ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

Design of Lightweight Stemmer for Odia Derivational Suffixes

Dhabal Prasad Sethi

Lecturer in Computer Science& Engineering

Government College of Engineering, Keonjhar, Odisha, India

Abstarct: Stemming plays an important role in morphological analyzer, information retrieval system and others. Stemming is the process of reducing all forms of word to its base form or stem. Derivational stemming is the process of removing the derivational suffixes from its derived word and get the stem/root word. Using stemmer the information retrieval system becomes faster. In this article, I designed the lightweight stemmer algorithm for Odia derivational suffixes and later that algorithm compared with simple suffix stripping algorithm.

Keywords: Derivational Suffixes, Stemmer, Lightweight, Krudanta, Tadhita, Information Retrieval

I.INTRODUCTION

Stemming is a process that conflates morphological forms into a single common form called stems or roots without doing complete morphological analysis. The term conflation means mapping variants of a word to a single term or stem. The most important tool used in information retrieval system (IRS) and morphological analyzer is a stemmer who reduces a word to its stem form to improve the system performance.

In Information Retrieval (IR), it doesn't mean stem generated are genuine words or not. The word "computation" might be stemmed to "comput" rather than "compute". "Comput" is the stem form of a word where "compute" is the lemmatize form of word. An algorithm which converts a word form to its linguistically correct root is called lemmatize.

Stemming algorithm can be classified into two categories 1) stem-based2) root-based. In stem-based algorithm removes prefixes and suffixes from words where root-based algorithms reduce stems to root. The researches named Al-Jlayl and Frieder [5] shown that stem-based retrieval is more effective than root-based. In root-based algorithm many surface word variants don't have similar semantic interpretation. These surface words are different in meaning; they originate from the same root. So it increases the word ambiguity.

In IR System, when the user enters the query word "fishing" as input, he actually wants to retrieve documents containing the related term "fisher" and" fished". Thus using stemmer the system improves the recall rate i.e. the number of documents retrieved in response to a query. It also decreases the size of index file on IRS (Information Retrieval System), since many related terms are mapped to one. The applications of stemmer are machine translation, document summarization and text classification.

The rest part of this paper is organized as: Section II describes its related work, Section III describes the Odia derivational morphology, Section IV describes as Odia lightweight stemmer algorithm, Section V describes as result and experiment, Section VI describes as conclusion and future work.

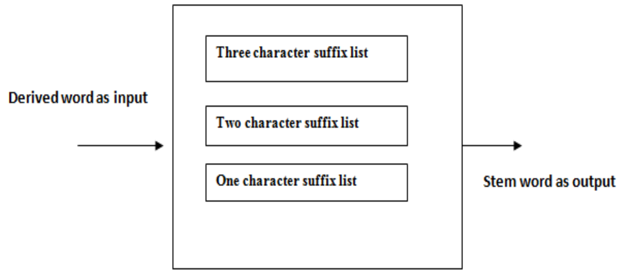

Derived word as input
Three character suffix list
Two character suffix list
One character suffix list
Stem word as output

Fig1: Shows Block Diagram of lightweight stemmer for Odia)

II.RELATED WORK

First the stemmer algorithm is developed by Lovins in 1968 for English language. His algorithm is based on dictionary based and the suffix list, root words are stored in dictionary .when word is checked for suffix removal, right-hand end of word is matched with suffix dictionary, if found then removed the suffix and then matched with root dictionary. The Lovins algorithm contains 294suffixes and 19 context-sensitive rules to remove the suffixes. Then in 1980 porter stemmer developed suffix stripping algorithm for stemmer. That algorithm gives better result than Lovins because minimal rules are applied e.g. 5 steps are applied for word transformation, approximately 60rules are used for programmer. That onwards all the foreign language and Indian languages used the same technique for their language and get better result for stemming. In India not much work has done about stemmer. But now in India stemmer for Gujarati, Bengali, Hindi, Tamil, Odia have developed. Among the Indian stemmer algorithm are implemented in inflectional or derivational words. Rare works have done both the stemmer for inflectional and derivational types.

For Indian languages: Juhi Ameta, Nisheeth Josi, Iti Mathur [4] presented a paper named "A light weight stemmer for Gujarati". They have shown the creation of rules for stemming and morphology that Gujarati possess.

