COMBINING HIDDEN MARKOV MODELS AND PHONETIC TRIGRAMS IN A THAI SOUNDEX SYSTEM

Tassawut Duangpanyasawang and Boonserm Kijsirikul
Machine Intelligence and Knowledge Discovery Laboratory
Department of Computer Engineering,
Chulalongkorn University,
Bangkok 10330, THAILAND
Email: {tassawut, boonserm}@mind.cp.eng.chula.ac.th

ABSTRACT

Most of previous methods on Thai soundex encoding are based on Odell and Russell's algorithm. These methods still have limitations in grouping words of similar sound. To improve the accuracy of Thai soundex encoding, we propose a new method that combines a Hidden Markov Model (HMM) and phonetic trigrams. Our soundex code consists of three features, i.e. initial consonant sound, vowel sound, and final consonant sound. The HMM and phonetic trigrams are used as a soundex converter. First, the HMM generates all possible soundex codes, and then the answers are re-ranked by combining probabilities from phonetic trigrams with probabilities from the HMM. The experimental results show that our algorithm surpasses the previous works and gives the best performance with 95% and 84% in precision and recall, respectively.

INTRODUCTION

Soundex is a technique for identifying words that have similar pronunciation, and is used in many applications such as word retrieval in census database [Green], spelling correction [Arunwong-Na-Ayuthaya, 1991; Tangkhawanwanich, 1991], cross-language retrieval [Suwanvisat & Prasitjutrakul, 1999], etc. For English, Odell and Russell's algorithm provides a good result of grouping words based on their sounds. The algorithm translates a string into a canonical form of at most four characters by ignoring vowels and 'w', 'h', 'y' and converting the rests into a phonetic code consisting of the first letter and three decimal digits.

For Thai soundex systems, there are two board classes of approaches: (1) a soundex converter using a nondeterministic finite automaton (NFA) [Karoonboonyanan, et al., 1997], and (2) soundex schemes based on Odell and Russell's algorithm. The NFA soundex converter generates all possible pronunciations but does not provide probabilities for ranking them. For this reason, the NFA soundex converter is impractical in most applications, because it cannot select the most probable answer.

Most of Thai soundex systems are based on Odell and Russell's algorithm [Lohjeerachoonhakul & Khuwinpan, 1982; Udompanich, 1983; Arunwong-Na-Ayuthaya, 1991; Suwanvisat & Prasitjutrakul, 1999]. The works of [Lohjeerachoonhakul & Khuwinpan, 1982] and [Udompanich, 1983] are almost the same as the original Odell and Russell's algorithm, i.e. they preserve the first character of the input word and convert the rest into decimal digits by ignoring vowels. The works of [Arunwong-Na-Ayuthaya, 1991; Suwanvisat & Prasitjutrakul, 1999] proposed improved versions of Odell and Russell's algorithm for Thai soundex systems by encoding both consonants and vowels into a phonetic code, but these systems still did not perform quite well. The following are specific characteristics of Thai that are overlooked in previous works.

(1) As soundex can be though of as a phonetic code of an input string, the smallest unit in the soundex code should encode each syllable of the input string. However, some Thai soundex systems did not take this into account, i.e. they ignored vowels, and thus may produce the same code for different words despite the fact that the words contain different syllables. For example, the word 'จีวร' [tɕiːwɔn] having two syllables and the word 'จร' [tɕɔːn] composed of only one syllable are encoded into the same soundex code [Lohjeerachoonhakul & Khuwinpan, 1982; Udompanich, 1983]. Thai soundex systems, which consider both consonants and vowels, still face the problem of Thai syllables [Arunwong-Na-Ayuthaya, 1991; Suwanvisat & Prasitjutrakul, 1999]. This is because usually a Thai string can be segmented into many different sequences of syllables, but the previous Thai soundex systems use only one sequence of syllables in the original form.

(2) There are special vowel structures which give the same pronunciation but are encoded into different soundex codes, such as vowels in the words 'เกา' [kau] and 'กาว' [kau] [Lohjeerachoonhakul & Khuwinpan, 1982; Arunwong-Na-Ayuthaya, 1991]. Generally, the sounds of vowels 'เ-า' [–au] and 'า' [-aː-] are different, but the combination of the vowel 'า' [-aː-] with a final consonant 'ว' [-u] provides the same sound as that of the vowel 'เ-า' [–au].

