ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

Analysing Morphology of Assamese Words using Finite State Transducer

Mirzanur Rahman[1], Shikhar Kumar Sarma[2]

Assistant Professor, Department of Information Technology, Gauhati University, Guwahati, Assam, India[1]

Professor, Department of Information Technology, Gauhati University, Guwahati, Assam, India[2]

ABSTRACT: Morphology of a word is basic required information for language processing. Finite State Transducer (FST) based approach is most efficient method for developing a morphological analyzer for highly inflection language like Assamese Language. In this paper we are trying to analyze the words of Assamese language using SFST (Stuttgart Finite State Transducer) tool for generating the FST. Lists of root words are created manually along with rule list for inflectional and derivational class of Assamese words. Our Morphological Analyzer gives approximately 84.64% correct result according to our test case.

KEYWORDS: Morphological Analyzer, Finite State Transducer , Assamese Language, SFST

I. INTRODUCTION

For processing a language Morphological Analyser is the basic required application. For digitizing, a language needs to be processed with proper Morpho-Phonological rules which will help to understand or find out the other linguistic information of a language easily.

Assamese is the major dialect or official language used in the state Assam for day to day conversation and the state is situated at the north–eastern part of the country, India. Assamese Language always placed as first choice while communicating among various discourse groups of the state. The language Assamese is an Indo-Aryan language originated from Vedic dialects, however the development of the language is in not clear yet. The language as it stands today, passes through tremendous modifications in all the component viz. phonology, morphology, conjunction etc. There are two variations of Assamese language according to dialectical regions i.e. Eastern Assamese and Western Assamese Language [2]. Both are different in terms of phonology and morphology. But still the written text is same for all the regions.

Morphological information of a word is an important component of any language. In language processing technology such as Machine Translation [3], Parsing, POS Tagger, Text summarization etc requires Morphological Analysers to find out the lexical component of a word. And lexical components are the very important parts of a grammar of a language.

In our work, only standard written Assamese literary data is considered for processing the language. We have used SFST tool (SFST tool is developed by Institute of Natural Language Processing, University of Stuttgart) for implementing the analyser. SFST tool supports UTF-8 character coding which is important for the implementation of Assamese Language Processing. In this paper we have discussed about how we implement Morphological Analyser for Assamese Language.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

II. MORPHOLOGY OF A LANGUAGE

Morphology is subject that deals with the study of form and structure of words in a language. In the context of language it refers to the word structure and word formation of that language.

Example in English language:
Cats=Cat + s
Root : Cat (category- noun)
's' (indefinite plural marker)

Example in Assamese language:
মানুহজন=মানুহ + জন
Root : মানুহ (category- noun)
'জন' (Singular Suffix marker)

In Morphological Analysis a word is breakdowns to its morpheme(s). Morphemes are known as smaller units of a word with meaning. There are two categories of Morpheme; root or stem and affix. Affix provides additional meaning after adding with root morpheme. Affix may falls into different categories; Prefixes (eg: in+active), Suffixes (eg: believe +able), Infixes (Infix is not found in both English and Assamese Language, but it quite common in Philipine and Tagalog Langauge), Circumfixes (un+believe+able). Prefixes precede the stem, suffixes follow the stem, circumfixes do both, and infixes are inserted inside the stem.

Based on the affixation; there are two categories of morphology; Inflectional and Derivational morphology. After affixation, if a word falls in to its same parts-of-speech category, it's known as Inflectional Morphology. If a word falls into different parts-of-speech category after adding with an affix, known as Derivational Morphology

Example for Inflectional Morphology
Boy (noun) + s (plural marker) = Boys (noun)

Example for Derivational Morphology
Kill (Verb) + er (Suffix) = Killer (noun)

III. RELATED WORK

In most well-studied languages, Morphological inflections usually take place at the right-hand end of a word-form and this has influenced the affix stripping approaches to extract the root form a given word. The Porter Stemmer [4] , an iterative rule-based approach, has found the most success and is used widely in applications such as spell-checking and Morphological analysis. This simple approach was first developed for English and later adapted to Germanic (German, Dutch) and Romance (Italian, French, Spanish and Portuguese).