Md. Zahurul Islam, Md. Nizam Uddin and Mumit khan [2] presented a paper named" A lightweight stemmer for Bengali and its use in spelling checker". They have used the suffix stripping algorithm according to the longest suffix. Ananthakrishnan Ramanathan, Durgesh D Rao [1] presented a paper named "A light weight stemmer for Hindi". They have removed the suffix from each word by longest possible suffix.

Kartik Subha.Dipti Jiangani, Pushpak Bhattacharyya [8] presented a paper named Hybrid inflectional stemmer and Rule-based derivational stemmer for Guajarati. They have used the rule-based technique for derivational suffixes and POS based and suffix-stripping algorithm for inflectional suffixes.

For Odia languages: Sampa Chaupatnaik, Sohang Sunder Nanda, Sanghamitra Mohanty [9] presented a paper named" A suffix stripping Algorithm for Odia stemmer". They have used the suffix stripping algorithm to remove the inflectional suffixes from noun, and verb.

R.C Balbantray, B Sahoo, M.Swain, D.K Sahoo [10] presented" IIIT-Bh FIRE submission: MET Track Odia". They have used the affix removal technique.

III. ODIA DERIVATIONAL MORPHOLOGY

a) Odia Morphology:

Odia morphology deals with the analysis, identification and description of structure of morpheme. In Odia, morphemes are called Rupeme (ରୂପିମ) . Morpheme (Rupeme) is the smallest component/unit of Odia language which carries and conveys a unique meaning and is appropriated by grammar. For example the word BALAKAMANE the morphemes are BALAKA, MANE. Morpheme is not necessary to form a meaningful word in Odia. Every morpheme is a base/root word, prefix or suffix. Morphemes are classified into five types:1)free morpheme,2)bound morpheme,3)complex or combined morpheme,4)mixed morpheme,5)marker morpheme. Here I have explained only free and bound morpheme. That morpheme which are independent called free morpheme. These morphemes are stand alone without other. It does not need to add with other to create a word.

Example

ଗୋପାଳ ଭାତ ଖାଉଛି e.g. ଗୋପାଳ ଭାତ(କୁ) ଖାଉଛି

Here the morpheme BHATA can stand alone without morpheme (KU).

Those morphemes added with another morpheme and give meaningful word are called bound morpheme. In Odia languages most of morphemes are bound type.

b) Study of Odia Derivational (Tadhita and Krudanta) Words and Suffixes (Prataya)

Those Odia words are derived from Sanskrit verbal root with addition of suffixes and are used in Odia language; these words are called"TATSAMA KRUDANTA" word. Example Darshana is derived form drush dhatu; Patha ପାଠ is derived from path ପାଠ୍ dhatu. Those Odia words are derived from Odia verbal root and the Odia verbal root are derived from Sanskrit verbal root, these Odia words are called TATABHABA KRUDANTA WORD. Example KANDANA କାନ୍ଦଣା is derived from Odia dhatu KANDA କାନ୍ଦ which is derived from Sanskrit KRANDA କ୍ରନ୍ଦ dhatu. The Odia words which are derived from Odia verbal root with addition of suffixes that words are called DESAJA KRUDANTA WORD. Example the Odia word KHASANDA ଖସଡା word is derived from Odia dhatu KHASA ଖସ୍ with addition of suffix ADA ଅଡା .Those suffixes are added with root word to create new word, these suffixes are called TADHITA SUFFIXES ତଦ୍ଧିତ ପ୍ରତ୍ୟୟ and newly generated words (which are different part of speech) are called TADHITA WORD ତଦ୍ଧିତ ଶବ୍ଦ . The tadhita word KULINA କୁଳୀନ is derived from root word KULA କୁଳ with addition of suffix (INA) ଈନ , KOULIKA କୌଳିକ word is derived from root word KULA କୁଳ with addition of IKA ଇକ

c) 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

1) Noun to Adjective ବିଶେଷ୍ୟରୁ ବିଶେଷଣ

Noun to adjective means when the derivational suffix added with noun word it changes to adjective category. It changes its part of speech. Here is the example

RUPA+ELI=PUPELI

ରୂପା+ଏଲି=ରୂପେଲି

2) Adjective to Noun ବିଶେଷଣରୁ ବିଶେଷ୍ୟ

Adjective to noun means derivational suffixes added with adjective word to form noun word (which is the derived form).

