A Corpus Based Unsupervised Bangla Word Stemming Using N-Gram Language Model

Tapashee Tabassum Urmi, Jasmine Jahan Jammy, Sabir Ismail
Department of Computer Science and Engineering
Shahjalal University of Science and Technology, Sylhet
Sylhet-3114, Bangladesh
tapashee.tagore@gmail.com, jammy.sust@gmail.com, sabir-cse@sust.edu

Abstract— **In this paper, we propose a contextual similarity based approach for identification of stems or root forms of Bangla words using N-gram language model. The core purpose of our work is to build a big corpus of Bangla stems with their corresponding inflectional forms. Identification of stem form of a word is generally called stemming and the tool which identifies the stems is called stemmer. Stemmers are important mainly in information retrieval systems, recommending systems, spell checkers, search engines and other sectors of Natural Language Processing applications. We selected N-gram model for stem detection based on the assumption that if two words which exhibit a certain percentage of similarity in spelling and have a certain percentage of contextual similarity in many sentences then these words have higher probability of originating from the same root. We implemented 6-gram model for the stem identification procedure and we gained 40.18% accuracy for our corpus.**

Keywords- **unsupervised learning; natural language processing; n-gram model; root word; stemming**

I. INTRODUCTION

Now-a-days text processing has achieved a vast amount of concentration of the researchers with the growing need of efficient human-machine interaction. As we have huge amount of data gathered in computers in the form of text, computers can be used to extract information from the textual data by analyzing text patterns. To build a computerized system in any particular language it is required to build efficient methods for text processing for that language.

Bangla text processing is very new in the sector of Natural Language Processing (NLP) and it is known that almost all the applications of text processing need stemmer which is able to identify the stem forms of words of a language efficiently and correctly. So, to construct advanced NLP applications for Bangla language construction of a stemmer is a must.

Stemming plays important role in information retrieval systems for appropriate document retrieval. For example, if a document entitled “Importance of Stemming” is indexed in a search engine and a user issues a query “stem” then there will be no match with the title. But if the title and the query both were stemmed then the query would have matched with the title and the document would have been shown. Similarly in web search engines query strings are stemmed and indexed for generating useful query suggestions. Recommender systems also use stemmers for creating user profiles and item profiles to generate appropriate recommendations for the users.

In this paper we present an efficient unsupervised approach for constructing a Bangla stemmer for Bangla language using 6-gram model. Our approach is completely corpus based and simple threshold technique is used for root identification. We initially assume that a root word and its modified forms are very similar in spelling and their contextual similarity is very high. Contextual similarity of two distinct words with high spelling similarity is calculated comparing their previous words and following words occurring in a sentence. To generate previous words’ list and following words’ list of a word, we take its previous five words and next five words from every sentence and add them to their corresponding lists. If contextual similarity score of two distinct words goes above a certain threshold value then the words are considered to be produced from the same root.

As Bangla is very morphologically rich language, implementing all the word formation rules for constructing a stemmer which can identify all the stem words correctly is very difficult. This difficulty can easily be avoided if we can make computers learn from the text data by analyzing the text patterns of highly inflectional languages.

II. RELATED WORKS

First generation of stemmers were designed as rule based stemmers. The first paper published on such approach was by Julie Lovins in the year 1968. He introduced approximately 260 rules for stemming English language. His approach was Iterative Longest Match heuristic. Later in 1980 Dr. Martin Porter presented a new and improved rule based stemming approach where he reduced the rules of Lovins to about 60 rules. This algorithm is known as Porter Stemmer Algorithm which is a process of stripping the commoner morphological and inflectional endings from words in English. Later on Dr. Porter developed Snowball program which is a small string processing language designed for creating stemming algorithms. Most of the modern stemming algorithms have been created using Snowball Compiler Program. Currently more than 13 languages have stemmers created using Snowball Compiler. Rule based stemmers require good morphological knowledge of the language. As the language of the sub-continent is rich in grammars with various rules, it seems very difficult to use such approach. So varieties of other approach have been introduced to make stemmers of such highly inflectional language.

Sajib Dasgupta and Mumit Khan [1] developed a morphological parser using PC-KIMMO which is very popular morphological parser among the researchers of computational linguistics. Their work is mainly on incorporating Bangla in PC-KIMMO.

