Design and implementation of an effective web-based hybrid stemmer for Odia language

Gouranga Charan Jena, Siddharth Swarup Rautaray
School of Computer Engineering, KIIT Deemed to be University, India

Article Info

Article history:

Received May 3, 2019
Revised Aug 20, 2019
Accepted Jan 5, 2020

Keywords:

Brute force
Derivational suffixes
Inflectional words
Information retrieval
Mula
Odia stemmer

ABSTRACT

Stemmer is used for reducing inflectional or derived word to its stem. This technique involves removing the suffix or prefix affixed in a word. It can be used for information retrieval system to refine the overall execution of the retrieval process. This process is not equivalent to morphological analysis. This process only finds the stem of a word. This technique decreases the number of terms in information retrieval system. There are various techniques exists for stemming. In this paper, a new web-based stemmer has been proposed named as "Mula" for Odia Language. It uses the Hybrid approach (i.e. combination of brute force and suffix removal approach) for Odia language. The new born stemmer is both computationally faster and domain independent. The results are favourable and indicate that the proposed stemmer can be used effectively in Odia Information Retrieval systems. This stemmer also handles the problem of over-stemming and under-stemming in some extend.

Corresponding Author:

Gouranga Charan Jena,
School of Computer Engineering,
KIIT Deemed to be University,
Bhubaneswar, Odisha, India.
Email: jenagouranga2000@gmail.com

1. INTRODUCTION

Stemming is a technique for removing inflectional or derived word to its stem. This technique is used to remove the suffix or prefix affixed in a word. This process finds stem of a word. Stemmer is essential for information retrieval system to refine the performance of the system. Its technique is not equivalent to morphological analysis. Its primary objective is to decreases the number of terms in an information retrieval system. Stemming technique can be used in information retrieval to decrease as many related words to a common form that is not in base form. For example, the English word "Computation" has different inflections such as 'Comput', 'Compute, 'Computing', 'Computes' etc. In this case stemmer can be used to reduce derived words into its root or stem word. Many stemmers had been developed for different languages, which reduce a word to its root/stem form. It ultimately reduces the index file size in an information retrieval system. In this way we can improve recall (i.e. the number of documents retrieved in response to a query.) of an IR system by effectively using stemmer in the background. Since many derivational words are mapping into one word i.e. root or stem. It ultimately reduces the volume of the index files in the IR system.

There are several types of stemming algorithms exists and it differs in respect to their performance and accuracy. There are various algorithms used to find a stem of a word i.e. (a) Brute-force algorithms: It uses a lookup table that contains derived words with their corresponding roots. To find the root/stem of a word, the table is queried to find a matching inflection word. If a matching inflection word is found, then corresponding root returned. Otherwise, it fails. (b) Suffix-stripping Approach: It does not depend on any lookup table that consists of derived or inflected words and their root word relations. It simply uses a set "rules" which drives the algorithm. It finds the root/stem of the given input word based on that rules. (c) Lemmatization algorithms: This technique also called as text normalization. In Lemmatization root word is called Lemma. The POS is first identified of that language and then an attempt will be made to find the stem. The stemming rules will change based on a word's POS of that language. Lemmatization process ensures that the root or stem of the inflected words belongs to that language (d) Stochastic algorithms: It is based on probability method to detect the root form of a word. This trained on a table of root words to inflected words relations to develop a probabilistic model. (d) Affix Removal Approach: The name clearly suggests this approach is related to removing the suffixes or prefixes of a word. An Affix may be a prefix or a suffix. It comes under truncating method of the stemming algorithm. We found affixes are connected with nouns in Odia language. One can opt any of the above technique while designing stemmer. (e) Hybrid approach: This technique combines more than two methods as discussed above. It may merge the rule-based technique along with the probability method. (f) N-Gram Modeling: Many stemming methods used in the n-gram technique of a word to select the correct stem for a word.

Stemming plays a vital role to handle the vocabulary mismatch problem of an IR system. In this said problem, the query words mismatch with the document words. For example, when a user input a query word and the word does not exist in the vocabulary of the document then it may cause unreliable result. To avoid this problem, we have developed a new web-based hybrid stemmer using brute force with enhanced suffix stripping algorithm union that can be adopt in the Odia information retrieval system. The new stemmer is both computationally faster as well as domain-independent.

