A Chunk-based n-gram English to Thai Transliteration

Wirote Aroonmanakun

Dept. of Linguistics, Facu;ty of Arts, Chulalongkorn University,
Phayathai Rd., Bangkok, Thailand 10330
Tel. +66-2-218-4696, Fax.: +66-2-218-4695
e-mail : awirote@chula.ac.th

Abstract

In this study, a chunk-based n-gram model is proposed for English to Thai transliteration. The model is compared with three other models: Table lookup model, decision tree model, and statistical model. The chunk-based n-gram model achieves 67% word accuracy, which is higher than the accuracy of other models. Performances of all models are slightly increased when an English grapheme to phoneme is included in the system. However, the accuracy of the system does not suffice for using as a public transliteration tool. The low accuracy of the system is caused by the poor performance of the English grapheme to phoneme module and the inconsistency of pronunciation in the training data. Some suggestions are provided for further improvement.

1 Introduction

English to Thai transliteration is a way to write English words in Thai alphabets. While English has 26 characters for 24 consonant and 20 vowel sounds[1], Thai has 44 characters for 21 consonant sounds, and 19 characters (including 3 consonant characters) for 24 vowel sounds (including 6 diphthongs), and 4 characters for tone markers. When transliterating English words into Thai words, it is usual to have different Thai written forms. For example, the word "internet" can be found written as อินเตอร์เน็ต, อินเตอร์เนท, อินเตอร์เน็ท, อินเทอร์เน็ต, อินเทอร์เนต, or อินเทอร์เน็ท. To standardize the transliteration, the Thai Royal Institute issued regulations of English-Thai transliteration in 1982. Nevertheless, many people tend to transliterate English words on their own rather than adhering to the regulations. It would be very useful if an English-Thai transliteration program that conforms to the Royal Institute's guideline is available. In this study, we aim to develop such a system. A corpus of transliterated words is created by collecting English and Thai word pairs from books published by the Royal Institute. A total of 8,181 word pairs are used in this study. In each word pair, Thai characters are aligned with their English correspondent characters. Alignments between English and Thai characters are first assigned by a program and then manually corrected. It is possible that more than one character in English or Thai is aligned, e.g. 'th'-'ท', 'ia'-'ีย'. Examples of aligned characters between word pairs are shown below. These data will be used for training the transliteration systems.

l/ i/ th/ o/ s/ o/ l/ s/	ล/ิ/ ท/ โ./ ซ/ อ/ ล/ ส์/
l/ i/ th/ ua/ n/ ia/	ล/ิ/ ท/ั/ว/ น/ี/ย/
l/ i/ v/ e/ r/ p/ oo/ l/	ล/ิ/ ว/ เ.อ/ ร์/ พ/ู/ ล/
l/ i/ v/ i/ ng/ s/ t/ o/ n/ e/	ล/ิ/ ฟ/ว/ิ/ ง/ ส/ ต/ โ./ น/ #/
l/ i/ v/ i/ u/ s/	ล/ี/ ว/ี/ อ/ั/ ส/

This paper first reviews previous models of transliteration systems. Table lookup, decision tree, and statistical models are briefly discussed. Then, a new approach of chunk-based n-gram model is described in section 3. The results when using each model are reported and compared in section 4. Since knowing English pronunciation is usually useful for

[1] Based on Gimson's pronunciation of English (2001)

transliteration, all the models are tested again by applying a module of English grapheme to phoneme. The new results are reported in section 5. Though the chunk-based n-gram model performs better than other models, the accuracy is not high enough to be used as a tool for the public. At the end, we will review and discuss the problems for further improvements.

2 Previous research

Since transliteration is basically a process of transforming one writing system into another writing system, approaches used in any transliteration systems as well as those used in grapheme to phoneme systems are relevant. In this study, three different approaches, namely table lookup, decision trees, and statistical model, are reviewed and implemented in this study.

2.1 Table Lookup Model

The model is based on Bosch and Daelemans' grapheme-to-phoneme conversion model (1993). It is used for transcribing English, French, and Dutch. Conversion of characters to phonemes is done by applying conversion rules, which are extracted from a training corpus. But in this study, the conversion rules are used for converting characters from English to Thai. Rules are store in a lookup table as a mapping from English characters to Thai characters, which is determined by the left and right contexts of the English characters. By using a training corpus composed of word pairs aligned between characters of the two languages, if the target language character can be uniquely determined from the source language character within its minimal context, the conversion rule will be stored in a lookup table. But when the same context does not uniquely determine the target language character, conversion are done by default mapping by selecting the most occurring target character in that context. In this study, lookup tables of various context sizes are implemented: 0-0, 0-1, 1-1, 1-2, 2-2, 2-3, 3-3, 3-4, 4-4, 4-5, and 5-5. (The two digits indicate the number of characters on the left and right contexts) Default mapping of 0-0, 1-1, and 2-2 contexts are used when no lookup table is applicable.

