Detection and Correction of Homophonous Error Word for Khmer Language

Chea Sok Huor, Top Rithy, Ros Pich Hemy
csh007@gmail.com, toprithy@gmail.com, pichhemy@gmail.com

Abstract

The ability to detect and correct the written error is a very crucial challenge for Khmer language computing because of the fact that Khmer does not separate words in its writing system. The richness of Khmer characters confuses users to write a word in many different ways. In this paper, we proposed a method to detect the homophonous non-word error using a Khmer Common Expression (in short KCE) and automatically correct it. The idea is to generate the same expression for every word that is likely to be confused in sound. Therefore, the string to map the word in the dictionary is not the real word string, but an expression, which has the same pronunciation as the target word.

1. Introduction

The detection and correction of errors for Khmer language is a vital key for the development of other NLP applications such as Information Retrieval, machine translation, OCR, etc…Since there is no obvious boundary in Khmer writing system, it is clear that most errors in the text cause abnormal segmentation. Consequently, if errors exist, segmentation result can be weird. For example, the segmentation of the word **សំរែច**, which is the incorrect word for **សម្រែច** results two different words **សំ-រែច**. To human perception, it is very easy to judge that it is a one-word error due to its pronunciation and context, but it is very difficult for the computer to identify the fact.

The Khmer Common Expression (KCE) is introduced. The expression is created to detect the sound-similarity errors. The main idea to solve the problem is to encode the misspelled word and the dictionary word list into an expression that is based on how it is pronounced. It means the string with the same pronunciation has the same expression.

For example, the set of *សំលេង, សម្លេង, សំឡេង* should be encoded into the same KCE.

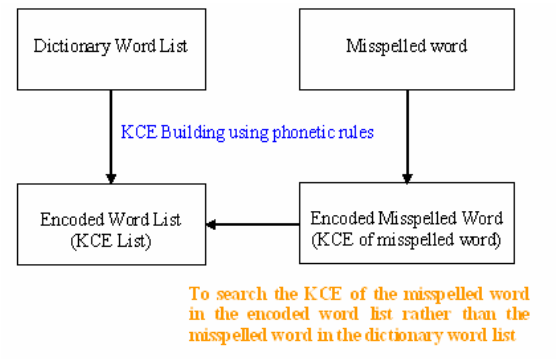

Figure 1: KCE building using phonetic rules

2. Method

2.1. Khmer text writing error behavior

Generally, Spelling errors are grouped into two types: non-word error and real word error. Real word errors are those typographic errors that result in a valid dictionary entry not intended by typist. For example, typing “form” when “from” was intended. Non-word errors are those error words that are invalid in the dictionary. During the research, we have assigned a group of people to type various articles from the books and another group to do the dictation. Here is our observation on the most common non-word misspellings:

1. Around 70% percent of the error is due to the phonetic similarity. This type of error occurs under these circumstances:
   - The users do not know how to write the word so they write something which has the same pronunciation to the targeted word. For example, the confusion of writing the word **សំឡេង** as **សំលេង** or **សម្លេង**
   - The omission of the character **ះ** when it is found at the end of the word, because mostly, Khmer do not pronounce this consonant when it is found at the end of the word.

- Some various signs do not have any effect in Khmer pronunciation so that it always is omitted or confused in writing.
- Etc…

2. Around 10% of misspelling is due to the shape similarity. For example, confusion of the subscript of TA (្ត) and the subscript of DA (្ដ) due to its similar shape
3. The rest is caused by the adjacent keys and others.

2.2. Khmer homophone problem analysis

2.2.1. Khmer phonetic sets. The following list is some of the homophonous sets that are extracted from the Khmer Language Grammar book by Khin Sok.

Table 1: Khmer homophone sets

Phone	Homophone Set	Example
[អាហ៎]	{[អ/អ៊]+ះ, [អ/អ៊] + ាស់}	កន្លះ, កន្លាស់
[អេហ]	{ែះ, ៃះ, ៃស}	កេះ, កែះ, កែស
[អុហ]	{ុះ, ុស}	ដុះ, ដុស
[អួហ]	{[អ៊]+ស់,[អ៊]+ោះ,[អ៊]+ួស }	ពស់, ពោះ, រលួស
[អ៊ា]	{[អ៊]+ា, [អ៊]+ៀ, [អ]+ៀ }	ទាន, ទៀន, តៀន
[អ៊ុំ]	{[អ៊]+ុំ, [អ៊]+ំ}	ទុំ, ទំ
[អ៊ូរ]	{[អ៊]+ូរ, [អ៊]+ៅ}	នៅ, នូរ
[អ៊ុង]	{[អ៊]+ង់, [អ៊]+ុង}	កំពង់, កំពុង
[អ៊ប់]	{[អ៊]+ប់, [អ៊]+ុប}	លប់, លុប

In Khmer language, independent vowel is a stand-alone character and has its own complete sound. It needs no vowel or subscripts or various sign. In writing, it is also one of the main causes for misspelling. For instance, the word ឬ is usually confused with the wrong word រឺ due to the sound similarity. The list below gives some similar phonetic sets of some independent vowel.