2. RELATED WORKS

In the study of information retrieval, researchers find stemming plays an important role. Stemming is not a new concept. Stemming techniques had invented since 1968. The first stemming algorithm was designed by Julie Beth Lovins [1]. After that many researchers continued investigating various approaches to this area of study and proposed several algorithms to improve its performance. Another stemmer in English was written by Martin Porter [2] in the year 1980. As compared to European languages as well as English, a few researches have been discovered in Indian Language. A Hindi stemmer [3] was proposed by Rao, Durgesh et al. based on suffix striping approach. A Bengali Morphological analyzer [4] was developed by Dasgupta et al. based on suffix striping approach. Stemming is the process by which the user inputs an inflected word to the trained model and the model produces the root/stem word according to its rule set. In this Paper we have developed a Stemmer based on Hybrid Approach.

3. LITERATURE SURVEY ON ODIA STEMMER

These are the few papers published on Odia Stemmer. Table 1 describes the paper details with key findings.

Table 1. Literature survey on Odia stemmer

Reference	Key Findings
Sampa et al. [5]	Published a paper on Stemmer for Odia language. They used the suffix stripping approach to remove the inflectional suffixes. The limitation of this algorithm was it only predicts 88% accuracy.
Balbantray, R.C. et al. [6]	Published a paper FIRE 2012 Submission: MET Track Odia. They had used the affix removal algorithm. The system reads input text files from the folder. Firstly, it removes stop words from the input files against the stop word dictionary then matched the token with the root word dictionary. After that the input matched the suffix dictionary then removes the suffix and match with the root word. If the root word found then there is no further processing required.
Balabantaray, R.C. et al. [7]	Presented a paper on Odia Text Summarization.
Sethi, Dhabal Prasad [8]	Published a paper on Lightweight Stemmer for Odia Derivational Suffix. He used suffix stripping method to find the stems.

4. ODIA DERIVATIONAL MORPHOLOGY

4.1. Odia morphology

The formal variants of a morpheme are called allomorphs of that morpheme. The variant may be phonologically or morphologically conditioned. A morpheme may be a free or a bound form. Alternatively, we can say that a word consists of one or more than one morpheme. From the point of view of its internal structure, a word may consist of (i) a root morpheme only (ii) a root and one or more non root morpheme or (iii) more than one root morpheme. The non-root morphemes are bound forms and are generally referred to as affixes. Roots enter into further morphological constructions and form a base while non-roots do not [9].

4.2. Word formation

Word formation is concerned with those words which comprise more than one meaningful component called morphemes. The common morphological processes, which are involved in word formation, are inflection, derivation, reduplication, echo formation and contraction. The word formation process is shown in Figure 1.

4.2.1. Inflection

Inflection is a morphological process by which words are formed with the help of bound forms, which are called inflectional affixes. Inflected words belong to the same form-class to which the root word belongs.

4.2.2. Derivation

Derivation is a morphological process, which is concerned with the structure of the stems. In other words, word stems are formed by derivation. Two types of this process are generally distinguished and they are compounding and derivation. Compounding is a derivational process in which a stem is formed with two roots, the resultant stem belonging to the form class of at least one of the constituent roots. Derivation is a process of word formation in which a stem is formed with two roots or a root and an affix and the resultant stem does not belong to the form class of any of the constituents. Both inflectional and derivational affixes are involved in affixation. Depending on their position of occurrence with respect to the root, the affixes are classified into prefixes, suffixes and infixes. Prefixes precede the root, suffixes follow it and infixes occur within the root.

4.2.3. Reduplication

Laurel J. Brinton in his structure of English: A Linguistic Introduction defines “Reduplication is a process similar to derivation, in which the initial syllable or the entire word is doubled, exactly or with a slight morphological change.” Reduplication is another morphological process in which a part of a root or the root itself is added to the root. This type of word formations is popular in Odia language.

4.2.3. Echo formation

The partial repetition of a phoneme or syllable of the base may be called an echo-formation. In other words, if the initial phoneme/syllable of the base is replaced by another phoneme or syllable it has neither any individual occurrence nor any meaning of its own. It may be called as echo-formation.

