Khmer Word Segmentation and Out-of-Vocabulary Words Detection Using Collocation Measurement of Repeated Characters Subsequences

Channa Van and Wataru Kameyama

1 Introduction

Word segmentation is one of the fundamental operations in the Natural Language Processing (NLP) research fields. It has been actively studied by many researchers for several different natural languages especially for the Asian languages where words are usually written without any explicit boundaries. Many NLP research fields rely on it, including information retrieval, information extraction, part-of-speech tagging, machine translation and so on. The most common approach in word segmentation is lexicon-based one where the longest matching algorithm or the maximum matching algorithm is used. The accuracy of this approach totally depends on dictionary being used. However, as dictionary cannot cover all words of a language, this raises an issue of unknown words or out-of-vocabulary (OOV) words.

Like many Asian alphabets, Khmer script is also written without any delimiter, and this causes many issues in Khmer NLP, particularly in Khmer word segmentation. Only a few researches on word segmentation for Khmer have been conducted and implemented so far. Identifying OOV words such as compound words, proper names, acronyms and new words are also the main challenging tasks. Among such a few works of Khmer word segmentation, we have found two interesting works. The first one has been investigated by Seng et al. [1] using multiple text segmentation for statistical language modeling where the longest matching algorithm and the maximum matching algorithm have been investigated. The more numbers of OOV words increases, the more segmentation performance decreases. The second work is the implementation of Khmer word segmentation that has been done by a research group of Cambodia PAN Localization [2]. Two types of approaches have been investigated in their works: the word bigram model and the orthographic syllable bigram model. As both works are amongst the first implementations, a lot of issues have not been solved yet. The issues including the detection of OOV word such as abbreviations, acronyms, proper names, derivative words and compound words are stated to be solved in the future by the authors.

Among the non-segmented languages, only Japanese and Chinese have been intensively studied in the field of OOV detection. For Chinese, the N-gram generative language modeling based approach proposed by Teahan et al. [3]. Gao et al. [4] uses class-based language for word segmentation where some word category information is incorporated. Zhang et al. [5] use a hierarchical hidden Markov Model to incorporate lexical knowledge. Xue [6] uses a sliding-window maximum entropy classifier to tag Chinese characters into one of four position tags. Peng et al. [7] use conditional random field to detect the new words. Their work is currently achieving the state-of-the-art of Chinese word segmentation. For Japanese, Uchimoto et al. [8] have incorporated a probabilistic unknown word models as the feature function of a maximum entropy based morphological analyzer. Asahara and Matsumoto [9] have used them as the feature of character-based chunking of unknown words using support vector machine. Murawaki and Kurohashi [10] have proposed a lexicon acquirer using Japanese morpheme analyzer.

From all the previous mentioned studies, various statistical-based approaches are widely used to detect the unknown words. While Chinese and Japanese NLPs have been intensively researched, the language resources such as corpus or dataset for training and evaluation are not the big issue, while it is the key issue in Khmer NLP. Therefore, it is really difficult to achieve it by applying the modern machine learning techniques in the meantime in our research as these approaches require many resources. The task of creating a dataset is an intensive work that requires a lot of people getting involved in. Therefore, at this time, creating a good dataset for Khmer hasn't been done, yet.

Therefore, in this research, we propose a rule-based approach obtained by statistical analysis as well as the specific

Manuscript received 24th October 2012; revised 21st January 2013; Accepted 6th February 2013
Graduate School of Global Information and Telecommunication Studies, Waseda University

linguistic rules of Khmer. The proposed approach aims to tackle the issues of OOV words, particularly the issues of detecting the compound words, proper names/acronyms, derivative words and new words in the environment of low resource language. First of all, a corpus is created by just accumulated the texts that are found in the Web. Then, based on this raw corpus, we have applied our proposed rule learning algorithm in order to detect the OOV words without using any predefined information such as the part-of-speech tags or words in the training dataset. Rules are obtained from the corpus by detecting the repeated character subsequences in text based on SEQUITURE algorithm[14]. A repeated character subsequence is a subsequence that occurs in a character sequence or a text more than once. The statistical measurements are used to measure the strength of each rule, which is clarified in section 7. Then, the linguistic rules are used to detect the possible OOV words from the text.

In the following sections, we first describe the Khmer writing system in section 2, and our methodology of building a Khmer text corpus in section 3. After that, an explanation of the Khmer OOV word is detailed in section 4. In section 5, the proposed approach is presented. Then, it is followed by the detail of the proposed approach that includes the proposed rule-learning algorithm in section 6, the proposed statistical measurements in section 7, and the word extraction in section 8. Next, in section 9, we describe the experimental setup and procedure, and discuss the experimental results for each type of proposed statistical measurements compared with the Khmer word segmenter of Cambodia PAN Localization [2]. Finally, we conclude in section 10.

2 Khmer Writing System and the Issue of Khmer Word Segmentation

2.1 Khmer Writing System

Unicode is the only existing encoding that can be used to encode the Khmer text. In the Unicode chart, the Khmer script consists of 35 consonants, 17 independent vowels, 16 dependent vowels, 13 diacritics, 7 punctuations, a special subscript sign and several other signs. Words can be formed by only a consonant or the combination of consonants, vowels, subscripts and diacritics together. These symbols are arranged in 5 layers as shown in Fig. 1.

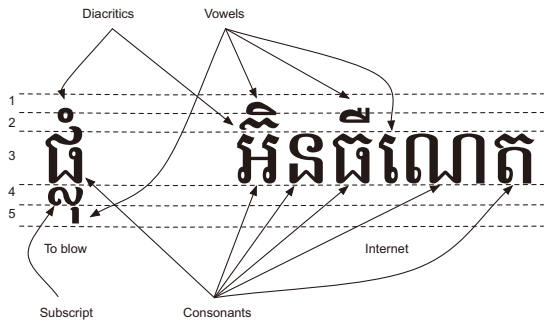

Fig. 1 The 5 Layers of Khmer Writing System.

