Designing a Bangla Stemmer using rule based approach

Conference Paper · September 2019

DOI: 10.1109/ICBSLP47725.2019.201533

Shahidul Shakib
Khulna University of Engineering and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

K. M. Azharul Hasan
Khulna University of Engineering and Technology
69 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Designing a Bangla Stemmer using rule based approach

MD Shahidul Salim Shakib
Computer Science and Engineering department
Khulna University of Engineering & Technology
Khulna,Bangladesh
salim1507034@stud.kuet.ac.bd

Tanim Ahmed
Computer Science and Engineering department
Khulna University of Engineering & Technology
Khulna,Bangladesh
ahmed1507113@stud.kuet.ac.bd

K. M. Azharul Hasan
Computer Science and Engineering department
Khulna University of Engineering & Technology
Khulna,Bangladesh
az@cse.kuet.ac.bd

***Abstract*—Stemming is a preprocessing task for natural language processing that involves normalizing inflected words representing the same concept of the original word. Steaming is a process of text normalization that has many applications. There are many techniques for steaming of inflected words for different languages but very few works for Bangla word steaming. Therefore, stemming Bangla word is a unsolved problem. There are many different situations that can occur in Bangla language for word steaming. In this paper, we present a rule based algorithm to stem Bangla words. We developed the rules for infection detection for verb inflection (বিভক্তি), number inflection (বচন), and others. Using our rules, we developed a system to find the root word of Bangla words and found good performance. Sufficient examples are provided to explain the proposed system.**

Keywords—stemming, text normalization, root word detection, corpus, NLP.

I. INTRODUCTION

Stemming is the process for removing the commoner morphological and inflectional endings from words. The main applications of stemmers are in information retrieval and to increase the recall rate of text mining. There are two categories of stemming namely rule based and corpus based[1]. The rule based stemming removes the suffixes by repeatedly replacing another word(s) or null string. On the other hand, the corpus-based system groups the similar sets that represent the same concept. Popular stemmers are mainly based on rule based system. This is because rule based system is faster and does not require any preprocessing once the rules are generated. The rule based system that encode a large number of language specific rules are available only for a few languages. The most popular and well known stemming approach for English is a rule-based approach namely Porter stemmer [2]. The Porter stemmer successively applies rules to convert a word form into its base form. For example, words such as *hopelessness will be* reduced to *hope* by removing the suffixes *ness* and *less*. There is no effective and full phase Bangla stemmer yet and it is in infant stage[3]. Although there are different kinds of the algorithms for different languages for stemming task, but there is no efficient algorithm for Bangla stemming. In present, there have some works for Bangla word stemming but they follow brute force system. And the accuracy of this system is poor. For the development of Bangla language in Natural Language Understanding(NLU) and Bangla Language Processing, stemming of Bengali words is a must. Due to the morphological structure of Bangla language,it has some difficult issues to design a stemmer. We have proposed a better solution for solving this problem. We developed some important rules to solve this problem. We have divided the words into some categories namely noun, verb, number and others.And proposed an algorithm that will perfectly stem the Bangla words. We also created an exception set to avoid the ambiguity. Our algorithm provides efficiency of 94% and the efficiency can be improved by increasing the terms in the rules. The scheme can be applied to text mining for sentiment analysis[4], Parse design[5], Spam filter[6] and many others.

II. RELATED WORKS

In 1980 "Porter Stemmer" was developed by Martin Porter[2]. Porter stemmer defines some rules these rules applied in a word for removing suffix. Five steps are followed in porter stemmer to stem any word. Several stemming algorithms exist for English such as Lovins Stemming[7], Praice/Husk[8], Dawson[9] and so on. Porter Stemmer is most recognized one that is applied English. Some other stemmers for other languages are Indonesian[10], Malay[11], Dutch[12], Slovene[13], Turkish[14], Latin[15] etc.. There remains some works for stemming Bangla words[16][3][17][1]. Bangla is a language that is highly inflected. It is commonly inflected with noun, verb, adjective. A Light Weight Stemmer for Bengali and its use in spell checker proposed in [17]. [16] describes a clustering-based approach to discover equivalence classes of root words and their morphological variants. Using their clustering technique, they define distance measures to identify equivalent classes. A design a rule-based stemmer for natural language text proposed in [3] using regular expressin syntax. They have another one rule-based stemmer developed for Hindi and Bengali[1] to measure the retrieval effectiveness.

III. Methodology

The proposed stemming approach follows some rules. These rules show how the stemming will be performed. And further an algorithm is developed. This algorithm works based on the rules. The rules are developed with the following general rule.

General rule

Let α is a part of a string of β (α⊆β and α≠β) . Where β is a word and αcan be found at the end of the β. We define the general rule for stemming of the form

α→ ν

where α is replaced by ν to get the non inflected form of β.If ν= ε, then we mean "ε" is an empty string.