4.2.4. Contraction

Contraction is a process of word formation in which a syllable is dropped from the root. In Odia words are formed using different morphological process viz., inflection, compounding derivation, affixation, reduplication and contraction. Both prefixes and suffixes occur in Odia. The prefixes are used to form derived adjectives, verbal noun, agent noun, collective and reciprocals. The suffixes denote gender, number, case, tense, aspect, and mood.

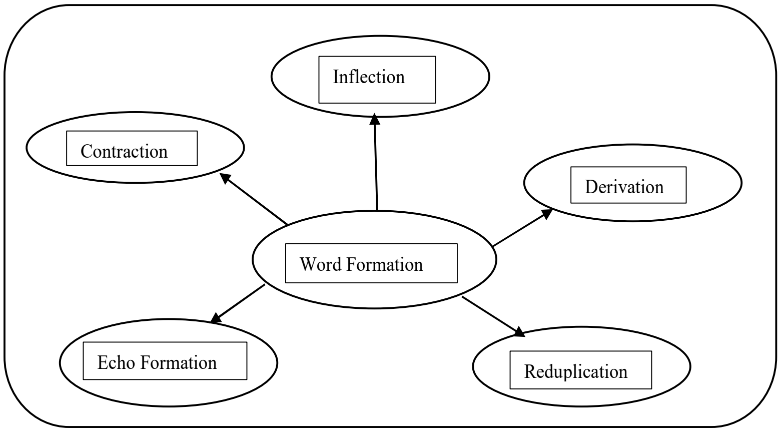


Figure 1. Word formation process

The formal variants of a morpheme are called allomorphs of that morpheme. The variant may be phonologically or morphologically conditioned. A morpheme may be a free or a bound form. Alternatively, we can say that a word consists of one or more than one morpheme. From the point of view of its internal structure, a word may consist of (i) a root morpheme only; (ii) a root and one or more non root morpheme or; (iii) more than one root morpheme. The non-root morphemes are bound forms and are generally referred to as affixes. Roots enter into further morphological constructions and form a base while non-roots do not [9]. Odia morphology deals with the analysis, identification and description of structure of morpheme. Morphology deals with the structure of words. The basic unit is the focus of study in morphology is morpheme. For example: The word ବାଳକମାନେ the morphemes are ବାଳକ, ମାନେ. Morpheme is not always conveying a meaningful word in Odia. Any morpheme in Odia should be a root word, prefix or suffix. Morphemes are divided into five categories shown in Figure 2.

The morpheme which are independent called free morpheme. Those morphemes are standalone in nature. It does not need to add with other to create a word. Examples of free and bound morpheme:

ରାମ ଭାତ ଖାଉଛି । ରାମ ଭାତ(କୁ) ଖାଉଛି|

The morpheme ଭାତ is a stand-alone morpheme and morpheme (କୁ) is a suffix. Most of the morphemes are bound type in Odia language.

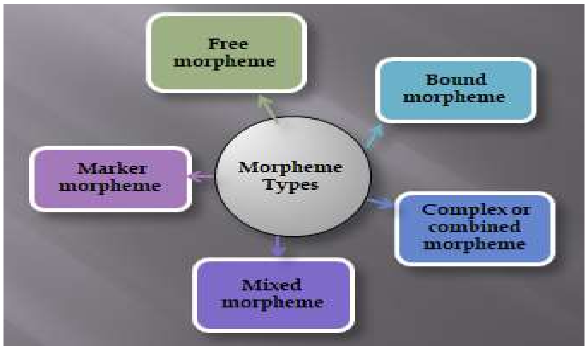


Figure 2. Type of morpheme

4.3. Odia derivational morphology

Derivational morphology deals with the addition of derivational suffixes with word stem to form word of different class (different part-of-speech). Like English, Odia derivational suffixes are added with root word to form different part-of-speech. They are in Table 2.