Table 2: Independent vowels homophone sets

Phone	Independent vowel	Equal sound
[អិ]	ឥ	អិ
[អុ]	ឧ	អុ
[អូរ]	ឱ	អូរ
[រឹ]	ឫ	រឹ
[រឺ]	ឬ	រឺ
[លឹ]	ឭ	លឹ
[លឺ]	ឮ	លឺ
[អែ]	ឯ	អែ
[អោ]	ឱ	អោ

2.2.2. Other phonetic set cases. Apart from the list above, some other common homophonous misspelling error is observed and listed:

1. The case of រ: Normally, រ is not pronounced when it is found at the end of the words. For example, the set of {គូរ, គូ}
2. Since some various signs are not pronounced in the word, they are likely to be forgotten during the writing. These consists of: ៍, ់, ៊, ់
3. The case of the consonant set {ណ, ន} and {ល, ឡ}: The two consonants are likely to be confused when its previous syllable has the sound [អ] A. For example, the confusion of writing the word សំណួរ as សំនួរ and the confusion of the writing សំឡេង as សំលេង

2.2.3. Sound shifting issue. Fundamentally, Khmer consonants are grouped into two main types of sound, [អ] (sound A) and [អ៊] (sound O). The sound type of the consonant and the subscript could affect the sound of the syllable when they combine with the vowel.

Taking as example of the Khmer vowel ◌ា "a:".

It is pronounced as "a:" when it is with the sound-A consonant and "i:ə" when it is with the sound-O consonant.

Example1:

កា (Sound A)	+	◌ា	→	កា
K	+	a:	→	ka:
But				
គា (Sound A)	+	◌ា	→	គា
K	+	a:	→	ki:ə

Example2:

សំ(A)	+	◌្ត (A)	+	◌ា	→	ស្តា
S	+	d	+	a:	→	sda:
But						
សំ(A)	+	◌្ត(O)	+	◌ា	→	ស្តា
S	+	p	+	a:	→	spi:ə

There are many more issues left concerning the sound shifting in the syllable. For more details, please see the Khmer Language Grammar Book by Khin Sok.

2.3. Erroneous word segmentation algorithm

Since there is no boundary between words in Khmer writing system, there is no easy solution for the task. Therefore, it is inevitable to take word segmentation to be part of the research. We proposed a word segmentation method that segments Khmer text into combinations of characters, called Khmer Character Cluster (in short KCC), and then merge those KCCs in possible word segmentations. The main advantage is that KCC is an inseparable unit and well defined so that the boundary of KCC might be the boundary of word also. This leads to a decrease in the number of accesses to the dictionary during the process of finding possible segmentations, which is time consuming.

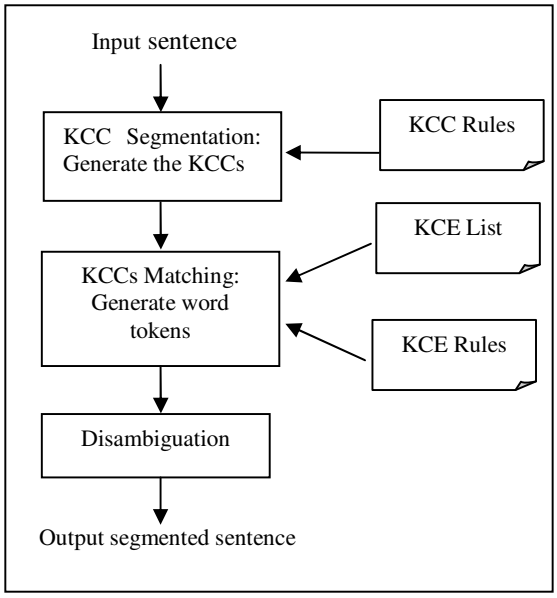

Figure 2: Flow chart of the system

Figure 2 above illustrates the main process of the algorithm. First, the input sentence is converted into a collection of KCCs. Then, KCC matching module reads each KCC one by one from left to right and match them. Then, it converts the KCCs into KCE string. The KCE string is used to look up if it exists or not in the dictionary. Therefore, multiple possible segmentations of the input text are generated. The disambiguation module will select the best segmentation among those candidates.

