A Computational Approach for Corpus Based Analysis of Reduplicated Words in Bengali

Apurbalal Senapati[1] and Utpal Garain[2]

[1] Central Institute of Technology, BTAD, Kokrajhar-783370, Assam, India
apurbalal.senapati@gmail.com
[2] Indian Statistical Institute, 203, B.T.Road, Kolkata – 700108, India
utpal.garain@gmail.com

Abstract. Reduplication is an important phenomenon in language studies especially in Indian languages. The definition of reduplication is the repetition of the smallest linguistic unit partially or completely i.e. repetition of phoneme, morpheme, word, phrase, clause or the utterance as a whole and it gives different meaning in syntax as well as semantic level. The reduplicated words has important role in many natural language processing (NLP) applications, namely in machine translation (MT), text summarization, identification of multiword expressions, etc. This article focuses on an algorithm for identifying the reduplicated words from a text corpus and computing statistics (descriptive statistics) of reduplicated words frequently used in Bengali.

Keywords: Reduplication, Bengali, Corpus, Descriptive statistics, Evaluation.

1 Introduction

Reduplication is one of the highly productive morphological processes in Bengali. It is frequently used in the language for various linguistic and pragmatic reasons and purposes. The use of reduplicated words in text or corpus is in different ways and manners to serve various means of information-sharing and communication. Although it is mostly used to express a sense of multiplicity of various countable items, it is also used as a process to refer to the act of continuation of an action or an event [1] or something else.

For example, S1: আপনি কোন গ্রামে যেতে চান ? / *aapni kon grame jete chan* ? (Which village do you want to visit?); S2: আপনি কোন কোন গ্রামে যেতে চান ? / *aapni kon kon grame jete chan* ? (Which are the villages you want to visit?). Clearly in sentence S2, the semantic changes to plural and it is due to the use of reduplication of word কোন /*kon* (which). Similarly for example S3: ঘরে কোন লোক নাই /*ghare kono lok nai* (There is no one in the house); S4: ঘরে ঘরে বেকার যুবক / *ghare ghare bekar jubak* (unemployment is in every house). The semantic meaning of reduplication of ঘরে/*ghare* (in house) in S3 and S4 are different. In S3, meaning of ঘরে /*ghare* is "in house" but in S4, the meaning of ঘরে ঘরে /*ghare ghare* is "in every house". Now it is clear that in many NLP applications especially in MT, the semantic of reduplication has to be considered carefully in order

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 456–466, 2015.
DOI: 10.1007/978-3-319-18111-0_34

to achieve high accuracy. For example, the machine translation (using Bengali to English Google translator, dated 28th January, 2015) of sentences S5: সে খেতে আসছে / *se khete aaschhe* and S6: সে খেতে খেতে আসছে / *se khete khete aaschhe* is "*He is coming to eat*" in both cases. But the actual translations are "*He is coming to eat*" and "*He is coming while eating*", respectively. It is obvious that the wrong translation producing in sentence S6 due to failure in capturing the semantic of reduplication. Similarly in many NLP applications the reduplication has to be tackled separately in order to reduce semantic analysis error.

2 Types of Reduplicated Words in Bengali

The process of reduplication is quite frequent in Bengali. A large number of words are capable of producing valid reduplication. But practically most of them are not used or used with very low frequencies. It is also observed that the reduplication can occur to all word categories including the pronouns and indeclinable.