Table 2. Detail description of Odia derivational morphology

Categories	Examples
Noun to Adjective	Noun word + Derivational suffix = Adjective category ରୁପ+ଏଲି = ରୁପେଲି
Adjective to Noun	Adjective word + derivational suffix = noun words ଆଧୁନିକ + ତା=ଆଧୁନିକତା
Adjective to Adjective	Adjective word + Suffix = Adjective word ସରୁ+ଆ=ସରୁଆ
Verb to Adjective	Verbal word + Derivational suffix =Adjective word ପାନ+ଇଅ= ପାନିଅ
Verb to Noun	Verbal word + Derivational suffix = Noun word ବିଞ୍ଚ+ଅଣା= ବିଞ୍ଚଣା ଖେଳ+ଆଳି=ଖେଳାଳି

In derivational stemming, words that are derived, either by adding affixes to that stems or by performing changes at the morpheme boundary, are reduced to their stem form. Odia language has strong inflectional system can be classified as nominal inflection and verb inflection. Here we represent the rules using Panini Grammar. For example, 'କଲମଗୁଡ଼ିକ' here stem କଲମ and suffix is ଗୁଡ଼ିକ. The details of nominal suffix and Verbal suffix are given in Table 3 and Table 4 [5]. In Odia we find some prefixes which is attached only on noun. There is 20 such type of prefixes in Odia. These are basically from Sanskrit. They are as shown in Table 5.

Table 3. List of nominal suffixes in Odia

ବିଭକ୍ତି **Inflection**	ଏକ ବଚନ **(Singular)**	ବହୁ ବଚନ **(Plural)**	କାରକ **(Case-Relationship)**	ଉପପଦ **(Noncase-Relationship)**
ପ୍ରଥମା (1st Inflection)	ଏଜ,ଟେ,ଟି,ଟା,ଟିଏ,ଟାଏ,◌୍ଞ	ଏଗୁଡ଼ିଏ,ଗୁଡ଼ାକ,ଗୁଡ଼ିକ,ମାନ ,ମାନେ ,	କର୍ତା	
ଦ୍ୱିତୀୟା (2nd Inflection)	କୁ,ତେ,ଟିକୁ,ଟାକୁ,ଙ୍କୁ,କି,ଙ୍କୁ, ଠାକୁଠିକି,ଠିକୁ,	ଙ୍କୁଗୁଡ଼ାକୁ,ଗୁଡ଼ିକୁ ,ମାନଙ୍କୁ ,	କର୍ମ	
ତୃତୀୟା (3rd Inflection)	ରେଙ୍କଦ୍ୱାରା,ଦେଇ,ଦ୍ୱାରା,	ଙ୍କଦ୍ୱାରା ,ମାନଙ୍କ ଦେଇ ,ମାନଙ୍କ ଦ୍ୱାରା , ମାନଙ୍କରେ	କରଣ	
ଚତୁର୍ଥୀ (4th Inflection)	କୁ,ଙ୍କ,ଲାଗି,ପାଇଁ,ଙ୍କି,କି,ଙ୍କୁ, ନିମନ୍ତେ ,ଙ୍କ ଯୋଗୁଁ ,ଙ୍କ ନିମନ୍ତେ ,ଙ୍କ ଲାଗି, ଙ୍କ ସକାଶେ ,ସକାଶେ	ଙ୍କୁ ,ମାନଙ୍କ ଲାଗି ,ଗୁଡ଼ାକୁ,ଗୁଡ଼ିକୁ ,ମାନଙ୍କୁ , ନିମନ୍ ,ମାନଙ୍କନିମନ୍ତେତେଯୋଗୁଁମାନଙ୍କ , ସକାଶେ	ସମ୍ପ୍ରଦାନ	
ପଞ୍ଚମୀ (5th Inflection)	ରୁଠଉଁ ,ଠୁଁ ,ଠାରୁ ,	ମାନଙ୍କଠାରୁ,ମାନଙ୍କଠୁଁ , ଙ୍କଠାରୁମାନଙ୍କ ଠୁ ,ମାନଙ୍କ ଠଉଁ ,	ଅପାଦାନ	
ଷଷ୍ଠୀ (6th Inflection)	ର,ଠି,ଠାଇଁ ,ଏ ,ଙ୍କର ,ଙ୍କ , ଠେଇଁଠିଁ ,ତହିଁ,	ମାନଙ୍କଙ୍କରି ,ମାନଙ୍କରି ,ମାନଙ୍କର ,		ସମ୍ବନ୍ଧ
ସପ୍ତମୀ (7th Inflection)	ରେ,ଠେଇଁ,ଠି,ଠାଇଁ,ଏ,ଠାରେ, ତହିଁଠିଁ ,	ମାନଙ୍କରେ ,ମାନଙ୍କ ଠାଇଁ ,ମାନଙ୍କଠାରେ, ମାନଙ୍କ ଠେଇଁ ,ମାନଙ୍କ ଠିଁ	ଅଧିକରଣ	