2.2 Decision Tree Model

Decision tree model converts symbols from one language to another by applying rules that are in the form of decision tree. Kang and Choi (2000) use this model for Korean-English transliteration system. Decision tree is created by applying a well-known machine learning technique, ID3. This method is often applied to many NLP systems, such as Thai grapheme to phoneme (Chotimongkol and Black 2000), word sense disambiguation (Pedersen 2004). In this study, Lenzo's (1998) decision tree model for English grapheme to phoneme is modified to create decision trees for English-Thai transliteration. The maximum depth of trees is set to 7. The left and right contexts are set to 3 characters.

2.3 Statistical Model

The third model is a statistical model, which is often used in transliteration research, such as Japanese-English back-transliteration (Knight and Graehl 1997), English-Arabic transliteration (Glover and Knight 1998), English-Korean transliteration (Kang and Kim 2000), English/Japanese transliteration (Fujii and Ishikawa 1999, 2001), English-Korean transliteration (Jung et al. 2000), etc. Transliteration problem is viewed as a probabilistic model. In this study, English-Thai transliteration can be viewed in a similar way as follows:

$$\arg\max_{Tw} P(T_w \mid E_w) = \arg\max_{Tw} \frac{P(T_w) * P(E_w \mid T_w)}{P(E_w)}$$

$$= \arg\max_{Tw} P(T_w) * P(E_w \mid T_w)$$

$$P(T_w) = P(Tc_1, Tc_2, ..., Tc_n) \approx \prod_{i=1,n} P(Tc_i \mid Tc_{i-2} Tc_{i-1})$$

$$P(E_w \mid T_w) \approx \prod_{i=1,n} P(Ec_i \mid Tc_i)$$

Transliteration from English to Thai is composed of two sub-models: P(T) and P(E|T). P(T) can be estimated by a trigram model, while P(E|T) is estimated from alignments between English and Thai characters in the training corpus.

In addition, Haizhou et al. (2004) joint source-channel model, which is used for English-Chinese transliteration, will be implemented as another variant of this model in this study. Unlike other models which capture how source words can be mapped to target words, this model uses both source and target

words simultaneously. The model can be formulated as follows:

$$\arg\max_T P(T \mid E) = \arg\max_T \frac{P(T,E)}{P(E)} = \arg\max_T P(T,E)$$
$$= \arg\max_T P(t_1 t_2 .. t_n, e_1 e_2 ... e_n)$$
$$= \arg\max_T P(< t_1, e_1 >, < t_2, e_2 >, ... < t_n, e_n >)$$
$$\approx \arg\max_T \prod_{i=1}^{n} P(< t_i, e_i > | < t_{i-2}, e_{i-2} >< t_{i-1}, e_{i-1} >)$$

Transliteration is viewed as a probabilistic model of transliteration pairs between English and Thai characters, which then can be estimated by a trigram model.

3 Chunk-based n-gram Model

The model proposed in this study is a chunk-based n-gram model. It is based on Kang and Kim's (2000) view of phoneme chunks and Haizhou et al.'s (2004) joint source-channel model. In this chunk-based model, alignment between English and Thai characters can have various lengths. For example, for the word pair "locarno" โลคาร์โน, beside the normal alignments l-ล, o-โ, c-ค, a-า, r-ร์, n-น, o-โ, alignments of larger units, i.e. lo-ลโ, oc-โค, ca-คา, ar-าร์, rn-ร์น, no-นโ, loc-ลโค, oca-โคา, car-คาร์, arn-าร์น, rno-ร์นโ, …, and locarno-ลโคาร์นโ are also generated. Like other statistical models, transliteration here is viewed as a probabilistic model of transliteration pairs between Thai and English. But in this model, the units of transliteration pair can be a chunk of characters. Probability of a sequence of transliteration pairs is estimated by a trigram model in this study. The sequence with the highest probability will be selected as the solution.

$$\arg\max_T P(T \mid E) = \arg\max_T \frac{P(T,E)}{P(E)} = \arg\max_T P(T,E)$$
$$= \arg\max_T P(t_1 t_2 .. t_n, e_1 e_2 ... e_n)$$
$$= \arg\max_T P(< t_{1..a}, e_{1..a} >, < t_{a+1..b}, e_{a+1..b} >, ... < t_{m+1..n}, e_{m+1..n} >)$$
$$= \arg\max P(< ct_1, ce_1 >, < ct_2, ce_2 >, ... < ct_n, ce_n >)$$
$$\approx \prod_{i=i}^{n} P(< ct_i, ce_i > | < ct_{i-2}, ce_{i-2} >< ct_{i-1}, ce_{i-1} >)$$

(ct_i *and* et_i *are a chunck of Thai and English characters*)

For example, when transliterating 'unitarian', there could be many possible sequences of transliteration pairs as follows:

<unita,ยูนิทเ.>,<r,ร>,<ian,เ็ียน>
<unita,ยูนิทเ.>,<r,ร>,<i,ไ.><an,อัน>
<unitar,ยูนิทเ.ร>,<ian,อาน>
<uni,ยูนไ.>,<t,ต>,<a,า>,<rian,รเ็ียน>
<unit,ยูนิท>,<arian,ารเ็ียน>
….

