A formal method for designing a Bangla Stemmer using rule based approach

Conference Paper · December 2019

CITATIONS
0

READS
97

3 authors, including:

Tanim Ahmed
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

A formal method for 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

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

***Abstract*—In ML-based NLP makes your training data denser. It reduces the very large dictionary (number of words used in the corpus) simple. For Natural Language Processing stemming is necessary. Although there have many techniques for stemming different language. But in Bengali, there has no expected algorithm for stemming the Bengali Language. We have proposed a system for stemming Bangla word. It is possible to build up a search engine using this corpus since search engine works with keywords. Our system can help to find those root word. It also has other applications such as sentiment analysis, spam filter etc.**

Keywords—stemming, search engine, sentiment analysis, corpus, NLP

I. Introduction

Stemming is the process for removing the commoner morphological and inflectional endings from words. Although there are different kinds of the algorithm for different languages for this task. Like as Porter Stemmer is a famous algorithm for stemming English words. But for Bengali almost there has no efficient algorithm. In the present, there have some works for Bangla Stemming but they are almost they follow brute force system. And the accuracy of this system is very poor. For the development of Bengali Language in Natural Language Understanding(NLU) and Natural Language Processing(NLP), stemming of Bengali Language is necessary. But it has some difficult issue to solve this problem. Because if you follow the brute force approach to solve this problem you accuracy and complexity will be very poor. But we have proposed a better solution for solving this problem. We have created some important rules to solve this problem. We have divided the words some categories(NOUN, VERB, NUMBER, etc.).And proposed an algorithm that will perfectly stem the Bengali words from dataset. And our algorithm provides better efficiency. The efficiency of our algorithm almost 94.35%. Table 0 shows some example how the words are stemmed with our techniques.

বিভক্তি(verb inflection) Example	(বচন)Number Example	Others Example
ায় → ɛ # করায়	টি --> ɛ # মেয়েটি (ব্যতিক্রমঃ খিটিমিটি, বৃষ্টি)	তে → ɛ # ছাড়তে (ব্যতিক্রমঃ হাতে,ভাতে)
াও → ɛ # করাও,পড়াও	টি --> ɛ # ছেলেটি	জের → জ # তেজের
াইস → ɛ # করাইস	টা --> ɛ # কলমটা	ার → া # বাঁধবার
াচ্ছ → ɛ # করাচ্ছ	খানা -->ɛ # খাতাখানা	ের → ɛ # যুগের
াচ্ছি → ɛ # করাচ্ছি	খানি -->ɛ # বইখানি,	তেই → ɛ # শুরুতেই
াছে → ɛ # করাইয়াছে,	রা --> ɛ # ছাত্ররা	খি → খ # দেখি
েছ → ɛ # করিয়েছ	রা --> ɛ # পাখিরা,	তে → ɛ # শেয়াকুলিতে,
াক → ɛ # করাক	রা-->ɛ # পিপীলিকারা,	নের →ন # আয়তনের
াও → ɛ # করাও	গুলো--> ɛ # আমগুলো,	ের →ɛ # গৌরবের
াইস → ɛ # করাইস	গুলো-->ɛ # টাকাগুলো,	তে → ɛ # দিঘিতে
ালেন → ɛ # করালেন ালি → ɛ # করালি	গুলো-->ɛ # ময়ূরগুলো,	টাই → ɛ # নামটাই
ালাম → ɛ # করালাম	গণ --> ɛ # দেবগণ,	তার → তা # দেবতার
াইতে → ɛ # করাইতে	সমূহ --> ɛ # বৃক্ষসমূহ	নের → ন # বিসর্জনের
াতে → ɛ # করাতে (ব্যতিক্রমঃ উৎপাতে)	বলি --> ɛ # পুস্তকাবলি	তার → তা # রাস্তার
	গুচ্ছ --> ɛ # কবিতাগুচ্ছ	পে → প # মাপে
	দাম --> ɛ # কুসুমদাম	ার → া # প্রতিমার

Table 1

II. Related Works

In 1980 “Porter Stemmer” was developed by Martin Porter[1]. Porter stemmer defines some rules these rules applied in a word for removing suffix. Five steps are followed in porter stemmer[2] to stem any word. Several stemming algorithms exist for English such as Lovins Stemming[3],Praice/Husk[4], Dawson[5] and so on. Porter

Stemmer is most recognized one that is applied English and others languages that are Indonesian[6],Malay[7],Dutch[8],Slovene[9],Turkish[10] and Latin[13].For Bengali, there remain some works in this field. Bengali 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 [15].A design a rule-based stemmer for natural language text proposed in [14].They have another one rule-based stemmer developed for Hindi and bengali[16].