Table 4. Odia verbal suffix

କାଳ (Tense)	ପୁରୁଷ (Person)	ଏକ ବଚନ (Singular Suffix)	ବହୁ ବଚନ (Plural Suffix)
ବର୍ତ୍ତମାନ କାଳ (Present Tense)	ପ୍ରଥମ ପୁରୁଷ	ଉଅଛି,ଉଛି,ଉଥାନ୍ତି,ଏ	ଉଛୁ, ଇଅଛୁ, ଉଥାନ୍ତୁ, ଉ
	ଦ୍ୱିତୀୟ ପୁରୁଷ	ଉଅଛୁ, ଉଛୁ, ଉଥାନ୍ତୁ, ର	ଉଅଛ, ଉଛ, ଉଥାନ୍ତ
	ତୃତୀୟ ପୁରୁଷ	ଉଅଛି, ଉଛି, ଉଅଛନ୍ତି, ଉଥାନ୍ତା, ଉଥାନ୍ତେ, ଉଥାଆନ୍ତେ, ନ୍ତି, ଉଆନ୍ତା	ଉଛନ୍ତି, ଉଅଛନ୍ତି, ଉଥାଆନ୍ତି, ନ୍ତି
ଅତୀତ କାଳ (Past Tense)	ପ୍ରଥମ ପୁରୁଷ	ଇଲି,ଇଛି, ଇଥିଲି, ଥିଲି,ଇଥାନ୍ତି,ଇଅଛି	ଗଲୁ, ଇଥିଲୁ,ଉଥିଲୁ, ଇଥାନ୍ତୁ
	ଦ୍ୱିତୀୟ ପୁରୁଷ	ଇଲୁ,ଇଛୁ, ଇଥିଲୁ, ଉଥିଲୁ, ଇଥାନ୍ତୁ	ଇଲ, ଇଛ, ଇଅଛ, ଇଥିଲ, ଉଥିଲ, ଇଥାନ୍ତ
	ତୃତୀୟ ପୁରୁଷ	ଇଲା, ଇଲେ, ଇଛି, ଇଛନ୍ତି, ଇଥିଲେ, ଇଥିଲଲ, ଉଥିଲେ, ଇଥାଆନ୍ତା, ଇଥାଆନ୍ତେ	ଇଲେ, ଇଛନ୍ତି, ଇଥିଲେ, ରଥିଲେ, ଇଥାଆନ୍ତେ
ଭବିଷ୍ୟତ କାଳ (Future Tense)	ପ୍ରଥମ ପୁରୁଷ	ଇବି, ଇଥିବି, ଥାନ୍ତି, ଉଥିବି	ଇବୁ, ଇଥିବୁ, ଥାଆନ୍ତୁ, ଉଥିବୁ, ଇବା
	ଦ୍ୱିତୀୟ ପୁରୁଷ	ଇବୁ, ଇଥିବୁ, ଥାଆନ୍ତୁ, ଉଥିବୁ	ଇବ, ଇଥିବ, ଥାଆନ୍ତୁ, ଉଥିବ
	ତୃତୀୟ ପୁରୁଷ	ଇବ, ଇବେ, ଇଥିବେ, ଇଥିବ, ଥାଆନ୍ତ, ଥାଆନ୍ତେ, ଉଥିବ, ଉଥିବେ	ଇବେ, ଇଥିବେ, ଥାଆନ୍ତେ, ଇଥିବେ

Table 5. List of Odia prefixes

ପ୍ର	ପରା	ଅପ	ସମ୍	ନ
ଅଧି	ସୁ	ନିର୍	ଉତ୍	ପରି
ପ୍ରତି	ଅବ	ଅନୁ	ଦୁର୍	ବି
ଅଭି	ଅତି	ଅପି	ଉପ	ଆ

5. PROPOSED METHODOLOGY FOR ODIA STEMMER

We have proposed a new web-based stemmer based on hybrid approach (i.e. combination of brute force and suffix removal approach) [10] for Odia language. The proposed stemmer is both computationally inexpensive and domain independent. The algorithm of the proposed stemmer is described in Figure 3.