In Khmer text, words are written continuously without any delimiter. Fig. 2 shows an example of Khmer phrase, which consists of 3 words: Cambodia, kingdom and wonder. The first line is the original text written in Khmer script, and the second line shows the boundary of the words in the sentence by the vertical lines.

កម្ពុជាព្រះរាជាណាចក្រអច្ឆរិយៈ

កម្ពុជា|ព្រះរាជាណាចក្រ|អច្ឆរិយៈ

Cambodia the kingdom of wonder

Fig. 2 An Example of Words in a Khmer Phrase.

2.2 Issues of Khmer Word Segmentation

There are two obvious issues in Khmer word segmentation: the over-segmentation and the word-segmentation ambiguity. The over-segmentation issue is the most common one. It is mostly caused by the OOV words found in the text. Fig. 3 shows the two example of the over-segmentation. The first one the name Obama is segmented into 3 different clusters of characters by using dictionary-based approach. Also the second example the OOV word "supermarket" is incorrectly over-segmented into two different words. The issue dramatically decreases the precision and recall of the word segmentation as it enlarges the number of incorrect segmented terms while decreases the correct segmented words.

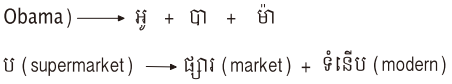

Fig. 3 Example of Over-segmented of OOV Words.

The word-segmentation ambiguity issue is rather rare in Khmer word segmentation. But occasionally, two words may be incorrectly segmented into a single word. Fig.4 shows an example of word-segmentation ambiguity.

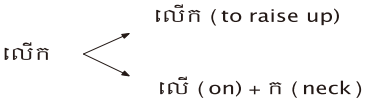

Fig. 4 Example of Word-segmentation Ambiguity

As the majority of the issues in Khmer word segmentation are caused by the over-segmentation mainly by the OOV words, solving the problem of OOV words is the most crucial work in order to increase the performance of word segmentation.

3 Corpus Collection

Like many other minor Asian languages, language resource is one of the major issues in the development of Khmer NLP. There is no standard Khmer text corpus, which can be used for hypotheses testing or statistical analysis. In order to achieve the goal of our proposed approach, a large collection of Khmer text corpus is required. Hence, the building of a Khmer text corpus is the first task to be completed. Our corpus-building task is focus on collecting the Khmer text encoded in Unicode.

Building a large text corpus for Khmer is quite a challenging task due to the current limitation of available Khmer digital texts as well as the lack of Khmer optical character reader that prevents us from using hard copy resources such as books and magazines. Therefore, the Internet Web pages are the only resource from which we can collect Khmer digital texts. Most of the texts in our corpus have been collected from the daily-published newspapers Websites. However, most of the Websites in Khmer Unicode text are just recently available; therefore, the number of the contents are still limited. The collection has been done for 2 years from various sources of Websites in Khmer.

We have built a corpus to be useful not only for our own research but also for any other Khmer NLP researches. Thus the corpus has been built including the annotations of words, sentences and part-of-speeches. These annotations have been done based on the previous researches of Khmer NLP. An implementation of Khmer word segmentation from Cambodia PAN Localization [2] has been used to segment words, while the part-of-speech annotation is based on the Khmer part-of-speech tagger proposed by Nou et al. [11]. Furthermore, to ensure the extensibility of the corpus in the future, the eXtensible Corpus Encoding Standard (XCES) [12] has been applied for the corpus encoding. We have achieved about 3 million words within twelve different domains as shown in Table 1.

Table 1 The Corpus Statistics.

Domain	Articles	Sentences	Words
Newspaper	5523	66397	2341249
Magazine	52	1335	42566
Medical	3	76	2047
Technical	15	607	16356
Culture	33	1178	43640
Law	43	5146	101739
History	9	276	7778
Agriculture	29	1484	30813
Essay	8	304	8318
Short Story	108	5642	196250
Novel	78	12012	236250
Other	5	134	5522
Total	**5906**	**94591**	**3000139**

4 Khmer Out-of-Vocabulary Word Rules

4.1 Out-of-Vocabulary Word

In this subsection, some Khmer OOV words are presented in order to understand Khmer better. We divide Khmer OOV words into 4 groups: compound word, proper name/acronym, derivative word, and new word. The detail of each type of OOV words are described as follows:

4.1.1 Compound Word

Since compound words are simply created by combining two or more words together, it is very difficult to recognize automatically this type of OOV words. There are no specific rules of creating a compound word, while many compound words exist in Khmer. For example:

- ឡានក្រុង (bus) = ឡាន (car) + ក្រុង (town or city).
- ផ្កាយដុះកន្ទុយ (comet) = ផ្កាយ (star) + ដុះ (to grow) + កន្ទុយ (tail).

We define a compound word by its meaning. If two or more words combine together creating a new meaning, we consider that the combination of these words is a compound word.

4.1.2 Proper Name/Acronym

Proper name/acronym is a big challenge in OOV recognition not only in Khmer. Recognition of these words depends on the complex linguistic rules, which are different from language to language. This type of OOV words consists of person's name, location name, organization name and abbreviation. Due to the fact that there is no resource and no study on Khmer named-entity recognition, it is quite difficult to recognize them. In this research, some simple rules of Khmer named-entity are proposed. These rules are based on the simple patterns of Khmer named-entity for the names of people, locations and organizations. A person's name in Khmer is usually preceded by an honorific title. Different honorific titles are used according to gender, age and social status of each person. Similar to that of a person's name, location and organization names are also usually preceded by a word which indicates the place or the organization such as city, country, district, school, hospital, company and so on.