From the structural point of view there are six types of reduplication in the Bengali texts [1], which are as follows:

i) The repetition of same word as a second member without the addition of any suffix of inflectional properties with any member i.e. the proper reduplication. Examples, হাসি হাসি / *hasi hasi* (smiling) [হাসি/ *hasi* (smile)]; বছর বছর / *bachhar bachhar* (every year) [বছর / *bachhar* (year)]; লাল লাল / *lal lal* (red in plural sense) [লাল / *lal* (red in singular sense)]; ভালো ভালো / *bhalo bhalo* (good in plural sense) [ভালো / *bhalo* (good in singular sense)]; দিন দিন / *din din* (day by day) [দিন / *din* (day)] etc. Note that in this category, each individual word has a valid mining which is different (same on some cases) their reduplicated meaning.

ii) The first word is repeated and while first word carries no inflection but the second word carries an inflection. Examples, ধব ধবে /*dhab dhabe* (pure white color) [ধব/*dhab* and ধবে /*dhaeb* are not valid words], টক টকে/*tak take* (deep red color) [টক/*tak* (sour) but টকে/*take* (not a valid word)], লক লকে/*lak lake* (flickering / attractive) [লক/*lak* and লকে/*lake* are not a valid words] etc. Note that in this category, each individual word is not a valid word whereas the reduplicated words are meaningful.

iii) The first word is inflected and then inflected word is repeated. Example, ঘরে ঘরে / *ghare ghare* (in every house) [ঘরে / *ghare* (in house)], কানে কানে / *kane kane* (secretly) [কানে / *kane* (in ear)], গাছে গাছে / *gachhe gachhe* (in every tree) [গাছে / *gachhe* (in tree)] etc. Note that, this category is also proper reduplication and in this case also the semantic behavior is same as of category i).

iv) A semantically or almost similar word is added with the first word to generate the reduplicated word. Example, চাল চুলো / *chal chulo* (economically poor) [চাল/chal (rice), চুলো / *chulo* (cooking burner)], চুরি চামারি / *churi chamari* (robbery) [চুরি / *churi* (theft), চামারি / *chamari* (illegal work)], অলি গলি / *ali gali* (narrow lane with complicated direction) [অলি / *ali* (narrow lane), গলি / *gali* (narrow lane)] etc. Note that, in this case the semantic meaning of each individual word is almost same and their re-duplicated meaning is also almost similar to the individual word.

v) An eco word is added as the second member with the first word to generate the reduplicated word. Example, জল টল/ *jal tal* (water, beverage etc.) [জল / *jal* (water) and টল/ *tal* (eco word)], খাবার দাবার/ *khabar dabar* (varieties food) [খাবার/ *khabar* (food) and দাবার/ *dabar* (eco word)], মাছ টাছ/ *mach tachh* (egg, fish, meat etc.) [মাছ/ *mach* (fish) and টাছ/*tachh* (eco word)] etc. Note that, in this case the first word has a specific meaning but after adding the eco word the meaning changes. Also note that the composite meaning is almost similar to the first word but in plural form but this property does not follow in all cases.

vi) Onomatopoeic words made with two words of identical structures. Examples, ছম ছম / *chham chham* (feeling of sound of silence), খিল খিল /*khil khil* (sound of laugh), ঝিন ঝিন /*jhin jhin* (jingling) etc. Note that, this category is also proper reduplication and in this case the semantic meaning is related to sound (real or virtual) of different events.

Whereas, the reduplicated words can be classified in other perspective like phonological perspective, morphological perspective, lexical perspective, constructional perspective, etc. In functional point of view, reduplicated can be classified based on the part of speech also. But our present work only concentrate on the computational aspect of identifying the reduplicated words from the corpus.

3 Existing Work on Reduplicated Word in Bengali

Most of the existing works on reduplication is contributed by the linguistic people and it has started long back in many Indian languages. Ananthanarayana [2] describes the reduplication in Sanskrit and Tamil, Abbi [3] focuses on the different aspect of reduplication on south Asian language, Murthy [4] worked on Kannada language, etc. The work on Manipuri reduplicated is found in identification of multiword expressions by in Nongmeikapam [5] work. In Bengali language, the linguistic study found from Chattopadhyay [6] , Chaudhuri [7], Thompson's [8] work. In computational point of view, Bandyopadhyay [9] has studied reduplicated words for semantic based analysis. Senapati [10] has studied the reduplicated pronoun in their anaphora resolution task in Bengali.