ADHUNIKA+TA=ADHUNIKATA

ଆଧୁନିକ+ତା=ଆଧୁନିକତା

3) Adjective to Adjective ବିଶେଷଣରୁ ବିଶେଷଣ

Adjective to adjective means when some adjective words which when added with derivational suffixes it changes its part of speech e.g. adjective word+ suffix=adjective word(derived form).
SARU+A=SARUA
ସରୁ+ଆ=ସରୁଆ

4) Verb to Adjective କ୍ରିୟାରୁ ବିଶେଷଣ Verb to adjective means when derivational suffixes added with verbal root word it changes its meaning. E.g. verbal word+ suffixes=Adjective word (derived form).
PAN+IYA=PANIYA
ପାନ+ଇୟ=ପାନୀୟ

5) Verb to Noun କ୍ରିୟରୁ ବିଶେଷ୍ୟ

Verb to noun means some verbal word which when added with some derivational suffix, it changes its part-of-speech e.g. it converted verb to noun word.
BINCH+ANA=BINCHANA ବିଞ୍ଚ+ଅଣା=ବିଞ୍ଚଣା
KHEL+ALI=KHELALI ଖେଳ୍+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 [8].

IV.ODIA LIGHTWEIGHT STEMMER ALGORITHM

The simple suffix stripping algorithm for Odia derivational suffixes is:

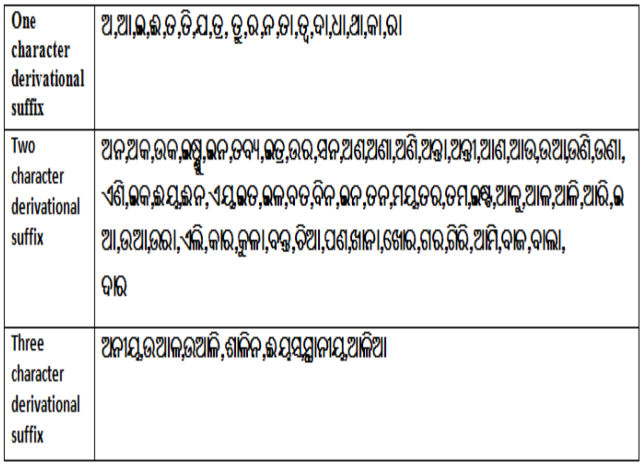
```
START
Step1: Input the derived word to be stemmed
Step2: If derived word is matched with one of suffix list, remove the suffixes
Step: Get the stem
STOP
```

Implementing the above algorithm I found many over-stemming and under-stemming errors. Over-stemming means those words are not morphological variants are conflated. Under-stemming means those words are morphological variants are not conflated. Example of over-stemming, if the user wants to remove the suffix UA ଉଆ , but the machine removes the suffix AA ଆ before removing UA ଉଆ , because in the two suffixes the character AA ଆ is matched.

To solve this over-stemming problem, I designed a prototype of Odia lightweight stemmer that removes only derivational suffixes. I identify a set of suffixes that should be removed based on the grammatical functions of suffixes. I generate three derivational lists of suffixes which consist of one character, two characters and three characters respectively. In the Odia derivational suffix list, I identify 18one character, 55 two character and 7 three character suffixes that should be removed recursively in stemming. To remove the prefixes in word, the algorithm works non-recursively. Since my work is to remove only derivational suffixes from words, it uses recursive technique. The new enhanced version of suffix stripping algorithm is

```
START
Step1: Enter the derived word to be stemmed
Step2: Recursively removes the three character suffixes in the order of presentation if word length greater than three before removing the suffixes.
Step3: Recursively remove the two character suffixes in the order of presentation if the word length greater than two before removing the suffix.
Step4: Recursively remove the one character suffixes in the order of presentation if the word length greater than two before removing the suffixes.
STOP
```

Again I explained the above new algorithm clearly. When the user enters the derived word as input, it removes longest length suffix (three characters), two character and then one character recursively and gets the stem.

