Word Bigram Vs Orthographic Syllable Bigram in Khmer Word Segmentation

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

Abstract

This paper discusses the word segmentation of Khmer written text based on Bigram model. This research will carry out two extended methods from the Bigram Model, Word Bigram and Orthographic syllable (technically called as Khmer Character Cluster or KCC in short) Bigram. Many major issues in the segmentation process of Khmer text will be discussed. The sound similarity errors identification of our previous research will be combined in this research to improve the accuracy of the segmentation.

1. Introduction

Khmer writing system does not separate words in the sentence. It posed a major problem for many natural language processing applications such as information retrieval, machine translation, speech processing, etc… where words boundary is very important. It is very easy for the native speaker to judge the break point of the word in the sentence, but it is a different case for the computer. Therefore, automatic word segmentation is critical for Khmer natural language applications. Two major problems occur in the automatic Khmer word segmentation.

The first problem is the ambiguity issue. Since there is no delimiter between words in writing, the computer is more confused by word boundary due to a great quantity of ambiguities. Most of the words can be co-located to form a new word. Hence, word or sentence can be segmented in various ways.
For example:

1. លើក = លើក or លើ | ក

2. ប្រជាជាតិខ្មែរ = ប្រជា|ជាតិ|ខ្មែរ or ប្រជាជាតិ|ខ្មែរ or ប្រជា|ជាតិខ្មែរ

The second problem is the identification of unknown word. Unknown word refers to word that does not exist in the dictionary. It causes segmentation error because since the word does not exist in the dictionary, it could be incorrectly segmented into shorter words or pieces of single syllable. For example, ដោះស្រាយ would be segmented into ដោះ-ស្រាយ after dictionary look-up. Unknown words can be categorized into the different types such as error words, abbreviation, proper names, derived words, compounds, Numeric type compounds etc...

There are not many researches on the topic for Khmer language. Our previous research is the detection and correction of Homophonous errors in Khmer language. The main objective of the research was to detect the sound similarity errors in Khmer writing text. The accuracy of the detection reaches 92 percent. The research used the maximum matching algorithm for the word segmentation. It is a dictionary-based algorithm, which selects the segmentation with minimum of words as the result.

This research will incorporate the disambiguation module using Bigram model with the homophonous error segmentation of our previous research.

2. Methods

2.1. General algorithm of the Khmer word segmentation

The main algorithm of the Khmer word segmentation is the same as the previous research. Only disambiguation module is changed.

First, as illustrated in Figure 1, the input sentence is segmented into combinations of character, called Khmer Character Cluster (in short KCC). 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. Here, Bigram model is used. Trained text corpus is required. The more text corpus we collected and trained, the higher the accuracy of the segmentation.

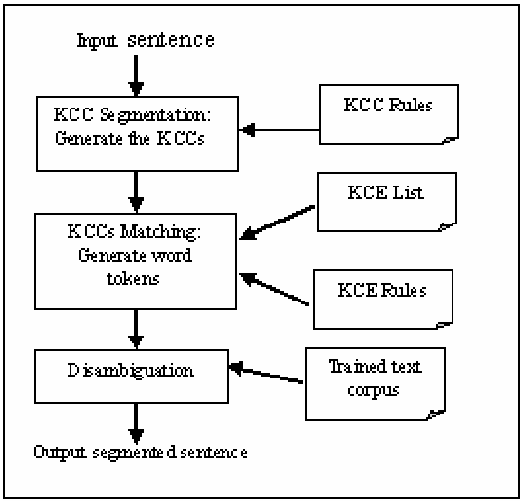
Input sentence
KCC Segmentation: Generate the KCCs
KCC Rules
KCE List
KCCs Matching: Generate word tokens
KCE Rules
Disambiguation
Trained text corpus
Output segmented sentence

Figure 1: Flowchart of the system

2.2. Bigram model understanding

Bigram model is used in the disambiguation module to decide the most appropriate segmentation among the list of candidates. The main idea is to assume that the next word can be predicted given the previous word. Therefore, the probability function is as followed:

$$\boxed{P(w_1^n) = \prod_{k=1}^{n} P(wk \mid wk-1)}$$

For example, given the Bigram probabilities table shown in Table 1, The probability of the sequence "I have dinner" is:

$$\begin{aligned} P(\text{I have dinner}) &= P(\text{I}|\text{<BOS>})\times P(\text{have}|\text{I}) \times P(\text{dinner}|\text{have}) \\ &\quad \times P(\text{<EOS>}|\text{dinner}) \\ &= 0.3 \times 0.2 * 0.25 * 0.13 \\ &= 0.00195 \end{aligned}$$

Table 1: Bigram probability table

Bigram	Probability value
<BOS> I	0.3
I have	0.2
have dinner	0.25
dinner <EOS>	0.13

Note: <BOS>: Beginning of sentence
<EOS>: End of sentence

2.2.1. Overflow issue: Since the probabilities are all less than one by definition, the product of many probabilities gets smaller. Practically, this might risk the numerical overflow, if the probability of a very long sequence is computed. Logarithm value of each probability should be applied. Therefore, multiplication operation is changed to addition operation.

