IndiLem@FIRE-MET-2014: An Unsupervised Lemmatizer for Indian Languages

Abhisek Chakrabarty
CVPR Unit
Indian Statistical Institute
203 B.T. Road, Kolkata, India
abhisek0842@gmail.com

Sharod Roy Choudhury
Narula Institute of Technology
Kolkata, India
sharod.roy@gmail.com

Utpal Garain
CVPR Unit
Indian Statistical Institute
203 B.T. Road, Kolkata, India
utpal@isical.ac.in

Abstract

An unsupervised and language independent lemmatization procedure has been developed for major Indian languages (Bengali, Hindi etc) which are morphologically very rich and agglutinative in nature. The task of a lemmatizer is mapping an inflected surface word to its appropriate dictionary root word and it is a pre-requisite for implementing several NLP tools like Word Sense Disambiguation system, Machine Translation system, etc. Here the proposed method builds a trie structure using the root words from dictionary and tries to find out a potential lemma of a surface word by efficiently searching in the trie. In this present work, our lemmatization system is tested on Bengali.

1 Introduction

Unlike English, the most popular and widely used language in the world, there exist various inflectional and derivational morphological variants of a root word in the premier Indian languages causing a lot of problems for computational processing. In the lexical knowledge-bases like dictionary, WordNet, etc, the entries are usually root words with their morphological and semantic descriptions and therefore, when a surface word is encountered in a raw text, its meaning cannot be obtained unless and until its appropriate root word is determined through lemmatization. Thus, lemmatization is a basic need for any kind of semantic processing for Indian languages. So far, there have been several researches conducted on stemming (Majumder et al., 2007), (Dolamic and Savoy, 2012), (Paik and Parui, 2011), (Paik et al., 2011) and (Ganguly et al., 2012)) in the context of information retrieval, but no good lemmatizer is still available for Indian languages. For Bengali, few works (Loponen et al., 2013 and Sarkar and Bandyopadhyay, 2012) on lemmatization and morphological analysis are found in the literature. (Loponen et al., 2013) proposed a lemmatizer (i.e., GRALE) but no algorithmic details are given in their paper. The work of (Sarkar and Bandyopadhyay, 2012) is based on a set of Bengali-specific rules and therefore, it is not workable for other languages. Our present work, in a little way, similar to the recent work of (Bhattacharyya et al., 2014) as they also built a trie structure consisting of the words in the WordNet of a language and took two searching strategies in the trie, firstly direct searching and secondly back-tracking for finding lemma of a surface word, but we have brought more sophistication in searching process in the trie.

This article is designed as follows. Section 2 gives the details of our proposed method and in section 3, we report the result of our system for Bengali in the MET-FIRE-2014. Section 4 discusses the limitations of our method and concludes the paper.
2 The Proposed Lemmatization Algorithm

Our algorithm requires a dictionary of the language concerned. For Bengali, we have used the dictionary available with the University of Chicago and processed it. There are 47,189 distinct root words in the dictionary.

At first, we have created a trie structure using the dictionary root words. Each node in the trie corresponds to an unicode Bengali character and the nodes that end with the final character of any root word are marked as final nodes. The rest nodes are marked as non-final nodes.

To find the lemma of a surface word, the trie is navigated starting from the initial node in the trie and navigation ends when either the word is completely found in the trie or after some portion of the word there is no path present in the trie to navigate. While navigating, some situations may occur, depending on which we take decision to determine the lemma. Those situations are described below.

1. The surface word is a root word. In that case, the surface word itself is the lemma.
2. The surface word is not a root word. In that case, the trie is navigated up to that node where the surface word completely ends or there is no path to navigate in the trie. We call this node as the end node. Now again two different cases may occur here.
   (a) In the path from the initial node to the end node, if one or more than one root words are found i.e. if one or more final nodes are present in the path, then pick that final node which is closest to the end node. We measure the distance between two nodes by the number of edges between them. The word represented by the path from initial node to the picked final node is considered as the lemma. For example, consider two dictionary root words ‘অংশ’/‘angsha’ and ‘অংশীদার’/‘angshidar’ and take two inflected words ‘অংশর’/‘angsher’ and ‘অংশীদারর’/‘angshidarer’.

   ‘অংশ’ = ‘অ’ + ‘ং’ + ‘শ’.
   ‘অংশীদার’ = ‘অংশ’ + ‘ী’ + ‘দ’ + ‘া’ + ‘র’.
   ‘অংশর’ = ‘অংশ’ + ‘র’.
   ‘অংশীদারর’ = ‘অংশীদার’ + ‘র’.

