Analysis and Evaluation of Stemming Algorithms: A case Study with Assamese

Navanath Saharia ∗
Department of CSE
Tezpur University
Napaam, India-784028
nava_tu@tezu.ernet.in

Utpal Sharma †
Department of CSE
Tezpur University
Napaam, India-784028
utpal@tezu.ernet.in

Jugal Kalita ‡
Department of CS
University of Colorado
Colorado Springs, USA-80918
kalita@eas.uccs.edu

ABSTRACT
Stemming is the process of automatically extracting the base form of a given word of a language. Assamese is a morphologically rich, relatively free word order, Indo-Aryan language spoken in North-Eastern part of India that uses Assamese-Bengali script for writing. As it is among the less computationally studied languages, our aim is to extract stem from a given word. We adopt the suffix stripping approach along with a rule engine that generates all the possible suffix sequences. We found 82% accuracy with the suffix stripping approach after adding a root-word list of size 20,000 approximately.

Keywords
Stemming, Inflectional language, Assamese

1. INTRODUCTION
In any information retrieval system (IR), one of the first tasks is to extract and index the information available in the document in the form of words or terms. As most Indian languages are highly inflectional (as a result morphologically rich), they are adversely affected by the abundance of words appearing in various morphological forms. Indexing all the words is tedious as well as uninformative for further processing. To reduce the morphological variation in indexing the most common method is to represent the words in a normalized representative form. One such approach is the process of finding the stem or root. Though linguistically root and stem has a different meaning, in this report we alternately use both words to mean stem word from an inflected form.

∗Navanath Saharia is currently pursuing his Ph.D. in Computer Science and Engineering, Tezpur University, India.
†Utpal Sharma is working as an Associate Professor in the Department of Computer Science and Engineering, Tezpur University, India.
‡Jugal Kalita is a Professor in the Department of Computer Science, Colorado University at Colorado Springs, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICACCI’12, August 3-5, 2012, Chennai, T Nada, India.
Copyright 2012 ACM 978-1-4503-1196-0/12/08…$10.00.

It is one of the initial steps in analyzing the morphology of words. A number of approaches have been reported by researchers for different language groups. The reported approaches can be classified into two broad categories. Rule based technique[10, 14], unsupervised technique[7, 16]. The Rule based approach produces the best result either relatively fixed word order or less inflectional languages. But for highly inflectional language, we need more complete linguistic knowledge of the formation of words, to handle more complex derivational and inflectional morphology. To extract the root from an inflected word, we should have detailed knowledge of word formation of that language. For highly inflectional languages some probable ways of forming words are listed below.

1. Combining more than one root word.
2. Partially combining more than one root word.
3. Adding tense, aspect or mood markers.
4. Word-to-word translation from another language.
5. Partially combining more than one root word from two languages.

After passing through any of the steps mentioned above, a root word has changed either in terms of morphology, syntax or semantics. An unsupervised approach needs linguistics resource including corpus. Thus an unsupervised approach may not produce better results for resource poor languages like most Indian languages including Assamese. There are also a number of published works that combine both the above mentioned approach, these we refer as hybrid.

Assamese is one of the major languages in the north-eastern parts of India with approximately 30 million native speakers. In this study, we take into consideration the problem of stemming Assamese texts in which stemming is particularly hard due to the common appearance of single letter suffixes as morphological inflections. In our study we observed that more than 50% of the inflections in the Assamese language appear as single letter suffixes. Such single letter morphological inflections cause ambiguity while predicting the underlying root word.

The rest of the paper is organized as follows. In section 2 we describe previous work related to stemming followed
by linguistic characteristics of Assamese. Next section 4 describes our approach and section 5 give the result and analysis of the approach. Finally, Section 6 concludes our paper.

2. RELATED WORK
Among the reported work, Porter stemmer [10], an iterative rule based approach is most successful and used widely in different applications such as spell-checking and morph analyzing. Lovin’s approach [5] and minimum description length [3] are among other popular stemming algorithms.

In the Indian language context a number of (although very few) hand-crafted rule based stemmers have been reported, whose primary idea is to strip off suffixes. Among these [12] used a hand crafted suffix list and stripped off the longest suffix for Hindi and reported 88% accuracy for their approach using a dictionary of size 35,997. The work reported by [7] learned suffix stripping rules from a corpus and used clustering to discover the nearest class of the root word for Bengali, English and French. They described a centroid based approach that rewards the longest common prefix to form similar word clusters based on a threshold value. The work of [8] focused on some heuristic rules for Hindi and reported 89% accuracy. [1] proposed a hybrid form of [7], [8] for Hindi and Gujarati with precisions of 78% and 83%, respectively. Their approach took both prefixes as well as suffixes into account. They used a dictionary and suffix replacement rules and claimed that the approach is portable and fast.