(3) There are many ambiguous pronunciation rules in Thai that cause low precision in the previous works. Vowels 'เ' [-eː-], 'แ' [-ɛː-], 'ไ' [-ai], 'ใ' [-ai], 'โ' [-oː-] can provide more than one pronunciation. For example, a word 'เกษม' may be read as 'เก-ษม' [keːsoːm] or 'กะ- เษม' [kaseːm], and in this case the correct one is 'กะ-เษม' [kaseːm]. However, a word 'เกษร' which can also be read as 'เก-ษร' [keːsoːn] and 'กะ-เษร' [kaseːn], but the correct one is 'เก-ษร' [keːsoːn].

(4) The *reduced form* of words is also an important cause of a limitation of Thai soundex systems. For instance, words such as 'วิทยา' [wittajaː] and 'อัตรา' [ʔàttraː] contain hidden pronunciation that need to be read as if they were 'วิท**ทะ**ยา' [wittajaː] and 'อั**ตร**ตรา' [ʔàttraː], respectively.

In this paper, we propose a new method for Thai soundex encoding that can solve the previous limitations. First, to solve the problem of syllable segmentation, we segment an input string into all possible sequences of syllables. Next, we handle the problem of special vowel structures and the ambiguous in pronunciation rules by applying a Hidden Markov Model (HMM) and phonetic trigrams to generate all possible pronunciations and select the most probable one. Finally, to solve the problem of the reduced form of words, we employ a preprocessing method that adds the hidden pronunciation back into the words.

To evaluate the effectiveness of our method, we construct a corpus consisting of two parts: the first one is for training and the other is for testing. The experimental results show that a combination of HMM and phonetic trigrams provides high precision and recall and our algorithm outperforms the previous works.

COMBINING AN HMM AND PHONETIC TRIGRAMS

Below we describe our method for soundex encoding. We first give the definition of soundex, then explain the algorithm which employs an HMM and phonetic trigrams.

THE ENCODING OF SOUNDEX

We employ the encoding scheme in [Karoonboonyanan, et al., 1997] which defines the properties of *similarly-pronounced syllables* as being composed of two main features, i.e. (1) *common features* consisting of initial consonant sounds, vowel sounds and final consonant sounds, and (2) *variations* consisting of clusters, vowel lengths and tones. Soundex encoding should preserve common features and eliminate variations. Therefore, a unit in the soundex code is composed of three letters representing an initial consonant sound, a vowel sound, and a final consonant sound. In Thai, there are 20 initial consonant sounds, 12 vowel sounds, and 9 final consonant sounds as shown in Table 1,2,3 respectively.

Table 1: Initial consonant encoding.

Consonant(s)	Phonetic Values	Soundex Code
ก กล กร กว	[k]	ก
ข ค ฆ คล คร คว ขว ขร ขล	[kh]	ค
ง หง	[ŋ]	ง
จ จร	[tɕ]	จ
ช ชร ฌ ฉ	[tɕh]	ช
ส ซ ษ ศ ศร ศล สร สล ทร ซร	[s]	ซ
ย อย หญ ญ หย	[j]	ย
ด ดร ฎ ฑ	[d]	ด
ต ตร ตล ฏ	[t]	ต
ท ทร ฐ ถ ฑ ฒ ธ	[th]	ท
น ณ หน	[n]	น
บ บร บล	[b]	บ
ป ปร ปล	[p]	ป
พ พร พล ผ ผล ภ	[ph]	พ
ฝ ฝล ฝร ฟ ฟล ฟร	[f]	ฟ
ม หม	[m]	ม
ล ร ฬ หล หร ฤ	[r],[l]	ล
ว หว	[w]	ว
ห ฮ	[h]	ฮ
อ	[ʔ]	อ

* We group 'ร' (thrill [r]) and 'ล' (lateral [l]) together due to their close pronunciation.

Table 2: Vowel encoding.

