

A Morphological Analyzer and a stemmer for Nepali

Bal Krishna Bal, Prajol Shrestha

bal@mpp.org.np, prajol@mpp.org.np

Madan Puraskar Pustakalaya, Nepal

Abstract

This paper discusses the design and implementation issues as well as the linguistic aspects of the Morphological Analyzer and a stemmer for Nepali.

1. Background

Natural language Processing(NLP) is a new activity for research and development in Nepal. Precisely speaking, in a large scale, it started just in the year 2005 with the release of the first Spell Checker for Nepali and the "Dobhase" English to Nepali machine translation project, respectively developed by Madan Puraskar Pustakalaya (MPP, <http://www.mpp.org.np>) and with the Kathmandu University(<http://ku.edu.np>) in collaboration. In the same year, further works on language engineering like corpus building and annotation for Nepali, Text-To-Speech System for Nepali, digitized Nepali dictionary also got started under the NeLRaLEC (Nepali Language Resources and Localization for Education and Communication) Project , also known as the Bhasha Sanchar Project (<http://www.bhashasanchar.org>) and currently being run at Madan Puraskar Pustakalaya, Nepal . No doubt , these works proved to be substantial in triggering the awareness and interest for further research and development of Natural Language Processing, NLP tools, however , a full-fledged focus on the NLP development had not been able to get realized due to the domain-specific requirements of the above mentioned activities. It was on the light of the Nepali Grammar Checker development, an activity under the PAN Localization Project(<http://www.PAN10n.net>) that the necessity of an adequate research and development of NLP was felt. This led to the study of the structure of Nepali Grammar and Language as well as the components required for a Grammar Checker. The findings of the study revealed that research and development of the NLP tools and resources in Nepal, for instance, a machine-readable and a multi-purpose Nepali lexicon, morphological analyzer, Parts-Of-

Speech, POS tagger, chunker, parser etc., are primarily in the infant stage and that the development of a Nepali Grammar Checker would necessarily require the research and development of the intermediate NLP components, some or all of these right from scratch.

With the above mentioned NLP tools developed, further doors of possibilities would open up for the research and development of several useful Natural Language Processing Applications like Machine Translation systems (currently we just have a uni-directional one. from English to Nepali), Question Answering systems, Grammar Checker for Nepali, Information Retrieval Systems, Expert Systems and so on and so forth. Given the socio-economic realities and constraints of Nepal and the Nepalese, such a substantial growth in the focus towards the research and development of NLP applications is certain to bring about some positive impacts, some of which include a concrete contribution towards bridging the existing digital-divide and the development of the expertise in local language computing.

2. Introduction

A Morphological Analyser, MA, is a program or algorithm which determines the morpheme(s)¹ of a given inflected or derived word form including the analysis of the bound morphemes in its grammatical form. For the MA to function, a stemmer should be priorly present in the system. Hence, wherever required, we would also be talking about the stemmer throughout the document. Stemmers are used to find the root of the inflected or derived word form. MA uses the information of the stemmer and keeps track of the bound morpheme(s) present in the original inflected or derived word and in addition to this

1 Morpheme is the smallest linguistic piece of a word with a grammatical function[10] . We could divide morphemes into two parts, one being the base morpheme which can standalone and the other which attaches to the base morpheme called bound morpheme like affixes, infixes, circumfixes.

provides grammatical information. MA are generally used for information retrieval and other Natural Language Processing Applications like the Parts-Of-Speech (POS) Tagging, Machine Translation etc. MA have been designed and implemented for several languages of the world. Complexities in design and development, however, prevail for inflectionally and derivationally rich languages like Nepali.

The stemmer and consequently the MA being developed for Nepali currently does not handle compound words formed as a result of the concatenation of individual words.

3. Available stemming algorithms

Several stemming algorithms are available for the English language. Given below is a list of some of the popular stemming algorithms:

- 1) Krovetz Stemming Algorithm(1993);
- 2) Paice/Husk Stemming Algorithm(1990);
- 3) Porter Stemming Algorithm(1980);
- 4) Dawson Stemming Algorithm(1974);
- 5) Lovins Stemming Algorithm(1968);

All of the above mentioned stemmers are suffix removal stemmers and hence do not deal with the prefixes in a given word. Iteration and context awareness is very crucial in the stemmer development. These basically relate to looking for more than one prefixes and suffixes in a word and consequently producing a valid free morpheme out of the input word. Stemmers are further classified into two major types, respectively the context-sensitive and the context-free types. The context-sensitive prevents the production of insensible and invalid roots whereas the context-free may end up into some invalid and insensible roots or free morphemes.

The above algorithms are for the English language and cannot be applied to the Nepali Language. For each language there will be different stemming algorithm, hence a unique morphological analyser. Currently, there is not any algorithm available to stem a Nepali word. The stemmer and the morphological analyzer that we are currently developing is a data driven one with a core engine, set of rules for stemming, a free morpheme list and an affix list.

4. Prerequisites of the Stemmer

The prerequisites of the Stemmer Module would be the following:

- i) POS Tagset;
- ii) Tokenizer;
- iii) Free morpheme based lexicon;
- iv) Two set of affixes each for the suffix and prefix;
- v) A Database of word breaking Grammatical rules.

