DE GRUYTER OPEN
RIGA TECHNICAL UNIVERSITY
Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
May 2017, vol. 21, pp. 61–65
doi: 10.1515/acss-2017-0008
https://www.degruyter.com/view/j/acss

Analysing the Methods of Dzongkha Word Segmentation

Parshu Ram Dhungyel[1], Jānis Grundspeņķis[2]
[1]*College of Science and Technology, Royal University of Bhutan, Bhutan,* [2]*Riga Technical University, Latvia*

***Abstract* – In both Chinese and Dzongkha languages, the greatest challenge is to identify the word boundaries because there are no word delimiters as it is in English and other Western languages. Therefore, preprocessing and word segmentation is the first step in Dzongkha language processing, such as translation, spell-checking, and information retrieval. Research on Chinese word segmentation was conducted long time ago. Therefore, it is relatively mature, but the Dzongkha word segmentation has been less studied by researchers. In the paper, we have investigated this major problem in Dzongkha language processing using a probabilistic approach for selecting valid segments with probability being computed on the basis of the corpus.**

***Keywords* – Dzongkha word segmentation, maximal matching, n-gram, natural language processing.**

I. Introduction

Dzongkha has been used as a popular language for communication in Bhutan since the 17th century. In 1972, the Third King of Bhutan, His Majesty the King Jigme Dorji Wangchuck declared it as the national language of Bhutan [1]. To preserve the culture and to enhance the language, computerisation of Dzongkha must be performed. From the review, we have found that only work on Dzongkha Unicode has been done; however, further processing of Dzongkha language needs a parser. Among the phases of Natural Language Processing (NLP), the segmentation of a sentence into a meaningful individual word is one of the necessary pre-processing tasks [2], [3] because a word is both syntactically and semantically fundamental unit for analysing the language structure. The segmentation phase can be considered as a training phase where different words within a particular natural language are introduced to the system, allowing the computer to recognise valid words.

The word introduction to the system can be done based on a dictionary and a text corpus. During the segmentation phase, the given article or sequence of words is broken down into paragraphs, then further to sentences and then to the individual words. The accuracy of higher phases of the project like POS tagging depends on the tokenisation phase. There are various approaches for word segmentation [4], among which we have adopted segmentation based on a probabilistic approach.

The probabilistic approach is used to select the valid segments, with probability being computed on the basis of the corpus. The corpus contains a structured set of texts. The Dzongkha Corpus developed by Dzongkha Development Commission of Bhutan (DDC) contains the valid words along with their POS tags. The corpus is collected from the domains listed in Table I.

TABLE I
Domain Area of Collected Corpus

Sl. No	Domain	Amount, %
1	World Affairs	12 %
2	Social Science	2 %
3	Arts	9 %
4	Past Work	20 %
5	Adventure	1 %
6	Culture	2 %
7	Story	20 %
8	Newspapers	25 %
9	Religious Texts	4 %
10	Songs and Poems	5 %

The accuracy of the segmentation depends on the type of document domain, as well as the size and quality of the corpus. The following text is a small fragment of the Dzongkha Corpus developed by DDC.

The corpus without annotations looks like:

ཉེ་མ་ གཡུས་ཅིག་དང་ མཉ་ཅིག་དང་མཉ་ཅིག་ཡོད་པ་མས། ཁོང་གཉིས་ལུ་གཅས་ག་ཞིང་དང་རྒྱུ་ཅ་ལ་ལེ་ཤ་ཡོང་རྟང་ མ་ལུ་ཕྱད་རྩོག་ཅིག་ཡང་ཤེད་པར་སེམས་གནམ་ཤེད་ལ་མེད་སྐྱོ་སྐྱེ་སྡོད་ཀྱག མ་རྟག་ར་ཁོང་ར་གཉིས་ གཅིག་གིས་གཅིག་ལུ་བཀྱ་སྐྱེ་ང་བཅས་ར་གཉིས་ཏན་ཏས་ལ་ནེམ་ཅིག་ཁར་སྡོད་ངང་ ཏུ་དང་བུམོ་ གཅིག་ཡང་ཤེད། ཤུལ་ལས་བྱཤ་མ་ཁོངས་དང་ གག་གིས་བཀྱ་ཅིག་གིས་གཅིག་ལུ་སྨན་ཅི་ར་སེམས་ སྐྱོ་སྐྱེ་སྡོད་ཀྱག