Probability of each sequence is calculated based on trigram statistics of all transliteration pairs in the training data. Using this model, it is likely that a sequence with fewer chunks will have higher probability than a sequence with longer chunks and chunks with high occurrences will have higher probability than chunks with low occurrences.

4 Experiments

The corpus of 8,181 English-Thai pairs is divided into five data sets. Four of them are used as the training data, while the other one is used as the test data. Each data set is tested on all systems. Then, the results from five tests will be averaged as the performance of each system. This indicates the performance on unseen data. Each system is also tested for seen data (All data sets are used for both training and testing). Performance is measured in two ways: word accuracy (W.A.) and character accuracy (C.A.) (Kang and Kim 2000). W.A. is counted from the exact match of the generated words and the correct word. C.A. is calculated on the basis of edit distance between the two words. C.A. = (L – (i + d +s)) / L where L is the length of a word, and i, d, s is the number of insertion, deletion, and substitution that are needed to change the result to match the target word. The results are shown in Table 1.

	Unseen		Seen	
	W.A.	C.A.	W.A.	C.A.
TB	60.6%	85.8%	97.2%	99.2%
DT	61.7%	86.8%	95.7%	99.0%
N-gram	37.6%	77.1%	48.2%	82.1%
Joint	50.1%	84.4%	67.2%	90.0%
Chunk	67.4%	88.9%	99.8%	99.9%

Table 1: Result of English-to-Thai transliteration

It can be seen that the chunk-based n-gram performs better than other systems for both unseen and seen data. For unseen words, the accuracy at the word level is 67.4%, and the

accuracy at the chracter level is 88.9%. For seen words, word accuracy of the chunk-based n-gram is 99.8% and character accuracy is 99.9%.

5 Adding E2P module

Knowing how the word is pronounced is usually useful for transliteration from English to Thai. In fact, to transliterate an English vowel correctly, it is necessary to know how it is pronounced. For example, vowel form 'i' can be transliterated to three Thai vowel forms, ิ ี ไ depending on the pronunciation of that vowel. Therefore, the systems are tested again by adding a module of English grapheme to phoneme (E2P) as a part of the system. But only three models, TB, DT, and Chunk-based, are tested at this time. The systems are implemented by considering both English characters and phonemes in this new test. E2P module is created by applying lookup tables, which are generated from a CMU pronunciation dictionary. The accuracy of the E2P is measured at 56.7% for W.A., and 90.8% C.A. The new result of the systems when including E2P is shown in Table 2.

	Unseen		Seen	
	W.A	C.A	W.A	C.A
TB	65.3%	88.4%	99.7%	99.9%
DT	64.4%	88.0%	97.5%	99.4%
Chunk	68.1%	88.7%	99.8%	99.9%

Table 2 : Result when E2P is included

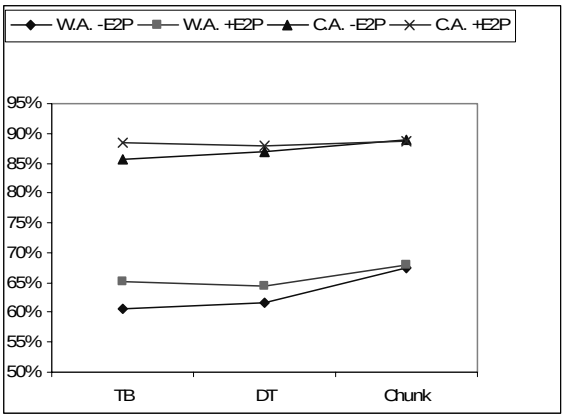

Figure 1: Results with and without E2P

It can be seen that all systems perform a bit better when English G2P is included. And the chunk-based system is still the best model in this study.

6 Discussions

Although the chunk-based n-gram model performs better than other models, the accuracy of the system is still not high enough for using as a transliteration tool for the public. To improve the performance, error analysis is needed to understand why some generated words do not match the correct words.