Reported work of [4] for Punjabi used a dictionary of size 52,000 and found 81.27% accuracy using a brute-force approach. [6] described a hybrid method (rule based + suffix stripping + statistical) for Marathi and found 82.50% precision for their system. Work in Malayalam [11] used a dictionary of size 3,000 and reported 90.5% accurate system using finite state machines. We have found only [16, 9] as example of state of the art of Assamese stemming, where [16] use unsupervised approach to acquire the language specific stemming rules.

3. LANGUAGE RELATED ISSUES
Assamese is a morphologically rich Indic language spoken in north-eastern part of India. Among the reported work on Assamese [15, 16, 13, 14] are the main. Like most Indic language, in Assamese also, only suffixes and prefixes are attached to the starting and ending of a word. Though it seems simple, there are certain characteristics that complicate the Assamese stemming process.

1. A sub-string may not always appear as a suffix in word construction of Assamese. For example <bor> and <mAn> denotes indefinite plural marker but in case of kethor, kisumAn we cannot extract <bor> and <mAn> as suffix. Likewise <jan> is used to mark singular number masculine gender animated noun. But it is not the case with the word prayjan, we can not extract <jan> as a singular number masculine gender animated noun denoted suffix. In this sentence the bold hyphenated token, i.e., dAngar-dAngar kAjiyA karise.

2. The corpus we use to process has a number of misspelled words and sometimes some suffixes are written separately from the root word. For example we found suffix <samUr> more than 500 times occurring as a single word in a corpus of size 2.6 million, which is grammatically wrong. There are a number of other irregularities specifically in case of hyphenated words.

3. Sometimes a whole word can appear as a separate suffix or we can say a suffix can appear as a separate word. For example <dal> is used to mark indefinite plural marker animated noun, when <dal> appears as a separate word it means a kind of aquatic grass or a group of people.

4. There are some suffixes with single characters such as <-r> indicates the genitive case marker, <-t> denotes locative case marker these create problems with different types of non-inflectional noun or verb words. For example the word mAnuhar ends with <-r> and is in genitive case whereas amar is also ends with <-r>, but here <-r> is not a case marker, this <-r> is the part of the stem.

5. In case of some onomatopoeic words or re-duplicative hyphenated words both parts may take inflection to emphasize the word. For example:

dAngare-dAngare kAjiyA karise.

TF1 : elder+NOM-elder+NOM quarrel do+PPT ET2 : elders are quarrelling.

In this sentence the bold hyphenated token, i.e., dAngar-dAngar is inflected with a nominative case marker (NOM) to emphasize on the “elder”, which actually caused to implies a plural sense. In certain cases numbers, foreign words and symbols are also inflected depending on use. In such situations suffix stripping method fails.

6. A root can take more than one suffix in a sequence. Our study reveals that an Assamese noun root can have maximum 30,000 inflected forms in the worst case. So our aim is to extract the sequence of suffixes that a root can have. During addition of a suffix sequence, based on the sound some morphophonemic changes are made. For example <kar>(kar: to do) is a verb root ending with a consonant, and any suffix string added after the root kar that starts with a full vowel (for example <owA>, 2nd person causative verb marker) the new inflected form will be karowA. The full vowel <-o> is changed to vowel modifier. As the new form ends with a vowel sound <-a>, any addition after that does not undergo morphophonemic change.

\[
\text{karowaiCil} = \text{kar}+\text{owA} + \text{Cil} = \text{do} + 2\text{nd person causative verb marker} + \text{Past perfect tense marker.}
\]

Here the bold faced o is vowel modifier and the bold faced i is full vowel. Thus identifying this type of inflection is very tedious or sometime impossible using the simple suffix stripping method.

\text{1}\text{Transliterate form}

\text{2}\text{Approximate English Translation}
4. OUR APPROACH

In this study we use a part of EMILLE\(^3\) corpus of size 123753 words. Table 1 provides basic statistics of the dataset used and Table 2 shows the top most frequent words along with part of speech tag and frequency from the. For this experiment we consider the suffix stripping approach to find stems in Assamese. We found that the suffix stripping approach can stem words with accuracy 61%. After that we added a root-word list of size approximately 20,000 in the second approach.