ཉེནམ་ཅིག་མཚ་དེ་རྒྱ་ལེན་པར་འགྱོ་ཤ་དང་ མཚ་འདི་འགྱོ་གའི་ལས་ཁར་མ་ལུ་རྒྱང་ཀྱ་ཅིག་ཐྱ་སྐྱེ་སྡོད་ས་ མཚོང་ནུགས་ལུ་འདི་འཐུ་འབག་སྐྱེ་ཁྱིམ་ནང་ཁོངས་ཞེནམ་ལས་ མ་ཁ་ལུ་སྐྱོནམ་ད་མཁ་དེ་གིས་ དཁའི་མ་ལུ་འདི་ལུ་ཁཚ་དང་སྨན་ཆ་གུ་ཡང་ཤེད་པ་མས། མ་ལུ་འདི་ ང་བཅས་ར་གཉིས་ཀྱིས་གསོ་གས་ལ་ཁྱེ་ཤེས་ སྨན་སྐྱེ་མ་ལུ་འདི་ཉེནམ་རིང་ཚད་ཤེད་པར་རྒྱ་འཐུ་ཡི་ར་ཚོ་ར་ཀྱི་ཚོམ་བྱེན་ཡི་ར་གསོ་བཞག་ནུག [5].

The corpus with annotations looks like:

ཉེ་མ་=avt}# གཡུས་=nc}ཅིག་=det}ནང་=nr}# མཉ་=nc}ཅིག་=det}དང་=cjc}མཚ་=nc}ཅིག་=det}ཡོད་པ་=ex+ne}མས=eep}|=sb}* ཁོང་=ppr}གཉིས་=cd}ལུ་=cal}གཅས་ག་ཞིང་=nc}དང་=cjc}# རྒྱུ་ཅ་ལ་=nc}ལེ་ཤ་=ajl}ཡོང་=ex}རྟང་=cjv}# མ་ལུ་=nc}ཕྱད་རྩོག་=ncl}ཅིག་=det}ཡང་=et}མེད་=exn}པར་=nfv}# སེམས་^གནམ་མེད་ག་མེད་=avd}^སྐྱོ་=vit}=vp}སྡེ་=cn}སྡོད་=vt}ནུག=tp}* མ་རྟག་ར་=avt}# ཁོང་ར་=pr}གཉིས་=cd}# གཅིག་གིས་གཅིག་ལུ་=prp}བཀྱ་=vt}སྐྱེ་=cn}# ང་བཅས་ར་=pr}གཉིས་=cd}ཏན་ཏས་=ajd}ལ་ཤེམ་ཅིག་ཁར་=phr}སྡོད་=vt}རྟང་=cjv}# ཏུ་=nc}དང་=cjc}ཁུམོ་=nc}གཅིག་=cd}ཡང་=et}མེད་=exn}|=sb}# ཤུལ་ལས་=avt}#བྱཤ་=na+ne}^མ་=neg}^འབངས་=vt+ne}=vp}ད་=pfv}# ག་=icq}གིས་=caa}བཀྱ་=vt}ཅི་=tf}ཟེར་=sr}#གཅིག་གིས་གཅིག་ལུ་=prp}སྨན་=vt}ཅི་=tp}ར་=et}# སེམས་སྐྱོ་=vit}སྐྱེ་=cn}སྡོད་=vt}ནུག=tf}*&