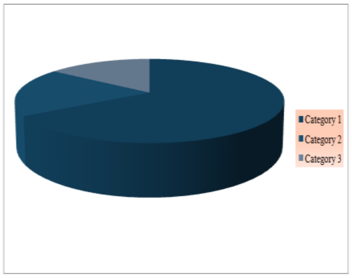


One character derivational suffix	ଅ,ଆ,ଇ,ଈ,ତ,ତି,ଯ,ତ୍ର, ତୁ,ର,ନ,ତା,ତ୍ୱ,ବା,ଧା,ଆ,କା,ରା
Two character derivational suffix	ଅନ,ଅକ,ଉକ,ଇଷ୍ଣୁ,ଇନ,ତବ୍ୟ,ଇତ୍ର,ଉର,ସନ,ଅଣ,ଅଣା,ଅଣି,ଅନ୍ତା,ଅନ୍ତୀ,ଆଣ,ଆଉ,ଉଆ,ଉଣି,ଉଣା, ଏଣି,ଇକ,ଇୟ,ଈନ,ଏୟ,ଇତ,ଇଳ,ବତ,ବିନ,ଇନ,ତନ,ମୟ,ତର,ତମ,ଇଷ୍ଠ,ଆଳୁ,ଆଳ,ଆଳି,ଆରି,ଇଆ,ଉଆ,ଉରା,ଏଲି,କାର,କୁଳା,ବନ୍ତ,ଟିଆ,ପଣ,ଖାନା,ଖୋର,ଗର,ଗିରି,ଅମି,ବାଜ,ବାଲା, ଦାର
Three character derivational suffix	ଅନୀୟ,ଉଆଳ,ଉଆଳି,ଶାଳିନ,ଇୟସ୍ଥାନୀୟ,ଆଳିଆ

[Table1: show the Odia derivational suffix list]

V.RESULT AND EXPERIMENT

I have taken 20 derived words for each suffix in testing. There are 80 derivational suffixes in Odia language approximately e.g. 1600 derived words are taken for testing out of which using simple suffix stripping algorithm predicted 1060 words (53suffixes) correct stem, 300words (15suffixes) are over-stemming, 240words (12suffixes) did not stem due to difficulty word construction. So the accuracy of stemming for derived word is 66.25%, over-stemming error percentage is 18.75% and 15%words did not stem. Using lightweight stemming algorithm the accuracy of correct stem is 85%, because this algorithm solves over-stemming problem and 15%did not stem.Fig2 show the percentage of correct stemming using simple suffix-stripping algorithm where fig3 shows the percentage of stemming using lightweight algorithm.



[Fig2:Category1 is percentage of correct stemmed e.g. 66.25%, category2 is the over-stemming error percentage e.g. 18.75% and category3 is the word that did not stem e.g. 15%.]

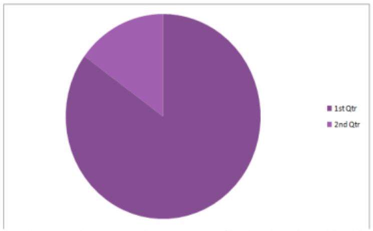


Fig3: 1st Qtr shows the percentage of correct stem e.g. 85%, 2nd Qtr shows the word that did not stem e.g15%.

VI.CONCLUSION

In this article I designed a lightweight stemmer algorithm for Odia which removes derivational suffixes from derived word. The algorithm removes the suffixes recursively first three character, then two characters and last one character which is a new enhanced approach of simple suffix removal technique. Implementing the simple suffix stripping algorithm for derived word gives the result 66.25%, whereas the proposed algorithm predicts 85%words correct stem approximately. The future work is to use this algorithm in different stemmer.

ACKNOWLEDGEMENT

I would like to thanks my school teacher named Padmanava Rout who was teaching the Odia language, has given his valuable time to prepare the article. I would like to thank all my well wishers.

REFERENCES

[1]“A lightweight stemmer for Hindi” developed by Ananthakrishan Ramanathan from national center for software technology, rain tree marg,sector7,CBD Belapur, Navi Mumbai and Durgesh D Rao from DR system,S-27,Lane1,Sector9,CBD Belapur, Navi Mumbai 400614,India

[2]”A Light weight stemmer for Bengali and its use in spelling checker” by Md . Zahurul Islam,Md. Nizam Uddin and Mumit Khan from center for research on bangle language processing, BRAC University, Dhaka, Bangladesh.