Syllable Form	Soundex Code
อะ[-a] อั-[-a-] อรร-[-a-] อ[-a] อำ[-am] ใอ[-ai] ไอ[-ai] ไอย[-ai] เอา[-au] อา-[-aː-]	a
อิ-[-i-] อี-[-iː-] -ฤ-[-ri-]	i
อึ-[-ɯ-] อื-[-ɯː-] -ฤ-[-rɯ-]	v
อุ-[-u-] อู-[-uː-]	u

เอะ[-e] เอ็-[-e-] เอ-[-eː-]	e
แอะ[-ɛ] แอ็-[-ɛ-] แอ-[-ɛː-]	x
โอะ[-o] อ-[-o-] โอ-[-oː-]	o
เอาะ[-ɔ] อ็อ-[-ɔ-] ออ-[-ɔː-] อ[-ɔː] อร[-ɔːn]	c
เออะ[-ɣ] เออ-[-ɣː-] เอิ-[-ɣː-] เอย[-ɣːi] ฤ-[rɣː-]	d
เอียะ[-iə] เอีย-[-iə-]	j
เอือะ[-ɯə-] เอือ-[-ɯːə-]	w
อัวะ[-uə] อัว[-uːə] อว-[-uːə-]	$

Table 3: Final consonant encoding.

Group	Consonant(s)	Soundex Code
kok [-k]	ก ข ค กร คร	ก
kong [-ŋ]	ง	ง
kom [-m]	ม มิ	ม
kon [-n]	น ณ ร ญ ล ฬ	น
koew [-u]	ว	ว
koey [-I]	ย	ย
kot [t]	จ ช ซ ฎ ฏ ฐ ฑ ฒ ด ต ถ ท ธ ส ศ ษ ษฐ ตร ชร ทร ติ ตุ ทธิ ฒิ ทธ รถ	ต
kop [-p]	บ ป พ ฟ ภ	บ
Open syllable	ฮ	ฮ

With the above classification and all variations elimination applied, here are some encoding examples:

ไน [nai]	is encoded นaย
อัตรา [ʔàttraː]	is encoded อaดตaฮ
ธรรมะ [thammáʔ]	is encoded ทaมมaฮ

OUR ALGORITHM

We now describe our algorithm for converting a string into its soundex code. The algorithm is shown in Table 4.

Table 4: The algorithm for soundex encoding.

1. Preprocess the input word by:
    1.1 eliminating tones and cancelled letters,
    1.2 recovering the hidden form, and
    1.3 creating new words for special vowels.
2. Segment each word into all possible sequences of syllables.
3. Apply a Hidden Markov Model (HMM) and phonetic trigrams to generate soundex codes, and output N-best answers according to the probabilities calculated from the HMM and phonetic trigrams.

As shown in Table 4, the algorithm for generating soundex codes is composed of three steps.

1. The first step is to preprocess the input word by eliminating tone markers and cancelled letters. We also solve the problem of the reduced forms of words by generating new words from the input word with hidden-pronunciation adding. For example, the preprocessing of 'วิทยา' [wittajaː] will generate two new words: 'วิทยา' [wittajaː] itself and 'วิททะยา' [wittajaː] with the hidden-pronunciation (ทะ) added. Furthermore, the ambiguous pronunciation of special vowels 'เ' [-eː-], 'แ' [-ɛː-], 'ไ' [-ai], 'ใ' [-ai], 'โ' [-oː-] is also solved in this step. For an input word containing these vowels, we generate all possible pronunciations for these vowels. For example, for an input word 'เกษม' [kaseːm], the algorithm will produce the word 'เกษม' [kaseːm] itself and another word 'กเษม' [kaseːm]. Therefore, the preprocessing of 'เกษมวิทยา' will provide four outputs: 'กเษมวิทยา', 'กเษมวิททะยา', 'เกษมวิทยา' and 'เกษมวิททะยา'.

2. The second step is to segment the words from Step 1 into all possible sequences of syllables. In other words, given a string of characters $S = c_1c_2 \cdots c_n$ from Step 1, we want to find a set $W = \{ o_1o_2 \cdots o_m \mid o_1, o_2, \ldots, o_m \in O$ and $o_1o_2 \cdots o_m = S \}$, where O is a set of valid Thai syllable forms.

3. The last step is to apply an HMM and phonetic trigrams to encode soundex for each sequence of syllables, and select the N-best answers according to the probabilities calculated from the HMM and phonetic trigrams. The HMM and phonetic trigrams are described below.

HIDDEN MARKOV MODEL

An HMM is a well-known and widely used statistical method, especially for speech recognition tasks [Rabiner & Juang, 1993]. Here, we apply an HMM as a soundex converter, by defining each state as a soundex code and outputs of each state as all syllable forms which have pronunciation matching with the state. A transition between two states is defined as the probability that two soundex codes are connected to each other. Figure 1 shows a part of our HMM.