4 Our Contribution

From the literature survey it is clear that most of analysis is on the linguistic point of view and the works are common in nature i.e. analysis of reduplicated words and tried to capture their semantic meaning. Whereas the computational works are limited. But some basic issues like how many reduplicated words are there in Bengali or what are the frequencies in which reduplicated words appear in Bengali, etc. i.e. the corpus based statistics are still not studied. We have proposed an algorithm to identify the reduplicated words from a text corpus and also proposed a dictionary based tuning technique to enhance the accuracy of identifying such word in the corpus. Finally, the frequencies of reduplicated word have been calculated in word level as well as in sentence level.

5 Computational Approach to Identify Reduplication in Bengali

Our computational approach is based on the morphological similarities of the duplicated words. In our work, the morphologically similar reduplicated words implies that the similar or almost similar words in terms of their word length and use of characters or use of vowel modifiers in the words. In section 2, we have seen that in category (i), (iii) and (vi) the formation of reduplication by the repetition of same word i.e. of the form "*w w*" where "*w*" is word in the corpus. Also we observe that, in category (ii), (iv) and (v) the formation of reduplication by the repetition of almost similar word. And hence from the computational aspect we define the reduplicated words of two types. The proper reduplication i.e. when the repetition of same word; for example, খেতে খেতে / *khete khete* (continue eating), যেতে যেতে / *jete jete* (continue going), where above category (i), (iii) and (iv) come under this type. The other type is the partial reduplication i.e. first and second word is not exactly same but almost similar; for example, খাবার দাবার/ *khabar dabar*/ (food etc.), চাল চুলো / *chal chulo* (economically poor), where above category (ii), (iv) and (v) come under this type. There are some exceptional cases, e.g. মাথা মুন্ডু/*matha mundu* (meaning lass), লোটা কম্বল/*lota kambal* (belonging of poor man), etc. and we are now not considering these cases. The computational approaches for identifying two different types of reduplication are also different and handled by two different algorithms. Finally, to reduce the error we have used a dictionary and frequency based tuning technique. The details descriptions of the algorithms are given below.

Table 1. Algorithm to find the proper reduplication from the text corpus

	ALGORITHM	
s1:	$w_i \leftarrow$ word from corpus	
s2:	if w_i contains "-" then	
s3:	if w_i is of the form "*w*-*w*" then	// type 2
s4:	print "**re-duplication**";	
s5:	frequency= frequency+1;	
s6:	end if	
s7:	else if w_i is of the form "*ww*" then	// type 3
s8:	print "**re-duplication**";	
s9:	frequency= frequency+1;	
s10:	end if	
s11:	else	
s12:	$w_{i+1} \leftarrow$ next word from corpus	
s13:	if "w_i *is equal to* w_{i+1}" then	// type 1
s14:	print "**re-duplication**";	
s15:	frequency= frequency+1;	
s16:	end if	
s17:	end if	

For algorithmic approach, first we analyzed the proper reduplication in terms of morphological similarity. In lexical point of view the proper reduplication is of three

types. The first type is of the form "*w w*" i.e. repetition of same word with a single space; for example, খেতে খেতে / *khete khete* (continue eating). The second type is of the form "*w-w*" (or "*w - w*") i.e. repetition of same word with a "-" separation, for example, ধীরে - ধীরে/*dhire dhire* (slowly) and the third type is of the form "*ww*" i.e. repetition of same word without any space; for example, গজগজ/*gajgaj* (feeling of irritation). The formal algorithm of this category is given in Table 1. Also note that the algorithm also calculating the frequencies of reduplicated words separately.