4.1.3 Derivative Word

Due to the heavy influence of the old Indian languages (Sanskrit and Pali) in Khmer, some grammar aspects from these languages have also been imported, especially the derivation rules of words. The derivation is generally used to modify the part-of-speech of a word. It is similar to English like the word "creation" is derived from the verb "create". The derivation is carried out by prefixing and suffixing [13].

The prefixing and suffixing rules are shown as follows:

- <*Prefix*> + <*Noun, Verb or Adjective*> = <*Noun*>
- <*Verb or Adjective*> + <*Suffix*> = <*Noun*>

The most common prefixes in Khmer are ការ, ភាព, សេចក្តី, អ្នក, ជន, សភាព, ក្តី, while the common suffixes are កម្ម, ភាព, សាស្ត្រ, វិទ្យា, កិច្ច. For example, to create a noun of the verb រាំ (to dance), we just do the prefixing: ការ <prefix> + រាំ = ការរាំ (a dance), and the derivative word សង្គមវិទ្យា (social science) is derived from សង្គម (social) + វិទ្យា <suffix>.

4.1.4 New Word

Most of the new words in Khmer, which are usually technical terms, are borrowed from other languages, especially English. These words are included in Khmer lexicons by using the transliteration technique. For example, កុំព្យូទ័រ (computer) and អ៊ិនធឺណេត (Internet) are transliterated from English.

4.2 Rules

As presented in the subsection 4.1, it is very difficult to define any specific rules of OOV words for the groups of compound words and new words because there are no any specific recognizable patterns in these groups. On the other hand, we can define some rules for the proper name/acronym, and derivative words based on the grammatical rules as follows.

4.2.1 Proper Name/Acronym Rule

As mention in subsection 4.1.2, most of the proper names/acronyms are usually preceded by some specific words. So, we predefine a set of these words called indicative words that are served to detect the OOV words following them. In general, a proper name/acronym is incorrectly split into different parts, which are characters or strings, during the word segmentation process. In other words, a proper name/acronym are segmented into a series of characters and strings, where the last term of the series is determined when its following term is recognized as a word by using a word list. Therefore, we define a rule of the proper name/acronym detection as follows:

$$t_n + t_{n+1} + \cdots + t_{n+k} \rightarrow w \; if \; t_{n-1} \in I \; and \; t_{n+k+1} \in D \quad (1)$$

Where t_n to t_{n+k} are the sequence of characters or strings obtained by the incorrect segmentation, t_{n-k} and t_{n+k+1} are respectively the preceding term of t_n and the following term of t_{n+k}, w is a new OOV word, I is a set of predefined indicative words, and D is a dictionary.

4.2.2 Derivative Word Rule

The derivative word rules are also based on the grammatical rules of Khmer, specifically the prefixing and suffixing. A set of prefixes and suffixes are predefined, and any terms following a prefix or leading a suffix are grouped together with the prefix or the suffix to form a new OOV word as follows:

$$if \; t_n \in P \; then \; t_n + t_{n+1} \rightarrow w \quad (2)$$

$$if \; t_n \in S \; then \; t_{n-1} + t_n \rightarrow w \quad (3)$$

Where t_{n-1}, t_n and t_{n+1} are the terms in a segmented text, w is a new OOV word, while P and S are the predefined sets of prefixes and suffixes, respectively.

5 The Proposed Approach

The OOV word recognition is very important in the word segmentation in order to achieve high accuracy of segmentation. Therefore, the goal of our proposed approach is to identify the OOV words, which are not detectable by using dictionary-based approach, as many as possible. In general, as most of the OOV words are widely scattered in texts, the character subsequence of each OOV word shall be identifiable using some specific techniques to scan through a large collection of texts. We propose a rule-based approach that rules can be easily trained by using a large collection of texts based on a specific rule-learning algorithm. A rule learning algorithm based on the SEQUITUR algorithm [14] is used to create the rules of the repeated character subsequences. The statistical measurements are also incorporated in order to measure the frequency of the collocation of the rules. Moreover, the specific linguistic rules of Khmer OOV including the Khmer named-entity rules and the derivative rules, which are described in subsection 4.2, are also applied for the better achievement of OOV recognition as well as the word segmentation.

Fig. 5 illustrates our proposed approach. In the rule-learning step, the system extracts rules of repeated character subsequences by using a rule-extracting algorithm. The Khmer text corpus described in the section 3 is used for the training. After completing the rule learning, the rule tagging step and the word extraction step are carried out. The rule tagging is done based on different kind of statistical measurements that are entropy, mutual information

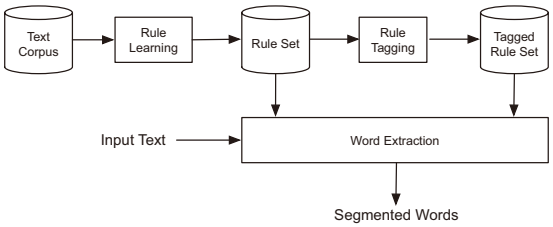

Fig. 5 The Proposed Approach.

[15], mutual dependency, log-frequency biased mutual dependency [16], and chi-square test. These statistical measurements are independently applied to weight the strength of each rule to be a word. A text is segmented into words based on the matching of the rules in the word extraction step. If a subsequence in the text matches to a qualified rule, that subsequence is taken as a word and is segmented. Finally, the linguistic rules are applied to optimize the segmentation result. The detail of the rule learning, rule tagging and word extraction are described in section 6, 7 and 8.

