Chapter 4

4 STEMMING

Stemming is the term used as a tool for information retrieval and linguistic morphological analysis to describe the process for reducing inflected (or sometimes derived) words to their word stem, based on root form generally a written word form. The stem sometimes don't matches to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Computer Science department is studding different algorithms of stemming since the 1960s. Conflation is a process of treating words with the same stem as synonyms as a kind of query expansion. Many search engines use this conflation.

4.1 Types of Stemming

There are several types of stemming algorithms [Kashif Riaz 2010] which are mentioned below; each of these groups has a typical way of finding the stems of the word variants.

Brute Force Algorithms:

In Brute force stemmers, a lexicon which contains a relation between root form and inflected form. To stem a word, the lexicon is queried to find a matching inflection. If a matching inflection is found, the associated root form is returned.

Suffix Stripping Algorithms:

Suffix stripping algorithms based on a typically small list of "rules" stored in an array or a lexicon. This list provides a path for the algorithm to find out the root form of a given input word.

Lemmatization Algorithms:

This process involves first determining the part of speech category (e.g.: noun, verb, adjective, adverb etc) of the input word, and applying different normalization rules for each part of speech category. The part of speech

is first detected and finally the stemming rules change depending on word's part of speech category.

Stochastic Algorithms:

In this method machine first trains the inflected word along with root word and produces some probabilistic internal rule set. Based on these internal rules machine finds out the most probable root word from a given input word and most of the cases it removes the affixes from the word.

Hybrid Approaches:

As the name clearly suggests that these methods are combination of two or more of the approaches described above. A simple example is a suffix tree algorithm which first consults a lookup table using brute force.

Affix Stemmers:

In linguistics concern the term affix refers to either prefix or suffix. In addition to dealing with only suffixes, several approaches are there which discussed in literature survey. If many of the approaches mentioned above can strip prefix as well as suffix then they are called affix stemmer.

Matching Algorithms:

These types algorithm use a stem database (for example a set of documents that contain stem words). These stems are not necessarily valid words themselves (but rather common sub-strings). In order to stem a word the algorithm tries to match it with stems from the database, applying various constraints, such as on the relative length of the candidate stem within the word.

4.2 System Description

For explaining the proposed system three lexicons are taken into consideration: prefix, suffix, root lexicon and a set of hand written rules for prefix and suffix stripping, especially for inflected and derived word. Lexicon means

dictionary, where various entries of words of any language are included. Words in the lexicons are manually created.

Suffix Lexicon

Suffix means those words that come after the root or stem. Approximately 120 suffixes are taken into consideration.

Prefix Lexicon

Prefix means those words which are added to the front of the word. Around 25 words are taken to build prefix lexicon.

Root Lexicon

Root lexicon contains over 1000 words.

4.3 Proposed Algorithm for STEMMER

Since domains are taken from Nepali corpus so obviously these texts will have sufficient number of Nepali punctuations, Nepali and English digits and Single-letter-words. But while developing a Stemmer we need not take these unnecessary characters into consideration .So before doing actual stemming it will be efficient to remove these characters step by step and this process is called cleaning. This will be the pre-processing steps of a good Stemmer.

Tokenization

Tokenization is the process of breaking the sentences as well as the text file into word delimited by white space or tab or new line etc. Outcome of this tokenization phase is a set of word delimited by new line.

Punctuation Removal

A document may contain lots of Nepali punctuations in the text. These characters have no importance for stemming.

Digit Removal

In general Nepali text file may contain Nepali as well as English digits. But as meaningful Nepali words do not contain digits.

Single Letter-Word Removal

There exist a lot of words having a single letter. Most of these Single-Letter-Words are Stop-Words (those words which have extremely high term frequency in a corpus are known as Stop Words). As a step of stemming the stop-words need to be removed before further processing. So the Single-Letter-Words are removed in this phase.

For above all pre-processing steps list of digits both English and Nepali, list of stop words and list punctuations are maintained in a text file.