   The trie is built using the two root words ‘অংশ’/‘angsha’ and ‘অংশীদার’/‘angshidar’. For ‘অংশর’/‘angsher’, the end node is ‘শ’/‘sh’ node and for the word ‘অংশীদারর’/‘angshidarer’, the first ‘র’/‘r’ node from left is the end node. For ‘অংশর’/‘angsher’, ‘শ’/‘sh’ node is the only final node present between the initial node and the end node and hence ‘অংশ’/‘angsha’ is taken as the lemma. In the case of ‘অংশীদারর’/‘angshidarer’, ‘শ’/‘sh’ and ‘র’/‘r’ are two final nodes present between the initial node and the end node. As ‘র’/‘r’ node is closer to the end node than ‘শ’/‘sh’ node, so we take ‘অংশীদার’/‘angshidar’ as the lemma for ‘অংশীদারর’/‘angshidarer.

   (b) If no root word is found in the path from the initial node to the end node, then find the final node in the trie which is closest to the end node. If more than one final nodes are found at the closest distance then pick all of them. Now, generate the root word(s) which is/are represented by the path from initial node to those picked final node(s). Finally among the generated root word(s), pick

\[\text{http://dsal.uchicago.edu/dictionaries/list.html}\]
Table 1: Results of our lemmatization system on Bengali in MET FIRE 2014

<table>
<thead>
<tr>
<th>TOTAL. Precision</th>
<th>TOTAL. Recall</th>
<th>TOTAL. F1-measure:</th>
</tr>
</thead>
<tbody>
<tr>
<td>56.19%</td>
<td>65.08%</td>
<td>60.31%</td>
</tr>
</tbody>
</table>

the root word(s) which has/have maximum overlapping prefix length with the surface word. By the phrase ‘overlapping prefix length’ between two words, we mean the length of the longest common prefix between them. Even at this stage if more than one roots are selected, then select any one of them arbitrarily as the lemma as it is very rare to have more than one root words in this stage and if more than one root exist, then all are viable candidates. For example, consider the dictionary root words ‘শুনা’/‘shuna’, ‘শুনানি’/‘shunani’ and ‘শুনানো’/‘shunano’ and took an inflected word ‘শুনে’/‘shune’.

\[
\begin{align*}
\text{শুনা} & = \text{শ} + \text{ু} + \text{ন} + \text{া} \\
\text{শুনানি} & = \text{শুনা} + \text{ন} + \text{ি}
\end{align*}
\]

\[
\begin{align*}
\text{শুনানো} & = \text{শুনা} + \text{ন} + \text{ো} \\
\text{শুনে} & = \text{শ} + \text{ু} + \text{ন} + \text{ে}
\end{align*}
\]

The trie is built using the three root words ‘শুনা’/‘shuna’, ‘শুনানি’/‘shunani’ and ‘শুনানো’/‘shunano’. For the inflected word ‘শুনে’/‘shune’, the first ‘ন’/‘n’ node from the initial node in the trie is the end node. From the end node, ‘ে’/‘e’ is the closest final node in the trie and hence the corresponding root word, ‘শুনা’/‘shuna’, is considered as the lemma of ‘শুনে’/‘shune’.

### 3 Experimental Results

Based on the evaluation of the Morpheme Extraction Task - FIRE 2014, the results obtained on Bengali data using our lemmatization system are given in the Table 1. The process of linguistic evaluation goes as follows. Word pairs with same roots are compared and scores are calculated on the number of matched thus obtained. For obtaining precision, a set of 1000 words are sampled from the result files generated by the lemmatization system. For each of the sampled words, another word having same root is chosen from the result file and these pairs are compared to the gold standard data. A point is giving for every word pair that has common root in the gold standard as well. The number of points for each word is normalized to 1. Now, precision is calculated as the ratio of total number of points obtained to the total number of sampled words. Similarly for calculating recall, a set of 1000 words are sampled, but this time from the gold standard data. For each of these words, another word having the same morpheme is chosen from the gold standard data randomly. The word pairs are then compared to the analyses in the results generated by our lemmatization systems. A point is given to each sampled word pair having common morpheme. This process is carried out several times and the average values of precision and recall are taken. The final score for the system in this evaluation is taken as the F1-measure i.e., the harmonic mean of precision and recall.

### 4 Conclusion and Future Work

We have designed our lemmatization algorithm in such a way so that it does not depend much on language-specific features. Inspite of that, it has some limitations which are as follows. Compound words and out-of-vocabulary words are not considered in our algorithm. Root words are taken from dictionary but if the coverage of the dictionary
used is not good, then that will cause errors. However, as there is no such good language
independent lemmatizer for Indian languages, we hope our effort is a positive contribution
and we will try to overcome the drawbacks in future.

References