Sandipan Sarkar and Sivaji Bandyopadhyay [2] proposed rule-based stemming for Bangla. A detail analysis of corpus and grammar is presented in their paper.

Md. Zahurul Islam, Md. Nizam Uddin and Mumit Khan [3] built a light weight stemmer for Bangla spell checker. They marked the suffixes responsible for inflection of nouns, verbs and adjectives and stripped the longest suffix from a word to obtain a stem.

Amitava Das and Sivaji Bandyopadhyay [4] presented an approach for Bangla stem identification by use of clustering technique. They used K-means clustering where the center of a cluster is declared to be the root of that cluster. This is an iterative approach and they gained accuracy of 74.6%.

Sabir Ismail, M. Shahidur Rahman, Md. Abdullah Al Mumin [5] developed an algorithm for constructing POS tagged dictionary in Bangla. As a part of their work they identified root words from their sorted unique word list. They manually generated a suffix list initially and they tested if a word is ended by a valid suffix with the help of the suffix list.

Sabir Ismail, M. Shahidur Rahman [6] proposed an algorithm for Bangla word clustering where they used n-gram model which can also be effectively implemented in identifying root words.

Mohd. Shahid Husain [7] proposed an unsupervised stemming approach for Urdu and Marathi language. He presented an unsupervised suffix learning procedure using an Urdu corpus. After collecting all the valid candidate suffixes suffix stripping strategy was applied to get the stem words.

Utpal Sharma, Jugal Kalita, Rajib Das [8] worked on Assamese language where they used a valid suffix list and an Assamese corpus. They firstly checked the ending of a word against each valid suffix. If the ending of the word matches a valid suffix in the suffix list the ending is stripped and the validity of the word is determined by searching the word or the modified form of the word ending with other valid suffixes in the corpus. They also devised another approach [9] based on unsupervised learning for highly inflectional languages like Assamese. In this approach they attempt to build lexicons and suffixes by analyzing the text input of a language without any manual specification of that language.

Many other approaches are presented for stemming and morphological parsing for other languages like English, Hindi, Russian, Turkish etc.

III. Problem Definition

A. *Stem/Root Form Of A Word*

Root words/stems are base forms of words to which affixes (suffix, prefix, etc.) can be attached. A stem does not necessarily have to be the valid root of a word. It is a form of word which can be modified by using the inflection and affixation rules of word formation. Our method is based on the assumption that a word can have only one stem.

Consider the following group of words.

- অনুমোদনের
- অনুমোদনও
- অনুমোদনহীন

The appropriate stem for these words is অনুমোদন.

B. *Difficulties In Bangla Text Processing And Stemming*

Bangla text processing is very new in the field of NLP. That is why we face the different kinds of problems in building advanced NLP applications in Bangla which need root form of words in their implementations. For Example:

- We have no root word dictionary which contains all possible roots in Bangla. So any NLP application which needs root word forms has to depend on stemmers. So the performance of that NLP application may vary with the decision of selection of a stemmer.
- A stem word generated by only suffix checking [4] produces erroneous results. For example, although অক্ষ and অক্ষত are two different words অক্ষ will be considered to be the root of অক্ষত as ত is a valid suffix, which is wrong.
- We have several rule based stemmers which can generate only a limited number of root words.
- As there is a huge amount of word formation rules with many exceptions in Bangla it is very difficult to build a stemmer which can extract the stem forms of all kinds of Bangla words. Machine learning procedure can make this task simple. We used N-gram model for this purpose.

C. *N-gram Model*

In the fields of computational linguistics, an *n-gram* is a contiguous sequence of *n* items from a given sequence of text. An *n-gram* model can be imagined as placing a small window over a sentence or a text in which only *n* words are visible at a time. For different values of *n* we can define different *n-gram* models. An *n-gram* model is called *unigram* model for *n*=1. Similarly *bi-gram* is defined for n=2, *tri-gram* for (n=3) etc. As our work is based on 6-gram model we consider the following text for example and extract all 6-grams from this text.