Lovins, [5] introduced a suffix dictionary to assist in finding stems of words. The right-hand end of a word is checked for the presence of any of the suffixes in the dictionary.

A probabilistic stemming approach is described by Dincer and Karaoglan [6] for Turkish. String distance-based stemming, an alternative to language-specific stemming is proposed by Snajder and Basic [7], where stems are classified using a string distance measure called the dice coefficient-based on character bigrams from a corpus.

For language-specific stemming, additional resources (like dictionary) are also often used by Krovetz [8] to group Morphologically related words. In the Indian language context, a few lemmatizer and stemmers have been reported. Among these Ramanathan and Rao [9] use a hand crafted suffix list and strip off the longest suffixes for Hindi and report 88% accuracy using a dictionary of size 35,997.

The work reported by Majumder [10] learns suffix stripping rules from a corpus and uses a clustering-based method to find equivalent categories of root words. They show that their results are comparable to Porter's and Lovin's stemmers for Bengali and French.

The work of Pandey and Siddiqui [11] focuses on heuristic rules for Hindi and report 89% accuracy. Aswani and Gaizauskas [12] propose a hybrid form of Majumder [10] and Pandey and Siddiqui [11] for Hindi and Gujarati with precisions of 78% and 83% respectively. Their approach takes both prefixes as well as suffixes into account. They use a dictionary and suffix replacement rules and claim that the approach is portable and fast.

Sharma [13] describes an unsupervised approach that learns morphology from an unannotated corpus and report 85% precision. They discuss salient issues in Assamese morphology where the presence of a large number of suffixal determiners, sandhi, samas and the propensity to use suffix sequences make more than 50% of the words used in written and spoken text inflected.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

Snigdha Paul, Nisheeth Joshi and Iti Mathur [14] has developed a lemmatizer which generates rules for removing the affixes along with the addition of rules for creating a proper root word. They use the rule based approach by creating knowledgebase which contains all the Hindi words that are commonly used in day to day life. According to them their system gave 91% of accuracy. Among the reported work on Indian languages, the result may vary widely as each author may have individual rules and corpus for the same reported language.

In [15] authors describe implementation of Hindi Morphological analyzers along with Hindi POS tagger using FST method. Their Morphological analyzer gives approximately 97% correct results and POS tagger gives an accuracy of approximately 87% for the sentences that have the words known to the trained model file, and 80% accuracy for the sentences that have the words unknown to the trained model file.

As we have considered Assamese language for our research work, which is more resource-poor language in the world, we work with a rule-based and a supervised approach and implement Morphological Analyzer with the help of SFST tool.

IV. IMPLEMENTATION

A FST based Morphological analyzer can be divided into three phases 1) Generation of lexicon file 2) Creation of rule list for inflection and derivational morphology 3) Morphological processor.

Lexicon file or root word file is generated manually by collecting word list from different sources. Following table (Table 1) shows the source list for collecting our training corpus.

1.	Collected text corpus from Language Technology Development Project, department of IT, Gauhati University
2.	Asamiya Abhidhan [16]
3.	Xobdo online Dictionary [17]

Table . 1. Sources we are using for collecting text corpus

1) Generation of lexicon file:

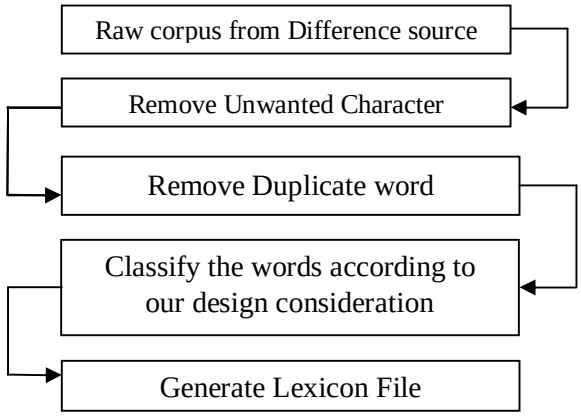

Fig.1. Generation of Lexicon file

The above figure (Fig 1) shows the different steps for generating a lexicon file. After collecting the corpus from different sources, we remove the duplicate words from the corpus; i.e the word list contains unique Assamese words. After that we manually classified the words in to different classes of Assamese Tag set. In our current work we are considering only "*Noun*", "*Pronoun*", "*Verb*", and "*Adverb*" and all other words are categorized as "*Other*".