[3]”A Light Weight stemmer for Urdu language: A scarce Resourced Language” by Sajjad Ahemad khan, Waqas Anwar, Usama Ijaz Bajwa, Xuan Wang from COMSATS Institute of Information Thechnology, Abbottabad, Pakistan, Harbin Institute of Technology, Shenzhen Graduate School, P.R China.

[4]“A Lightweight stemmer for Gujarati” by Juhi Ameta, Nisheeth, Iti Mathus from Department of Computer science, Apaji Institute, Banasthali university, Rajasthan, India.

[5]“Arabic light stemmer: A New enhanced approach” by Hayder K. Al Ameed,shaikha O. Al ketbi,Amna A. Al kaabi,khadija S. Al shebli,Naila F. Al shamsi,Noura H. Al Nuaimi, Shaikha s. Al Muhari from software engineering Dept. college of Information Technology, UAE University.

[6]”Building an Arabic stemmer for information retrieval” by Aitao chan,school of information management and systems, university of California at Berkeley,CA94720-4600,USA and Fredric Gey,UC Data Archive & Technical Assistsant ,University of California at Berkeley,CA94720-5100,USA

[7]”Design and Development of stemmer for Tamil Language: Cluster Analysis” by M. Thangarasu,Dr. R.Manavalan,Department of Computer Science and Application, K.S.Rangasamy college of Arts and Science, Tiruchengode, India.

[8]”Hybrid Inflectional Stemmer and Rule-based Derivational stemmer for Gujarati” by kartik suba,dipti jiandani, Department of computer engineering, Dharmsinh Desai University, Pushpak Bhattacharyya, department of computer science and engineering, Indian Institute of Technology, Bombay

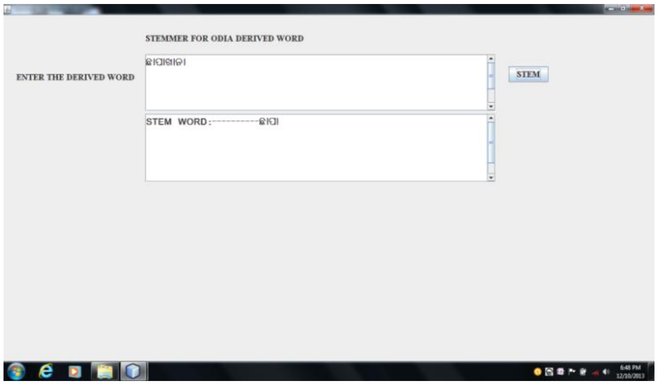
[9]”A suffix stripping Algorithm for Odia Stemmer” by Sampa Chaupattnaik, Sohang Sunder Nanda, Sanghamitra Mohanty, P.G Department of Computer Science and Application. Utkal University.

[10]”IIIT-BH FIRE 2012 Submission: MET Track Odia” by R.C Balabantaray, B.sahoo,M.Swain, D.K.Sahoo form IIIT Bhubaneswar.

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

[12]Baleswari Odia Dialect Identification using Rule Based Technique by Dhabal Prasad Sethi at International Journal of Computational Linguistic and Natural Language Processing,Aug,2013 edition.

[13]Morphological Analyzer for Sambalpuri Odia Dialect Inflected Verbal Forms by Dhabal Prasad Sethi at International Journal of Advanced Reseach in Computer Science and Sofiware Enginnering,October,2013.



ଛାପାଖାନା

STEM WORD:------------ଛାପା

(Snapshot of Odia Derivational Stemmer)

BIOGRAPHY

Dhabal Prasad Sethi is working as a lecturer in Computer Science & Engineering at Government College of Engineering, Keonjhar, Odisha .He has completed his Bachelor of Engineering in Computer Science &Engineering from BIET, Bhadrak in2006. He has completed his Master of Engineering in Computer Science& Engineering with Specialization in Knowledge Engineering from PG Department of Computer Science and Application, Utkal University, Bhubaneswar in 2011.He has more than two years teaching experience. He has presented two numbers of papers in international journals. This is his third numbers of his own credit. His research area of interest is Natural Language Processing, Information Retrieval, Data Mining and Software Engineering.