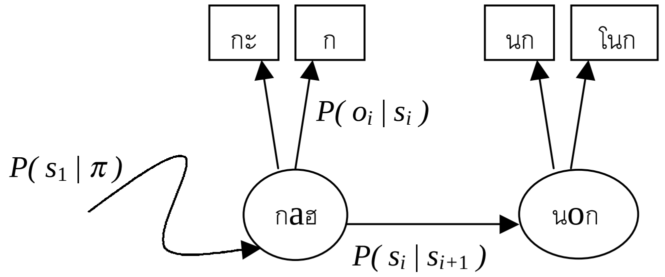

Figure 1: A part of our HMM.

States 'กaฮ' and 'นoก' in the figure represent soundex codes pronounced as 'กะ [ka]' and 'นก [nok]', respectively. The outputs of the state 'กaฮ' are the syllable forms ('กะ' and 'ก') of which pronunciation are 'กะ [ka]'. Similarly, 'นก' and 'โนก', which are outputs of the state 'นoก', have the same pronunciation 'นก [nok]'. The transition between the states 'กaฮ' and 'นoก' is a probability that the state 'กaฮ' is followed by the state 'นoก'.

Each soundex code contains three characters representing an initial consonant sound, a vowel sound, and a final consonant sound. Since there are 20 initial consonant sounds, 12 vowel sounds and 9 final consonant sounds, the total number of states is 20 x 12 x 9 = 2,160. Therefore, the total number of transitions is 2,160 × 2,160 = 4,665,600, in case of a fully-connected HMM. However, the transitions in our model are constructed from training data (not fully-connected). In our experiment, the model contains 10,896 transitions by the end of the training process. We create outputs of each state by combining all initial consonant forms, vowel forms, and final consonant forms shown in Table 1. For example, outputs of a state ‘งUง’ [ŋuːŋ] are ‘งูง’ [ŋuːŋ], ‘หงูง’ [ŋuːŋ], ‘งุง’ [ŋuːŋ], and ‘หงุง’ [ŋuːŋ], since there are 2 initial consonant forms ‘ง’ [ŋ] and ‘หง’ [ŋ], 2 vowel forms ‘ ุ ’ [-u-] and ‘ ู ’ [-uː-], and 1 final consonant forms ‘ง’ [ŋ].

To apply the HMM for soundex encoding, we define the task as follows:

Given:

- a Hidden Markov Model $H = \{\pi, T, O\}$
- a sequence of syllable forms $X = o_1 o_2 \cdots o_m$

where π is a set of states, T is a set of transitions, and O is a set of outputs of states, and $o_1, o_2, \ldots, o_m \in O$.

Find:

- the state sequence $S = s_1 s_2 \cdots s_m$ (the soundex code of the given X) that maximizes $P(S \mid X)$ by using the following equation:

$$P(S \mid X) = P(s_1 \mid \pi) \prod_{i} P(s_i \mid s_{i+1}) P(o_i \mid s_i) \tag{1}$$

where $P(s_1 \mid \pi)$ is a probability that state s_1 is chosen as an initial state,
$P(s_i \mid s_{i+1})$ is a transition probability between state s_i and s_{i+1},
$P(o_i \mid s_i)$ is a probability that output symbol o_i will be emitted at state s_i.

Note that all these probabilities are calculated from training examples.

TRIGRAM MODEL

We employ *phonetic trigrams* for enhancing the performance of our method. Though various trigram models, e.g. word trigrams, character trigrams, have been successfully applied to many tasks [Golding & Schabes, 1996; Meknavin, et al., 1998], to our knowledge this is the first attempt to use a phonetic trigram model. To apply a phonetic trigrams to select the most probable answer, we define the task as follows. Given a state sequence $S = s_1 s_2 \cdots s_m$ from the HMM, the probability of this sequence can be calculated by using the phonetic trigrams defined as follows:

$$P(S) = \prod_{i} P(s_i \mid s_{i+1}, s_{i+2}) \tag{2}$$

where s_i, s_{i+1} and s_{i+2} are a state in state sequence S, the state next to s_i and the state next to s_{i+1}, respectively.

We then define the combined probability (CP) that combines the phonetic trigrams with the HMM by using Equation (1) and Equation (2) as:

$$CP = P(s_1 \mid \pi) \prod_{i} P(s_i \mid s_{i+1}) P(o_i \mid s_i) \times \prod_{i} k P(s_i \mid s_{i+1}, s_{i+2}) \tag{3}$$

where *k* is a constant that determines the relative values of trigrams's probabilities versus HMM's probabilities.