Two factors are likely the causes of low accuracy: accuracy of E2P and different accents of pronunciation. The accuracy of E2P module developed in this study is not really high. It yields only 56.7% for word accuracy and 90.8% for phoneme accuracy. Creating a good E2P module is difficult. Black et al. (1998) also reported the accuracy of their E2P system at 57.80% for word accuracy and 91.99 for phoneme accuracy when using CMU pronunciation dictionary in their tests, while the accuracy when using Oxford Advanced Learners Dictionary of Contemporary English is higher (74.56%). Black et al. explained that the difference lied on the fact that CMU dictionary includes a lot more proper names. According to Llitjós (2001), systems that produce high accurate results are those that are dictionary-based. Transliteration rules are used only when the input word is not listed in the dictionary. Therefore, to improve E2P performance, a pronunciation dictionary should be used directly. However, in English-Thai transliteration, many inputs are proper names. It is unlikely to have all names listed in the dictionary. It is still necessary to create a good E2P module that is not a dictionary-based.

Different accents of pronunciation could also be another cause of low accuracy in this study. Since transliteration is partly based on the pronunciation of English words and the same word can be pronounced with different accents, i.e. British or American pronunciations, the transliterated words then could be written differently. For example, the word "Leonard" is found transliterated in the Royal Institute's books as เลนาร์ด, เลนเนิร์ด, and เลียวนาร์ด. Although only one form that we think is most conformed to the guideline is stored in the corpus in this case, it does not exclude the possibility of different accents entailed in different words. This inconsistency of the training data could result in inefficiency of transliteration rules.

In addition, when the generated words do not exactly match the correct words, it is possible that the generated word is another form of acceptable transliterations. For example, the word "ballast" is transliterated by the chunk-based system as บัลลาสต์ , while the correct word is แบลลาสต์. The difference of vowel form in this case is resulted from different accents of pronunciation. In this example, the generated word is considered an acceptable result. Therefore, the results are manually checked whether they are acceptable transliteration. Using this acceptable criterion, it is found that the accuracy of the chunk-based model gains up to 84%.

7 Conclusion

Although the chunk-based model performs better than other systems, 68-84% W.A. is not a satisfying result. To be released as an English-Thai transliteration tool for the public, the program should have high accuracy up to 98%. The system has to be improved by employing a good E2P. And it might be necessary to manually clean up the training data to make English pronunciation in the transliterated words consistent with one particular accent. Beside, the chunk-based model uses more resources than other models. It is needed to be improved in terms of processing speed.

Acknowledgment

This research is supported by a grant from the Thailand Research Fund and the Commission on Higher Education in July 2003 – June 2005.

References

Black, A. W., Lenzo, K. and Pagel, V. 1998. Issues in Building General Letter to Sound Rules. *3rd ESCA Speech Synthesis Workshop*, pp. 77-80, Jenolan Caves, Australia.

Bosch, A. van den and W. Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 45-53, Utrecht, Netherland.

Carnegie Mellon University Pronouncing Dictionary v0.6 (http://www.speech.cs.cmu.edu/cgi-bin/cmudict)

Chotimongkol, Ananlada and Alan W Black. 2000. Statistically trained orthographic to sound Models for Thai, In *Proceedings of ICSLP 2000*, Beijing, China October 2000.

Fujii, Atsushi and Tetsuya Ishikawa. 1999. Cross-Language Information Retrieval for Technical Documents. In *Proceedings of the Joint ACL SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 29-37,

Fujii, Atsushi and Tetsuya Ishikawa. 2001. Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration. In *Computers and the Humanities, 35(4)*: 389-420.

Gimson, A.C. 2001. *Gimson's pronunciation of English.* Sixth Edition. Revised by Alan Cruttenden. London: Arnold.

Glover, Bonnie and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages.* pp. 34-41, Montreal, Quebec, Canada.

Haizhou, Li, Zhang Min, and Su Jian. 2004. A Joint Source-Channel Model for Machine Transliteration. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 159-166, Barcelona, Spain.

Jung, SungYoung, SungLim Hong, and Eunok Paek. 2000. An English to Korean Transliteration Model of Extended Markov Window. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 383-389, Saarbrücken, Germany.

Kang, B. J. and K. S. Choi (2000) Automatic transliteration and back-transliteration by decision tree learning. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athnes, Greece.

Kang, In-Ho and GilChang Kim. 2000. English-to-Korean Transliteration using Multiple Unbounded Overlapping Phoneme Chunks. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp.418-424, Saarbrücken, Germany.

Knight, Kevin and Jonathan Graehl. 1997. Machine Transliteration. In *Proceedings of 35th*

Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics, pages 128-135, Madrid, Spain.

Lenzo, Kevin A. (1998) ”s/ ($text) / speech $1 / eg;” In *The Perl Journal*, Issue 12, pp.26-29.

Llitjós, Ariadna Font. 2001. *Improving Pronunciation Accuracy of Proper Names with Language Origin Classes.* Master Thesis. Carnegie Mellon University.

Pedersen, Ted. 2001. Lexical Semantic Ambiguity Resolution with Bigram-Based Decision Trees. In *Proceedings of CICLing 2001*, Mexico City, Mexico, Febuary 18-24, 2001, 157-168.