To identify the partial reduplicated word is relatively complicated compared to proper reduplication and hence first we studied the features of partial reduplication to setup our algorithm. In earlier work some people have been used some heuristic rules. According to Bandyopadhyay [9] the partial reduplication are of three types, (i) change of the first vowel or the matra (vowel modifier) attached with first consonant, (ii) change of consonant itself in first position or (iii) change of both matra and consonant. They have also identified some exceptions e.g. আবল-তাবল/ *aabol-taabol* (irrelevant) etc. According to the linguistic study of Chattopadhyay [6], we found the rule formation of partial reduplication i.e. the consonants that can be produced after changing are ট, ফ, ম, স. Now from the above studied and from our observation on reduplicated words, the common features of partial reduplication are:

(i) Most of the cases the length of the individual words are same e.g. কখনো সখনো/*kakhano sakhano* (sometimes) e.g. *length(*কখনো*) = length(*সখনো*)* or length of reduplicated word is one more than the first word e.g. ধব ধবে */dhab dhabe* (pure white color) where *length(*ধব*)+1 = length(*ধ, ে*)*

Table 2. Algorithm to find partial reduplication from corpus

ALGORITHM	
s1:	w_i and w_{i+1} ← word from corpus
s2:	if (*length*(w_i) == *length*(w_i)) then
s3	count ← *charecterWiseDifferent*(w_i, w_{i+1});
s4	differentCharecterPair ← (c_1, c_2); //mismatch character pair in w_i & w_{i+1}
s5	if (*count ==1*&& (c_1 & c_2 *both vowel modifier or both alphabet*) then
s6:	print "**re-duplication**";
s7	frequency= frequency+1;
s8:	end if
s9	else if (*length*(w_i)+*1* == *length*(w_{i+1})) then
s10:	count ← *charecterWiseDifferent*(w_i, w_{i+1});
s11:	if (*count ==1*) then
s12:	misMatchChar ←(w_i , w_{i+1}); // mismatch character
s13:	if (*misMatchChar is vowel modifier*) then
s14:	print "**re-duplication**";
s15:	frequency= frequency+1;
s16:	end if
s17:	end if
s18:	end if

(ii) The difference between the reduplicated words in character wise is either a letter [e.g. কখনো সখনো/*kakhano sakhano* (sometimes) where difference character pair is (ক, স)] or a vowel modifier [e.g. খুচ খাচ/*khuch khach* (little bit) where difference character pair is (ু,া)]

(iii) Numbers of characters differs in one and

(iv) Most of the cases this letter is a consonant of specific types like ট, ফ, ম, স, etc.

Now based on these observations we have incorporated the features i.e. (i), (ii) and (iii) in our algorithm to identifying the partial reduplication and is given in Table 2. In this algorithm, the function *charecterWiseDifferent*(w_i, w_{i+1}) returns the number of mismatch between two words w_i and w_{i+1} character wise and also calculating the frequencies of each reduplicated word. Note that, this algorithm also considering cases like w_i-w_{i+1} (or w_i - w_{i+1}) but not shown in algorithm separately.

6 Corpus Based Study of Reduplicated Words in Bengali

For the corpus based study of reduplication in Bengali, the Technology Development for Indian Languages (TDIL) corpus [12] has been used. The TDIL corpus is developed by the Department of Electronics, Govt. of India for Bengali language (http://tdil.mit.gov.in/). This corpus contains texts from Literature (20%), Fine Arts (5%), Social Sciences (15%), Natural Sciences (15%), Commerce (10%), Mass media (30%), and Translation (05%). Where each category has some sub categories e.g. Literature includes novels, short stories, essays etc.; Fine Arts includes paintings, drawings, music, sculpture etc.; Social Science includes philosophy, history, education etc.; Natural Science includes physics, chemistry, mathematics, geography etc.; Mass Media includes newspapers, magazines, posters, notices, advertisements etc. Commerce includes accountancy, banking etc., and translation includes all the subjects translated into Bengali. The size and the number of reduplicated words found using above algorithms in the corpus are given in the following table (Table 3).