III. Methodology

The stemming approaches follows some rules. These rules show how the stemming will be performed. And further an algorithm is developed. This algorithm works based on the following rules.

Number (বচন) CFG :

Rule 1: A # Remove A if it appears at the end of word (where A = টি , টা,খানা,খানি (table 1))

Example:
ছেলেটি -> ছেলে
বইখানা -> বই
Example with sentence:
কলমটি সুন্দর
কলম সুন্দর

Bivokti(বিভক্তি) CFG:

Rule 1: A # Remove it if A appears at the end of word (where A =াই, াও,াইস table 1)

Example:
করাই -> কর
পড়াও -> পড়
Example with sentence:
কাজটি করাও
কাজ কর
শিক্ষক পড়াই
শিক্ষক পড়

Others(বিবিধ) CFG:

Rule 1: A # Remove it if A appears at the end of word (where A = তেই,তে(table4))

Example:
শুরুতেই -> শুরু
ভাতে -> ভাত
Example with sentence:
মাছে ভাতে বাঙালি
মাছ ভাত বাঙালি

Rule 2: B -> C # Replace B with C if B appears at the end of a word. (B =পে, খি, C = প, খ(table 4))

Example:
মাপে -> মাপ
দেখি -> দেখ
Example With Sentence:
মাপে ঠিক আছে
মাপ ঠিক আছে

Don't stemmed:

IF the data of the table 4 is contained in the suffix then there is no need no change the word. Because if we strip these words then they will lose their real meaning These word will be directly used as root word.
Example: খোটা,যারা,তারা (table 5)

ক্রিয়া-বিভক্তি
ে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

Table 2

বচন(বিশেষ্য-সর্বনাম)
বৃন্দ | মন্ডলী | কুঞ্জ | পুন্জ | 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 | গেন → ε

Table 3

বিবিধ		
তে → ɛ	তার ---> তা	লেই ---> লা
ঠল → ঠ	বার ---> ɛ	নে ---> ন
টাই ---> ɛ	োর ---> ো	োচ্ছিল ---> ɛ
ইতেছি ---> ɛ	পে ---> প	রিতে ---> রি
নোর ---> ɛ	মার ---> মা	ললে ---> ল
টার ---> ɛ	োতে ---> ো	েছি ---> ɛ
সছে ---> স	তে ---> তা	লেম ---> ɛ
লের ---> ল	দ্দর ---> দ্দ	খে ---> খ
েও ---> ɛ	ঠল ---> ল	র্বে ---> র্ব
ও ---> ɛ	মল ---> ম	নীর ---> নী
রলে ---> র	দের ---> ɛ	লের ---> ল
য়ে ---> ɛ	শে ---> শ	নের ---> ন
রে ---> র	ড়ে ---> ড়	লায় ---> লা
ড়াতে ---> ড়	েছে ---> ɛ	ষ্ঠে ---> ষ্ঠ
তেই ---> ɛ	েরই ---> ɛ	লায় ---> লা
খি ---> খ	রছে ---> র	ময়ে ---> ময়
ষ্ঠে ---> ষ্ঠ	টিয়ে ---> টা	জের ---> জ
েধেছে ---> োধ	োতেও ---> ɛ	রের ---> র
তে ---> ɛ	জে ---> জ	য়ে ---> য়
টা ---> ɛ	লতে ---> ল	ের ---> ɛ
কুল ---> ɛ	োরে ---> র	েছে ---> ɛ

