus (1.00Train) and the 1.25-times-stretched NECTEC-ATR (1.25Train). The result indicates that the trigram model achieves higher performance than the bigram model in the mix-type and the open-type environments while it is not helpful in the closed-type environment (see Table 8). The correctness improvements of the trigram over the bigram in the mix-type environment are 4.79% and 3.75% for the 1.00Train and the 1.25Train, respectively.

6. Error Analysis

For the closed-type, mix-type and open-type language models, the spelling recognition results are quite straightforward. The highest recognition rates are obtained in the closed-type environment where the system has already known the 136 test names. For the open-type environment, the language model is trained by more than 5,000 names excluding the test set and the recognition rates are the worst. In this work, we focus on the mix-type language model (LM2 and Tri-LM2), which are trained by 6,107 names including the test names. In this section, we consider four experiments on the bigram and trigram models based

Table 13. Experiments and their details in error analysis.

Experiment	Abbr.	Language model	Training corpus
ATR-bigram	AB	LM2	original NECTEC-ATR
ATR-trigram	AT	Tri-LM2	original NECTEC-ATR
1.25ATR-bigram	1.25AB	LM2	1.25-times stretched NECTEC-ATR
1.25ATR-trigram	1.25AT	Tri-LM2	1.25-times stretched NECTEC-ATR

on the mix-type environment. The details of these experiments are shown in Table 13.

As a consequence of setting the weight ratio between the acoustic model and the language model to be a low value, forcing the language model to be more important than the acoustic model, the insertion errors are dominantly reduced. By this setting, the main errors are primarily substitution errors. There are 48 alphabets from 66 Thai letters used in the test set. The numbers of each letter vary from 12 to 528. The substitution errors of these 48 letters are investigated. By using dynamic programming matching (DP matching), we can draw out the substitution errors of each letter from each experiment. To focus on the major errors of each letter, the letters that appear in the test set less than 15 times are eliminated and the letters that gain low recognition rate with substitution error greater than or equal to 50% in each experiment, called *erroneous letters,* are discussed. The erroneous letters with their percentage of substitution in four experiments are shown in Table 14, with the order in the sequence of Thai alphabets. There are 8 erroneous letters that are 5 consonants (‘ป’, ‘ฟ’, ‘ฤ’, ‘อ’, ‘ฮ’) and 3 vowels (‘อะ’, ‘อิ’, ‘อุ’). Although the percentage of substitution occurrence of the vowel “ อิ ” is less than 50%, we still consider it as the erroneous letter. This is because the percentage of substitution occurrence of this vowel is almost 49.07% and there are quite a large amount of this vowel appearing in our test set. Related to the results in Table 14, Table 15 shows the set of substitutions letters for each erroneous letter in each experiment. For instance, ATR-bigram experiment (AB) has 7 erroneous letters (‘ป’, ‘ฟ’, ‘ฤ’, ‘อ’, ‘ฮ’, ‘อะ’, ‘อุ’), where their percentages of substitution occurrence are greater than or equal to 50%.

From these two tables, the following conclusions can be made. First, the numbers of erroneous letters are reduced when the trigram is applied instead of the bigram. For instance, ATR-bigram (AB) has 7 erroneous letters, but ATR-Trigram (AT) has merely one erroneous letter. Second, by using the stretched NECTEC-ATR corpus as the training set, the recognition of vowels is obviously improved, compared with the original corpus. This result matches with the

Table 14. List of erroneous letters and their percentages of substitution errors.

Err letter	Phonetic sound	Total Amt	% Substitution occurrence			
			AB	AT	1.25AB	1.25AT
ป	*/p-@@0//pl-aa0/*	90	**50.00**	26.67	**57.78**	30.00
ฟ	*/f-@@0//f-a-n^/*	18	**72.22**	**61.11**	**77.78**	**72.22**
ฤ	*/r-@@0//r-v3/*	18	**100.0**	5.56	**77.78**	11.11
อ	*/z-@@0//z-aa1-ng^/*	162	**63.58**	40.12	**56.79**	35.19
ฮ	*/h-@@0//n-o3-k^//h-uu2/*	30	**56.67**	13.33	**60.00**	20.00
อะ	*/s-a1//r-a1//z-a1/*	54	**55.56**	40.74	33.33	18.52
อิ	*/s-a1//r-a1//z-i1/*	432	49.07	19.68	23.61	10.88
อุ	*/s-a1//r-a1//z-u1/*	138	**75.36**	28.99	24.64	11.59

Table 15. List of substituted letters for each erroneous letter.

