An improved stemming approach using HMM for a highly inflectional language

Navanath Saharia[1], Kishori M Konwar[2], Utpal Sharma[1], and Jugal K Kalita[3]

[1] Department of CSE, Tezpur University, India; {nava_tu, utpal}@tezu.ernet.in.
[2] Department of MI, University of British Columbia, Canada; kishori82@yahoo.com.
[3] Department of CS, University of Colorado at Colorado Springs, USA; jkalita@uccs.edu.

Abstract. Stemming is a common method for morphological normalization of natural language texts. Modern information retrieval systems rely on such normalization techniques for automatic document processing tasks. High quality stemming is difficult in highly inflectional Indic languages. Little research has been performed on designing algorithms for stemming of texts in Indic languages. In this study, we focus on the problem of stemming texts in Assamese, a low resource Indic language spoken in the North-Eastern part of India by approximately 30 million people. Stemming is hard in Assamese due to the common appearance of single letter suffixes as morphological inflections. More than 50% of the inflections in Assamese appear as single letter suffixes. Such single letter morphological inflections cause ambiguity when predicting underlying root word. Therefore, we propose a new method that combines a rule based algorithm for predicting multiple letter suffixes and an HMM based algorithm for predicting the single letter suffixes. The combined approach can predict morphologically inflected words with 92% accuracy.

1 Introduction

Most information retrieval systems represent documents as a set of words. The efficiency of such systems is adversely affected by the abundance of words appearing in various morphological forms either as a result of inflection or derivation. To reduce this detrimental effect of morphological variations, one common method is to represent the text in a normalized form. One such approach is the process of finding the root word from an inflected form. It is an initial step in analyzing the morphology of words. A number of approaches have been proposed by researchers for stemming, e.g., affix stripping, co-occurrence computation, dictionary look-up, longest suffix matching and probabilistic. Most approaches are first developed for English, and later adapted for other languages. So these approaches may not work properly for highly inflectional Indic languages.

Assamese, is a rarely studied low resource language, spoken in the north-eastern parts of India. Approximately 30 million people speak Assamese. In this study, we address the problem of stemming Assamese texts. Stemming in Assamese is difficult due to the common appearance of single letter suffixes as morphological inflections. Our experiments show that more than 50% of inflections in Assamese appear as single letter suffixes. Such single letter morphological inflections cause ambiguity when one predicts 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 brief linguistic characterisation of Assamese, our experimental test-bed in Section 3. Section 4 describes our approach and Section 5 provides the results and analysis of the approach. Section 6 concludes our paper.

2 Previous Work

Porter stemmer [1], an iterative rule based approach has found great success and is used widely in various applications such as spell-checking, and morphological analysis. In the Indian language context, a few hand-crafted rule-based stemmers have been reported to strip off suffixes. Among these, [2] use a hand crafted suffix list and strip off longest suffixes for Hindi and report 88% accuracy using a dictionary of size 35,997. [3] learn suffix stripping rules from a corpus and use clustering to discover the nearest class of the root word for Bengali, English and French. They describe a centroid based approach that rewards the longest common prefix to form similar word clusters based on a threshold value. [4] focus on heuristic rules for Hindi and report 89% accuracy. [5] propose a hybrid form using approaches reported in [3] and [4] for Hindi and Gujarati with precisions of 78% and 83%, respectively. Their approach takes both prefixes as well as suffixes into account. They use dictionary and suffix replacement rules, and claim that the approach is portable and fast.

Kumar and Rana [6] use a dictionary of size 52,000 and obtain 81.27% accuracy in Punjabi using a brute-force approach. Majgaonker and Siddiqui [7] describe a hybrid method (rule based + suffix stripping + statistical) for Marathi and claim 82.50% precision for their system. Sharma et. al [8], [9], [10] describe an unsupervised approach, that learn morphology from unannotated Assamese corpus and report 85% precision value. The method discussed by Saharia et al. [11] and [12] for parts-of-speech tagging has three basic steps: brute-force determination of suffix sequences, suffix sequence pruning and suffix stripping. Table 1 enumerates the statistics reported by the different methods. In this paper, we extend this method for stemming inflected words in Assamese by using HMM for single character inflections.