Table 4

Do not stem
ক্লাস ,দিয়ে, শাসন , পরিচয়, আমাদের, নতুন ,তাদের ,থেকে, এদের, তারা ,যারা, সন্ধান, মার ,পারে ,দিতে, দরকার ,খিটিমিটি, নিয়ে ,খোঁটা

Table 5

Algorithm 1:

1: procedure: BanglaStemmer(string1, string2, string3) ◅ /*An algorithm to remove all possible suffixes from a word*/

2: Input: st: String of Bangla sentence having n words , string1:Set of all longest Bivokti(বিভক্তি) suffix , string2: Set of all smallest Bivokti(বিভক্তি) suffix, string3:Set of all bochon(বচন) suffix, string4: Set of all Extra suffix, do_not_stem: Dataset of do not stem these word

3: Output: A sentence with stemming word

4: data[i] ← Tokenize (st)/* i=1 to n */

5: data2[i] ← Remove stop word from the data[i]

6: for each word ← data2[i] do

7: if (word doesn't match with do_not_stem dataset) then

8: stemmed_word ← Take Suffix of this word and match with the string1, return stemmed word

9: stemmed_word ← Take Suffix of this word and match with the string2, return stemmed word

10: stemmed_word ← Take suffix of this word and match with the string3, return stemmed word

11: stemmed_word ← Take suffix of this word and match with the string4, return stemmed word

12: end if

13: end for

IV. Results

For checking accuracy we have taken dataset from "যোগাযোগ(রবীন্দ্রনাথঠাকুর)"[18].Firstly we have found the accuracy close to 85%.Then we have modified our rules. After some iteration we have found accuracy 94.35%. Table 5 shown accuracy test.

Accuracy test
Total: 172 stemming word(target)
stemmed :162 (True positive)
false stemmed:5 (True negative)
Should be stemmed but not stemmed : 5(False positive)
Accuracy: 94.35% ((TP+TN)/(T+N)) (Using confusion matrix) [17]

Table 5

Future Work

We will add if any rules that we have not yet added such the accuracy will be close to 100%. And try to propose our modified algorithm such that this algorithm gives best time complexity and space complexity and we use this to build Bangla Word2Vec.

Conclusion

Finally, we have perfectly build up an efficient algorithm and technique for Bengali stemming. That is more perfect than the exists technique in the perspective of accuracy and complexity. Although we have faced some problem during

the rules generation after adding some exceptional rules we have overcome it successfully. This techniques algorithm can be used in many aspects for the development of the system(Sentiment analysis, Search engine, Spam filter, etc.) that is based on the Bengali Language.

REFERENCE

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

[2] C.J. van Rijsbergen, S.E. Robertson and M.F.Porter, "New models in probabilistic information retrieval", *British Library Research and Development Report, no. 5587*, 1980.

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

[4] 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.

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

[6] 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.

[7] 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.

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

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

[10] 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.

[11] I. Moulinier, A. McCulloh and E. Lund, "Non-English monolingual retrieval in Cross language information retrieval and evaluation", In the Proceedings of the *CLEF 2000 workshop*, C. Peters, Ed.: Springer Verlag, 2001, pp. 176-187.

[12] C. Monz and M.de Rijke, "Shallow morphological analysis in monolingual information retrieval for German and Italian in Cross-language information retrieval and evaluation", In the *Proceedings of the CLEF 2001 workshop*, C. Peters, Ed., Springer Verlag, 2001.

[13] 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.

[14] 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.

[15] M. Islam, M. Uddin, M. Khan, et al., "A light weight stemmer for bengali and its use in spelling checker," 2007.

[16] D. Ganguly, J. Leveling, and G. J. Jones, "Dcu@ fire-2012: rule-based stemmers for bengali and hindi," 2012.

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

[18]https://bn.wikisource.org/wiki/%E0%A6%AF%E0%A7%8B%E0%A6%97%E0%A6%BE%E0%A6%AF%E0%A7%8B%E0%A6%97/%E0%A7%A7

[19]বাংলা ব্যাকরণ (নবম ও দশম শ্রেণি) Bangla Bakaron Class 9-10.