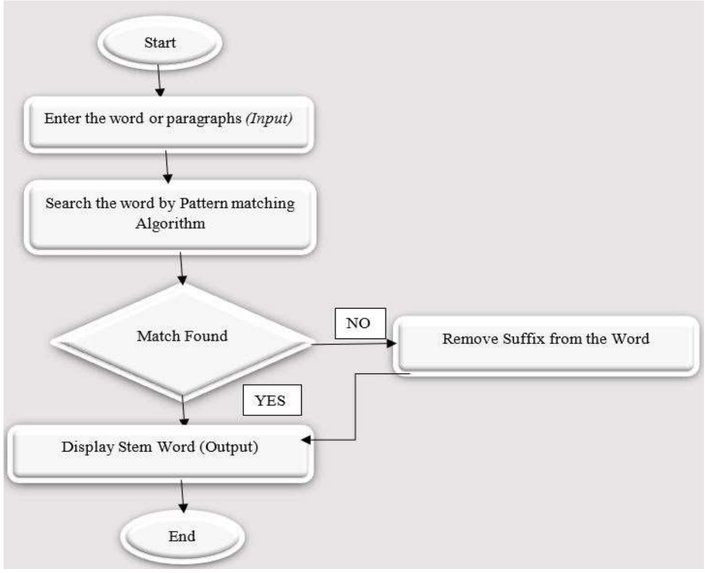

Start
Enter the word or paragraphs (Input)
Search the word by Pattern matching Algorithm
Match Found
NO
Remove Suffix from the Word
YES
Display Stem Word (Output)
End

Figure 3. Flowchart of the algorithm

Brute force search is called as exhaustive search. It searches all the possible solution from the data. Here it searches the root words present in the database. This technique uses a lookup table which contains inflected words and root words mapping. This technique [11] we create and store maximum possible inflected words along with their corresponding root word in a database table. When we give input to the system then brute force search is carry out and it inspects that whether the derivational words exist in the database. If the word is present in that table then it will give its corresponding stem or root word. If the word is not present in the table then it will go for suffix removal method to handle those words. Suffix removal is a rule-based approach in that certain rule set is defined. By applying those rule set suffixes are removed from the inflected or derived word, to find the stem/root. The new enhanced approach of suffix stripping algorithm. Figure 4 shows the stemmer user interface.

Start
Step 1: Enter derivational word that to be stemmed
Step 2: The system removes the 3 characters suffixes, 2 character suffixes and 1 character suffixes from the derivational word if word length greater than three, and two respectively recursively.
End

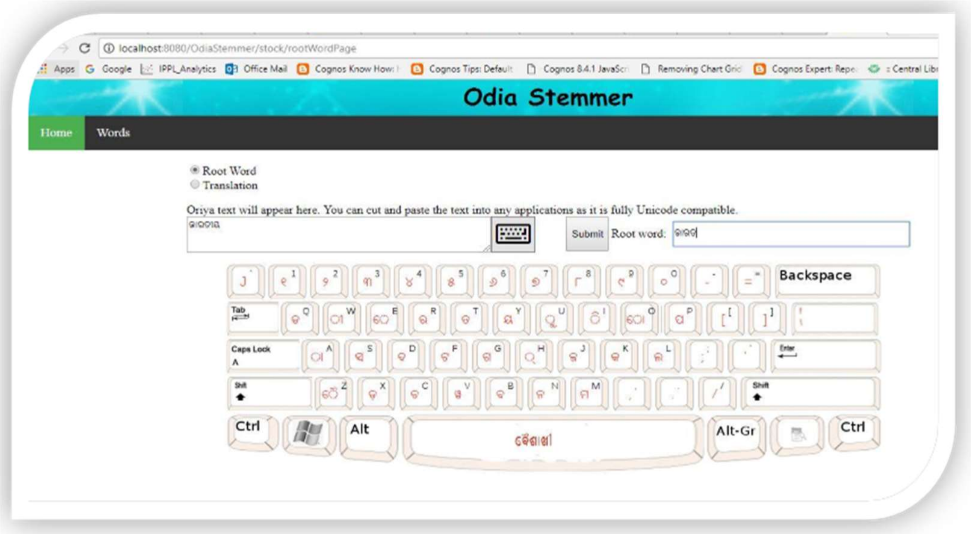
The inflected word is processed by the stemmer in three steps. The steps are shown below.