Table 3. Reduplicated words in TDIL corpus

Corpus	# Files	# Sentences	# Words	# Reduplicated words (unique)	# Frequency
TDIL	1362	334260	4429574	6196	61647

The Table 3 shows that the percentage of reduplicated words in the corpus is 1.4% and at a glance it looks like quite low but while it will be consider in sentence level then it shows that, 18.44% of the sentences contain reduplicated words. Since the semantic of sentence highly depends on the presence of reduplicated word and hence this percentage shows that it cannot just ignore in any NLP application.

7 Tuning Technique

Though our algorithm has potential to identifying the reduplicated words compared to other existing approaches but still in order to reduce the error we have used a dictionary and frequency based tuning technique. Table 3 shows that there are a large number of reduplicated words with high frequency in the corpus. But our observation is that many of them are erroneous or not reduplicated word at all. And some of them occur with very low frequency and can be ignored without loss of generality. For example, the algorithm produces output "*2424*" or "ऽऽऽऽ" or "((" as reduplicated words, since these are strings of the form "*ww*", but actually not the reduplicated words. Hence in order to improve the efficiency we have used the tuning technique. Also we have used a technique to identify the reduplicated with eco words. This identification is very helpful in many NLP applications especially in MT.

Frequency measure: The frequency measure is an important technique to validate the word or association of words in a corpus. The general phenomenon is that the high frequencies of two words occur together, then that is evidence that they have a special function that is not simply explained as the function that results from their combination. Based on this phenomenon have we fixed a threshold frequency (T_f) and hence if the frequency of reduplicated words exceeds the threshold frequency i.e. > T_f then we only consider them in our experiment. Whereas to fix the threshold value many factors has to be considered like, the size of the corpus, domain of the corpus etc. Note that in our experiment we have defined T_f = 5 by applying the random sample technique in the corpus. Using this technique many irrelevant entries has been eliminated. For example, পি. ভি./p.v. (abbreviation of a name), বৃদ্ধ বৌদ্ধ /*bridha boudha* (irrelevant word) etc. structurally look like reduplicated but actually not.

Online dictionary: In this case we have eliminated the incorrect words using an online dictionary in the following techniques. We validate "*ww*" in online dictionary [13] and if "*ww*" found a valid word in the dictionary then we reject "*ww*" i.e. do not consider it as a reduplicated word. For example, the algorithm will produce output "বাবা", "দাদা", "দিদি", "মামা" etc. as reduplication word, since these are strings of the form "*ww*". Now, once these words are checked in online dictionary and found as valid words, they are rejected as reduplicated words. Following this method, all (erroneous words like, বাবা/*baba* (father), দাদা/*dada* (elder brother), দিদি/*didi* (elder sister), মামা/*mama* (maternal uncle) etc. are eliminated.

Identification of eco words: In case of w_1-w_2 form, system splits it into w_1 and w_2 separately and validate in online dictionary separately. If it shows that the first one i.e. w_1 is a valid word but second one i.e. w_2 is not a valid word then identified is as eco words. For example, অঙ্ক-টঙ্ক/anka-tanka (maths etc.) [অঙ্ক/anka (maths), টঙ্ক/tanka (eco word)], আত্মীয়-টাত্মীয়/aantiya-taantiya (relatives) [আত্মীয়/aantiya (relative), টাত্মীয়/taantiya (eco word)], ব্যাপার-স্যাপার/bapar-sapar (matters) [ব্যাপার/bapar (matter), স্যাপার/sapar (eco word)], etc.

The interesting observation is that, after applying the tuning techniques the number of reduplicated words is reduced significantly and most of the erroneous entities are eliminated, and the revised result is shown in Table 4.

Table 4. Reduplicated words in TDIL corpus (after tuning)

Corpus	# Files	# Sentences	# Words	# Reduplicated words (unique)	# Frequency
TDIL	1362	334260	4429574	794	37919

After tuning, Table 4 shows that the percentage of reduplicated words in the corpus is 0.71% and in sentence level then it shows that 9.4% of the sentences contain reduplicated words. Clearly after tuning process system eliminates about 50% of reduplication produced by the above algorithm. Next section shows that improvement of accuracy after tuning technique.