3 Suffixes in Assamese

In the context of stemming, the most common property of Indic languages is that, they take a sequence of suffixes after the root words. We give an example from Assamese below.

নাতিনীয়েককেইজনীমানেহে→ নাতিনী + য়েক + কেইজনী + মান + ে + হে

Report	Language	Dictionary Size	Accuracy	Used technique
[1]	English		90.00%	Suffix Stripping
[13]	Arabic		96.00%	Rule base
[14]	Dutch	45000	79.23%	Porter Stemmer
[10]	Assamese		85%	Unsupervised approach
[3]	Bengali		90.00%	Suffix Stripping
[6]	Punjabi	52,000	81.27%	Brute Force Approach
[7]	Marathi		82.50%	Rule based + Statistical
[15]	Gujarati		90.00%	Unsupervised + Rule based
[16]	Malayalam	3,000	90.5%	Finite State Machine
[2]	Hindi	35,997	88.00%	Suffix Stripping
[4]	Hindi		90.00%	Unsupervised

Table 1. Reported performance of stemmers in some highly inflectional languages (except English)

nAtinIyekkeijanImAnehe → *nAtinI* +*yek* +*keijanI* +*mAn* +*e* +*he*
nAtinIyekkeijanImAnehe → noun root+ inflected form of kinship noun[4] + indefinite feminine marker + plural marker + nominative case marker + emphatic marker. (Approximate English meaning: only a few granddaughters)

These sequences of suffixes can easily be stripped off using algorithm proposed by [12]. A major drawback of the prior method is that it is not able to identify the single letter suffix well. For example, the method removes ৰ from the words অমৰ (*amar* : immortal) and মানুহৰ (*mAnuhr* : man+genitive marker), whereas the first word is a root word form, but the second word is inflected, with -ৰ (*ra*) as an genitive case marker. We have found that, in Assamese, a noun root word may potentially take more than 15,000 different inflections and up to 5 sequential suffixes after the noun root. Likewise, a verb may potentially also have more than 10,000 different inflectional forms. The frequency of appearance of single-letter inflections in Assamese is higher than multiple-letter inflections.

Among major Indic languages, Bengali is the closest to Assamese in terms of spoken and written forms. Table 2 tabulated an important observation around 2000 words collected from different news articles of English, Assamese, Bengali and Hindi. The forth column describes the inflected unique words in terms of number.

[4] All relational nouns in Assamese have the inflection য়েক (*yek*) in 3^{rd} person. For example in 3^{rd} person relational noun ভাই (*bhAi : younger brother*) is inflected to ভায়েক (*bhAyek*), ককাই (*kakAi : elder brother*) is inflected to ককায়েক (*kakAyek*). Bora [17] reports that Assamese has the highest numbers of kinship nouns among Indo-Aryan languages.

[5] http://timesofindia.indiatimes.com (*Access date : 22-Nov-2012*)

[6] http://janasadharan.in (*Access date : 22-Nov-2012*)

[7] http://www.anandabazar.com (*Access date : 22-Nov-2012*)

[8] http://www.jagran.com (*Access date : 23-Nov-2012*)

Language	Sent.	Words		Inflection type			Source of text
		Total	Unique	Single	MS*	Multiple	
English	82	2012	843	06.88%	-	18.50%	Times of India[5]
Assamese	132	2164	1293	28.21%	09.49%	13.06%	Dainik Janasadharan[6]
Bengali	202	2205	1246	17.97%	07.22%	18.37%	Anandabazar Patrika[7]
Hindi	116	2162	795	12.07%	03.14%	12.82%	Dainik Jagaran[8]

Table 2. A random survey on single letter inflection. *MS** : Suffix sequence or multiple suffix end-with single letter suffix.

We observe that the compression rates for English, Assamese, Bengali and Hindi are 41.89%, 59.75%, 56.50% and 36.77% respectively. We also see that among the languages Assamese has the highest single letter inflectional suffixes. This behoove us to develop an algorithm to improve the accuracy of detecting single-letter suffixes and use it in combination with the algorithm in [12]. The next section discusses the an Hidden Markov Model based approach we use to handle single-letter suffixes better.