a. Input: The inflected Odia word/paragraph is entered as an input to the web-based system. Here "ଆଧୁନିକାତା" is given as an Input word.
b. Processing: Derivational/inflected word is searched by brute force method. It matches with the user searched word with the words exist in the database table. If the matching word is exist in the database then it will provide the stem of the word as output. If mismatch found then it searches for the alternate method called suffix stripping method i.e. the algorithm removes the suffixes recursively first 3 characters, then 2 characters and last 1 character with a condition that the inflected word must be greater than the suffix to find the stem/root of the word.
c. Output Unit: In Output Unit, the result comes after the processing of word. The result after processing is "ଆଧୁନିକ."

One Character derivational Suffix: ଅ, ଆ, ଇ, ଈ, ତ, ତି, ଯ, ତ୍ର, ତ୍ରୁ, ନ, ତା, ତ୍ୱ, ଥା, ଦା, ଧା, କା, ରା

Two Character derivational Suffix: ଅନ, ମୟ, ଅକ, ତର, ଉକ, ତମ, ଇନ, ଇଷ୍ଟ, ଇଷ୍ଣୁ, ଆଳୁ, ତବ୍ୟ, ଆଳ, ଇତ୍ର, ଆଳି, ଉର, ଆରି, ସନ, ଇଆ, ଅଣ, ଉଆ, ଅଣା, ଉରା, ଅନି, ଏଲି, ଅନ୍ତା, କାର, ଅନ୍ତୀ, କୁଳା, ଆଣ, ବନ୍ତ, ଅଉ, ଚିଆ, ଉଆ, ପଣ, ଉଣି, ଖାନା, ଉଣା, ଖୋର, ଏଣି, ଗର, ଇକ, ଗିରି, ଈୟ, ଆମି, ଈନ, ବାଜ, ଏୟ, ବାଲା, ଇତ, ଦାର, ଇଳ, ବିନ, ଇନ, ତନ, ବତ.

Three Character derivational Suffix: ଅନୀୟ, ଉଆଳ, ଉଆଳି, ଶାଳିନ, ଈୟସ, ସ୍ଥାନୀୟ, ଆଳିଆ.



localhost:8080/OdiaStemmer/stock/rootWordPage

Odia Stemmer

● Root Word
○ Translation

Oriya text will appear here. You can cut and paste the text into any applications as it is fully Unicode compatible.

ରାଜାର

Root word: ରାଜା

Figure 4. Stemmer user interface

6. EVALUATION

We have evaluated the stemmer by taking different set of words i.e. 100 words, 200 words, 300 words and so on to calculate the time taken to extract the root words as shown in Figure 5. We have not compared our Odia stemmer with any of the existing stemmer available for Odia language. Nowhere had we found the existing result to compare with the proposed stemmer. Table 6 shows the time to extract Odia root words.

Table 6. Time taken to extract Odia root words

Set No	No of Words	Time Taken (Sec.)
Set-1	100	5.07
Set-2	200	8.92
Set-3	300	13.27
Set-4	400	17.18
Set-5	500	20.29
Set-6	600	24.84
Set-7	700	29.11
Set-8	800	32.85
Set-9	900	37.66
Set-10	1000	40.48

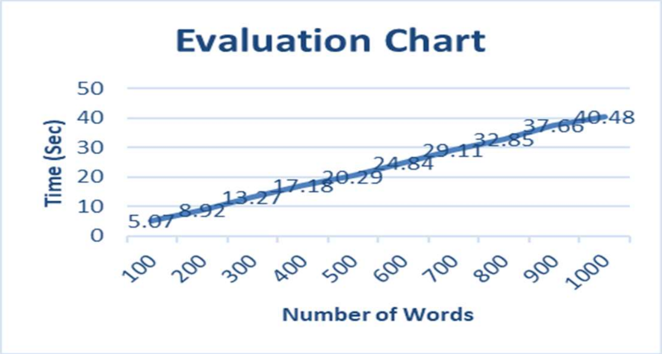


Figure 5. Evaluation graph

7. APPLICATION OF STEMMER

7.1. Information retrieval

Stemmer can be used in information retrieval [12] to reduce as many related words to a common form which is not in base form.