***** If you give me money I will fix your car *****

TABLE I: LIST OF 6-GRAMS

List of 6-grams
1. ***** If
2. **** If you
3. *** If you give
4. ** If you give me
5. * If you give me
6. If you give me money I
7. you give me money I will
8. give me money I will fix
9. me money I will fix your
10. money I will fix your car
11. I will fix your car *
12. will fix your car **
13. fix your car ***
14. your car ****
15. car *****

In the above text * is considered to be a null string and total fifteen 6-grams have been found in the text.

In our method, for each word found in a sentence we collect five consecutive words preceding it and keep it in a list called *previous word list*. Similarly five consecutive words following it are kept in a list called *following word list*. Thus we generate for every unique word in a text, list of its preceding and following words in order to calculate the contextual similarity of two different words.

Our approach can solve the problem of অক্ষ and অক্ষত. Because although they are similar in spelling, their surrounding words will show dissimilarity as their meanings are different. And also our method is able to gather a huge number of stem words by learning procedure using large amount of Bangla texts available in the internet.

IV. PROPOSED APPROACH

We worked on a custom made Bangla corpus which was generated by parsing Bangla texts from different sources like Bangla news contents, Bangla blogs and Bangla literature. Our method is designed for the root words to which only suffixes can be attached.

Initially we collect all unique words and sort those using Unicode values to simplify our computation.

Then for each unique word we generate two separate lists one of which stores its previous words and another stores its following words. For every occurrence of a word in the corpus we take its previous five words and add them to its previous words' list and take its next five words and add them to its following words' list. Let a word be at ***kth*** position in the corpus. So we store ***(k-1)th***, ***(k-2)th***, ***(k-3)th***, ***(k-4)th***, ***(k-5)th*** word in the list of its previous words and ***(k+1)th***, ***(k+2)th***, ***(k+3)th***, ***(k+4)th***, ***(k+5)th*** word in the list of its following words.

We denote ***ith*** unique word as w_i. For every pair of different unique words if we find at least a fixed percentage of resemblance in their spelling then we compute their Jaccard similarity using their previous and following word lists. This similarity actually means their similarity in use in any text.

For finding spelling resemblance between w_i and w_j we initially assume that a root word can be modified only by joining suffixes. This implies that, a derived word must contain its root at its beginning. So resemblance in spelling can be obtained by comparing prefixes of a pair of words. So, we can call this as prefix resemblance of two words.

Let $S(w_i,w_j)$ gives the percentage of prefix resemblance. The following equation gives this percentage after comparing the prefixes of two unique words.

$$S\,(w_{i,},w_j)=\frac{length(matchPrefix(\boldsymbol{wi},\boldsymbol{wj}))}{Min(length(\boldsymbol{wi}),length(\boldsymbol{wj}))}\times 100 \;\ldots\ldots\ldots..\;(1)$$

If (1) gives at least a fixed percentage of prefix resemblance then we go to the next step to determine the contextual similarity between w_i and w_j, which is measured with respect to their previous words' lists and following words' lists generated previously. To calculate the contextual similarity between a pair of words we first compute the total number of words common in their their previous words' lists and the total number of words common in their following words' lists.

$$matchPrv = count(matchPreviousList(\boldsymbol{wi},\boldsymbol{wj}))\ldots..\;(2)$$

Here *matchPrv* gives the total number of common words in the previous words' lists of w_i and w_j.

$$matchNxt = count(matchFollowingList(\boldsymbol{wi},\boldsymbol{wj}))\ldots.\;(3)$$

Here *matchNxt* gives the total number of common words in the following words' lists of w_i and w_j.

Now we compute two similarities:

- The similarity of the first word with the second word.
- The similarity of the second word with the first word.

Let $P\,(w_i,w_j)$ be the similarity of w_i with w_j which is given by:

$$P\,(w_{i,},w_j)=\frac{matchPrv+matchNxt}{count(Previous(\boldsymbol{wi}))+count(Following(\boldsymbol{wi}))}\times 100\ldots(4)$$

Likewise, the similarity of w_j with w_i is given by:

$$P\,(w_{j,},w_i)=\frac{matchPrv+matchNxt}{count(Previous(\boldsymbol{wj}))+count(Following(\boldsymbol{wj}))}\times 100\ldots(5)$$

If the maximum value of (4) and (5) exceeds a predefined value and the minimum value of (4) and (5) exceeds another predefined value then we decide that w_i and w_j originate from the same root and we cluster them into the same group.