4 HMM Based Approach

In this paper, we extend the algorithm in [12] to classify Assamese nouns and verbs. In this previously published work, Saharia et al. could automatically detect sequences of suffixes from inflected nouns and verbs and stem correctly with an accuracy 81%. Experimental result from [12] are given in Table 3. The algorithm accurately stems multiple character suffixes, but fails to handle well single character suffixes such as ৰ (*ra : genitive case marker*), and ক (*ka : accusative case marker*). These single letter morphological inflections, in Assamese are similar to post-positions in the English language.

We model Assamese text as a sequence of words produced by a generator with two possible states, *non-morphological* and *morphological*. When a morphological affix is present in a word, the state determines whether the affix is a part of the root word (in state *non-morphological*) or is a morphologically inflected word (in state *morphological*). In the current study, we present an HMM based algorithm to predict the hidden states of the generator. Our experiments show that our approach can stem inflected word with single character suffixes with an accuracy of 91%.

Our formulation of the problem in the form of a Hidden Markov Model parallels the well-known problem of "*Fair Bet Casino*", where a sequence of rolls of a dice find whether a dealer uses a fair dice or a loaded one. We model the commentator or writer as a generator of a sequence of words, $w_0, w_1, \cdots, w_{n-1}$, i.e., the words of a corpus in the order it is intended to be read. Each word w_i can be broken down as $p_i \circ s_i$, where p_i is a root word; s_i an inflectional suffix and $\circ$ the concatenation operation between two strings. We denote the set of inflectional suffixes by S, including the empty string ϵ. If w is a root word, $p \circ \epsilon$ is also the root word. For any word $w \equiv p \circ s$ if $s = \epsilon$, we say word w is a *root*

	Accuracy in %		
Correctly stemmed	81%		
No inflection (ϵ)		43%	
One character inflection (S_1)		36%	
Multiple character inflection (S_m)		21%	
Wrongly stemmed	19%		
There is no inflection but stemmed as inflected		66%	
Mark as one character inflection (S_1)			62%
Mark as multiple character inflection (S_m)			38%
There is one character inflection, but stemmed wrongly		27%	
Mark as no inflection (ϵ)			83%
Mark as multiple character inflection (S_m)			17%
There is multiple character inflection, but stemmed wrongly		17%	
Mark as one character inflection (S_1)			32%
Mark as multiple character inflection (S_m)			12%
Mark as no inflection (ϵ)			56%

Table 3. Calculated result for Assamese using [12] approach.

word. On the other hand, $s \in S$ and $s \neq \epsilon$, for any word w $(= p \circ s)$ implies that word w ends with an inflectional suffix but does not necessarily mean that $p \circ s$ is not a root word or the converse. We model this problem of predicting if a word in a sentence is morphologically inflected or not as being able to model the sense of the generator of the sentence when the word was written. Suppose we are given a set of inflections S in the language, not necessarily all inflections in the language. We can represent any given word w as $p \circ s$ such that $s \in S$. If $s = \epsilon$ is the only possible string of S that satisfies $w = p \circ s$, we say the generator G does not produce meaning leading to a morphological inflection for the word. On the other hand, if there is an inflection $s \in S$ and $s \neq \epsilon$ such that $w = p \circ s$, we say w is morphologically inflected whether the generation is meaningful. Therefore, we define two states of the generator at the time of generating the word, *viz.*, *morphologically inflected* (M) and *morphologically not inflected* (N). We associate with a corpus of some length ℓ, $w_0, w_1, \cdots w_{\ell-1}$ a series of states with labels N and Ms as $q_0, q_1, \cdots, q_{\ell-1}$ such that $q_i \in Q \equiv \{N, M\}$.For example in Table 4, we described the series of states of a sentence, “নবীনহঁতৰ ঘৰ আমাৰ ঘৰৰ পৰা এমাইলমান দুৰত”
TF: *nabinhatar ghar aAmAr gharar parA emAilmAn durat.*
WT: nabin’s(plural) house our house from one-mile distance