EXPERIMENTAL RESULTS

To evaluate the effectiveness of our proposed method, we construct a corpus which contains about 26,000 words (100,000 syllables) from the databases of Telephone Organization of Thailand and Thai Royal Institute dictionary. About 70% of the whole corpus are used as a training set and the rest is used as a test set. We run the experiment to compare three algorithms: (1) an algorithm described in [Suwanvisat & Prasitjutrakul, 1999] which is based on Odell and Russell's Soundex algorithm (SP99), (2) the HMM alone (HMM), and (3) the combination of the HMM and phonetic trigrams (HMM+TRI). The results are shown in Table 4.

The performances of the above methods are evaluated by standard precision (P) and recall (R). We also use F_1-*measure*(F1) that combines recall and precision with an equal weight and is defined as follows:

$$F1 = \frac{2PR}{P+R}$$

Table 4. The comparison of the performances of three algorithms.

SP99			HMM			HMM+TRI		
P (%)	R (%)	F1	P (%)	R (%)	F1	P (%)	R (%)	F1
68.30	80.30	73.82	89.70[*1]	82.84[*1]	86.13[*1]	95.20[*1]	83.61[*1]	89.03[*1]
			82.80[*2]	92.32[*2]	87.30[*2]	83.87[*2]	94.71[*2]	88.96[*2]
			66.46[*3]	96.93[*3]	78.85[*3]	58.41[*3]	99.70[*3]	73.66[*3]

[*1], [*2] and [*3] are the values obtained when the number of N-best answers is set to 1, 2 and 3, respectively.

The results show that our methods, both HMM and HMM+TRI, give better results than those of SP99, in both precision and recall. SP99 is a very fast method and usually finds the expected words, but it also gives a lot of false hits, resulting in high recall and low precision. Among these three algorithms, HMM+TRI performs best and significantly improves the use of HMM alone.

HMM+TRI gives the best performance when using only the first answer, and in this case, HMM+TRI yields a satisfactory result of 95.20% and 83.61% in precision and recall respectively. The recall is very high (99.70%) when we use the best three soundex codes, but it causes very low precision (58.41%).

CONCLUSIONS

We have proposed a method for improving Thai soundex system by using the soundex code that considers the initial consonant sound, vowel sound, and final consonant sound. We solve the problems of previous works by considering the syllable boundary and employing a preprocessing technique to handle the ambiguities in Thai pronunciation rules. We have applied the HMM with phonetic trigrams for Thai soundex encoding. The experimental results, comparing our algorithms and previous works, reveal that the combination of the HMM and phonetic trigrams gives the best result with 95% precision and 84% recall.

REFERENCES

Arunwong-Na-Ayuthaya, N., (1991) Transformation on Thai strings using soundex principle (in Thai), *Senior Project*, Faculty of Engineering , Chulalongkorn University, Bangkok, Thailand.

Green, J. H., Finding treasures in the U.S. federal census, http://www.firstct.com/fv/uscensus.html.

Golding, A. R., and Schabes, Y., (1996) Combining trigram-based and feature-based methods for context sensitive spelling correction, *Proceedings of the 34th Annual Meeting of the Associated for Computational Linguistics*, pp. 71-78, Santa Cruz, California, USA.

Karoonboonyanan, T., Somlertlamvanich, V. and Meknavin, S., (1997) A Thai soundex system for spelling correction, *Proceeding of the National Language Processing Pacific Rim Symposium 1997*, Phukhet, Thailand.

Lohjeerachoonhakul, V., and Khuwinpan, C., (1982) Thai soundex algorithm and Thai syllable separation algorithm (in Thai), *Research Report*, Faculty of Applied Statistics, The National Institute of Development Administration, Bangkok, Thailand.

Meknavin, S., Kijsirikul, B., Chotimonkol, A., and Nuttee, C., (1998) Combining trigram and winnow in Thai OCR error correction, *Proceedings of COLING 1998*.

Rabiner, L., & Juang, B.H., (1993) *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

Suwanvisat, P., and Prasitjutrakul, S., (1999) Transliterated word encoding and retrieval algorithms for Thai-English cross-language retrieval, *Proceeding of the NCSEC99*, Thailand.

Tangkhawanwanich, N., (1991) Spelling correction (in Thai), *Senior Project*, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand.

Udompanich, W., (1983) Applying homonymity to searching Thai sound-alike strings (in Thai), *Master Thesis*, Chulalongkorn University, Bangkok, Thailand.