7.2. Indexing

Stemmer can ultimately reduce the indexing size [13] of the documents and thus the retrieval process become faster.

7.3. Auto text summarization

It reduces a text document to its summary. It can be used for text summarization [14].

7.4. Cross-Language Information Retrieval (CLIR)

Stemmer can be used in cross-language information retrieval [14] to reduce as many related words to a common form which is not in base form. Example suppose the user enter the query in English, it retrieves relevant document written in Odia.

8. CONCLUSION

This stemmer can be played a vital role for the performance of an Odia IR System. It can efficiently handle the problem of understemming and over stemming. In future we can extend this research by merging few other techniques by including some more data in to the database and also by using extra rules for suffix stripping approach. Then we can compare the results with this stemmer result. In this way we can conclude which merging technique is computationally faster. Accordingly, we pick the approach for building an Odia IR system. In this paper we designed a stemmer algorithm for Odia which removes derivational suffixes from derived word. This algorithm uses brute force approach and a new enhanced approach of simple suffix removal technique.

REFERENCES

[1] J.B. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics*, Vol. 11, No. 1 and 2, pp. 22-31, 1968.

[2] M.F. Porter, "An algorithm for suffix stripping," *Program*, Vol. 14, No. 3, pp. 130-137, 1980.

[3] A. Ramanathan and Durgesh D. Rao, "A lightweight stemmer for Hindi," 2003.

[4] S. Dasgupta and V. Ng, "Unsupervised morphological parsing of Bengali," *Lang. Resources & Evaluation*, Vol. 40, pp. 311-330, 2006.

[5] S. Chaupattnaik, S.S. Nanda, and S. Mohanty, "A suffix stripping Algorithm for Odia Stemmer," *Int. Journal of Computational Linguistics and Natural Language Processing*, Vol. 1, No. 1, 2012.

[6] R.C. Balabantaray, B. Sahoo, M. Swain, and D.K. Sahoo, "IIIT-BH FIRE 2012 Submission: MET Track Odia," 2012.

[7] R.C. Balabantaray, B. Sahoo, M. Swain, and D.K. Sahoo, "Odia Text Summarization using Stemmer," *Int. Journal of Applied Information Systems (IJAIS)*, Vol. 1, No. 3, 2012.

[8] D.P. Sethi, "Design of Lightweight Stemmer for Odia Derivational Suffixes," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, No. 12, 2013.

[9] Odia grammar book of class 9th in BSE, ODISHA.

[10] A. Mohammad, S. Oqeili, and A. A. Rawan, "Occurrences Algorithm for String Searching Based on Brute-force Algorithm," *Jordan Journal of Computer Science*, Vol. 2, No. 1, pp 82-85, 2006.

[11] D.P. Sethi, "Morphological Analyzer for Sambalpuri Odia Dialect Inflected Verbal Forms," *International Journal of Advanced Reseach in Computer Science and Sofiware Enginnering*, Vol. 3, No. 10, 2013.

[12] M. Erritali, "Information Retrieval: Textual Indexing Using an Oriented Object Database," *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 2, No. 1, pp.205-214, April 2016.

[13] I Gusti Ayu Triwayuni, I Ketut Gede Darma Putra, and I Putu Agus Eka Pratama, "Content Based Image Retrieval Using Lacunarity and Color Moments of Skin Diseases," *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 9, No. 1, pp. 243-248, January 2018.

[14] P. Bajpai, P. Verma, Q. Abbas Syed, "Two Level Disambiguation Model for Query Translation," *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 8, No. 5, pp. 3923-3932, October 2018.