<table>
<thead>
<tr>
<th>Number of assertive sentences</th>
<th>7689</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of interrogative sentences</td>
<td>218</td>
</tr>
<tr>
<td>Number of exclamatory sentences</td>
<td>112</td>
</tr>
<tr>
<td>Total number of words</td>
<td>123753</td>
</tr>
<tr>
<td>Number of words in the longest sentence</td>
<td>1</td>
</tr>
<tr>
<td>Number of words in the shortest sentence</td>
<td>11</td>
</tr>
<tr>
<td>Unique words in the dataset</td>
<td>25111</td>
</tr>
<tr>
<td>Mean word length</td>
<td>5.85</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the test dataset

4.1 Approach-1

As mentioned above an Assamese root can take a series of suffixes sequentially, so our first aim is to find the probable sequence of suffixes that a root has. We manually collected all possible suffixes and categorised them into four basic groups, viz., case marker, plural marker, classifier and emphatic marker. After that the rule engine generates a list of suffixes. The rule engine is a module that generates all probable suffix sequences that may be attached after a root, based on the affixation rule of Assamese. For example an Assamese noun root may take the following sequence of suffixes -

1. root + plural marker (PL)
   Example: \(m\)Anuhbor = \(m\)Anuh + bor

2. root + case marker (CM)
   Example: \(m\)Anuhbar = \(m\)Anuh + r

3. root + plural marker + case marker + emphatic marker (EM)
   Example: \(m\)Anuhkborarhe = \(m\)Anuh + bor + r + he
   etc..

We confirmed 14 such rules and generated approximately 18,194 suffix sequences for Assamese. The next phase was to extract the longest possible sub-string from the given word using a suffix-list look-up. Among the generated suffixes, 11 suffixes were single word suffix and this single word suffixes caused rapid the down-fall of the accuracy, as they created the problem mentioned in section 3. The pseudo-code for this approach given in 1.

After this experiment we found that, 61% words are correctly stemmed. It is observed that the error rate for inflected word with suffix length greater than 4 is less that 1%, whereas the error rate for inflected words with suffix length equal to 1 is highest, 56%. It is clear from this experiment that the error rate will decrease with increasing suffix length. As the error rate of single character suffix is highest, one possible solution to increase the accuracy is to add a root-word list, which is discussed in section 4.2.

4.2 Approach-2

With the existing rule-set or knowledge base we were not able to classify all the words of language into a well-defined category. Each language has such words that cause trouble. To handle this type of exception we maintain a word-list, where most frequent and exceptional root words are stored. Thus in this approach first the word-list is checked against each word to be stemmed after that we apply algorithm 1. The main advantage of the approach is that it minimizes the over-stemming and under-stemming error. We give the steps mentioned in 2.

| Correctly stemmed | 61% |
| Incorrectly stemmed | 39% |
| Stemmed as no inflection | 24% |
| Stemmed as single character inflection | 56% |
| Stemmed as multiple inflection | 20% |

5. EXPERIMENTAL RESULT

The accuracy found using approach 1 is 61%, and using approach 2 is 82%. The complete statistics of our experiment is tabulated in Table 3 and Table 4. In comparison with [16, 9] the result produced by approach 2 with only 20,000 root-word list look-up is considerably better.

Table 3: Obtained result using approach-1

We evaluate the strength of approach 1 and approach 2 using [2]. Table 5 and Table 6 shows the evaluation result of the both approaches. The index compression factor for approach
In comparison with [16, 9], the result produced by approach
fix stripping, which increase the accuracy from 61% to 82%.
In the second approach we use a frequent root-word with suf-
fi
x generated by the rule engine, there is always a problem

su
ffi
su

approach we describe the su

x stripping approach with the

niques with a dataset of size 123753 words. In the first

In this report we analyse and evaluate two stemming tech-

iques: A case study for Marathi language. 

Linguistics

Mechanical Translation and Computational

Learning of Morphology.

Natural Language

Engineering, 1, 1998.

D. Kumar and P. Rana. Design and development of a
stemmer for Punjabi. International Journal of

Mechanical Translation and Computational

M. M. Majgaonker and T. J. Siddiqui. Discovering
suffices: A case study for Marathi language.
International Journal on Computer Science and
Engineering

International Journal of


P. Majumder, M. Mitra, S. Parui, G. Kole, P. Mitra,
and K. Datta. YASS: Yet another suffix stripper. ACM

A. Pandey and T. Siddiqui. An unsupervised Hindi
stemmer with heuristic improvements. In In
Proceedings of the second workshop on Analytics for

M. Parakh and R. N. Developing morphology analyser
four Indian languages using a rule based suffix
stripping approach. Language In India, pages 13–15,
2011.