Err letter	Substituted Letters			
	AB	AT	1.25AB	1.25AT
ป	ค (*/kh-@@0//kw-aa0-j^/*) จ (*/c-@@0//c-aa0-n^*) ช (*/ch-@@0//ch-aa3-ng^/*) พ (*/ph-@@0//ph-aa0-n^/*) ม (*/m-@@0//m-aa3/*) อ (*/z-@@0//z-aa1-ng^/*)		ค (*/kh-@@0//kw-aa0-j^/*) จ (*/c-@@0//c-aa0-n^*) พ (*/ph-@@0//ph-aa0-n^/*) อ (*/z-@@0//z-aa1-ng^/*)	
ฟ	พ (*/ph-@@0//ph-aa0-n^/*)	พ	พ (*/ph-@@0//ph-aa0-n^/*)	พ
ฤ	ร (*/r-@@0//r-vva0/*)		ร (*/r-@@0//r-vva0/*)	
อ	ค (*/kh-@@0//kw-aa0-j^/*) จ (*/c-@@0//c-aa0-n^*) ช (*/ch-@@0//ch-aa3-ng^/*) ป (*/p-@@0//pl-aa0/*) พ (*/ph-@@0//ph-aa0-n^/*) ม (*/m-@@0//m-aa3/*)		ค (*/kh-@@0//kw-aa0-j^/*) ช (*/ch-@@0//ch-aa3-ng^/*) ป (*/p-@@0//pl-aa0/*) พ (*/ph-@@0//ph-aa0-n^/*)	
ฮ	ง (*/ng-@@0//ng-uu0/*) น (*/n-@@0//n-uu4/*)		ง (*/ng-@@0//ng-uu0/*) น (*/n-@@0//n-uu4/*)	
อะ	อา (*/s-a1//r-a1//z-aa0/*)			
อิ	อี (*/s-a1//r-a1//z-ii0/*)			
อุ	อู (*/s-a1//r-a1//z-uu01/*)			

intuition that recognition of vowels is sensitive to duration since in Thai there are short vowels and their long vowels for the same phonetic sound. They have same phones but different durations. Therefore, after adjusting the durations of the training utterances to be consistent with the durations of the test utterances, the recognition of these vowels is improved. Third, letters that substitute each erroneous letter are similar even in different environments. For example, the letter 'ป' is mostly substituted by {'ค', 'จ', 'ช', 'พ', 'อ', and 'ม'}, the letter 'ฟ' is always substituted by 'พ', the letter 'ฤ' is substituted by 'ร', the letter 'อ' can be substituted by {'ค', 'จ', 'ช', 'พ', 'ป', and 'ม'}, and the letter 'ฮ' is substituted by 'ง' or 'น'. In case of the letter 'ป' (sound: */p-@@0//pl-aa0/*) and 'อ' (sound: */z-@@0//z-aa1-ng^/*), we can define these letters and their substituting letters as one mixed set {'อ', 'ป', 'ม', 'พ', 'ช', and 'จ'}. One potential cause of these substitution errors in this set is that these letters are pronounced with two syllables and they share the same vowel phone @@ in the first syllable and the same vowel phone *aa* in the second syllable. For the letter 'ฟ' (sound: */f-@@0//f-a-n^/*), which is always substituted by 'พ' (sound: */ph-@@0//ph-aa-n^/*), can be explained by two reasons: (1) the phone *f* is quite similar to *ph*, (2) they share the phone @@ in the first syllable and get mixed between short vowel (*/-a-n^/*) and long vowel (*/-aa-n^/*) in the second syllable. In case of 'ฤ' (sound: */r-@@0//r-v3/*) and 'ร' (sound: */r-@@0//r-vva0/*), they share the same phone */r/*. For the letter 'ฮ' (sound: */h-@@0//n-o3-k^//h-uu2-k^/*), which can be substituted by the letter 'ง' (sound: */ng-@@0//ng-uu0/*) and 'น' (sound: */n-@@0//n-uu4/*). There are two potential reasons: (1) they share the same phone /@@/ in the first syllable and */uu/* in the last syllable and (2) the phones *h*, *ng* and *n* are similar because they are all nasal phones.

Another substitution error is caused by the confusion in the duration of a vowel pair. The vowel letter 'อะ' (sound: */s-a1//r-a1//z-a1/*) is often misrecognized by its corresponding long vowel 'อา' (*/s-a1//r-a1//z-aa0/*). The vowel alphabet 'อิ' (sound: */s-a1//r-a1//z-i1/*) is usually recognized by its corresponding long vowel pair 'อี' (sound: */s-a1//r-a1//z-ii0*). In the same way, the vowel alphabet 'อุ' (sound: */s-a1//r-a1//z-u1/*) is generally recognized by 'อู' (sound: */s-a1//r-a1//z-uu0*). After compensating the duration difference between training and test utterances by stretching training utterances to be 1.25Train, these substitution errors are dominantly reduced.