ཉེནམ་=na}ཅིག་=det}མཚ་=nc}དེ་=det}#རྒྱ་=nc}ལེན་=vt}པར་=nfv}འགྱོ་=vt+ne}ད་=pfv} མཚ་=nc}འདི་=det}འགྱོ་=vt}ས་=nm+cag}ལས་=nc}ཁར་=nr+cal}# མ་ལུ་=nc}རྒྱང་ཀྱ་=ajd}ཅིག་=det}#ཐྱ་=vit}སྐྱེ་=cn}སྡོད་=vt}ས་=nm}མཚོང་=vt}ནུག=tp}* མ་ལུ་=nc}འདི་=det}#འཐུ་འབག་=vt+vt}=vcp}སྐྱེ་=cn}#ཁྱིམ་=nc}ནང་=nr}ཁོངས་=vt}ཉེནམ་ལས་=cn}# མ་ཁ་=nc}ལུ་=cac}སྐྱེནམ་

=vt+ne}ད་=pfv}# ཨཡ་=nc}དེ་=det}གེས་=caa}# དབའི་=ij}#ཨ་ལུ་=nc}འདི་=det}ལུ་=cac}#ཤཨ་=nc}དང་=cjc}སྒྲན་ཆ་=nc}#ཀ་ཡང་=pi}མེད་ལ་=exn+ne}མས=eep}།=sb}*ཨ་ལུ་=nc}འདི་=det}# ང་བཅས་ར་=pr}གཉིས་=cd}གྱིས་=caa}གསོ་བཞག་=vt+vt}=vcp}ཏེ་=meh}སྣོ་=ipq}ཟེར་=sr}སྨས་=vt}སྟེ་=cn}#ཨ་ལུ་=nc}འདི་=det}#ཉེན་ས་རིང་ཚད་མེད་པར་=avt}ཚུ་འཛུ་=na+vt}=vcp}ཡི་=tp}ར་=et}#རོར་=nc}གྱི་=cag}ཨོམ་=nc}བྱེན་=vt}ཡི་=tp}ར་=et}གསོ་བཞག་=vt+vt}=vcp}ནུག=tp}*& [5].

The tag delimiters used in the Dzongkha Corpus are shown in Table II.

TABLE II
TAG DELIMITERS

Delimiter	Meaning
=	assign operator
}	word delimiter
+	fusion
*	clause/ sentence marker
#	pause/ phrase marker
&	paragraph marker

II. RELATED WORK

Segmenting the sentences into tokens in the Asian languages can be complicated [6]. Like the Chinese, Japanese and Korean (CJK) languages, Dzongkha script is written continuously without any word delimiters and this causes a major obstacle in natural language processing tasks [3]. Dzongkha sentences are sequences of syllables separated by "Tsheg" characters (·), where each word is made up of one or more syllables. For example, Dzongkha word "རྒྱལ་པོ" meaning "king" is a two-syllable word. Unlike in the English language, the word boundaries are not separated by spaces, making segmentation of Dzongkha words more challenging. There are many approaches used for segmentation; since this is preliminary work related to automated Dzongkha word segmentation, basic maximal matching technique followed by probabilistic N-gram technique [3] has been adopted.

III. METHODOLOGY

We have implemented the maximal matching algorithm to segment Dzongkha words. The basic maximal matching technique works accurately if the possible segmented word is found within the dictionary; otherwise, it will return an invalid word. For example, if the possible segmented word is name of a person which is not contained in the dictionary, it will be considered an invalid word [7]. The ordinary dictionary can also be replaced by a corpus [3] comprising multiple documents from various fields such as agriculture, science, sports, news, etc.

To resolve the problem of segmenting compound words [3], we use the N-gram probabilistic technique to further enhance the segmentation process. For example, the Dzongkha word "སེམས་རྟོག་ར" meaning "name of the place called Semtokha" is a compound word where it has been formed from the combination of words "སེམས" meaning "heart", "རྟོག" meaning "deduce" and "ར" meaning "mouth". The basic maximal matching technique used alone will list all four possible words; yet only one is required. To overcome such ambiguity, the N-gram technique is used to select the most appropriate word among the list of choices. Furthermore, the segmentation task can be enhanced by using a Markov chain where the current word determines the following words [6].

A. Maximal Matching Algorithm

The Dzongkha words are segmented by using the maximal matching algorithm, which is based on an algorithm published in [3], which first takes the input sentence and generates all possible segmentations and then selects the segments with the shortest or minimum number of word tokens. After that the dictionary lookup is done for accurate matching.