Then we sort the words of each group lexicographically and the word with the smallest length in the lexicographic order of any group is considered as the root of the words of that group.

For example, we have a corpus which contains the following four sentences:

1. ছাত্রছাত্রী এবং শিক্ষকগণ উক্ত অনুষ্ঠানটিতে অংশ নেন
2. ছাত্রছাত্রী এবং শিক্ষকগণ উক্ত অনুষ্ঠানটিতে অংশগ্রহণ করেন
3. অভিভাবকদের অনুষ্ঠানটিতে অংশ নিতে বিনীত অনুরোধ জানানো হয়েছে
4. শিক্ষাবিদদের অনুষ্ঠানটিতে অংশগ্রহণ করতে বিনীত অনুরোধ জানানো হয়েছে

Let us consider two unique words অংশ and অংশগ্রহণ.

Initially we generate previous words' list and following words' list for each of অংশ and অংশগ্রহণ. From the above four sentences whenever we find অংশ/অংশগ্রহণ we take its previous five words and add them to its previous words' list and similarly we take its following five words and add them to its following words' list.

Table II and Table III show the previous and the following words' lists for অংশ and অংশগ্রহণ from above four sentences.

TABLE II: PREVIOUS WORD LIST FOR অংশ AND অংশগ্রহণ

Word	Previous words found
অংশ	ছাত্রছাত্রী,এবং,শিক্ষকগণ, উক্ত, অনুষ্ঠানটিতে, অভিভাবকদের
অংশগ্রহণ	ছাত্রছাত্রী, এবং, শিক্ষকগণ, উক্ত, অনুষ্ঠানটিতে, শিক্ষাবিদদের

TABLE III : FOLLOWING WORD LIST FOR অংশ AND অংশগ্রহণ

Word	Following words found
অংশ	নেন, নিতে, বিনীত, অনুরোধ, জানানো, হয়েছে
অংশগ্রহণ	করেন, করতে, বিনীত, অনুরোধ, জানানো, হয়েছে

Now we find the prefix resemblance between অংশ and অংশগ্রহণ. The length of অংশ is 3 and the length of অংশগ্রহণ is 7. And the length of matched prefix is 3. Then the percentage of their prefix resemblance *S*(অংশ, অংশগ্রহণ) is given by (1).

$$S(\text{অংশ,অংশগ্রহণ}) = \frac{3}{Min(3,7)} \times 100 = 100$$

Our prefix resemblance threshold is 90% for the given corpus. This means, we can go to the next step of our method as prefixes of অংশ and অংশগ্রহণ show more than 90% resemblance.

Contextual similarity between অংশ and অংশগ্রহণ based on their previous and following words is given by (4) and (5) where the number of common words in their previous words' lists is 5 and the number of common words in their following words' lists is 4.

$$P\ (\text{অংশ, অংশগ্রহণ}) = \frac{5+4}{12} \times 100 = 75$$

$$P\ (\text{অংশগ্রহণ,অংশ}) = \frac{5+4}{12} \times 100 = 75$$

Maximum value of *P*(অংশ, অংশগ্রহণ) and *P*(অংশগ্রহণ, অংশ) is 75 and minimum value of *P*(অংশ, অংশগ্রহণ) and *P*(অংশগ্রহণ, অংশ) is also 75.

Many different pairs of threshold values were tested for the maximum and minimum of (4) and (5). Finally, for our given corpus our method gives the best result when thresholds are set to 30.0 and 0.75 respectively.

As maximum of *P*(অংশ, অংশগ্রহণ) and *P*(অংশগ্রহণ, অংশ) exceeds 30.0 and minimum of *P*(অংশ, অংশগ্রহণ) and *P*(অংশগ্রহণ, অংশ) exceeds 0.75, we consider that অংশ and অংশগ্রহণ are originated from the same root and they are clustered into the same group. As অংশ is the smallest of অংশ and অংশগ্রহণ, so অংশ will be the root of অংশ and অংশগ্রহণ.