In our manual classification, we have classified unique words according to their inflection, and derivations types. The final lexicon file total 22400 Noun word, 210 Pronoun words, 2010 Verb words and 250 Adverb is properly stored after verification by the linguist. In our work we create four lex file namely "*noun.lex*", "*pronoun.lex*", "*verb.lex*" and "*adverb.lex*". We stored one word at a line in the lex file.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

2) FST Rule:

We write rule file for our FST manually for inflected words (for both inflectional and derivational). It is not a good practice to add the entire lexicon falling under different category and the rules in a same file. So we create separate lexicon file and rule file. In the rule file we processed the entire lexicon file according to their Morphological behaviour. This file is saved with .fst extension (say "*assamese.fst*" in our case)

```
$name-reg-infl$=<Noun>:<> (\
{<DefiniteArticles>}:{খন} |\
{<DefiniteArticles>}:{খিলা} |\
{<Future-1stPerson>}:{িম} |\
{<Future-1stPerson>}:{ম} |\
{<IndefinteArticles>}:{মান} |\
{<IndefinteArticles>}:{চেৰেক} |\
{<PastPerfect-Peer2ndPerson>}:{ীলা} |\
{<PastPerfect-Peer2ndPerson>}:{ুলা} |\
{<root>}:{})

$verb-reg-infl$=<Verb>:<> (\
{<Declensor>}:{এ} |\
{<IncompleteTense>}:{য়} |\
{<IncompleteTense>}:{ই} |\
{<IndefinteArticles>}:{এই} |\
{<IndefinteArticles>}:{সৌ} |\
{<VerbEmphaticMarker>}:{ছোন} |\
{<VerbEmphaticMarker>}:{ও} |\
{<VerbEmphaticMarker>}:{লৈ} |\
{<Root>}:{} )

"pronounlist.lex" $pronoun-reg-infl$ |\
"verblist.lex" $verb-reg-infl$ |\
"namelist.lex" $name-reg-infl$
```

Fig.2. Content of assamese.fst file

The above figure (Fig 2) shows few lines of codes of "*assamese.fst*" file, where noun suffixes are stored under "*name-reg-infl*" FST variable name and verb suffixes are stored under "*verb-reg-infl*" FST variable . The last three lines of codes (fig 2) shows merging of lexicon entries with suffixes.

3) Morphological processor.

Morphological processor is the main processing unit for finding out the lexical form of an inputted surface form of a word. In our work we are using SFST tool to design Morphological processor (Fig 3 shows the block diagram). Two file is required for processing a word in SFST. Lexicon file and rule file. These two required file is already created manually and we input these two file to SFST tool.

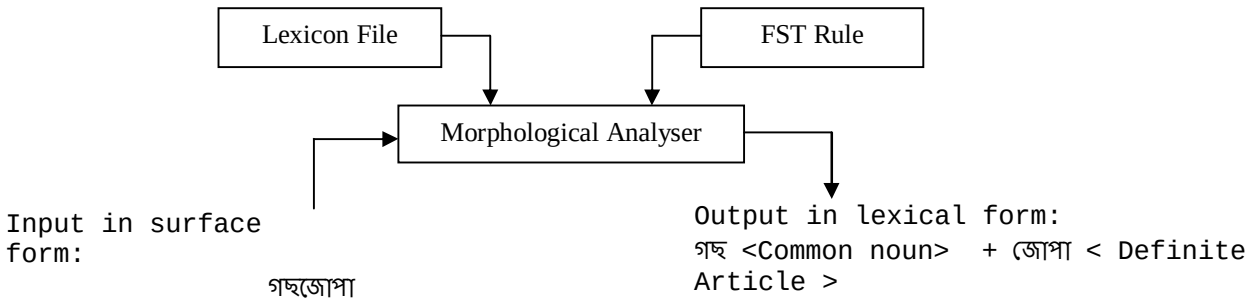

Fig. 3. Morphological Processor

- **Compilation of fst file:**
  The .fst file (*assamese.fst*) should be first compile before using it for Morphological Analysis process. For compilation process the command used