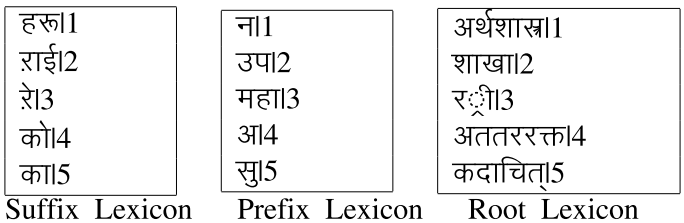
Lexical Look up Approach

After processing all the above steps, words (tokens) are ready for stemming. The stemming system was developed using lexical look up based approach using three different lexicons suffix, prefix and root. The system mainly works in two steps:

Firstly the input word is queried in the root lexicon; if it is found, then it is considered as a root word e.g. "मामा". Secondly if it fails then the system queried into prefix and suffix lexicon for affixes, if it is present then its rule number is retrieved and do affix striping according to rule which were written in the lexicon against retrieving rule number e.g. in the word "विदेशी", one prefix "वि" and one suffix "ी"is present. Rule for "वि" prefix is strip off "वि" from the input word "विदेशी" and keep it same but for "ी" suffix, strip off "ी" from the input word "विदेशी" replace "ी" by "ई". If the root word is found after stripping suffix/ prefix or both then the system will store root word along with its constituents parts i.e. prefix and suffix into output file. If one or more prefix/suffix present and the root word is not found then try combining the suffix/prefix one by one with the remaining part of the word and again search in the root lexicon for the root. This process will be continuing until it finds the root word or its prefix/suffix lists are empty otherwise the system will keep the word

as it is and store the word into output file.

Lexicons are in column format, i.e. a word per line in a sentence by sentence fashion along with serial numbers; a separator "|" is used to separate the word and number. Following are samples of prefix, suffix and the root lexicon:



Suffix Lexicon Prefix Lexicon Root Lexicon

All the root, suffix and prefix lexicons are stored into hash table with two field's key and value. The key of the hash function was generated from the summation of the ASCII values of each character of every suffix, prefix and root. Using this key, probable position was generated to store each suffix, prefix and root. If there was a previously stored suffix, prefix and root in that position, then collision occurred. In case of collision the next free space was searched and the suffix, prefix and root, were stored in that position. We found that the collision is minimum here [M. N. Karthik, Moshe Davis 2004].

The architecture of the core engine of the stemmer is presented below:

Table 6: Test Cases

Test No	Domain	No.of Words
1	Economics	400
2	Health	600
3	political	800

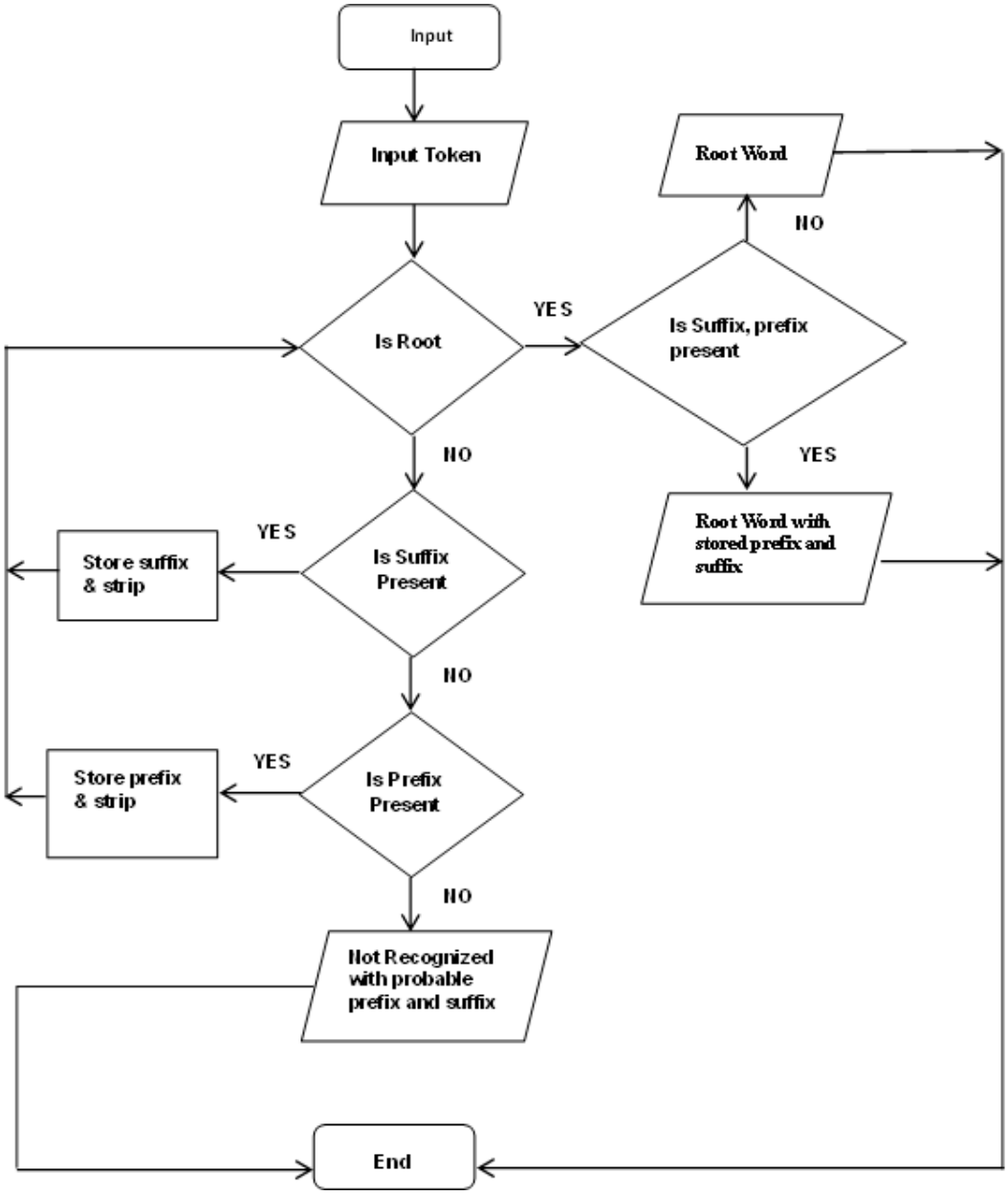


Fig 2: Architecture of Stemmer

4.4 Experimental Results and Discussion

System performance was evaluated based on different domains of news. These domains include news on Economics, Health and Political. The system was evaluated on 1,800 words. The overall accuracy achieved by the system is 90.48%. Three test data sets were taken from original corpus for testing, which was built by Technology Development for Indian Languages (TDIL) [Kashif Riaz 2010]. The following table shows the different test cases for testing:

The evaluation metrics for the data set is precision, recall and F-Measure. These are defined as following:-

Recall (R) = Total number of words stemmed by the system / Total number of words.

Precision (P) = Total number of words correctly stemmed by the system / Total number of words.

$$F - Measure = (\beta^2 + 1)PR/(\beta^2 R + P)$$

Where β is the weighting between precision and recall and typically $\beta = 1$.

Table 7: Accuracy of System on different Test Cases

Data Set No.	Recall	Precision	F-Measure
Data Set-1	92.35%	89.95%	91.13%
Data Set-2	91.56%	88.80%	89.82%
Data Set-3	91.78%	89.26%	90.50%

Accuracy measurement of stemmer

During stemming process one text file is generated for each test case: stem.txt. If the system stems the word then it will store its root part into stem.txt otherwise store the word as it is and print them in the

stem.txt. From this file we can get total number of stem as well as unknown words. Also in case of getting total number correct root word, one small matcher program is written which sequentially reads stem.txt and root lexicon. The correct root words are those which match in both the files and the remaining words are unknown words. The program will match word by word and increment the count if the word matched in both the files.

The overall accuracy is measured by calculating the mean of three F-measure values. F-measure which combines the precision and recall to give a single score. It is defined to be the harmonic mean of the precision and recall.