2.2.2. Data sparseness issues: Data sparseness is a very serious and frequently occurring problem since the size of the corpus never seems to get enough. It means that if there is unseen Bigram, the probability of the sequence is zero due to the equation above. The smoothing technique make the distribution more uniform and redistribute probability mass from higher to lower probabilities. In this research, Witten-Bell smoothing technique is selected to handle the issue. The basic idea of Witten-Bell smoothing technique is that an unseen n-gram is one that just did not occur yet, so when it does happen, it will be the first occurrence. Therefore, the probability of the unseen n-gram is the probability of seeing a new n-gram.

$$P^{WB}(w_i \mid w_{i-1}) = \begin{cases} \dfrac{T(w_{i-1})}{Z(w_{i-1})(N(w_{i-1}) + T(w_{i-1}))} (\text{if} C(w_i w_{i-1}) = 0) \\ \dfrac{c(w_{i-1} w_i)}{N(w_{i-1}) + T(w_{i-1})} (\text{if} C(w_i w_{i-1}) > 0) \end{cases}$$

Where

- $C(w_i w_{i-1})$ is the number of occurrence of the word W_{i-1} followed by W_i
- $T(W_{i-1})$ number of Bigram types starting with w_{i-1}
- $N(W_{i-1})$ is the actual frequency of Bigrams beginning with W_{i-1}
- $Z(W_{i-1})$ Number of Bigrams starting with W_{i-1} that were not seen

For example, Table 2 shows the chart of the occurrences of the words in the corpus

Table 2: Chart of occurrences of the words in the corpus

	A	B	C	D	…	$N(w_1)$	$T(w_1)$	$Z(w_1)$
A	10	10	10	0		30	3	1
B	0	0	30	0		30	1	3
C	0	0	300	0		300	1	3
D								
…								

+ The probability value of the seen Bigram AB is

$$P(B \mid A) = \frac{C(AB)}{(N(A) + T(A))} = \frac{10}{(30 + 3)} = 0.3030$$

+ The probability value of the unseen Bigram that start with C is:

$$P(A \mid C) = \frac{T(C)}{Z(C)(N(C) + T(C))} = \frac{1}{3(300 + 1)} = 0.0011$$

$$P(B \mid C) = \frac{T(C)}{Z(C)(N(C) + T(C))} = \frac{1}{3(300 + 1)} = 0.0011$$

3. Experiment Result

At present, many texts with different genres were collected and manually segmented. The current size of the training corpus is 10.6 MB with 673295 words and 20991 vocabularies.

Usually, three measures are used to evaluate the segmentation accuracy: Precision rate, Recall rate and error rate. A perfect method will have an error rate of zero and recall and precision of 100%. The formula is defined by Pevzner et al (2002).

Precision rate(P) = C/M
Recall rate(R) = C/N
Error rate = E/N
F-Measure = (P*R*2)/(P+R)

Where,

N Number of words occurring in the manual segmentation

E Number of words incorrectly identified by the automatic method

C Number of words correctly identified by the automatic method

M=C+E Number of words identified by automatic segmentation

Some random texts were selected from the newspaper and the book for evaluating the accuracy of approach. The text is about 190 KB in size with 13025 words. For experimentation, we compare the automatic segmented text with the manual segmented text. The result of the two approaches on the testing text is as shown in Table 3.

Table 3: Results

	Word Bigram	**KCC Bigram**
No. of words	12760	11602
Precision Rate	91.562	72.327
Recall Rate	92.138	72.438
F-measure	91.849	72.382

4. Problems with KCC bigram model

One advantage to use the KCC Bigram model is that it requires less storage space. Since word is the combination of one or more KCCs, the total number of KCC tokens is predictably less than that of words and so do the co-occurrence of the two consecutive KCCs. However, it is the reason for high ambiguity. As illustrated in the result chart in the experimentation section, the ability to disambiguate using KCC Bigram model is low compared to another approach.

Another issue concerning with the KCC Bigram model is its high computation. The method requires more lookup for the co-occurrence between two consecutive KCCs. For example, let suppose each English character represents a syllable and we have a sequence of words **abc-de-fg**.

In order to compute the probabilities of the above word sequence in word Bigram model, we need to lookup two times for the co-occurrence between two consecutive words: Co(abc,de), Co(de,fg). However, in KCC Bigram model, we needs 7 lookups: Co(a,b), Co(b,c), Co(c,EOW), Co(d,e), Co(e,EOW), Co(f,g), and Co(g,EOW).

5. Conclusion

This document described the researches and techniques for the Khmer word segmentation. Many problems had been solved concerning the issues faced including the word segmentation ambiguities and unknown word identification.