Rule for number (বচন) inflection reduction

The general for of the rule is

α→ ε .

For example if β is ছেলেটি then rule is টি→ ε where "টি"is replaced by ε("ε" is an empty string) to get ছেলে. Some possible values of α can be as follows.

টি→ε,[মেয়েটি]ব্যতিক্রমঃখিটিমিটি,বৃষ্টি)

টি→ε,[ছেলেটি]

খানা→ε,[খাতাখানা]

খানি→ε,[বইখানি]

রা →ε,[পাখিরা]

গুলো→ε,[আমগুলো]

গণ →ε,[দেবগণ]

সমূহ→ε,[বৃক্ষসমূহ]

বলি →ε,[পুস্তকাবলি]

গুচ্ছ→ε,[কবিতাগুচ্ছ]

Therefore, we summarize the number (বচন)inflection reduction rules

Rule 1:

বৃন্দ /মন্ডলী /কুঞ্জ /পুঞ্জ /গুচ্ছ /বৃন্দ /সমুদয় /সমূহ /বর্গ / রাশি /আবলি /খানা /গুলি /রাজি /নিকর /খানি /নিচয় / গুলো /মালা /যূথ /সকল /বলি /দের /এরা /দিগ /পাল /দাম / কূল/ টা/ রা/ সব/ গণ/ টি/ ◌ান→ε

Rule for bivokti(বিভক্তি)inflection reduction

The rule for bivokti(বিভক্তি)inflection reduction is α →ε . For example if β is করাইthen rule is ◌াই→ε where "◌াই" is replaced by ε to get করাই to কর.Some possible values of α can be as follows.

◌ায় →ε,[করায়]

◌াও→ε,[করাও,পড়াও]

◌াইস→ε, [করাইস]

◌াচ্ছি →ε,[করাচ্ছি]

◌াক →ε,[করাক]

◌াও →ε,[করাও]

◌াইস→ε,[করাইস]

◌ালেন→ε,[করালেন]

◌ালাম→ε,[করালাম]

◌াইতে→ε,[করাইতে]

◌াতে→ε,[করাতে](ব্যতিক্রমঃ উৎপাতে)

Therefore, we summarize the rule for bivokti(বিভক্তি) inflection reduction as follows.

Rule 2:

◌েছ/ ◌িয়েছ/ ◌ায়/ ◌াচ্ছি/ ◌ায়েছি/ ◌াচ্ছে/ ◌াতিস/ ◌ালাম/ ◌ালেন / ◌াইলে / ◌াইবি / ◌াবেন / ◌াইতে / ◌াতেন / ◌াচ্ছ / ◌াইলি / ◌াতাম / ◌াইবে / ◌াইব / ◌ালি / ◌ালে / ◌াউক / ◌াবো / ◌াইস/ ◌ায়ো/ ◌াবে/ ◌াইও/ ◌াইয়/ ◌াবি/ ◌াতে/ ◌াছে/ ◌াক/ ◌াও / ◌াইয়েছিলাম / ◌াইতেছিলেন / ◌াইয়েছিলাম / ◌াইতেচ্ছিস / ◌াইতেছিস / ◌াইয়েছিলেন / ◌াইয়াছিলেন / ◌াইতেছিলাম / ◌াইয়াছিলে / ◌াইতিছিলি / ◌াইয়েছিলি / ◌াইয়াছিলি/ ◌াইয়েছিলে/ ◌াচ্ছিলাম/ ◌াইতেছিলে/ ◌াচ্ছিলেন / ◌াইয়েছিস/ ◌াইয়েছিল/ ◌াচ্ছিলে/ ◌াইয়াছিস/ ◌াইতেছেন /◌াইয়াছেন/ ◌াচ্ছিলি/ ◌াচ্ছিস/ ◌াইয়াছি/ ◌াচ্ছেন/ ◌াইতেছি/ ◌াইয়েছি/ ◌াইয়াছে/ ◌াইয়েছে/ ◌াইতেছে/ ◌াইতেছ/ ◌াইতেন / ◌াইলেন/ ◌াইতাম/ ◌াইতিস/ ◌াইবেন/ ◌াইলাম/ ◌াইয়েছ/ ◌াইয়াছ / ◌িয়েছেন / ◌িয়েছ / ◌িয়েছিস / ◌িয়েছি / ◌াইলি / ◌োচ্ছিল / ◌িতেছিল / ◌িয়াছিলে / ◌েছিলেন / ◌েছিলে / ◌িয়াছিলি / ◌েছিলি / ◌িতেছিল / ছিল / ◌িয়াছিলেন / ◌িয়াছিলি / ◌েছিলি / ◌িয়াছিলাম / ◌িবে / ◌িবি / ◌িবে / ◌িয়া / ◌িবেন / বেন / ◌িয়া / ◌িস / ◌িয়েছেন / ◌িয়াছি / ◌াইল→ε