6 Rule Learning

Rule learning is a process of learning rules from a training text corpus. There are two main steps in the rule training process: subsequence extraction and rule extraction as shown in Fig. 6. First, the longest word matching algorithm is used to segment the texts by using a Khmer word list. The outcome is an array of extracted terms. Then we apply a rule-extracting algorithm to discover the rules of the subsequences that appear more than once from the array of extracted terms. Finally, a rule set is obtained based on the training corpus.

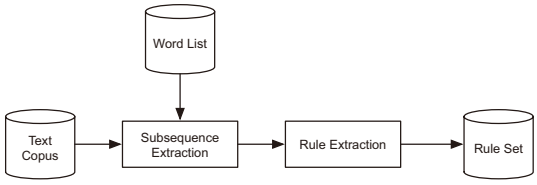

Fig. 6 The Rule-Learning Process.

6.1 Subsequence Extraction

In order to learn rules for the training corpus, first of all we need to split all texts in the corpus into terms. To do this, the longest word matching algorithm is used. The algorithm extracts a term from texts based on a word list. The word list consists of 32000 words based on the Khmer spelling dictionary of the National Language Institute [16]. The algorithm starts at the first character in a text using a word list, and attempts to find the longest word in the word list starting with that character. If a word is found, the word is extracted, and then the algorithm restarts the same longest match search starting at the next character of the match. If no match is found in the word list, the algorithm simply segments that characters as an extracted subsequence, and begins the search starting at the following character [18]. The output of the algorithm is an array of extracted subsequence based on the word list.

6.2 Rule Extraction

6.2.1 The Rule-Extracting Algorithm

The rule-extracting algorithm is based on the SEQUITUR algorithm proposed by Manning and Witten [14]. The SEQUITUR algorithm detects the hierarchical structure of a text by discovering the repeated subsequences from the text, and each repeated subsequence is replaced by a unique rule. The algorithm also forms a grammar or a rule set which consists of the rules of repeated symbols. There are two constraints of the rule formations. First, no pair of symbols appears more than once in the grammar. Second, every rule is used more than once. The first constraint requires for uniqueness of each rule, while the second constraint ensures the usefulness of each rule.

As the SEQUITUR algorithm is able to discover the rules, i.e. the repeated subsequence of s, we apply the algorithm in our rule extracting process. However, unlike the original goal of the SEQUITUR algorithm, our rule-extracting algorithm only attempts to discover the repeated subsequences in the training text and creates rules of the discovered subsequences. Thus, a database of rules is just created by the algorithm. Each rule is to be unique, and it can be expressed in 2 different forms: the expansion form and the bigram form. In the expansion form, a rule is a character subsequence. While in the bigram form, a rule is a bigram of rules and/or subsequences shown as follows:

$$R_i \leftarrow XY \tag{4}$$

Where X is a subsequence or a rule, Y is a subsequence or a rule and R_i is the i^{th} rule.

Due to the different goal of our algorithm and the SEQUITUR algorithm, a part of the SEQUITUR algorithm cannot be used. Therefore, only the first constraint of the SEQUITUR algorithm is used in the formation of rule in order to ensure the uniqueness of each rule, while the second constraint is excluded. This is because that the second constraint can create a rule, which is not a bigram of rules and/or subsequences as shown in the equation 4. Furthermore, the second constraint possibly removes the some rules, which are very useful the collocation measurement. Each subsequence represented by rule is very important, as they are the candidates of OOV words. Thus, all the rules have to be kept for the OOV word detection using statistical analysis later on.

Moreover, while the original SEQUITUR algorithm is based on a sequence of characters to find the repeated subsequences of characters, our SEQUITUR-based algorithm is applied to the array of subsequences to find the rules of the repeated subsequence of the subsequences. This approach is very useful to detect the OOV words such as compound words because of the high collocation probability

between the strings in compound words. Therefore, the array of subsequences, which is obtained by the subsequence extraction process as mentioned in subsection 6.1, is used for the rule extracting.

Algorithm 1: Rule-Learning

Input: A training text and an old rule set.
Output: A new rule set.
ArraySeq = LongestWordMatching (*TrainingText*);
ArraySeq = Initialize (*ArraySeq, RuleSet*);
index = 0;
RuleIndex = Length (*RuleSet*);
while *index* < (Length (*ArraySeq*) − 1) **do**
  Subsequence = *ArraySeq* [*index*] + *ArraySeq* [*index* + 1];
  if CountSequence (*Subsequence, ArrayString*) > 1 **then**
    RuleSet [*RuleIndex*] = *Sequence*;
    RuleIndex++;
    ArraySeq = ReplaceSequence (*Subsequence, ArraySeq*);
  else
    index++;
  end
end
return *RuleSet*;

Algorithm 1 describes the proposed algorithm of rule extraction in a pseudo code. The algorithm starts with the subsequence extraction using the longest word-matching algorithm. Next, it does the initialization on the extracted subsequences by replacing subsequences with rules that are matched to rules in the rule set. Then, starting from the first sequence of the two tokens of the extracted subsequences, it attempts to discover the number of occurrences of the subsequence. If there is only one occurrence, it proceeds to the next token. Otherwise, a new rule representing the sequence is created. Then, it goes through the extracted subsequences, and tries to replace all the same sequences with the new created rule. After that, it restarts the loop in order to search for other frequent subsequences from the current token position. This action is repeated until it reaches the last subsequence of two tokens. Finally, the outcome is a new rule database based on the training texts.

6.2.2 Functional Words

Functional words are words that have a little lexical meaning or an ambiguous meaning, but serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker. They signal the structural relationships between words and are the glue that holds sentences together. Thus, they serve as important elements to the structures of sentences. Some examples of functional word in Khmer are pronouns, conjunctions and ordinal counters. These words are less important for rule extraction due to its less probability to collocate with other words to form a new word. Therefore, the rule-extracting algorithm skips all the found functional words.

