Chapter 5

PROPOSED POS TAGGER

5.1 POS Tagger Architecture

The proposed method is based on hybrid approach; it combines the Rule-Based method presented with HMM probabilistic Techniques . The hybrid tagger for Nepali runs on three phases to POS tag input text. In the first phase, thetagger tokenizes the input text into independent words, and if a word is found in the lexicon, then the entered word itself will be the output. Because the lexicon contains only the morpheme(root word) and the valid word which has its own POS category. In the second phase the tagger assigns the POS category of each token with the help of HMM probabilistic techniques. Almost all the words get recognized by these two phases.However, some tokenswhich are stillconsidererd not analyzedwords (unknown) or disambiguate words will be analyzed in the third phase with the help of Rules of POS tagging.

5.2 Description of the proposed architecture

The different modules involved in this architecture are explained as follows:

5.2.1 The Proposed Design for Nepali Part of Speech Tagging:

There are a many approaches to implement part of speech tagger, i.e. Rule-Based approach, Statistical approach and Hybrid approach. In this project, We developed an effective POS tagger for Nepali language using hybrid based approach, .i.e. Hidden Markov Model (HMM) integrated with Rule-Based method.Our proposed POS tagging is

implemented by undergoing several distinct steps and the overall architecture of the system including the connections between the modules, flow diagram and its related explanation are given below:

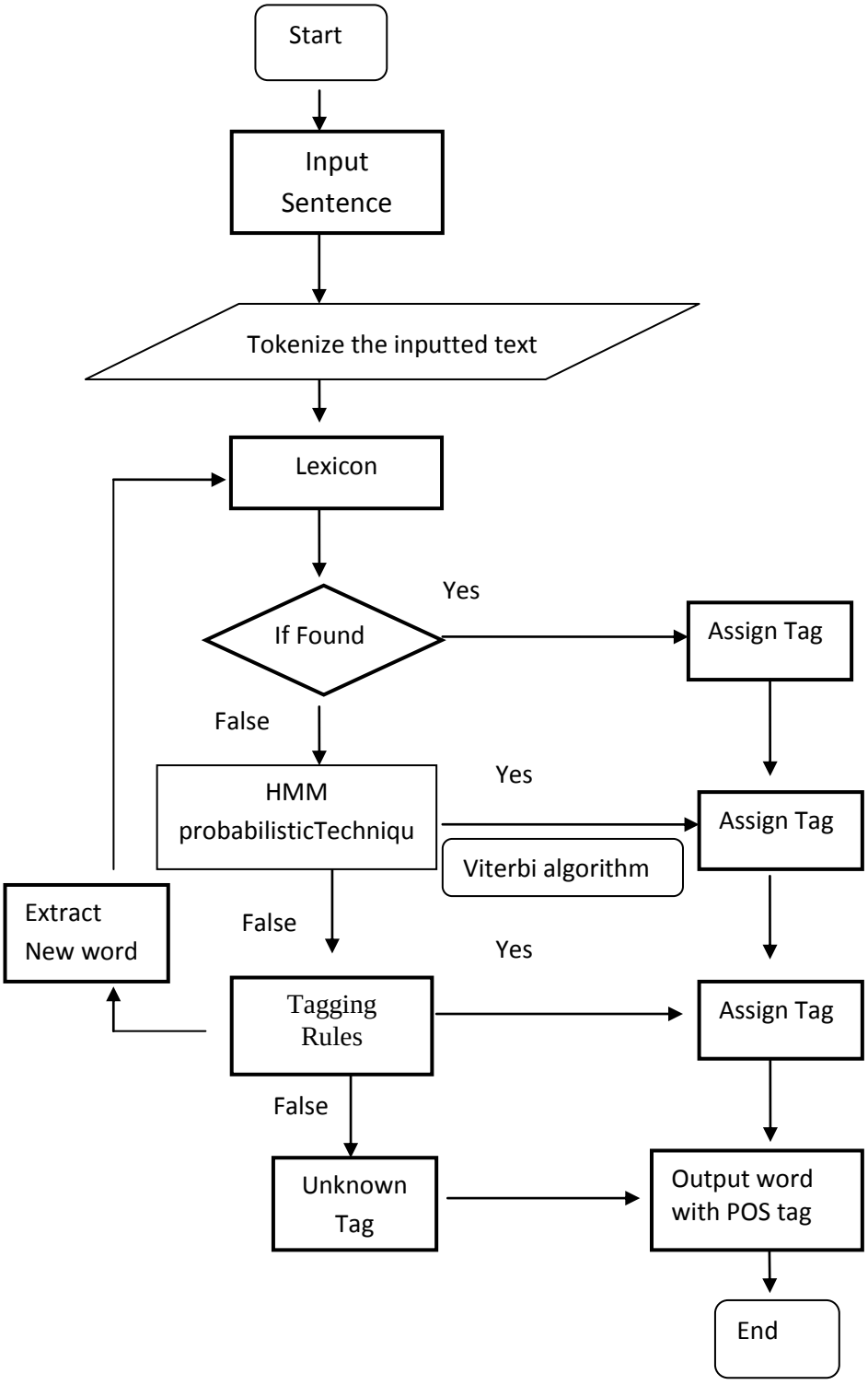

Fig. 5.1 Proposed System Design of Nepali POS Tagger

1. **Tokenization:** At first, we take a word from a sentence and tokenise or segregates words, punctuation marks and symbols of an input text, and subsequently assigns them into tokens by creating whitespaces between them.

2. **Lexicon**: In this step, all the Nepali words are defined in the lexicon. This lexicon includes all categories of POS tagging viz., Noun, Pronoun, prepositions, adverbs,Verb,conjunctions, interrogative particles, etc. This lexicon is developed manually by collecting limited words from Nepali books,newspapers, and dictionaries.In this phase, word is search in lexicon and if any word is found in the lexicon, then the entered word will be tagged or assigned with an appropriate tag. Else, it passes to the next step i.e Hidden Markov Model (HMM) probabilistic Techniques

3. **Hidden Markov Model (HMM) probabilistic Techniques:** The basic principal of HMM is probability and it is used to create tag sequences[73]. Basic idea of Hidden Markov Model is to determine or pick the most likely tag for a word in a sequence[10]. For to this purpose we have to calculate Transition probability.The Transition probability is generally calculated based on previous tags and future tags with the sequence provided as an input. The following equation (1) explains this phenomenon

$$P(ti/wi) = P(ti/ti-1).\ P(ti+1/ti).\ P(wi/ti) \quad \text{..... (1)}$$

Here, P (ti/ti-1) is the probability of current tag given previous tag and P (ti+1/ti) is the probability of future tag given current tag. P (wi/ti) is the Probability of word given current tag

It is calculated as-

P (wi/ti) = freq (ti, wi)/ freq (ti)…..... (2)

This is done because we know that it is more likely for some tags to precede the other tags. In Hidden MarkovModel, we consider the context of tags with respect to the current tag. HMM assigns the best tag to a word in a sentence by calculating the forward and backward probabilities of tags along with the sequence provided as an input. Powerful feature of HMM is context description which can decide the tag for a word by looking at the tag of the previous word and the tag of the future word.

Viterbi algorithm:It is a dynamic alogorithm and for finding the maximum probability HMM use the Viterbi Algorithm.Viterbi algorithm searches for the best tag for each word in order to find the best tag sequence W = (w1,w2,…,wn) based on the text corpus.

Let us consider this computation using the example "उसकी हृदयसुनजस्ते छ". In this Nepali sentence "सुन " is an ambiguous word which can either have an NN (Noun) or a VMF(verb) tag dependency of the possible tags.

