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. 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 board 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

[*] 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.

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

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 *ketbor, kisumAn* we can not 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 *prayojan*, we can not extract <*-jan*> as a singular number masculine gender animated noun denoted suffix.

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 <*-samUh*> 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.
   TF[1] : elder+NOM-elder+NOM quarrel do+PPT
   ET[2] : 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*>, 2^{nd} 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 <*-aA*>, any addition after that does not undergo morphophonemic change.

   kar**o**wa**i**Cil = kar+owA + iCil
   = do + 2^{nd} person causative verb marker
   + 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.

[1] Transliterate form
[2] 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 the 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.

Number of assertive sentences	7689
Number of interrogative sentences	218
Number of exclamatory sentences	93
Total number of words	123753
Number of words in the longest sentence	112
Number of words in the shortest sentence	1
Unique words in the dataset	25111
Mean word length	5.85

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: *mAnuhbor = mAnuh + bor*
2. root + case marker (CM)
   Example: *mAnuhar = mAnuh + r*
3. root + plural marker + case marker + emphatic marker (EM)
   Example: *mAnuhborarhe = mAnuh + 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.

Algorithm 1 Stemming Approach-1

```
1: Read a line from the corpus file.
2: Extract words (from this point we called it as token)
   from the line, clean the token, that is remove punctuation
   marker attached with token if there is one.
3: Look up suffix-list generated manually from the end of
   the token. If matched with the suffix-list extract and
   exit.
4: Go to step 1 until the end of the corpus.
```

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

Algorithm 2 Stemming Approach-2

```
1: Read a line from the corpus file.
2: Extract words (from this point we called it as token)
   from the line, clean the token, that is remove punctuation
   marker attached with token if there is one.
3: Check the dictionary. If a dictionary entry matches with
   the token, mark token as root word and exit otherwise
   execute the next step.
4: Look up suffix-list generated manually from the end of
   the token. If there is a match with the suffix-list extract
   and exit.
5: Go to step 1 until the end of the corpus.
```

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.

Approach -1	
Correctly stemmed	61%
Incorrectly stemmed	39%
Stemmed as no inflection	24%
Stemmed as single character inflection	56%
Stemmed as multiple inflection	20%

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

[3] http://www.emille.lancs.ac.uk/

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

Table 2: First 10 most frequent word in the dataset

Approach -2	
Correctly stemmed	82%
Incorrectly stemmed	18%
Stemmed as no inflection	23%
Stemmed as single inflection	57 %
Stemmed as multiple inflection	20%

Table 4: Obtained result using approach-2

1 is 0.28 whereas index compression factor for approach 2 is 0.31.

Approach -1	
Words in the test file	123753
Unique words before stemming	25111
Unique words after stemming	18012
Min. word length after stemming	1
Max. word length after stemming	14
Mean number of words	1.11
Mean stemmed word length	4.03
Index compression factor	0.28

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

Approach -2	
Words in the test file	123753
Unique words before stemming	25111
Unique words after stemming	17218
Min. word length after stemming	1
Max. word length after stemming	14
Mean number of words	1.36
Mean stemmed word length	4.55
Index compression factor	0.31

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

6. CONCLUSION

In this report we analyse and evaluate two stemming techniques 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 suffix 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.

7. REFERENCES

[1] N. Aswani and R. Gaizauskas. Developing morphological analysers for south asian languages: Experimenting with the Hindi and Gujarati languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC)*, 2010.

[2] W. B. Frakes and C. J. Fox. Strength and similarity of affix removal stemming algorithms. *SIGIR Forum*, 37(1):26–30, Apr. 2003.

[3] J. Goldsmith. An Algorithm for the Unsupervised Learning of Morphology. *Natural Language Engineering*, 1, 1998.

[4] D. Kumar and P. Rana. Design and development of a stemmer for Punjabi. *International Journal of Computer Applications*, 11, 2011.

[5] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11, 1968.

[6] M. M. Majgaonker and T. J. Siddiqui. Discovering suffixes: A case study for Marathi language. *International Journal on Computer Science and Engineering*, 04:2716–2720, 2010.

[7] P. Majumder, M. Mitra, S. Parui, G. Kole, P. Mitra, and K. Datta. YASS: Yet another suffix stripper. *ACM Transanctions and Information Systems*, 25, 2007.

[8] A. Pandey and T. Siddiqui. An unsupervised Hindi stemmer with heuristic improvements. In *In Proceedings of the second workshop on Analytics for noisy unstructured text data*, page 99âĂŞ105, 2008.

[9] 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.

[10] M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130âĂŞ137, 1980.

[11] V. S. Ram and S. L. Devi. Malayalam stemmer. In *Morphological Analysers and Generators*, pages 105–113, 2010.

[12] 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 Computatinal Linguistics for South Asian Languages*, 2003.

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

ACL-IJCNLP 2009 Conference Short Papers, pages 33–36, Suntec, Singapore, 2009.

[14] N. Saharia, U. Sharma, and J. Kalita. A suffix-based noun and verb classifier for an inflectional language. In *International Conference on Asian Language Processing*, pages 19–22. IEEE Computer Society, 2010.

[15] U. Sharma, J. Kalita, and R. Das. Root word stemming by multiple evidence from corpus. In *Proceedings of 6th International Conference on Computational Intelligence and Natural Computing (CINC-2003)*, 2003.

[16] U. Sharma, J. K. Kalita, and R. K. Das. Acquisition of Morphology of an Indic Language from Text Corpus. *ACM Transactions on Asian Language Information Processing (TALIP)*, 7, 2008.