2.4. Khmer Character Cluster (KCC)

Here, KCC is defined as a succession of characters with an inseparable unit. Since KCC is well defined, the boundary of the KCC might be the boundary of Khmer word also. It means that, a Khmer word is the combination of one or more KCCs. Below is the examples of the KCC in Khmer word:

- The word សាលាក្តី is the combination of three KCCs: សា + លា + ក្តី
- The word ធ្មុកចាប់ is the combination of four KCCs: ធ្មុ + ក + ចា + ប់
- The word ស្ត្រី is a one KCC word

The segmentation of text into KCC can be done by detecting the transition state for each character of

the input string. If there is no possible transition of any character, it means the end of KCC is reached. The rule for forming KCC is given in terms of type of characters.

<C|I> + [<Robat | Regshift>] + {COEUNG + <C + [Regshift] | I + [Regshift]>} + [[<ZWJ|ZWNJ>] + V] + {S} + [ZWJ + COEUNG + <C | I>]

The meanings of the symbols are:

Symbols	Meaning
{ }	Zero to two occurrences
[ ]	Zero to one occurrences
<x | y>	The choice of x or y
C	Consonant
I	Independent vowel
COEUNG	The COEUNG character (\u 17D2)
V	Vowel
Regshift	Registry shifter
S	Various sign
ZWNJ	ZERO WIDTH NON-JOINERS
ZWJ	ZERO WIDTH JOINERS
ROBAT	The Robat sign (\u 17CC)

2.5. Khmer Common Expression (KCE)

As stated, KCE is an expression created to make Khmer strings with the same pronunciation similar. Simple character-to-character mapping method could not be applied here because there are many cases of sound changes in Khmer pronunciation system. Therefore, KCC, which is the combination of Consonant, Robat, Subscripts, Consonant shifter, Vowels, Various sign, Zero width Non-Joiner, Zero width Joiner, is selected to be the fundamental component for building the KCE. After the study on the KCC structure and the problems listed above, we have categorized three types of rule for building the KCE. Each type of rule handles different blocks of the KCC.

- Type 1: focuses on Consonant and ROBAT block of the KCC
- Type 2: handles the subscript, consonant shifter block
- Type 3: handles the Vowels, Various signs.

For each type of rule, there are two main parts: matching rule and mapping rule. Matching rule is used to identify if the KCCs matched to the KCE rule or not. Mapping rule is used to map the original KCCs into another string.

The main process is illustrated in Figure 3. The system matches the input KCCs with all the rules in each type of rule. Only one rule in each type of rule could be matched. After getting the matched rules, it will map the input KCCs to other string according to the corresponding mapping rules. The string is called KCE.

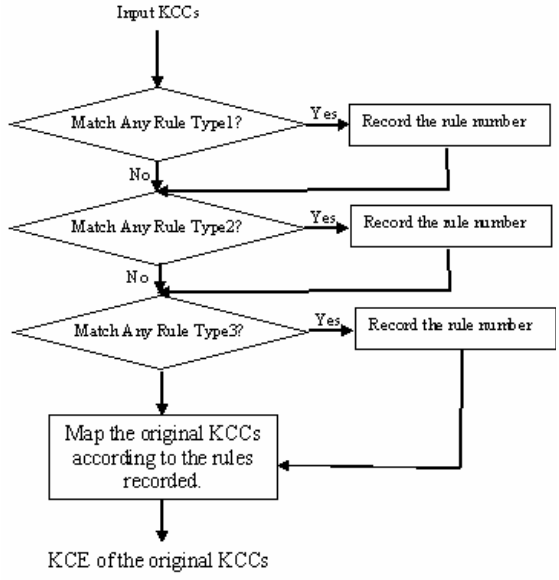

Figure 3: KCE building process

2.5.1. Matching rules. A matching rule is the combination of one or more KCC matching rules, which contains five main blocks.

The first block: is the consonant block. The value of the block could be

- A consonant
- 0 : The sound of KCC is [អ] (Sound A)
- 1 : The sound of KCC is [ឱ] (Sound O)
- 2 : The sound of the KCC is either [អ] (Sound A) or [ឱ] (Sound O)

The second-fifth block: The second block is the subscript block, the third is consonant shifter block, the fourth is the vowel block and the fifth is the various sign block. The value of each block could be:

- The character of corresponding block, for example, the vowel block should contain a vowel.