Two statistical based approaches had been proposed to solve the segmentation ambiguities, word

Bigram model and Orthographic syllable Bigram model. According to the experimentation, Word Bigram outperforms the Orthographic syllable Bigram approach. Most of the segmentation errors cause by unknown words. However, we expect higher segmentation accuracy rate if the size of the training corpus is bigger.

As part of error words identification, the sound-similarity error identification from the previous research is used. Many types of unknown word are not handled yet. It is very difficult to identify whether a string is unknown word or some possible words. Different approaches are needed to detect different categories of unknown word.

6. Further works

Plenty of work needs to be completed to improve both the performance and the accuracy of the automatic word segmentation.

6.1. Unknown word identification

The ability to identify the new words and compound words in the sentence is very important for other NLP application. Since there is no obvious boundary between words in Khmer writing system, it is clear that the unknown word cause the segmentation abnormal. Consequently, the unknown word will be segmented into a piece of single KCC words and shorter words. For example, the word ខុមហ្វ្រ៉ែល, which is the name of an organization, the segmentation may result ខុ-ម-ហ្វ្រ៉ែ-ល after the dictionary look up and disambiguation algorithm.

Only sound-similarity error is solved in this phase of the project, but it is a very small contribution. Many further researches need to be conducted about this topic to improve the Khmer segmentation issue such as

- Abbreviation and acronym identification
- Proper names identification such as name of places, name of peoples, name of organizations
- Derived word identification
- Compound word identifications
- Numeric compound identification

Error word identification other than sound-similarity error such typing errors, real word errors.

6.2. Text corpus

The current segmentation method depends strongly on the text corpus. The size of the current text corpus is about 10 MB, which a very small amount. Therefore, more Khmer documents need to be collected and tagged. The bigger the corpus size, the better the accuracy of the segmentation.

6.3. Segmentation speed

The current speed of the segmentation is not high as expected. The main problem is that all the Bigram data are stored in the hard disk. Every access to the frequency of a Bigram required opening a file and searching in it. The average access to a Bigram frequency is 0.5 milliseconds. Two solutions can be used to solve the problem. First is to eliminate the number search of the Bigram frequency during the segmentation process. Second is to find a better method to search for the Bigram frequency. This needs to be done in the further research.

7. References

[1] A. Chen, *Chinese Word Segmentation Using Minimal Linguistic Knowledge*, School of Information Management and Systems, University of California at Berkeley, Berkeley, CA 94720, USA

[2] 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

[3] B. Shen, Z. Zhang, Chunfa Yuan, “Person Name Identification in Chinese Documents Using Finite State Automata”

[4] C.S. Huor, T. Rithy, R.P. Hemy, “Detection and Correction of Homophonous Error in Khmer Language”, PAN Localization Cambodia, 2006

[5] C.D. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts, London, England, Second Edition, 1999

[6] CHUON NATH, “Dictionnaire Cambodgien”, Edition de L’INSTITUT BOUDDHIQUE, Phnom Penh, 1967

[7] C.P. Papageorgiou, Japanese Word Segmentation by Midden Markov Model, BBN Systems and Technologies, 70 Fawcett St. Cambridge, MA 02138

[8] D. Jurafsky and J.H. Martin, Speech and Language Processing, “An Introduction to Natural Language Processing”, Computational Linguistics and Speech Recognition, 2000

[9] J. Gao, M. Li and C.N. Huang, “Improved Source-Channel Models for Chinese Word Segmentation”, Microsoft Research, Asia, Beijing 100080, China

[10] K.J. Chen and M.H. Bai, “Unknown Word Detection for Chinese by a Corpus-based Learning Method”, Computational Linguistics and Chinese Language Processing, vol. 3, no. 1, February 1998, pp. 27-44

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

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

[13] 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

[14] P. Sojka and D. Antoˇs, “Context Sensitive Pattern Based Segmentation: A Thai Challenge”, Faculty of Informatics, Masaryk University Brno, Czech Republic

[15] Pevzner, L. and Hearst, M.A(2002), “A Critique and Improvement of an Evaluation Metric for Text Segmentation”, Computational Linguistics, Vol. 28, No.1, pp.19-36, March 2002

[16] S.F. Chen and J. Goodman, An Empirical Study of Smoothing Techniques for Language Modeling, Computer Science Group, Harvard University, Cambridge, Massachusetts, August 1998

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

[18] T. Theeramunkong and S. Usanavasin, “Non-Dictionary-Based Thai Word Segmentation Using Decision Trees”, Information Technology Program, Sirindhorn International Institute of Technology, Thammasat University, Pathumthani 12121, Thailand

[19] W.Y. Ma, K.J. Chen, “Introduction to CKIP Chinese Word Segmentation System for the First International Chinese Word Segmentation Bakeoff”, Institute of Information science, Academia Sinica

[20] W. Aroonmanakun, “Collocation and Thai Word Segmentation”, Department of Linguistics, Faculty of Arts, Chulalongkorn University Phyathai Rd., Bangkok, Thailand