V. RESULT ANALYSIS

Our procedure was implemented on a large Bangla corpus of 25,562,190 individual words which contain around 361,436 unique words. Our test corpus has about 1,593,398 sentences which is a mixture of varieties of topics like news, sports, blogs, websites, business magazines, journals etc. To obtain a decent result the main task was to generate good threshold values for minimum and maximum thresholds discussed in section IV.

A. Method Evaluation

As our data set was very large we chose about 5000 words from 361,436 unique words randomly and manually wrote the stem of each word. We were very careful and rigid about the correctness of each stemmed word. As our approach depends on two critical thresholds which greatly affect the quality of the stemmed words, we tested our method on different pairs of threshold values. The first threshold is the minimum required value of maximum of (4) and (5). Let it be th_{mx}. Similarly the second threshold is the minimum required value of the minimum of (4) and (5) which is labelled th_{mn}. During testing we observed that both the thresholds give more accurate results on a specific range. If we move higher or lower than this range the accuracy drops dramatically. Thus we considered this range to be the upper-bound and lower-bound of the thresholds. Table IV shows the accuracy for different pairs of thresholds. The values of the second row represent th_{mx} and values of the first column represent th_{mn}. The other cells of the table represent the exact number of words that were correctly stemmed by our method. Table IV shows that for our corpus the optimal result can be found for th_{mx}=30 and and th_{mn}=0.75 which give the accuracy of 40.18%.

TABLE IV : TOTAL NUMBER OF CORRECTLY STEMMED WORDS FOR DIFFERENT PAIRS OF THRESHOLD VALUES

th$_{mn}$	**th$_{mx}$**				
	30	***32***	***35***	***37***	***40***
0.5	2003	2005	2000	2004	2002
0.75	2009	2007	2002	2003	1998
1.0	2008	2004	1992	1988	1984
1.25	1999	1996	1984	1981	1978
1.5	1995	1995	1976	1974	1971

Table V shows some root words with their modified forms given by our approach for thresholds 30.0 And 0.75

TABLE V : SOME EXAMPLE OF CORRECTLY GENERATED ROOT FORMS AND THEIR MODIFIED FORMS FOR THRESHOLDS 30.0 AND 0.75

Root	Correctly Generated Modified Forms
অংশ	অংশটি, অংশে, অংশকে, অংশগ্রহণ, অংশবিশেষ
আকাঙ্ক্ষা	আকাঙ্ক্ষায়, আকাঙ্ক্ষাকে, আকাঙ্ক্ষাই
অনুষদ	অনুষদে, অনুষদভুক্ত, অনুষদসমূহ
আকর্ষণ	আকর্ষণের, আকর্ষণে
আওয়াজ	আওয়াজ, আওয়াজটা
উজ্জ্বল	উজ্জ্বলতা, উজ্জ্বলতম, উজ্জ্বলতা
ওয়েব	ওয়েবসাইট,ওয়েবসাইটটির,ওয়েবসাইটগুলোর,ওয়েবসাইটটি
খাদ্য	খাদ্যদ্রব্য, খাদ্যে, খাদ্যশস্য,খাদ্যপণ্য, খাদ্যশস্যের, খাদ্যনিরাপত্তা
ঘন্টা	ঘন্টায়, ঘন্টার
ছোট	ছোটদের, ছোট্ট, ছোটখাটো, ছোটো
ঝুঁকি	ঝুঁকিতে, ঝুঁকিপূর্ণ, ঝুঁকিমুক্ত, ঝুঁকিও
ঝামেলা	ঝামেলায়, ঝামেলার
আইন	আইনী, আইনি, আইনত, আইনমন্ত্রী, আইনগতভাবে, আইনশৃঙ্খলা
আকাশ	আকাশটা, আকাশকে, আকাশচুম্বী
ইচ্ছা	ইচ্ছার, ইচ্ছায়, ইচ্ছাকৃতভাবে, ইচ্ছামতো
ইস্যু	ইস্যুর, ইস্যুটি, ইস্যুভিত্তিক
ইসলাম	ইসলামের, ইসলামকে, ইসলামীর, ইসলামিক
এখন	এখনকার, এখনো, এখনই, এখনও
ঠিক	ঠিকমতো, ঠিকভাবে, ঠিকঠাক
ঢাকা	ঢাকার, ঢাকায়, ঢাকাকে
দখল	দখলে, দখলদার, দখলদারদের, দখলের, দখলমুক্ত
নদ	নদনদীর, নদী, নদীর
নজরুল	নজরুলের,নজরুলকে
ফুল	ফুলে, ফুলের
ফুটো	ফুটোটা, ফুটোর
মহিলা	মহিলাকে, মহিলাদের, মহিলার
এখতিয়ার	এখতিয়ারও, এখতিয়ারবহির্ভূত, এখতিয়ারভুক্ত
ঐক্য	ঐক্যজোটের, ঐক্যবদ্ধভাবে
ঐতিহ্য	ঐতিহ্যগত, ঐতিহ্যবাহী
কক্ষ	কক্ষগুলো, কক্ষটি
গাছ	গাছগুলো, গাছটি, গাছপালার, গাছপালা
ঝড়ো	ঝড়ো
চরমে	চরমে, চরমপন্থী, চরমভাবে
জমি	জমির, জমিজমা, জমিতে, জমিন
টুকরো	টুকরোর, টুকরোগুলো, টুকরোটি
তথ্য	তথ্যগুলো, তথ্যে, তথ্যের, তথ্যপ্রযুক্তি, তথ্যপ্রযুক্তির, তথ্যাদি
ধারাবাহিক	ধারাবাহিকটি, ধারাবাহিকে, ধারাবাহিকের
পদার্থ	পদার্থে, পদার্থের, পদার্থবিদ্যা, পদার্থগুলো