Others(বিবিধ)

In this section we present some rules for other inflection detection. The rule is of the form α→ε.For example if β is 'শুরুতেই' then rule is তেই→ ε where "তেই" is replaced by εto get শুরুতেইtoশুরু. Some examples are

তে→ε,[ছাড়তে] (ব্যতিক্রমঃহাতে,ভাতে)

◌ের→ε,[যুগের]

তেই→ε,[শুরুতেই]

তে→ε,[শেয়াকুলিতে]

◌ের→ε,[গৌরবের]

তে→ε,[দিঘিতে]

টাই→ε,নামটাই]

Therefore we summarize our rule 3 as.

Rule 3:

তে/ ◌ের/ তেই/ তে/ ◌ের/ তে/ টাই→ε

Exception

We develop an exception list a part of which is shown in Table 1 that contains the words that has same suffix with the rules we define. These exceptions are not candidate for stemming. This is because if we stem these words then they will lose their real meaning. These words are root words. For example: খোটা,যারা,তারা. Although 'টা' is a suffix in rule 1, but it is not suffix in the word 'খোটা'. Another situation occurs for exception is that '*the whole word*' i.e. ($\alpha=\beta$) for general rule. For example 'বলি', 'গুচ্ছ' can be a single word instead of suffix and this should not be stemmed.

Table 1: Exception list for not to stemming.

Exception
ক্লাস ,দিয়ে, শাসন , পরিচয়, আমাদের, নতুন ,তাদের ,থেকে, এদের, তারা ,যারা, সন্ধান, মার ,পারে ,দিতে, দরকার ,খিটিমিটি, নিয়ে ,খোঁটা

Table 2: Others stemming rules

বিবিধ (Example)		
ঠল → ঠ (উঠল)	রছে →র(করছে)	লের →ল(কমলের)
টাই→ε(পাঁচটাই)	জের →জ(তেজের)	েরই →ε (পরেই)
ইতেছি-->ε(করাইতেছি)	নে →ন(বনে)	রের →র(কবরের)
ও →ε(আমারও)	লেম →ε(শুনলেম)	টার →ε(জলটার)
রলে →র(করলে)	নের →ন(বিসর্জনের)	নীর →নী(ধরনীর)
য়ে →ε(পায়ে)	খে →খ(দেখ)	েছি →ε(করেছি)
রে →র(করে)	ণ্ঠে →ণ্ঠ(কণ্ঠে)	ড়ে →ড়(পড়ে)
ড়াতে-->ড়(পড়াতে)	লায় →লা(মামলায়)	দের-->ε(পরিবারদের)
তেই-->ε(শুরুতেই)	ময়ে →ময়(সময়ে)	খি →খ (দেখি)
মার-->মা(প্রতিমার)	ের →ε(গৌরবের)	েধেছে→ াধ(বেধেছে)
তে →ত(ভাতে)	লের →ল(সজলের)	তার → তা(রাস্তার)
দ্দর →দ্দ(বরাদ্দর)	লতে →ল(চলতে)	বার→ε (বাঁধবার)
	পে →প(মাপে)	

Algorithm1:

```
</*An algorithm to remove all possible suffixes from a word*/

Input: st: String of Bangla sentence having n words

Output: A sentence with stemming word

BanglaStemmer()
Begin
Word[i] ←Tokenize (st)/* i=1 to n */
Word[i] ←Remove stop word from the Word[i]
/* i=1 to n */
for i←  1 to n do{
α ← NULL;
     l ← length(word[i])

for j ← ( l-1) down to0 do
α← word[i]
    if(strcmp(α, NotSteamed_suffix)==0) break;
   if(strcmp(α, Bochon_suffix)==0){
    stem[i] = trim(word[i], α);
                // removes α from the end of  string
    break;
  }
    if(strcmp(α, Bivokti_suffix)==0){
    stem[i] = trim(word[i], α);
                // removes α from the end of  string
    break;
  }
    if(strcmp(α, Other_suffix)==0){
    steam[i] = trim(word[i], α);
                // removes α from the end of  string
    break;
  }
} // for j
}// for i
end.
```

After avoiding the exceptions shown in Table 1 we create another rule of the form $\alpha \rightarrow \gamma$, where α is replaced by γ to set the non inflected word. For example পে is replaced by প. Some examples are,

জের→জ,[তেজের]
খি→খ, [দেখি]

ার→া,[বাঁধবার]

নের→ন,[আয়তনের]

তার →তা,[দেবতার]

নের→ন,[বিসর্জনের]

তার → তা,[রাস্তার]

পে → প,[মাপে]

ার → া,[প্রতিমার]

Table 2 summarizes the rule for (বিবিধ) some other stemming rules for called rule 4. Using the rules defined above, we developed an algorithm for Bangla stemmer as shown in Algorithm 1.