Therefore, for a corpus generated by G the problem of deciding if a word is morphologically inflected, boils down to determining the state of G (N or M) at the exact moment of generating the word. We construct an HMM based algorithm to predict the states of G corresponding to the words of the corpus. Therefore, the problem has two steps: (a) training the HMM parameters with a training corpus and (b) applying the calibrated algorithm on a test corpus to detect morphologically inflected words.

w	w_0	w_1	w_2	w_3	w_4	w_5	w_6
words	নবীনহঁতৰ (*nabinhatar*)	ঘৰ (*ghar*)	আমাৰ (*aAmAr*)	ঘৰৰ (*gharar*)	পৰা (*parA*)	এমাইলমান (*emAilmAn*)	দুৰত (*durat*)
p	নবীন (*nabin*)	ঘৰ (*ghar*)	আমাৰ (*aAmAr*)	ঘৰ (*ghar*)	পৰা (*parA*)	এমাইল (*emAil*)	দুৰ (*dur*)
s	-হঁতৰ	ϵ	ϵ	-ৰ	ϵ	-মান	-ত
q	M	N	N	M	N	M	M

Table 4. An example sentence as modelled using our generative model of the text for the morphological inflections.

We know that the inaccuracy of the method in [12] comes mostly from single letter inflections. For multiple letter inflections, the ambiguity of being a true inflection versus a coincidental match of the word with the set of inflections is significantly low. We denote by S_1 and S_m the set of single letter and multi-letter inflections, respectively. In order to simplify our analysis, we consider the following partition of the set of inflections S as $\{\epsilon\}$, S_1 and S_m. Therefore, the appearance of a multi-inflection suffix on a word almost definitely generates the presence of morphological inflection. Hence, we can safely assume that if $s_i \in S_m$ for a word w_i, $q_i = M$. We can state the same notion as for $q_i = N$, $e_{q_i}(s) = 0$ for $s \in S_m$. Since we are essentially trying to predict the correct state of G for only single letter inflections (i.e., S_1), we assume that all inflections in S_1 are equivalent and, similarly the inflections in S_m are also equivalent to one another. So, we assume that our alphabet S in the Hidden Markov Model as $S' = \{\epsilon, s_1, s_m\}$, where s_1 and s_m are single-letter and multi-letter morphological inflections, respectively.

Estimating $a_{k\ell}$ and $e_k(b)$. We estimate the two needed parameters $a_{k\ell}$ and $e_k(b)$ from the training corpus. First we mark the states of the generator, G for every word in the corpus. Next, we identify the inflections, for every word, as belonging to $\{\epsilon\}$, S_1 and S_m if it has no inflection, has a single letter inflection or has a multi-letter inflection, respectively. Then, we calculate the number of times each particular transition and emission occurs in the training corpus. Let us denote these counts by $A_{k\ell}$ and $E_k(b)$. Then estimate the the parameters $a_{k\ell}$ and $e_k(b)$ as

$$\hat{a}_{k\ell} = \frac{A_{k\ell}}{\sum_{\ell'} A_{k\ell'} + \delta} \; and \; \hat{e}_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b') + \delta}$$

where δ is a very small positive number to avoid division by 0.

5 Results and Discussion

Preparation of training data. For our experiment, we used text from the EMILLE[9] Assamese corpus. We labelled approximately 2,000 words (144 sen-

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

tences) with 4 tags: words with multi-character inflection (M_{sm}), words with single character inflection (M_{s1}), words with no inflection (N_e) and words that have no inflection, but end with single character inflection marker (N_{s1}). Table 5 gives the details suffixes present in the training set. We found the suffix 'ৰ', *genitive case marker* and the suffix symbol ◌ে, *nominative case marker* are most frequently among single character suffixes.