7 Rule Tagging

The rule tagging is an automatic operation to tag each rule based on collocation measurement strength of the bigram of each rule. Six types of collocation measurements are independently used in order to find out the best suitable statistical approach for the rule tagging. These six collocation measurements are: left entropy, right entropy, mutual information, mutual dependency, log-frequency biased mutual dependency, and chi-square test. A threshold value is used for each measurement. If it is greater than the threshold value, the rule is tagged to be a word candidate, which means that the subsequence represented by that rule is a word candidate. Any subsequences that match these rules in the rule matching are segmented. In the following subsections, we describe each type of the proposed statistical measurements, one by one.

7.1 Entropy

Two types of entropies are calculated for each rule according to its adjacent strings: the left entropy and the right entropy. The left entropy is measured in association with any strings, which are found before the considered rule, while the right entropy is measured in association with any strings found after the considered rule. If the considered rule is a word, its preceding and following strings should be various, thus its left and/or right entropy shall be high enough. The left and right entropies are computed by the following formulas:

$$LE(xR) = -\sum_{\forall x \in S} P(xR|R)\ \log_2 P(xR|R) \tag{5}$$

$$RE(Ry) = -\sum_{\forall y \in S} P(Ry|R)\ \log_2 P(Ry|R) \tag{6}$$

Where R is the considered rule, S is a set of unique strings obtained by the longest word matching algorithm, x and y are any strings in S occurred on the left and right of the rule R respectively, LE and RE are the left and the right entropies, and P is the probability.

7.2 Mutual Information

The mutual information (MI) of two random variables is a quantity that measures the mutual dependency of the two variables [15]. If the mutual information of both elements in the bigram form of a rule is high enough, then both elements shall be a word rather than co-occurred by chance, and it is highly probable that the subsequence represented by that rule is a word. The mutual information of each rule is computed by the following formula:

$$I(X,Y) = \log_2 \frac{P(R)}{P(X)P(Y)} \quad (7)$$

Where R is the considered rule as $R \leftarrow XY$, I is the MI and P is the probability.

7.3 Mutual Dependency and Log-Frequency biased Mutual Dependency

In the study on the comparative evaluation of the collocation extraction metrics, Thanopoulos et al. [16] have investigated two information theoretic measures, that are mutual dependency (MD) and log-frequency biased MD (LFMD), against other statistical collocation measurements. They have stated that MI is actually a measure of independence rather than of dependence. They have suggested that the dependence can be identified by subtracting the self-information from the MI. The log-frequency biased mutual dependency (LFMD) is the combination of the frequency and the MD of information. The slight bias towards frequency can be beneficial reflecting statistical confidence; among similarity dependent bigrams, the most frequent one should be favored [16]. The equations of MD and LFMD are as follows:

$$MD(X,Y) = I(X,Y) - I(R) = \log_2 \frac{P^2(R)}{P(X)P(Y)} \quad (8)$$

$$LFMD(X,Y) = \log_2 P(R) + MD(X,Y) \quad (9)$$

Where R is the considered rule as R, $\leftarrow XY$, $I(X,Y)$ is the MI of X and Y, $I(R)$ is the self-information of the rule R, $MD(X,Y)$ is the MD of X and Y, $LFMD(X,Y)$ is the LFMD of X and Y and P is the probability.

7.4 Chi-square Test

The essence of the test is to compare the observed frequency table with the frequencies expected for the independence [19]. In our case, the chi-square test is applied to 2-by-2 table as shown in Table 2. This table shows the dependence occurrences of the X and Y of each rule with the frequencies of the bigram $w_i w_{i+1}$ where w_i can be X or others, and w_{i+1} can be Y or others.

The simpler form of chi-square test for the 2-by-2 table is shown as follows:

$$\chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11}+O_{12})(O_{11}+O_{21})(O_{12}+O_{22})(O_{21}+O_{22})} \quad (10)$$

Table 2 A 2-by-2 Table Showing the Dependence of Occurrences of the X and Y.

	$w_i = X$	$w_i \neq X$
$w_{i+1} = Y$	O_{11}	O_{12}
$w_{i+1} \neq Y$	O_{21}	O_{22}

Where N is the size of the tokens and O_{ij} is the frequency in the cell (i, j) of Table 2.

8 Word Extraction

This section describes the process of extracting words from a text based on the rule set which is obtained by the rule learning as describe above. Fig. 7 shows the overall process of word extraction. First, the subsequence-to-rule initialization is performed on the input text based on the rule set. Then in the rule matching, the rules obtained from the input text are matched to the rules in the tagged rule set. Finally, the linguistic rules applied to improve the OOV detection performance. The successful matching rules are extracted as the segmented words.

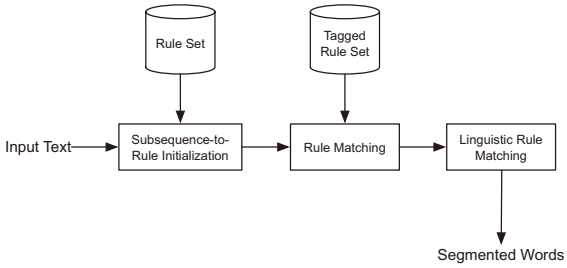

Fig. 7 The Words Extraction Process.

8.1 Subsequence-to-Rule Initialization

In this step, the input text is initialized based on the rule set obtained by the rule learning. First, the longest word-matching algorithm is applied to break the input text into subsequence based on a dictionary. Then, all the subsequences, which match to the rules, are replaced by the corresponding rules. Finally, we obtain an array of subsequence and rules of the input text using the rule initialization process.

8.2 Rule-Matching

ALGORITHM 2: Rule Matching

Input: The array of strings and rules of input text obtained by the rule extraction, a tagging rule set obtained by the rule tagging using a statistical measurement.