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

fst-compiler assamese.fst assamese.a

Here "*fst-compiler*" command is used to compile the rules and lexicons in to object fie; in our example "*assamese.fst*" file is compiled to "*assaemse.a*" object file. If any thing is modified in lex file or rule file, every time we need to execute above command to create a modified object file

- **Analysing input word(s):**

SFST tool provides two commands for morphological analysis "*fst-mor*" and "*fst-infl*". "*fst-mor*" command is used for analysing single word in terminal window. We are using "*fst-infl*" command, which is used to analyse batch of words.

fst-infl assamese.a write.txt > output.txt

In the above example, words to be analysed are stored in "*write.txt*" file. "*fst-infl*" command analyse the words from "*write.txt*" file one by one and finally we will get the analyse output in "*output.txt*" file. In the analyser we are providing words in its surface form and the analyser provides output in its lexicon form.

Working principle of FST:

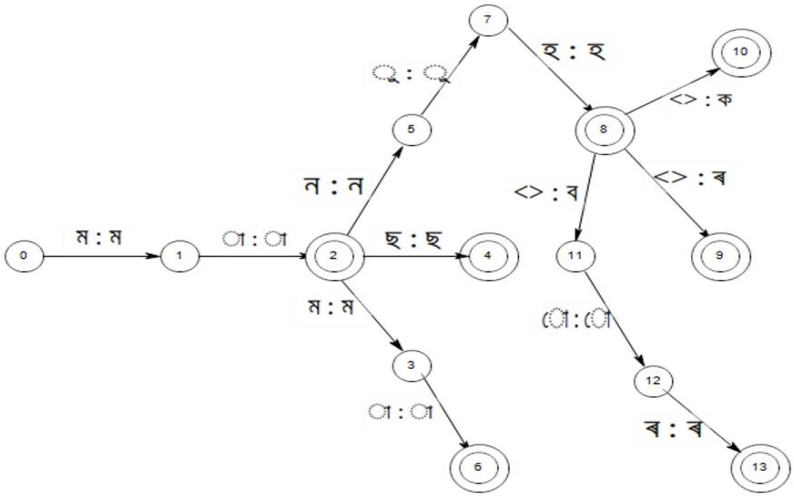

Fig.4. Diagrammatic representation of few Assamese Words

The above figure (Fig 4) shows the diagrammatic representation of Finite State Transducer of few Assamese words ("মাছ" (Maas) , "মামা" (Mama) , "মানুহ" (Manuh) , "মানুহৰ" (Manuhor) , "মানুহক" (Manuhok) and "মানুহবোৰ" (Manuhbor). Here 0 is the initial state and the set of final state include the state 2,4,6,8,9,10 and 13.

So the above FST will accept the words "মানুহৰ" as follows

মানুহৰ=ম +া + ন + ু + হ + ৰ

- At 0 state or initial state, FST will first match the first symbol of the word i.e. " ম ", since there is a transition from state 0 to state 1 for mapping of the symbol " ম " to " ম ", FST will pass the state to state 1.
- FST will try to match Assamese symbol "া" from state 1 to state 2 and pass to state 2, since there is a mapping of the symbol "া" to "া"

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

- State 2 is a final state for the Assamese word " মা " (Maa), but our FST will not take this state as final state, since our word "মানুহৰ" (Manuhor) is not completely parsed. From state 2, there are 3 transition state 2→5. 2→4, 2→3. FST will move to state 5 since next character to be recognized is " ন " and the mapping of symbol " ন " to " ন " is exist there.
- The next symbol for the word "মানুহৰ "(Manuhor) is "ু" and there is a transition from state 5 to state 7, having the mapping of symbol "ু" to "ু". So our FST will move to state 7.
- From state 7 there is a single transition to state 8 with the mapping of symbol "হ" to "হ" and since our next symbol to be parsed for the word "মানুহৰ " (Manuhor) is "হ", our FST will move to the state 8.
- State 8 is another final state for the word "মানুহ" (Manuh) which is the lemma of the word "মানুহৰ " (Manuhor). From state 8, there are 3 transition state 8→9. 8→10, 8→11. Our FST will move to state 9, which have a mapping of an empty symbol to the symbol "ৰ". Thus our FST will give output as "মানুহ" (Manuh) for the input word "মানুহৰ " (Manuhor) and the symbol "ৰ " as a suffix.