VI. CONCLUSION

To aid the computerization of Bangla language constructing an efficient Bangla stemmer is very essential. For this purpose we propose this approach which is easy to implement and makes use of large collection of Bangla text data already generated in computers. As our approach is entirely based on unsupervised learning, it is required to make use of as much data as possible to get more information about Bangla word formation patterns. Our method is dynamic and demands huge amount of data to produce more appropriate results. This method will be able to build a rich root word dictionary for Bangla if big amount of Bangla text is used and good threshold values are generated.

REFERENCES

[1] Dasgupta, S. and Khan, M. (2004). *Morphological parsing of Bangla wods using PC-KIMMO* [Online].Available: http://123.49.46.157/handle/10361/614#files-area

[2] Sarkar, Sandipan, and Sivaji Bandyopadhyay. "Study on Rule-Based Stemming Patterns and Issues in a Bengali Short Story-Based Corpus," presented at the International Conference on Natural Language Processing, Hyderabad, ICON. 2009.

[3] Md. Zahurul Islam, Md. Nizam Uddin and Mumit Khan. (2007). *A light weight stemmer for Bengali and its use in spelling checker* [Online]. Available: http://dspace.bracu.ac.bd:8080/xmlui/handle/10361/328

[4] Das, Amitava, and Sivaji Bandyopadhyay. "Morphological stemming cluster identification for Bangla," *Knowledge Sharing Event-1: Task* , 2003, vol. 3.

[5] Sabir Ismail et al., "Developing an automated Bangla parts of speech tagged dictionary," in *Computer and Information Technology (ICCIT), 2013 16th International Conference*, Khulna, ICCIT. 2013, pp. 355 – 359.

[6] Sabir Ismail, and Md Shahidur Rahman. "Bangla word clustering based on N-gram language model," in *International Conference Electrical Engineering and Information & Communication Technology*, Dhaka, ICEEICT. April 2014, pp. 1-5.

[7] Husain, Mohd Shahid. "An unsupervised approach to develop stemmer,"*International Journal on Natural Language Computing*, IJNLC. 2012.

[8] Utpal Sharma, Jugal Kalita, and Rajib Das. "Root word stemming by multiple evidence from corpus." *Proceedings of the 6th International Conference on Computational Intelligence and Natural Computation*, CINC. 2003.

[9] Utpal Sharma, Jugal Kalita, Rajib Das, "Unsupervised learning of morphology for building lexicon for a highly inflectional language," in *Proceedings of the ACL-02 workshop on Morphological and phonological learning,* 2002, vol. 6, pp. 1-10.