Output: The segmented words.

```
index = 0; WordIndex = 0;
while index < Length (ArrStrRule) do
  if IsString (ArrStrRule [index]) == True then
    SegWords [WordIndex] = ArrStrRule [index];
    WordIndex++;
  end
  else
   if match (ArrStrRule[index], TaggedRuleSet) == True then
    SegWords [WordIndex] = ArrStrRule [index];
    WordIndex++;
```

```
    end
  else
    RuleElements = Decompose (ArrStrRule [index]);
    for each Element in RuleElements then
      RuleMatching (Element, TaggingRuleSet);
    end
  end
  index + +;
 end
end
return SegWords;
```

In this step, we try to match the rules, which are obtained by the rule tagging, to the extracted rules from the input text. Algorithm 2 shows the pseudo code of the proposed rule-matching algorithm. First, the algorithm starts from the first token of the subsequence and rules obtained by the rule extraction as described in 6.2. Then, it attempts to find the matching between rules of a tagging rule set and rules of the input subsequence. If the token is a string, it is extracted as a word. Otherwise, a rule matching is carried on. If the considered token matches to a rule in the set, then it is qualified to be a word. If not, it is decomposed into two elements where each can be a string or a rule. Next, the rule-matching algorithm is reapplied on each decomposed elements one by one. It is a recursive operation until the considered element is found as a string or matches to a rule in the rule set. When the rule matching of the first token is completed, the algorithm moves to the next token, and it continues the matching process until it reaches the last token of the sequence. By doing this, we finally obtain the segmented words of input text based on a tagging rule set.

8.3 Linguistic Rule-Matching

In addition to the matching of rules, which are obtained by the statistical learning, the Khmer linguistic rules including named entity, rules and derivation rules as described in section 4, are also employed. The linguistic rule matching is done at the final stage of the word extraction in order to optimize the result of OOV recognition.

9 Experiments and Results

9.1 Test Data

We have randomly selected 20 articles of Khmer texts from the web. It consists of 6446 words by manual segmentation. These words are considered as the ground truth to evaluate the segmentation performance based on our proposed approach. Among these words, 1066 words are OOV words that are 16.54% of the total words. The majority of the OOV words are compound words, which account 42% followed by 25%, 25% and 8% for proper name/acronym, derivative word, and new word, respectively. We divide the 20 articles into two groups. One group of 5 articles is used in the threshold tuning in order to find out the optimum values of the threshold, while the other is used to test the segmentation performance of our proposed approach. Fig. 8 shows the distribution of known words and OOV words in the test data.

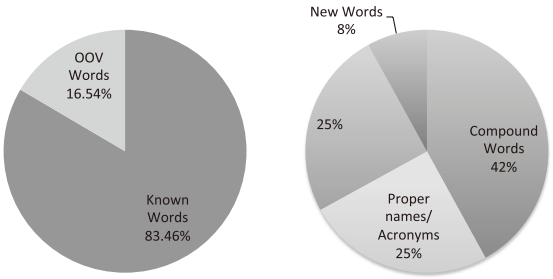

Fig. 8 The Distribution of Known Words and OOV Words (left) and the Distribution of the OOV Words by Category in the Test Data (right).

9.2 Experimental Setup and Procedure

The experiments were conducted on the test data based on the different tagging rule sets obtained by different kinds of statistical measurements presented in section 7. Precisions, recalls and F-measures were calculated for each type of proposed statistical measurements to evaluate the segmentation performance. These experiments are done based on the learning statistical rules only in order to find the best statistical measurement. After that, the best result is selected for applying the linguistic rules, and the evaluation of OOV recognition is done based on the obtained result. Furthermore, we also carried out the comparison of our results to the baseline and the current state-of- the-art which is Cambodia PAN Localization (PAN) [2]. The baseline refers to the proposed approach carried out without the statistical measurements. In other words, no rule tagging has been done to obtain the baseline result. We introduce the baseline in order to evaluate the improvement obtained by the proposed statistical measurements.

9.3 Threshold Tuning

In order to observe the variation of segmentation performance for each statistical measurement, different threshold values were employed in the rule tagging. These threshold values were selected amongst the best observed accuracy of word segmentation of each statistical measurement. Over 100 thousands of rules have been extracted from the training corpus. The tuning is based on 5 articles of the test data. Table 3 shows the best 5 threshold values for each proposed collocation measurement we have ever achieved.

Table 3 The Threshold Values Used in Each Type of Statistical Measurement.

	Thresholds				
	1	2	3	4	5
Right Entropy (RE)	1.5	2	2.5	3	3.5
Left Entropy (LE)	1.5	2	2.5	3	3.5
Mutual Information (MI)	1	2	3	4	5
Mutual Dependency (MD)	−11	−10	−9	−8	−7
Log-Frequency biased MD (LFMD)	−30	−27	−25	−23	−21
Chi-square Test	4000	5000	7500	10000	12500

9.4 Experimental Results

Table 4, Table 5, Table 6 and Table 7 clearly show the results of Khmer word segmentation obtained by our proposed approach using different kinds of statistical measurements. Comparing to the baseline, the proposed approach can significantly achieve higher recalls and F-measures in all cases, while the better precisions can be observed in a few cases. The proposed statistical measurements noticeably

Table 4 The Comparative Results of Precision between Each Statistical Measurement.

	Threshold Number				
	1	2	3	4	5
RE	0.755	0.754	0.752	0.751	0.733
LE	0.768	0.773	0.767	0.759	0.746
MI	0.777	0.774	0.772	0.760	0.741
MD	0.775	0.766	0.762	0.745	0.726
LFMD	**0.788**	0.783	0.770	0.752	0.725
Chi-Square	0.765	0.764	0.760	0.754	0.747

Table 5 The Comparative Results of Recalls between Each Statistical Measurement.