The following steps are included:

1. Read the input string of text. If there are multiple sentences, break them using sentence separator markers '*'.
2. Further split the input string of text with "Tsheg" characters into syllables.
3. Then take the next syllables and generate all possible strings.
4. If the string is greater than *n* for some value *n*:
   - Look up the series of strings in the dictionary and assign some weight-age accordingly.
   - Then shorten the string with the given weight-age.
   - Exclude the strings with low count.
5. If the combined syllables are not present in the dictionary, it is looked up in the corpus.
   - If it is present in the corpus, the probability for a particular word is computed and stored in the stack along with it.
6. Repeat Step 2 till all the syllables are processed.

The input string is searched in the dictionary and the lower half of the code will be executed; if the input string in not in the dictionary it is searched in the corpus.

The code for the maximal matching algorithm is shown in Fig. 1 and Fig. 2. The algorithm for the segmentation phase is depicted in Fig. 3.

```
public int COMPARE(String[] dum, String str){
int f=0;
/*try //(BufferedReader br = new BufferedReader(new FileReader("parshu.txt")))
    {*/
public static void main(String[] args) throws IOException{
BasicMaxiMatch seg = new BasicMaxiMatch();
seg.doit();
}
```

Fig. 1. Code to compare the words after retrieving from lexicon.

```
String s = "Parshu is going to university";
    String[] dum={"à½€","à½","parshu","is","going","to","university","high","dorji"};
    int f=dum.length;
    String s="";
    try{
    FileInputStream fs = new FileInputStream("dictionary.txt");
    BufferedReader br = new BufferedReader(new InputStreamReader(fs, "utf-8"));
    String dicRead;
    while ((dicRead = br.readLine()) != null) {
        // Print the content on the console
        s=s+dicRead;
        }
    String st="";
    try{
    FileInputStream fs = new FileInputStream("textfile.txt");
    BufferedReader br = new BufferedReader(new InputStreamReader(fs, "utf-8"));
    String strLine;
    while ((strLine = br.readLine()) != null) {
        // Print the content on the console
        st=st+strLine;
        System.out.print(st);
        }
```

Fig. 2. A section of code for maximal matching algorithm.

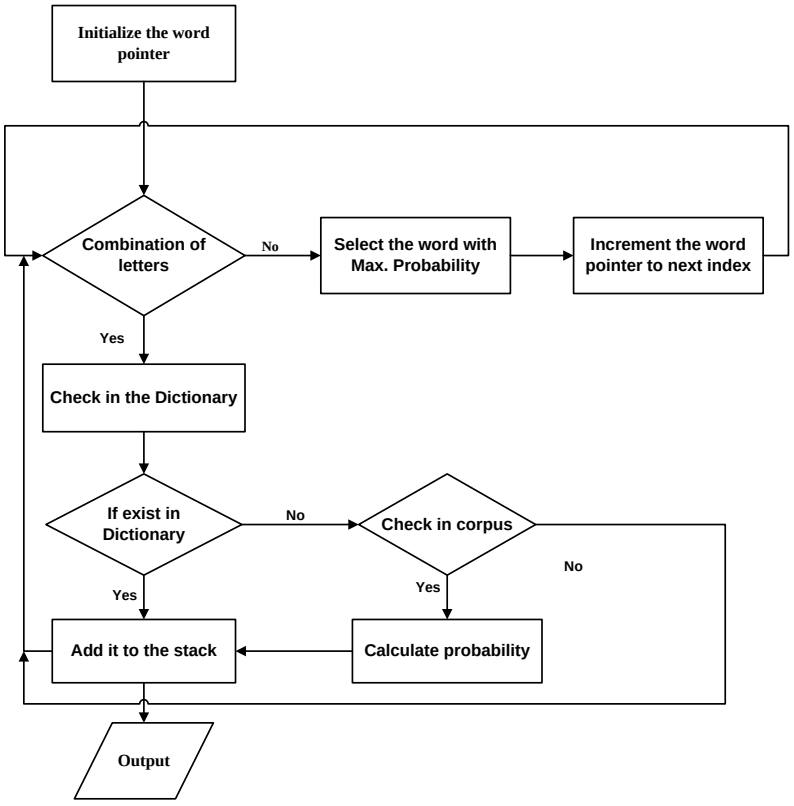

Fig. 3. The segmentation algorithm.