M. F. Porter. An algorithm for suffix stripping.
Program, 14:130aÂŠ137, 1980.

V. S. Ram and S. L. Devi. Malayalam stemmer. In
Morphological Analysers and Generators, pages

A. Ramanathan and D. D. Rao. A lightweight
stemmer for Hindi. In Proceedings of the 10th
Conference of the European Chapter of the Association
for Computational Linguistics (EACL), on
Computational Linguistics for South Asian Languages,
2003.

N. Saharia, D. Das, U. Sharma, and J. Kalita. Part of
speech tagger for Assamese text. In Proceedings of the

6. CONCLUSION
In this report we analyse and evaluate two stemming tech-
niques with a dataset of size 123753 words. In the first
approach we describe the suffix stripping approach with the
suffix generated by the rule engine, there is always a problem
of over-stemming and under-stemming with this approach.
In the second approach we use a frequent root-word with suf-
fix stripping, which increase the accuracy from 61% to 82%.
In comparison with [16, 9], the result produced by approach
2 with only 20,000 root-word list look-up is considerable.

Table 2: First 10 most frequent word in the dataset

<table>
<thead>
<tr>
<th>Sl.</th>
<th>Assamese word</th>
<th>Approximate Translation</th>
<th>POS tag</th>
<th>Frequency</th>
<th>is inflected?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>aAru</td>
<td>and</td>
<td>Particle</td>
<td>2249</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>ma</td>
<td>me</td>
<td>Pronoun</td>
<td>1105</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>ai</td>
<td>this/it</td>
<td>Demonstrative</td>
<td>1087</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>mor</td>
<td>me + Genitive marker</td>
<td>pronoun</td>
<td>1063</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>teon</td>
<td>he</td>
<td>Pronoun</td>
<td>765</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>aAsil</td>
<td>be + Past tense</td>
<td>Verb</td>
<td>751</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>teonr</td>
<td>he + Genitive case marker</td>
<td>pronoun</td>
<td>641</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>kari</td>
<td>do + Present Participle</td>
<td>Verb</td>
<td>640</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>ces</td>
<td>that</td>
<td>Demonstrative</td>
<td>639</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>kintu</td>
<td>but</td>
<td>Particle</td>
<td>583</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 4: Obtained result using approach-2

<table>
<thead>
<tr>
<th>Approach -2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly stemmed</td>
<td>82%</td>
</tr>
<tr>
<td>Incorrectly stemmed</td>
<td>18%</td>
</tr>
<tr>
<td>Stemmed as no inflection</td>
<td>23%</td>
</tr>
<tr>
<td>Stemmed as single inflection</td>
<td>57%</td>
</tr>
<tr>
<td>Stemmed as multiple inflection</td>
<td>20%</td>
</tr>
</tbody>
</table>

Table 5: Evaluation of approach-1 using [2]

<table>
<thead>
<tr>
<th>Approach -1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Words in the test file</td>
<td>123753</td>
</tr>
<tr>
<td>Unique words before stemming</td>
<td>25111</td>
</tr>
<tr>
<td>Unique words after stemming</td>
<td>18012</td>
</tr>
<tr>
<td>Min. word length after stemming</td>
<td>1</td>
</tr>
<tr>
<td>Max. word length after stemming</td>
<td>14</td>
</tr>
<tr>
<td>Mean number of words</td>
<td>1.11</td>
</tr>
<tr>
<td>Mean stemmed word length</td>
<td>4.03</td>
</tr>
<tr>
<td>Index compression factor</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 6: Evaluation of approach-2 using [2]

<table>
<thead>
<tr>
<th>Approach -2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Words in the test file</td>
<td>123753</td>
</tr>
<tr>
<td>Unique words before stemming</td>
<td>25111</td>
</tr>
<tr>
<td>Unique words after stemming</td>
<td>17218</td>
</tr>
<tr>
<td>Min. word length after stemming</td>
<td>1</td>
</tr>
<tr>
<td>Max. word length after stemming</td>
<td>14</td>
</tr>
<tr>
<td>Mean number of words</td>
<td>1.36</td>
</tr>
<tr>
<td>Mean stemmed word length</td>
<td>4.55</td>
</tr>
<tr>
<td>Index compression factor</td>
<td>0.31</td>
</tr>
</tbody>
</table>

7. REFERENCES