	Threshold Number				
	1	2	3	4	5
RE	0.782	0.795	0.811	0.829	0.832
LE	0.766	0.788	0.799	0.815	0.830
MI	0.756	0.756	0.761	0.762	0.767
MD	0.801	0.817	0.835	0.840	0.842
LFMD	0.796	0.825	0.847	**0.855**	0.847
Chi-Square	0.817	0.824	0.831	0.835	0.835

Table 6 The Comparative Results of F-measure between Each Statistical Measurement.

	Threshold Number				
	1	2	3	4	5
RE	0.768	0.774	0.780	0.788	0.779
LE	0.767	0.780	0.782	0.786	0.785
MI	0.766	0.765	0.766	0.761	0.753
MD	0.787	0.790	0.796	0.789	0.780
LFMD	0.792	0.803	**0.807**	0.800	0.781
Chi-Square	0.790	0.792	0.793	0.792	0.788

Table 7 The Segmentation Result of PAN and Baseline.

	Precision	Recall	F-measure
PAN	0.718	0.788	0.751
Baseline	0.777	0.755	0.765

boost the effectiveness of segmentation. In addition, when compared with PAN, a remarkable improvement can be also observed according to the tables. The baseline slightly outperforms the PAN in terms of precision and F-measure, while a slightly better recall can be observed for PAN. All in all, LFMD outperforms the other statistical measurements as well as PAN and the baseline where the maximum accuracy can be reached when using a threshold equals to −25.

Furthermore, Fig.9 explicitly demonstrates the comparative graph of the best F-measure results between the proposed approach using LFMD (threshold=−25), the baseline and PAN. The case of applying the linguistic rules is also shown in the comparison. Up to 0.807 of F-measure can be acquired by using LFMD with a threshold equals to −25. When the linguistic rules are used, 0.835 of F-measure can be reached. On the other hand, 0.765 of F-measure is able to achieve in the baseline while only 0.751 of F-measure can be obtained by using PAN. This obviously shows a notable amelioration of the proposed approach without or with the proposed statistical measurements as well as the linguistic rules when compared with the current stat-of-the-art.

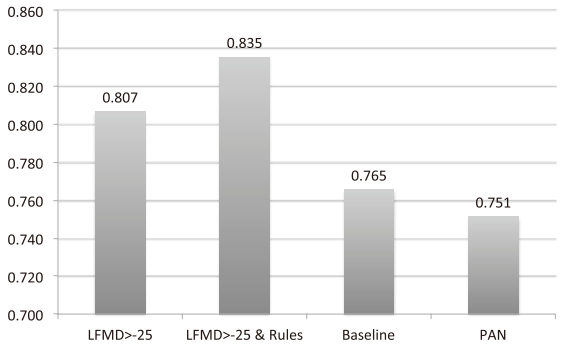

Fig. 9 The Best Results of F-measure of the Baseline and PAN.

9.5 Discussion

Based on the experimental results, the baseline can achieve a higher precision, while a lower recall is obtained. This is due to the decrease in number of the segmented strings as well as the correct segmented words. However, a significant improvement of F-measure is obtained comparing to PAN, which is based on the word bigram model. It shows that the proposed algorithm of rule extraction, which is based on the SEQUITUR algorithm,

helps to achieve a better rate of OOV word recognition as well as the word segmentation. Moreover, by employing the proposed statistical measurements, better results are able to attain with increasing the number of correct segmented words, i.e. the recall. In general, most of the proposed statistical measurements remarkably show the increase in F-measure except for the MI where a slight improvement is acquired and is worse than the baseline in some cases. It proves that these statistical measurements assist to precise the strength of each rule whether it shall be used for the matching or not. Moreover, the linguistic rules also help to achieve more accurate results in terms of OOV word recognition.

Next, we discuss the best word segmentation result of our proposed approach. It is obtained by using LFMD combining with the linguistic rules in order to evaluate the OOV detection rate. In total, there are 460 wrong segmented words where 28.91% are compound words, 17.83% are proper names/acronyms, 2.83% are derivative words, 7.83% are new words, and 42.61% are wrong detective words. Nevertheless, we can observe a significant improvement of OOV word detection as illustrated in Fig.10. Our proposed approach with LFMD and the linguistic rules can achieve the ameliorations of detection rate by 52.20%, 60.19%, 90.09% and 42.86% of compound word, proper name/ acronym, derivative word, and new word, respectively. These percentages show that the LFMD and the linguistics rules remarkably improve the OOV word detection especially in case of derivative word, compound word and proper name/acronym. Furthermore, the outstanding results of LFMD and MD compared with the other measurements reveal that the dependency of the bigram in a rule is an important factor to evaluate its strength as a word. It is similar to the result of Thanopoulos et al. [16] where they have found out that LFMD outperforms the other statistical measurements in the case of English.

Next, we discuss the best word segmentation result of our proposed approach. It is obtained by using LFMD combining with the linguistic rules in order to evaluate the OOV detection rate. In total, there are 460 wrong segmented words where 28.91% are compound words, 17.83% are proper names /acronyms, 2.83% are derivative words, 7.83% are new words, and 42.61% are wrong detective words. Nevertheless, we can observe a significant improvement of OOV word detection as illustrated in Fig.10. Our proposed approach with LFMD and the linguistic rules can achieve the ameliorations of detection rate by 52.20%, 60.19%, 90.09% and 42.86% of compound word, proper name/ acronym, derivative word, and new word, respectively. These percentages show that the LFMD and the linguistics rules remarkably improve the OOV word detection especially in case of derivative word, compound word and proper name/