7. Conclusion

This paper presented a general framework of Thai speech recognition enhanced with spelling recognition. We also gave an analytical introduction to four styles of

spelling Thai words. An HMM-based method was proposed to recognize spelling utterances of the first spelling style using an existing continuous speech corpus as the training speech. To find the optimal condition for recognition, the weighting ratio between acoustic and language models was also explored. The best ratio was used later for all experiments. Focused on utterance speed difference between spelling utterance and continuous speech utterance, a speed compensation method was applied to improve recognition performance. A number of experiments were made to examine various time-stretching factors of the original continuous speech that were used for training utterances. As the result, training with the 1.25-times stretched utterances achieved the best accuracy and correctness. Moreover, the experiments were done under three different environments; closed-type, open-type and mix-type language models. Unsurprisingly, the closed-type model achieves the highest performance while the open-type one gains the lowest one. The result of the mix-type model was close to that of the closed-type one. The result of the mix-type model indicated a promising performance of 87.37% correctness and 87.18% recognition accuracy when the 1.25-times stretched utterances are used as the training speech. They are 1.66% and 1.92% improvement over the baseline for correctness and accuracy, respectively. As a further investigation, we focused on utterances belonging to the one whose speech is dominantly different from the others in terms of spelling speed. The result showed that the optimal time-stretching factor was 1.43. Moreover, for this case, exploiting the original training utterances but compressing the spelling utterances instead is also explored. As a result, the system achieved up to 5.35% correctness and 6.47% accuracy improvement over the baseline. Instead of the bigram model, the trigram model was also investigated as the language model. With small perplexity, the trigrams can improve the recognition rate over the bigrams of the mix-type environment when using the original NECTEC-ATR by 4.79% correctness improvement. Finally, an analysis of recognition errors was made to investigate the cause of common substitutions. For our future works, we plan (1) to construct a system that can recognize several kinds of spelling methods, (2) to construct a corpus for spelling recognition purpose, and (3) to explore a way to incorporate spelling recognition into conventional speech recognition.

Acknowledgments

The authors would like to thank National Electronics and Computer Technology Center (NECTEC) for allowing us to use the NECTEC-ATR Thai Speech Corpus. This work has partly been supported by NECTEC under project number

NT-B-22-I5-38-47-04. The first author also would like to thank Phuket Community College, Prince of Songkla University for their support and allowance.

References

[1] R. San-Segundo, J. Macias-Guarasa, J. Ferreiros, P. Martin and J. M. Pardo, Detection of recognition errors and out of the spelling dictionary names in a spelled name recognizer for Spanish, in *Proceedings of EUROSPEECH 2001*, 2001.

[2] R. San-Segundo, J. Colas, R. Cordoba and J. M. Pardo, Spanish recognizer of continuously spelled names over the telephone, *Journal of Speech Communication*, 38, 2002, 287–303.

[3] F. Rodrigues, R. Rodrigues and C. Martins, An isolated letter recognizer for proper name identification over the telephone, in *Proceedings of 9th Portuguese Conference on Pattern Recognition*, 1997.

[4] C. D. Mitchell and A. R. Setlur, Improved spelling recognition using a tree-based fast lexical match, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* 2, 1999, 597–600.

[5] J. G. Bauer and J. Junkawitsch, Accurate recognition of city names with spelling as a fallback strategy, in *Proceedings of EUROSPEECH 1999,* 1999, 263-266.

[6] C. Pisarn and T. Theeramunkong, Incorporating tone information to improve Thai continuous speech recognition, in *Proceedings of International Conference on Intelligent Technologies,* 2003, 84–89.

[7] *Tellvoice Technology*, http://www.tellvoice.com.

[8] C. Pisarn and T. Theeramunkong, *Thai Continuous Speech: The Technical Report*, SIIT, 2004.

[9] S. Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, T. Jitsuhiro, G. Kikui and Y. Sagisaka, NECTEC-ATR Thai speech corpus, in *Proceedings of The Oriental COCOSDA 2003*, October 1–3, 2003.

[10] W. Verhelst and M. Roelands, An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech, in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* 2, 1993, 554–557.

[11] G. Pallone, Time-stretching and pitch-shifting of audio signals: Application to cinema/video conversion, http://www.iua.upf.es/activitats/semirec/semi-pallone/index.htm.

[12] *Wikipedia: The Free Encyclopedia*, Audio time stretching, http://www.ebroadcast.com.au/lookup/encyclopedia/au/Audio_time_stretching.html.

[13] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, *The HTK Book (for HTK Version 3.2.1)*, (Cambridge University Engineering Department, 2002).
[14] P. Taylor, R. Caley, A.W. Black and S. King, *Edinburgh Speech Tools Library System Documentation* (*Edition 1.2*) *for 1.2.0*, (1999) http://festvox.org/docs/speech_tools-1.2.0/x2921.htm.