V. RESULTS & DISCUSSION

SFST tool provides a command based analyzer for processing a word(s). Since command based processing is not user friendly, we have design an interface for our required input and output using JAVA language. In the interface, we need to select a file where our required content for processing is to be written. After analyzing, program will create a text file with its processed output and also displayed in the interface. For writing a text in Assamese language we have installed "*ibus-m17n*" package in Ubuntu operating system.

For testing our system we have collected around 2213 words from an Assamese story downloaded from wikisource.org [18]. After testing we have observed that, for known word FST based system provides best results but for unknown word it is unable for determine the word structure. The following table shows the result found for our test set documents.

Total words	2213
Correct	1873
Unrecognized	197
Wrongly recognized	143

Table . 2. Test Result

From the above table (Table 2) we can see that our Analyser provides only 84.64% correct results. All the known word for the system is shown as correct result or it is properly analysed. Unknown words are processed as unrecognized. Few words are wrongly recognized, this is due to the wrong analysis of tokenization and cleaning module of our Analyser. Since our analyser gets input as tokenized and clean word; whatever tokenization and cleaning module pass to analyser, it tries to analyse. As a testing result we can say that FST based model with supervised learning method is an adoptable method for highly inflected language like Assamese Language.

VI. CONCLUSION AND FUTURE WORK

In this paper, we are trying to describe a Morphological Analyzer for Assamese Language; a language commonly known as morphologically rich and relatively free-word order Indic language. Though the percentage of accuracy is not higher till now, we can increase it providing more and more information to the system.

We are currently trying to a design an interactive model for Assamese Morphological Analyzer by enhancing our current FST based model, where system will learn unknown words along with their required information from the user on run time.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

REFERENCES

1. Goswami Golockchandra (1982), Structure of Assamese, Gauhati University publication
2. Bani Kanta Kakati (1962), Assamese, Its Formation and Development
3. Kalyanee Kanchan Baruah, Pranjal Das, Abdul Hannan, Shikhar Kr Sarma, "Assamese- English Bilingual Machine Translation" International Journal on Natural Language Computing (IJNLC) Vol. 3, No.3, June 2014
4. Porter, M.F. : An algorithm for suffix stripping.
5. Lovins, J.B. : Development of a stemming algorithm.
6. Dincer, B.T. and Karaoglan, B. : Stemming in agglutinative languages : A probabilistic stemmer for Turkish.
7. Snajder, J. And Basic, B.D. : String distance-based stemming of highly inflected Croatian language
8. Krovetz, R. : Viewing morphology as an inference process.
9. Ramanathan, A. And Rao, D. : A lightweight stemmer for Hindi.
10. Majumder, P., Mitra, M., Parui, S.K., Kole, G., Mitra, P. And Datta, K. : YASS:Yet another suffix stripper.
11. Pandey, A. And Siddiqui, T.J. : An unsupervised Hindi stemmer with heuristic improvements
12. Aswani, N. And Gaizauskas, R. : Developing Morphological analysers for South Asian Languages
13. Sharma, U., Kalita, J.K. and Das, R.K. : Acquisition of morphology of an Indic language from text corpus.
14. Paul, S., Joshi, N. and Mathur, I. : Development Of a Hindi Lemmatizer.
15. Deepak Kumar, Manjeet Singh, and Seema Shukla: FST Based Morphological Analyzer for Hindi Language
16. Giridhar Sarma (1952), Asamiya Abhidhan
17. (2016) xobdo Online Dictionary, Available: http://www.xobdo.org/dic/
18. (2016) Wiki Source Assamese Story Page, Available: https://wikisource.org/wiki/বাঘ_আৰু_কেঁকোৱা).

BIOGRAPHY

Mirzanur Rahman, Assistant Professor, Department of Information Technology, Gauhati University, Assam, India.
Shikhar Kumar Sarma, Professor, Department of Information Technology, Gauhati University, Assam, India.