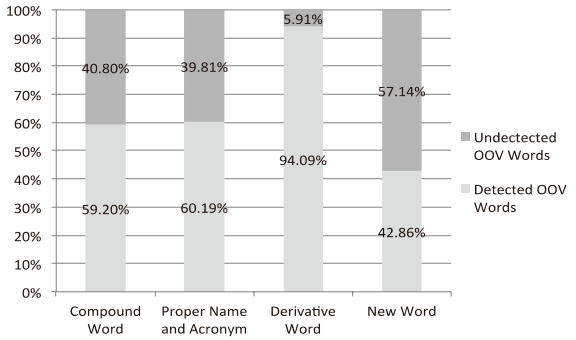

Fig. 10 The Detectable and Undetectable Rate of OOV words by Category Using LFMD with Threshold -25 Combining with Linguistic Rules.

acronym. Furthermore, the outstanding results of LFMD and MD compared with the other measurements reveal that the dependency of the bigram in a rule is an important factor to evaluate its strength as a word. It is similar to the result of Thanopoulos et al. [16] where they have found out that LFMD outperforms the other statistical measurements in the case of English.

10 Conclusion

We have presented and discussed a new proposed trainable and statistical rule-based approach in cooperated with the Khmer linguistic rules for the OOV word detection in Khmer word segmentation. A rule-learning algorithm has been shown to be an essential part of the proposed approach in order to detect the repeated character subsequences that are possibly words. In addition, different studies of the proposed approach based on different collocation statistical measurements also have been investigated. The experimental results show that our proposed approach can achieve significant accuracy of Khmer OOV word recognition as well as the Khmer word segmentation. It has also demonstrated that the linguistic rules are the major factors to improve the OOV word recognition. Besides the mentioned collocation measurements in our experiments, other types of measurements are also expected to increase the segmentation performance such as the maximum likelihood ratio, t-test and other collocation measurements that shall be investigated.

Furthermore, the proposed trainable approach significantly outperforms the implementation of Khmer word segmentation proposed by Cambodia PAN Localization, which is the current state-of-the-art. The outcomes from this fundamental research would be a great contribution to speed up the research of Khmer NLP while many are still struggling due to the lack of such important fundamental work. Many research fields of Khmer NLP including the information retrieval, information extraction,

part-of-speech tagging, machine translation and more, can fully take the benefits from this research.

REFERENCES

[1] Seng, S., Sam, S., Besacier, L., Bigi, B. and Castelli, E.: First Broadcast News Transcription System for Khmer Language, Proceedings of the Sixth International Language Resources and Evaluation (2008).

[2] Chea, S., Top, R. and Ros, P.: Word Bigram vs Orthographic Syllable Bigram in Khmer Word Segmentation, (online), available from http://www.panl10n.net/english/OutputsCambodia1.htm.

[3] Teahan, W. J., Wen, Y., McNab, R. and Witten, I. H. A Compression-based Algorithm for Chinese Word Segmentation. Computational Linguistics, 26 (3): 375–393 (2000).

[4] Gao, J., Li, M. and Huang, C. Improved Source-Channel Models for Chinese Word Segmentation. In Proceedings of the 41th Annual Meeting of Association of Computaional Linguistics (ACL), Japan (2003).

[5] Zhang, H., Liu, Q., Cheng, X., Zhang, H. and Yu, H. Chinese Lexical Analysis Using Hierarchical Hidden Markov Model. In Proceedings of the Second SIGHAN Workshop, pages 63–70, Japan (2003).

[6] Xue, N. Chinese Word Segmentation as Character Tagging. International Journal of Computational Linguistics and Chinese Language Processing, 8, (2003).

[7] Peng, F., Feng, F. and McCallum, A. Chinese segmentation and new word detection using conditional random fields. In Proc. of Coling-2004, pages 562–568, Geneva, Switzerland (2004).

[8] Uchimoto, K., Sekine, S. and Isahara, H. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In Procs. of EMNLP 2001, page 91-99 (2001).

[9] Asahara, M. and Matsumoto, Y. Extended models and tools for high-performance part-of-speech tagger. In Procs. of COLING 2000, page 21-27 (2000).

[10] Murawaki, Y. and Kuruhashi, S. Online acquisition of Japanese unknown morpheme using morphological constraints. In Procs. of EMNLP, pp 429-437 (2008).

[11] Nou, C. and Kameyama, W.: A Transformation-Based Approach with Hybrid Unknown Word Handling, Proceeding of International Conference on Semantic Computing (ICSC), Irvine, USA, pp.482–492 (2007).

[12] Ide, N., Bonhomme, P. and Rosmary, L.: XCES: An XML-Based Standard for Linguistic Corpora., Proceeding of Second Language Resources and Evaluation Conference (LREC), Athens, Greece, pp.825–830 (2000).

[13] Khin, S.: វេយ្យាករណ៍ភាសាខ្មែរ "Khmer Grammar", Royal Academy of Cambodia, first edition (2007).

[14] Nevill-Manning, C. and Witten, I.: Identifying Hierarchical Structure in Sequences, Journal of Artificial Intelligence Research, Vol.7, pp.67–82 (1997).

[15] Church, K.W., Robert, L. and Mark, L.Y.:A Status Report on ACL/DCL, pp.84–91 (1991).

[16] Thanopoulos, A., Fakotakis, N. and Kokkinakis, G.: Comparative Evaluation of Collocation Extraction Metrics, The 3rd Language Resources Evaluation Conference, pp.620–625 (2002).

[17] Long, S., Ev, C. and Chour, K.: Khmer Spelling Dictionary, Institute of National Language (2005).

[18] Indurkhya, N. and Damerau, F.J.: Handbook of Natural Language Processing, Chapman & Hall CRC, second edition (2010).

[19] Neville-Manning, C.D. and Schutze, H.: Foundations of Statistical Natural Language Processing, MIT Press, Cambridge (1999).