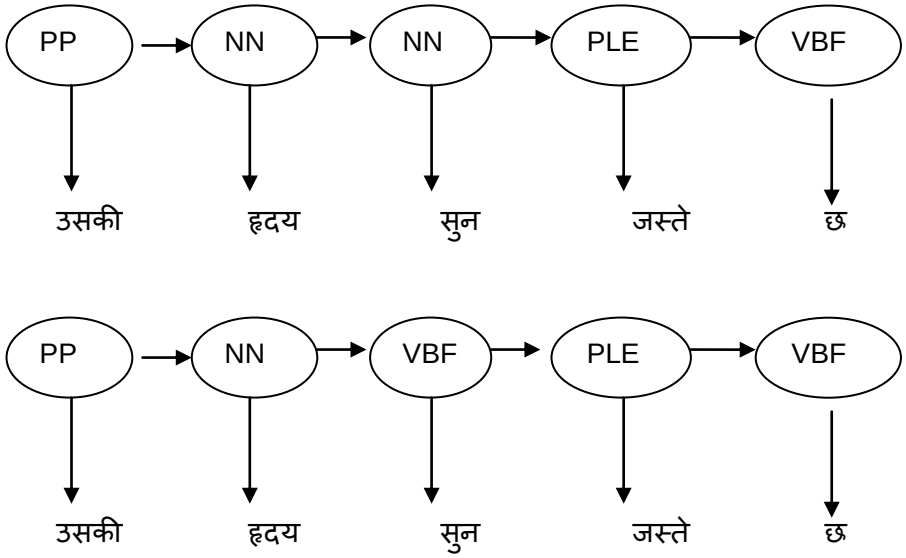

Fig 5.2 Tag transition probabilities

Since all the other word-tag combinations are same for the ambiguous words, their computation will also be the same. It is the ambiguous tag context which will make the difference to the final score. The one which is higher will get selected.

If the word is a noun, then the possibility of the next word to be noun is

P (NN|हृदय) = O.0712 (Data are taken from the Transition Probability Matrix table)

And the possibility of the next word to be verb is

P(VBF|हृदय) = 0.4716

The two possible tags VBF and NN, these correspond to the probabilities

*P (*सुन *|NN) and P(*सुन *|VBF)*

*P (*सुन *|NN) = .0.182*

*P(*सुन *|VBF) = 0.182*

We need to represent the tag sequence probability for the tag PLE for जस्ते and these are

P(PLE|VBF) = 0.5227

P(PLE|NN) = 0.7011

Finally *P(VBF |हृदय) X P(PLE|VBF)X P(*सुन *|VB) = 0.044863*

*P(NN|हृदय)XP(PLE|NN) X P(*सुन *|NN) = . 0.009085*

As the computation of VMF is higher than that of NN, VMF gets selected for "सुन".

Once all the tags are identified for the input words, they are displayed to the user.

4. **Rule based tagging:** Almost all words are recognized by previous two phases. However, some terms which are not analyzed(unknown) words or disambiguate words will be analyzed by third phase with the help of Rules of POS tagging. The tagger selects the proper tag by using the grammatical and morphological rules. We have applied different types of rules for Nepali in implementing this rule-based tagger. Two types of rules are very important for rule based tagger and they are lexical and contextual rules. Lexical rules are used for assigning an initial tag to unknown words, and context rules, for correcting tags on the basis of context.

   i) Lexical rules: These are the rules for generating lexical entries from base forms of words.A lexical rule is in a form of syntactic rule used within many theories of natural language syntax. These rules alter the argument structures of lexical items (for example verbs and declensions) in order to alter their combinatory properties.The ideal goal of the lexical module is to find rules that can produce the most likely tag for any word in the given language, i.e. the most frequent tag for the word in question considering all texts in that language. The problem is to determine the most likely tags for unknown words, given the most likely tag for each word in a comparatively small set of words. In this module a set of the affixes of each word are extracted. There will be two sets of affixes, one for the suffix and the other for the profix. Each set will contain an exhaustive list of the forms of affixes in the Nepali language. The form of affixes here mean the form in which the affixes are present in the inflected or derived form of the words.

ii) Contextual rules: A contextual rule is used to identify and label portions of text. A portion of text gets labelled if it satisfies a particular condition. The context rule is based on the relation between the untagged words and their adjacent words. In Nepali language words have some relations between current word and adjacent words. For example the preposition and interjections are always followed by nouns and this type of relations may allow tagging the words into its corresponding classes, that is, once the tagger has learned the most likely tag for each word found in the manually annotated training corpus and the method for predicting the most likely tag for unknown words. The contextual rules are learned for disambiguation and learner discovers rules on the basis of the particular environments (or the context).

The rules produced by the contextual learning module together with the lexical rules and several other files can then be used as input to the tagger to tag unannotated text. The set of `templates used and the names of the rule type for each template in parenthesis are given below

- *The preceding/following word is tagged with Z. (pretag/nxttag)*
-
- *One of the two preceding/following words is tagged with Z.(pre1or2/nxt1or2)*
- *One of the three preceding/following words is tagged with Z. (prev1or2or3tag/next1or2or3tag)*
- *The preceding word is tagged with Z and the following word is tagged with V.*

In this system design, different types of rules are applied for implementing this rule-based tagger. Bellow we mention some of these rules that are applied in our system.