For a given sequence of syllables (a sentence), the word pointer is initialised and the initial syllable is set to null. Every syllable is combined together to get all possible segmented valid words. If the combination has happened, the combined syllables are looked up in the dictionary to see if it can be a possible segment or not. If it exists in the dictionary, it is stored in the stack. If the combined syllables are not present in the dictionary, it is looked up in the corpus. If the combined syllables are present in the corpus, the probability for a particular word is computed and stored in the stack along with it. Otherwise, it is combined with the immediately following syllable, and the process loops until it reaches the end of the sentence. After the first iteration is completed, if only one valid word is stored in the stack then it is returned as a valid word, otherwise the word with the highest probability is chosen as an individual valid word. After the selection of valid word at each iteration, the word pointer is incremented by one and the process repeats until the word pointer equals the number of syllables in the sequence.

IV. Results and Discussion

The input given to the program code is as follows:

ང་ལེགས་ཏོག་ཅ་འཚོ་དོ།

The input sentence above is to be segmented. The segmented words obtained by different approaches are presented below.

A. Segmentation Using Basic Maximal Matching

The input sequence is broken down into individual syllables. Each syllable is combined with its immediate successor. Every combination is then checked in the dictionary to see whether the combination is a valid word or not. If it is found in the dictionary, then it is written as a valid word, otherwise void. Figure 4 shows the result of segmentation using basic maximal matching.

The basic maximal matching technique is simple and accurate if the possible segments are contained in the dictionary. However, it does not solve the ambiguity when there are two or more possible segments contesting for one valid word. This problem is solved by using the N-gram technique.

B. N-gram Technique

The ambiguity of word selection is resolved by using the N-gram technique. When there is more than one possible word, the one with the highest count (in the case of unigrams) or the highest probability (in the case of bigrams) is chosen as the valid word. Results of different intermediate steps in the implementation of the N-gram technique are discussed in the following subsections.

C. Unigram Counting

The process of counting the individual word occurrences in the corpus is called unigram counting. The count of individual unique words of sample corpus is shown in Fig. 5.

D. Bigram Probability

The probability of occurrence of a word, given the previous word [3], is computed as shown in (1).

$$\frac{P(W_i)}{P(W_{i-1})} = \frac{Count(W_{i-1}\ W_i)}{Count(W_{i-1})}. \quad (1)$$

The output is shown in Fig. 6.

E. N-gram Segmentation

The result obtained from segmentation using the N-gram technique is shown in Fig. 7.

```
1  ང
2  ལེགས
3  ལེགས་ཏོག་ཅ
4  འཚོ
5  དོ
6
```

Fig. 4. Segmentation output based on basic maximal matching.

```
1  ང : 2
2  ལེགས་ཏོག་ཅ : 2
3  འཚོ : 1
4  དོ : 1
5  । : 3
6  གི : 1
7  ལེགས : 1
8  འདི : 1
9  ཚེམ་བུག : 1
10 ཡིན : 1
11
```

Fig. 5. Unigram count output.

```
1  ང ལེགས་ཏོག་ཅ : 0.5
2  ང གི : 0.5
3  ལེགས་ཏོག་ཅ འཚོ : 0.5
4  ལེགས་ཏོག་ཅ འདི : 0.5
5  འཚོ དོ : 1.0
6  དོ । : 1.0
7  । ང : 0.3333333333333333
8  । ལེགས་ཏོག་ཅ : 0.3333333333333333
9  གི ལེགས : 1.0
10 ལེགས । : 1.0
11 འདི ཚེམ་བུག : 1.0
12 ཚེམ་བུག ཡིན : 1.0
13 ཡིན । : 1.0
14
```

Fig. 6. Bigram probability output.

```
1  ང
2  ལེགས་ཏོག་ཅ
3  འཚོ
4  དོ
5
```

Fig. 7. N-gram based segmentation output.