IV. RESULTS

For checking accuracy we have taken a part of the famous novel by Rabindronath Thakur "যোগাযোগ" as data set. Table 3 shows the accuracy test using the confusion Matrix. Table 3 shows promising results. The results can be found in [18].Same as we have taken another article from [19] and Table 3 shows promising results.

.

Table 3: Accuracy Calculation

Data Set	Target to stem	Truly Stemmed	Falsely Stemmed	Not Stemmed	Accuracy
D1	172	162	5	5	94.35%
D2	305	271	26	8	91.47%

V. Conclusion

We developed a rule based algorithm for Bangla stemmer. The approach is general and can be applied to any words. We create rules as well as exception list for word to stemming. The exception list makes the stemmer algorithm to work correctly and to produce good performance avoiding the ambiguity. We found good results in the experimental results. The approach can be made more powerful and accurate by increasing the terms in the rules for substituting α. This is a part of an ongoing work. We plan to make a parallel version of the algorithm to improve the response time of the algorithm.

Reference

[1] D. Ganguly, J. Leveling, and G. J. Jones, “Dcu@ fire-2012: rule-based stemmers for bengali and hindi,” In: FIRE 2012 Workshop, pp. 17-19, 2012.

[2] M.F. Porter, “*An algorithm for suffix stripping*”, Program, 14(3) 1980, pp.130−137.

[3] S. Sarkar and S. Bandyopadhyay, “Design of a rule-based stemmer for natural language text in bengali,” in Proceedings of the IJCNLP-08 workshop on NLP for Less Privileged Languages, 2008.

[4] SM Abu Taher, Kazi Afsana Akhter, K M Azharul Hasan “ N-Gram Based Sentiment Mining for Bangla Text Using Support Vector Machine”, In: International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1-5, 2018.

[5] K M Azharul Hasan, Al-Mahmud, Amit Mondal, Amit Saha, “Recognizing Bangla Grammar using Predictive Parser”, International Journal of Computer Science and Information Technology (IJCSIT), 3: 6. pp. 61-73 , 2011.

[6] Karl-Michael Schneider, A comparison of event models for Naive Bayes anti-spam e-mail filtering, Proceeding of EACL '03 Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics, pp. 307-314. 2003.

[7] J.B.Lovins, “Development of a stemming algorithm”, *Mechanical Translation and Computational Linguistics 11*, 1968, pp. 22-31.

[8] C.D. Paice, “An evaluation method for stemming algorithms”, In the *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1990, pp. 42 – 50.

[9] J. Dawson, “Suffix removal and word conflation”, *ALLCbulletin, 2(3)*, 1974, pp. 33–46.

[10] V. Berlian, S.N. Vega, and S. Bressan, “Indexing the Indonesian web: Language identification and miscellaneous issues”, In the *Tenth International World Wide Web Conference*, Hong Kong. 2001.

[11] S.Y. Tai, C.S. Ong, and N.A. Abdullah, “On designing an automated Malaysian stemmer for the Malay language”, (Poster) In the *Proceedings of the fifth international workshop on information retrieval with Asian languages*, Hong Kong, 2000, pp. 207-208.

[12] W. Kraaij and R. Pohlmann, “Viewing stemming as recall enhancement”, In the *Proceedings of ACM SIGIR96*, 1996, pp. 40-48.

[13] M. Popovic and P.Willett, “The effectiveness of stemming for natural language access to Slovene textual data”, *JASIS, 43 (5)*, 1992, pp.384-390.

[14] F.C. Ekmekcioglu, M.F. Lynch and P.Willett, “Stemming and n-gram matching for term conflation in Turkish texts”, *Information Research News, 7 (1)*, 1996, pp. 2-6.

[15] M. Greengrass, A.M. Robertson, S. Robyn, and Willett, “Processing morphological variants in searches of Latin text”, *Information research news, 6 (4)*, 1996, pp. 2-5.

[16] P. Majumder, M. Mitra , S. K. Parui,G. Kole, P. Mitra, K. Datta, “YASS: Yet Another Suffix Stripper”, ACM Transactions on Information Systems, Vol. 25, No. 4, 2007

[17] M. Islam, M. Uddin, M. Khan, et al., “A light weight stemmer for Bengali and its use in spelling checker,” BRAC University, Institutional repository, 2007.

[18]https://github.com/shahidul034/stemmer-accuracy-test

[19]https://www.bd-pratidin.com/first page/2019/09/03/453632