- 0 : The current KCC must not have the character of the corresponding block
- 1 : The current KCC must have at least one character of the corresponding block
- 2 : The current KCC must either have the character of the corresponding block or not

2.5.2. Mapping rules. A mapping rule is the combination of one or more KCC mapping rules, which contains five main blocks, the consonant block, subscript block, consonant shifter block, vowel block, and various sign block. The value of each block could be:

- The mapping character of the corresponding block, for example, the vowel block should contain a vowel.
- 0 : The mapping character is the same as the matching character
- 1: There is no mapping character.

Example: Given rules set as below

1. Type 1 rule: 0,2,2,2,2;ឡ,2,2,2,2=0,0,0,0,0;ល,0,0,0,0
2. Type 3 rule: 0,2,2,0,ំ=0,0,0,0,ម

The first rule means that the consonant ឡ could be replaced by ល when the sound of the first KCC is [អ] (Sound A). However, the second rule means that the vowel ំ is replaced by ម when in the KCC which contains the vowel has the A sound and there is no other vowel in the KCC.

Due to its similar pronunciation, the KCE of the three words សំលេង សំឡេង សម្លេង must be the same. The KCE representing the three words are:

សម្លេង = សមលេង

សំលេង = សមលេង due to the first rule.

សំឡេង = សមលេង due to the second rule.

Therefore, when the user falls to write one of the above sets, it is treated as the same.

Note: Khmer subscript is the combination of two Unicode character, \u17D2 and the consonant representing the subscript. In the process of building the KCE, all the subscript is represented only by the corresponding consonant. It is very advantageous for solving the transformation from TAMROUT words into SAMRAY words and vice versa.

3. Experiment result

As an experiment on the capacity of the approach, we collected some test sets by assigning some of our team to copy texts from textbook and newspaper. Some test sets are collected from the web. Therefore, homophonous error can be collected. Below is the result:

Total number of words in the test set=8956
Total Number of homophonous misspelling=436
Number of error correctly detected=403

$$\text{Rate} = \frac{\text{No. of errors correctly detected}}{\text{Total number of misspelling}}$$

$$= (403/436)*100\% = 92.43\%$$

4. Conclusion and further work

This paper described an approach for the detection and correction of homophonous non-word errors for Khmer language using the Khmer Common Expression KCE. The introduced approach has achieved a good result. However, it also raised a few ambiguities problems for word segmentation because rule based technique is used in the disambiguation module in this research.

For Khmer, plenty of work also has to be done to improve both the performance and accuracy of automatic detection and correction of error words.

There is a need to:

- Improve the KCE rules to advance the accuracy of the sound-similarity error words.
- Combine the current method with other different method to enable the non-homophonous errors detection such as keyboard adjacent errors and so on.
- Enable the ability to detect the real word errors so that we need to build contextual information.
- Use statistical based approach to improve the quality of the segmentation since there is no word boundary in Khmer sentence. Therefore, some ambiguity issues that is raised by the KCE approach is solved.

5. Acknowledgement

We would like to express our gratitude to Dr. Khin Sok for his helpful explanation on Khmer language grammar to our research programmer. We also would like to thank to the Royal Academy of

Cambodia for their support and suggestion on Khmer linguistic during our research on the project.

6. References

[1] A.B.A. Abdullah and A. Rahman, "A Generic Spell Checker Engine for South Asian Languages", *Research Report, Computer Science & Engineering*, BRAC University, Dhaka, Bangladesh.

[2] C. Nath, *Dictionnaire Cambodgien*, Edition de L'institut Bouddhique, Phnom Penh, 1967.

[3] K. Sok, *Khmer Language Grammar*, First Edition of Royal Academic of Cambodia, 2004

[4] L. Zhang, M. Zhou, C. Huang and H. Pan, "Automatic Detection/Correction errors in Chinese text by an approximate word-matching algorithm".

[5] P.K. Wong, and C. Chan, "Chinese Word Segmentation based on Maximum Matching and Word Binding Force", *Research Paper*, Department of Computer Science, The University of Hong Kong, Hong Kong.

[6] S. Bing and Y. Shiwen, "A Graded Approach for the Efficient Resolution of Chinese Word Segmentation Ambiguities", Institute of Computational Linguistics, Peking University, Beijing 1 00871, China.

[7] T. Naseem, "Spelling Checker", slides available online at: http://www.panl10n.net/Presentations/Afghanistan/UrduSpellChecking.pdf.