V. Conclusion

The paper presents a word segmentation method based on a basic maximal matching technique, which resolves the issue of a lack of word separation, which makes tokenising and further processing of Dzongkha language difficult. However, it does not solve the ambiguity when there are two or more possible segments contesting for one valid word. This ambiguity of word selection is resolved by the N-gram technique, where in case of more than one possible word, the one with the highest count (in the case of unigrams) or the highest probability (in the case of bigrams) is chosen as a valid word.

The result of our study shows that the accuracy of the output heavily depends on the quality of the tagged corpus and the dictionary. This experiment is carried out based on the corpus and the dictionary developed by Dzongkha Development Commission of Bhutan. As the corpus is at the development stage, the accuracy of the results will improve upon its completion.

VI. Acknowledgement

The research has been carried out as part of the Master Thesis at Riga Technical University, Latvia. The research would not be successful without the corpus and the dictionary developed by Dzongkha Development Commission of Bhutan and also quality resources obtained through the university.

I would like to thank my Professor, Dr. habil. sc. ing. Jānis Grundspeņķis for his consistent support and direction in achieving the research results.

References

[1] G. van Driem, "Language Policy in Bhutan," presented at conference "Bhutan: a traditional order and the forces of change", London, UK, March 1993.

[2] D. C. Chhoeden et al., "Dzongkha Text-to-Speech Synthesis System – Phase II," in *Conference on Human Language Technology for Development*, May 2–5, 2011, Alexandria, Egypt, pp. 148–153.

[3] S. Norbu et al., "Dzongkha Word Segmentation," in *8th Workshop on Asian Language Resources*, August 21-22, 2010, Beijing, China, pp. 95–102.

[4] H. Liu et al., "Tibetan Word Segmentation as Syllable Tagging Using Conditional Random Field," in *25th Pacific Asia Conference on Language, Information and Computation*, December 16–18, 2011, Singapore, pp. 168–177.

[5] C. Chungku, J. Rabgay, and P. Choejey, "Dzongkha Text Corpus," in *Conference on Human Language Technology for Development*, May 2–5, 2011, Alexandria, Egypt, pp. 34–38.

[6] G. Andrew, "A Hybrid Markov/Semi-Markov Conditional Random Field for Sequence Segmentation," in *Conference on Empirical Methods in Natural Language Processing*, July 22–23, 2006, Sydney, Australia, pp. 465–472.

[7] C. Chungku, J. Rabgay, and G. Faaß, "Building NLP resources for Dzongkha:A Tagset and A Tagged Corpus," in *8th Workshop on Asian Language Resources*, 21–22 August 2010, Beijing, China, pp. 103–110.

Parshu Ram Dhungyel graduated from Acharya Nagarjuna University, India in 2012 with B.Tech degree in Computer Science and Engineering. In 2016, he received MSc. degree in computer systems from Riga Technical University, Latvia. Currently he is working as a Lecturer at the College of Science and Technology of Royal University of Bhutan. He has worked as a Transmission Engineer at Bhutan Telecom for five years (2004–2009).
E-mail: sharmad99@gmail.com / parshuram.cst@rub.edu.bt

Jānis Grundspeņķis received the qualification of Electrical Engineer in Automation and Telemechanics in 1965 and the degree of Candidate of Technical Sciences in 1972 (recognised in 1992 as the degree of Doctor of Engineering Sciences) from Riga Polytechnical Institute, Latvia. In 1993, he received the degree of Habilitated Doctor of Engineering Sciences from Riga Technical University, Latvia.
Since 1994, he has been Professor of Systems Theory at Riga Technical University. He also holds the positions of the Dean of the Faculty of Computer Science and Information Technology and the Head of the Department of Artificial Intelligence and Systems Engineering of Riga Technical University, both since 1994. His research interests include agent based and multiagent intelligent systems, knowledge acquisition and representation, causal domain models for complex cyber-physical systems, and structural modelling.
He is a full member of the Latvian Academy of Science, senior member of IEEE, and member of ACM.
E-mail: janis.grundspenkis@rtu.lv