Words with single character inflection (S_1)	34%
Words with multiple character inflection (S_m)	21%
Words with no inflection (ϵ)	43%
Number of foreign words, numbers and symbols	2%

Table 5. Training corpus details used for experiment

Result and Analysis. The results obtained using the prior approach [12] have already been given in Table 3. Out of 19% words that the published method stems incorrectly, 27% of the words have single character inflection. After applying HMM to detect single character inflection, the overall accuracy increases by approximately 11.43%. The results obtained by combining previous approach with HMM are given in Table 6. Our test data set contains 1542 words (108 sentences) taken from EMILLE corpus. We manually evaluate the correctness of the output. We have to keep in mind that the previous approach used a frequent word lists of around twenty thousand words and a rule-base. In Table 6, "Stemmed as no inflection" means, either a single letter or multiple letter suffix was attached with the word and marked incorrectly as no inflection. "Stemmed as single character inflection" means that there was no inflection or multiple inflection, but the program separated the last character from the word incorrectly. Similarly "Stemmed as multiple inflection" means that there was no inflection or a single character inflection and the program separated a sequence of characters from the word incorrectly. The same test data was used to run the experiment with Morfessor [18] as well.

	[12]	Current paper	Morfessor
Correctly stemmed	81%	92%	82%
Incorrectly stemmed	19%	8%	18 %
Stemmed as no inflection	23%	36%	29%
Stemmed as single character inflection	57%	33%	19%
Stemmed as multiple inflection	20%	31%	52%

Table 6. Comparison of obtained result

The "transition probability" controls the way a state at time t is chosen over a given state at time $t-1$. Table 7 gives transition probabilities for the training set, where S_0 is the initial state, M is the inflected form of the word and N is the root form of a word. The "emission probability" is the probability of observing the input sentence or sequence of words W given the state sequence T, that is $P(W|T)$. Table 8 describes the emission probabilities for the training set, where S_0 is the initial state, M is the inflected form of the word, N is the root form of a word, ϵ is the zero inflectional form, s_1 is the single character inflection and s_m is the multiple character inflectional form.

	S_0	M	N
S_0	0.0000	0.5000	0.5000
M	0.0000	0.4269	0.5716
N	0.0000	0.4739	0.5261

Table 7. Transition probabilities for the training set.

	ϵ	s_1	s_m
S_0	1.0000	0.0000	0.0000
M	0.0000	0.6705	0.3295
N	0.5557	0.4443	0.0000

Table 8. Emission probabilities for the training set.

Evaluation. Comparison of Table 1 and Table 6, demonstrates that the performance of the current approach is better, for Assamese. We evaluate stemmer strength using [19]. Table 9 shows the evaluation results for both stemming techniques. According to [19], a conflation class is, the number of unique words before stemming (N) divided by the number of unique stems after stemming (S), i.e., the average size of groups of words converted to a particular stem. The index compression factor, $(N - S)/N$ takes into account the collection of unique words compressed by the stemmer. Thus high index compression factor represents a heavy stemmer. A heavy stemmer produces over-stemming,as it removes sequences of characters from words that are do not contain any suffix. For example, Morfessor separates words such প্ৰয়োজন (*prayojan* : need), আয়োজন (*aAyojan* : arrangement) and মানুহজন (*mAnuhjan* : the man) into a single group removing the suffix *-jan* from each of the words, Whereas the first two words are not inflected and are root words, the last word *mAnuhjan* is inflected with the definitive marker *-jan*, although all the words are ends with *-jan* suffix.

	[12]	Current Paper	Morfessor
Words in the test file	1542	1542	1542
Unique words before stemming	1010	1010	1010
Unique words after stemming	859	810	721
Min./Max. words length after stemming	1/18	1/18	1/18
Number of words per conflation class	1.17	1.24	1.40
Mean stemmed word length	5.36	6.03	4.94
Index compression factor	0.15	0.20	0.29

Table 9. Evaluation of stemmer strength using [19]

6 Conclusion

In this paper, we have presented a stemmer for Assamese, a morphologically rich, agglutinating, and relatively free word order Indic language. In this language, the presence of single letter suffixes is the most common reason for ambiguity in morphological inflections. Therefore, we propose a new method that combines a rule based algorithm for predicting multiple letter suffixes and an HMM based algorithm for predicting single letter suffixes. The resulting algorithm uses the strengths of both algorithms leading to a higher overall accuracy of 92% compared to 81% for previously published methods for Assamese. The 92% result is better than the published results for all other Indian languages(see Table 1). Future work will include calibrating the parameters of the HMM model with a much larger training corpus. In addition, it would be interesting to explore the possibility of modelling all possible morphological variations using Conditional Random Fields, which has been very successful in similar situations. It will also be useful to apply the method to other highly inflectional languages.