8 Evaluation

The system has been evaluated by the stratified simple random technique on the TDIL corpus. The technique is due to Sharon [11]. The technique in brief is as follows. The corpus is partitioned into non-overlapping groups and then groups are selected in random. Now from a selected group the manual output and the system output have been considered for the final evaluation. The Precision, Recall and F-score have been used as evaluation metric and result shows in Table 5. Note that, though the system is identifying the eco words separately, we are not evaluating the performance of eco word identification separately.

Table 5. Result for identification Reduplicated words in TDIL corpus

	Corpus	Precision	Recall	F-score
Before Tuning	TDIL	0.63	0.85	0.72
After Tuning	TDIL	0.93	0.84	0.88

9 Error Analysis

In order to find the weakness of our algorithms the error analysis has been carried out. This analysis not only measures the error in terms of number of wrongly identified but also identified the major source of errors in different phases of the system. Broadly we have identified the source of errors in two phases; the error generated by the system output and the error generated in tuning phase.

The Table 6 and Table 7 are the confusion matrixes for identification of reduplicated words before and after applying the tuning technique respectively. Table 6 shows that there were 45685 reduplicated words in the corpus and the system capable to capture only 38719 instances correctly and identified 22928 instances wrongly. Note that, “Actual False (X)” shown in Table 6 in first row indicates that number of non reduplicated words present in the corpus. Since, this number is not relevant in our measure and hence it is omitted and similarly the value of “true negative (X)” is also not calculated.

Table 6. Confusion matrix before applying tuning technique used in TDIL corpus

	Actual True (45685)	Actual False (X)
System Identified true	true positive (38719)	true negative (X)
System Identified false	false negative (6966)	false positive (22928)

Table 7 shows the result after applying tuning process. Note that in this table the number of actual reduplicated words is 42230 i.e. it reduces 3455 (45685 - 42230) true instances for tuning technique. Based on our observation, major contribution of this elimination is due to the instances with low frequency i.e. below threshold level. Also note that, applying tuning technique system has eliminated 20273 (true negative) false instances. In this case, the major contribution of this elimination is due to the use of dictionary entries. Note that some very common word (false instances like, বাবা/*baba* (father) with frequency 1342, দাদা/*dada* (elder brother) with frequency 327, দিদি/*didi* (elder sister), with frequency 325 etc.) i.e. instances with very high frequencies are eliminated and results improve the system performance. The error analysis in algorithmic level is given below.

Table 7. Confusion matrix after applying tuning technique used in TDIL corpus

	Actual True (42230)	Actual False (22928)
System Identified true	true positive (35474)	true negative (20273)
System Identified false	false negative (6756)	false positive (2655)

The error produced by the algorithms can be categorized of two types. We consider the first type is the *false negative* i.e. the algorithm fails to identify the reduplicated words. Actually the algorithm is designed based on the analysis of lexical features (details is in section 5) of reduplicated words. But these features does not cover all types of reduplicated words, especially those reduplicated words, where first and second words having morphological variant. For example, the reduplicated words like মাথা মুন্ডু/*matha mundu* (meaning lass), লোটা কম্বল/*lota kambal* (belonging of poor man), etc. are not covered by the algorithms. And hence it affects the accuracy in terms of recall and it reflects the recall value shown in Table 5. The other type of error is the *false positive* i.e. the algorithm wrongly identify the reduplicated words. For examples, consider the system generated output with their frequencies, দমদম/*dumdum* (Dumdum, name of a place) [frequency 50], টাটা/*tata* (Tata, name of a place) [frequency 50], শ্রীশ্রী/srisri (Mr. like term come before a name of male person) [frequency 17] etc. Clearly, দমদম/*dumdum* (Dumdum) will not be eliminated in tuning mechanism, because দমদম/*dumdum* (Dumdum) is not a valid word in online dictionary and since its frequency greater than threshold frequency ($50 > T_f = 5$). Hence it contributes errors and it affects the accuracy in terms of precision and it reflects the precision value shown in Table 5.