Verb Identification Rules: -If current word is not tagged and next word tagged as an auxiliary verb, then there is high probability that current word will be main verb.

For Example: खाने

In above example खाने is verb and छुis auxiliary verb.

Prefix Rule*:* If any word has prefix 'हरु' then there is high probability of that particular word will be noun.

Computational Model of the above rule can be written as:

```
if (word.tag.starts With ("हरु") )
        {
                word.tag.set ("NNP")
        }
```

Single Proper Name Rule*:* If current word is name and next word is surname then we tagged them as single proper name.

For Example: - मनिकादेवी<NNP>Manikadevi

In above example मनिका (Manika) is name and देवी (Devi) is surname.

Compound word Rule: If a compound word combine with two noun words and is separated with '-'then word will be tagged as noun.

e.g आमा-बाबा

5) Extraction of new words: In this architecture, the tagger first tag the words which are in the lexicon and the words which are not available in the lexicon are tagged by HMM probabilistic Techniques and then applying rules. If the HMM probabilistic Techniques and rules fail to tag such words then the new words are given a specific tag as "UNK" i.e.; unknown which can be extracted from the tagged output text. The new words are then entered in to the lexicon and new rules are created for the new words.

6. Output: Finally the tagger display the tag output and save tag structure into HTML file.

5.3 The Proposed Algorithm

Algorithm used for this tagging is as follows:

Step 1: Enter the Nepali Input text.

Step 2: Tokenise or segregates words of input Nepali text.

Step 3: Search for the words in lexicon in lexicon for a match.

Step 4: If the word is found in the lexicon then the assigned the word.

Step 5: If the word is not singular form or if word is in the form of affixation, derivation and compounding then segregates the word

into independent words and checks the word with the lexicon for a match.

Step 6: After segregating, search for token in lexicon and if the word is found in the lexicon then word will be the output as appropriate tag.

Step 7: If not found in lexicon, or multiple tags exists for a single word, mark those tokens. Find the POS Category of tokens by using HMM probabilistic Techniques.

Step 8: Identify and extract those terms which has been misclassified or unanalyzed during Step 7.

Step 9: Apply Part Of Speech rules for those words and returned the tagged output text by using rules.

Step 10: Make the new entry for the unknown /new word to the lexicon

Step 11: Repeat step 5 and 10 till the end of the input text.

Step 12: Display all the tag output.

5.4 Website & GUI Tool

5.4.1 Website

A Web site is a related collection of World Wide Web (WWW) files that includes a beginning file called a home page[69]. A company or an individual tells you how to get to their Web site by giving you the address of their home page and from the home page we can get to all the information about the page. The Nepali Part-of-Speech tagging work in

Nepali language has been highlighted by researcher on the website created by him **www.researchnlp.com.**

Figure 5.3 Nepali POS Tagger Domain

5.5.2 Graphical User Interface (GUI) Tool

GUI part of speech tagger named "NEPOST" has been developed by using PHP. The front-end of the tool has been implemented in PHP and its interface is connected with a text file of Nepali lexical items called "lexicon" as the back-end. The selection of textual database is for simplicity and to extend support for multiple platforms without the need of

the installation of any DBMS server like MYSQL etc. by the end user. Each lexical item entry in “lexicon” file has two fields: ITEM: CATEGORY:

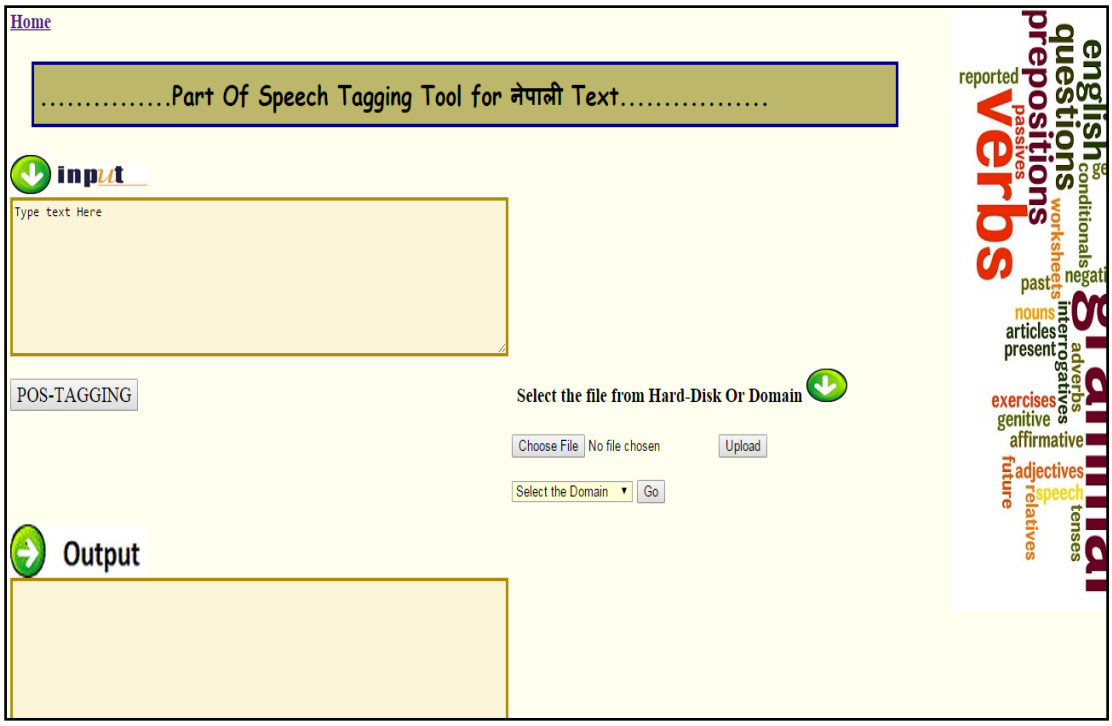

Figure 5.4 Nepali POS Tagging tool

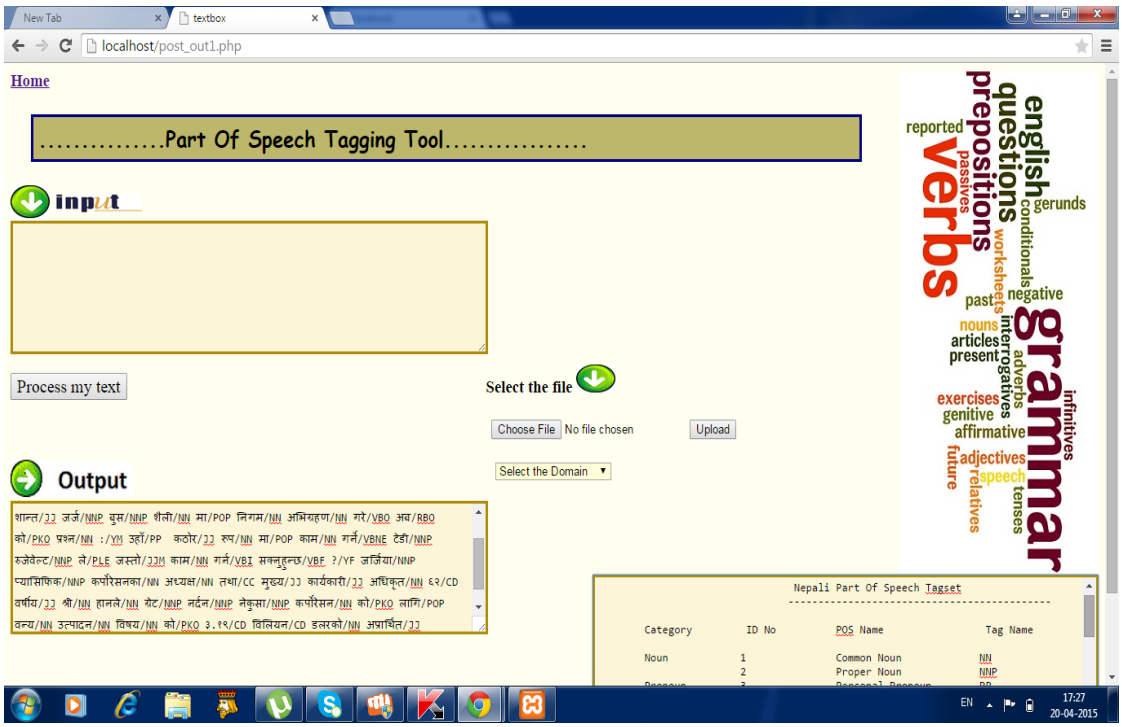

Figure 5.5 Online Nepali POS Tagging Tool