References

1. Porter, M.F.: An algorithm for suffix stripping. Program **14** (1980) 130–137
2. Ramanathan, A., Rao, D.: 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, Budapest (2003) 43–48
3. Majumder, P., Mitra, M., Parui, S.K., Kole, G., Mitra, P., Datta, K.: Yass: Yet another suffix stripper. ACM Trans. Inf. Syst. **25**(4) (October 2007)
4. Pandey, A.K., Siddiqui, T.J.: An unsupervised Hindi stemmer with heuristic improvements. In: Proceedings of the second workshop on Analytics for noisy unstructured text data. AND ’08, Singapore (2008) 99–105
5. Aswani, N., Gaizauskas, R.: 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), Malta (2010) 811–815
6. Kumar, D., Rana, P.: Design and development of a stemmer for Punjabi. International Journal of Computer Applications **11**(12) (December 2010) 18–23
7. Majgaonker, M.M., Siddiqui, T.J.: Discovering suffixes: A case study for Marathi language. International Journal on Computer Science and Engineering **04** (2010) 2716–2720
8. Sharma, U., Kalita, J., Das, R.: Unsupervised learning of morphology for building lexicon for a highly inflectional language. In: Proceedings of the ACL-02 workshop on Morphological and phonological learning, Philadelphia (2002) 1–6
9. Sharma, U., Kalita, J., Das, R.: Root word stemming by multiple evidence from corpus. In: Proceedings of 6th International Conference on Computational Intelligence and Natural Computing (CINC-2003), North Carolina (2003) 1593–1596
10. Sharma, U., Kalita, J.K., Das, R.K.: Acquisition of morphology of an indic language from text corpus. ACM Transactions of Asian Language Information Processing (TALIP) **7**(3) (June 2008) 9:1–9:33

11. Saharia, N., Sharma, U., Kalita, J.: Analysis and evaluation of stemming algorithms: a case study with Assamese. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ICACCI '12, Chennai, India, ACM (2012) 842–846
12. Saharia, N., Sharma, U., Kalita, J.: A suffix-based noun and verb classifier for an inflectional language. In: Proceedings of the 2010 International Conference on Asian Language Processing. IALP '10, Harbin, China, IEEE Computer Society (2010) 19–22
13. Al-Shammari, E.T., Lin, J.: Towards an error-free Arabic stemming. In: Proceedings of the 2nd ACM workshop on Improving non english web searching. iNEWS '08, New York, NY, USA, ACM (2008) 9–16
14. Gaustad, T., Bouma, G.: Accurate stemming of Dutch for text classification. Language and Computers **14** (2002) 104–117
15. Suba, K., Jiandani, D., Bhattacharyya, P.: Hybrid inflectional stemmer and rule-based derivational stemmer for Gujrati. In: 2nd Workshop on South and Southeast Asian Natural Languages Processing, Chiang Mai, Thailand (2011)
16. Ram, V.S., Devi, S.L.: Malayalam stemmer. In Parakh, M., ed.: Morphological Analysers and Generators, LDC-IL, Mysore (2010) 105–113
17. Bora, L.S.: Asamiya Bhasar Ruptattva. M/s Banalata, Guwahati, Assam, India (2006)
18. Creutz, M., Lagus, K.: Induction of a simple morphology for highly-inflecting languages. In: Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology. SIGMorPhon '04, Barcelona, Spain, ACL (2004) 43–51
19. Frakes, W.B., Fox, C.J.: Strength and similarity of affix removal stemming algorithms. SIGIR Forum **37**(1) (April 2003) 26–30