The error produced by the tuning technique also can be categorized of two types. The first type is the *false negative* i.e. the tuning technique elements the true reduplicated words. This will happened because, if some reduplicated words present in the corpus with low frequency ($\leq T_f$). For example, consider the system generated output with their frequencies মড়মড়/*marmar* (sound of break) [frequency 4], ঝিরিঝিরি/*jhirjhir*

(sound of rain in slow motion) [frequency 4], গুটিগুটি/*gutiguti* (slowly) [frequency 4], etc. Though all these are valid reduplicated words but will be eliminated by tuning mechanism because of the low frequency ($\leq T_f$). Obviously it affects the accuracy in terms of recall and it reflects the recall value shown in Table 5. The other error type is the *false positive* i.e. the tuning technique fail to eliminate the false reduplicated words. The examples describes above, like দমদম/*dumdum* (Dumdum) [frequency 50], টাটা/*tata* (Tata) [frequency 50], etc. will not eliminated by the tuning technique.

10 Conclusion

This paper presents a pioneering attempt to develop a computational approach for corpus based study of reduplicated word in Bengali. The paper also shows the frequencies of reduplicated words and also shows that how frequently the reduplicated words are present in a corpus as well as at the sentence level. It also identified with examples that the affect of reduplication in MT system and has focused an untouched issue in Bengali-English MT. The algorithms used for identifying the reduplicated words are very simple. Though, the performances of the algorithms are not very high but after applying the tuning techniques the performance has improved to the satisfactory level. The error analysis part also identified the weaknesses of the system and hence there is future scope to improve the accuracy further.

Acknowledgement. The authors sincerely acknowledge Prof. B.B. Chaudhuri of Indian Statistical Institute, who kindly shared his expertise on Bengali reduplicated words with the authors.

References

1. Dash, N.: A Descriptive Study of Bengali Words, pp. 225–251. CUP (2015)
2. Ananthanarayana, H.S.: Reduplication in Sanketi Tamil OpiL, vol. 2, pp. 39–49 (1976)
3. Abbi, A.: Reduplicated Adverbs of Manner and Cause of Hindi. Indian Linguistics 38(2), 125–135 (1977)
4. Murthy, C.: Formation of Echo-Words in Kannada. In: All India Conference of Dravidian Linguistics(eds.) (1972)
5. Nongmeikapam, K.: Identification of Reduplication MWEs in Manipuri, a rule-based approach. In: 23rd International Conference on the Computer Processing of Oriental Languages, California, USA, pp. 49–54 (2010)
6. Chattopadhyay, S.K.: Bhasa-Prakash Bangala Vyakaran, 3rd edn. Pupa publication (1992)
7. Chaudhuri, B.B.: Bangla Dhwanipratik: Swarup o Abhidhan (Bangla Sound Symbolism: Properties and Dictionary). Paschimbanga Bangla Academy, Kolkata (2010)
8. Thompson, H.R.: Bengali: A Comprehensive Grammar, pp. 663–672. Routledge publication (2010)
9. Bandyopadhyay, S.: Identification of Reduplication in Bengali Corpus and their Semantic Analysis: A Rule-Based Approach. In: Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010), Beijing, pp. 72–75 (2010)

10. Senapati, A., Garain, U.: Anaphora Resolution in Bangla using global discourse knowledge. In: Int. Conf. of Asian Language Processing, Hanoi, Vietnam (2012)
11. Sharon, L.L.: Sampling: Design and Analysis, 2nd edn. Advanced Series, pp. 73–101 (2010)
12. TDIL Corpus: A nation-wide consortium for machine translation of Indic languages is being funded by the Ministry of Information Technology, Govt. of India (1995), http://www.tdil-dc.in
13. Digital Dictionaries of South Asia, http://dsal.uchicago.edu/dictionaries/biswas-bangala/