5. POS Tagset

The NLP team at Madan Puraskar Pustakalaya, Nepal has developed a relatively simplified POS Tagset of 91 tags. The tagset has been developed targeting the Grammar Checker for Nepali. The tagset development guidelines and experiences of Hindi, Urdu and some others like the British National Corpus have been consulted in developing the current POS Tagset. The POS tag coverage test of the developed POS TagSet is currently being tested. For the testing purpose, the entries in the free morpheme list and the affixes list are being POS tagged.

6. Tokenizer

The tokenizer takes an input text from a file and creates tokens². The tokens are stored in a XML format for better representation of each element. The tokenizer separates sentences, words, numbers and punctuations in the given text. The MA takes each token and processes it and annotates each element in the XML format with it's morphemes and grammatical categories. The different tags used for the XML representation of the tokens are as given below:

<text> for text
<s> for sentence
<w> for word
<n> for number
<sp> for space
<p> for punctuation
<nl> for new line

2 Token here is indicating to the useful information that we could use later for further processing. We have considered that we would take sentences, words, numbers, punctuations, spaces, newlines as tokens.

Free morpheme based lexicon

The morpheme based lexicon would have the free or unbound morphemes of the Nepali language. Apart from placing the free morphemes (root form of words) in the lexicon, they would be assigned their respective syntactic categories or parts of speech with the pipe sign used as a delimiting symbol.

Given below is a possible list of entries for the free morpheme based lexicon.

कलम|NN

खा|VV

मीठो|ADQ

म|PFS

Abbreviations:

NN – Common Noun

VV – Verb Base Form

ADQ – Adjective Qualitative

PFS– Pronoun First Person

6.1. Set of affixes

There will be two sets of affixes, one for the prefix and the other for the suffix. 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. For example, if a word 'सुत्' : 'sut' is a root and it is combined with the suffix 'एको' : 'eko' the resulting form becomes 'सुतेको' : 'suteko'. The suffix 'एको' : 'eko' changes its form to become 'ेको' : 'eko'. Here 'ए' is a vowel and 'े' is a vowel symbol of 'ए'. So the set of suffixes will contain 'ेको'. These affixes represent the category of bound morphemes and indicate certain syntactic category when combined with suitable free morphemes or roots. This is most common with verbs. However, not all bound morphemes essentially may be assigned the syntactic categories.

Given below is a list of entries in the set of affixes.

Prefix set (file):

उप|10

नर|11

सु|12

Suffix set (file):

ेको|1

नु|2

दै|3

ई|4

एको|5

ले|6

लाई|7

शील|8

हरू|9

The number that is present after the suffix and the prefix, delimited by the '|' pipe sign is to point to the rule present in the word breaking grammatical rules file.

6.2. Grammatical rules for word breaking

Nepali compound words forming as a result of the combination of the free and bound morphemes are not always regular in terms of formation and consequently in breaking. Insertion and deletion of one or more free vowels and vowel symbols or dependent vowels is a common phenomenon.

Below, we try to illustrate some of the instances of the above operations in word formation.

सु+आगत=स्वागत

In the above, सु and the character आ combine to form स्वा and in doing so the vowel sign ु is deleted, the consonant स is reduced to स् or half character and the vowel आ is substituted by the cluster वा.

Some more examples:

Irregular word breaking:

खवाइ = खा+आइ

भनाइ = भन्+आइ

गराइ= गर्+आइ

Regular word breaking:

शहरीया = शहर+ईया

उपरथी = उप+रथ+ई

विदेशी = वि+देश+ई

These word formation patterns should be noted as rules, which need to be taken into consideration while breaking the words into the respective morphemes.

6.3. Database of the word breaking rules

The insertion and deletion rules mentioned above and which are associated with the prefixes and the suffixes may be formulated as follows:

String sequence in the input word	Position of the string sequence in the input word	Associated affix	Type of affix	Action required
1) ी	At the end of the word, the very last letter.	-ई	regular	1) Strip off ी from the input word. Record ई as the suffix associated. For example, (रथी=रथ+ई, शहरी=शहर+ई)
2) ीय/ीया	At the end of the word from the end.	-इय/-इया	regular	2) Strip off ीय/ीया from the input word. Record इय/इया as the suffix associated. For example,

String sequence in the input word	Position of the string sequence in the input word	Associated affix	Type of affix	Action required
				उदाहरणीय=उदाहरण+इय शहरीया=शहर+इया
3) ाइ	At the end of the word from the end.	-आइ	irregular	1) Strip off ाइ from the input word. Add ् to the end of the resulting word if the last letter of the word formed is a consonant. Record आइ as the suffix associated. For example, गराइ=गर्+आइ 2) Exception holds the letter व. If the the last character of the resulting word is व, strip it off and add ा . Record आइ as the suffix associated. For example, खवाइ=खा+आइ 3) If the

String sequence in the input word	Position of the string sequence in the input word	Associated affix	Type of affix	Action required

String sequence in the input word	Position of the string sequence in the input word	Associated affix	Type of affix	Action required
				insert ा in front of the character which is followed by ाइ अँकाइ=आँक्+ आइ
4) े	At the end of the word.	-ए	regular	1) Strip off े from the end of the word. Look for the resulting word in the free morpheme list. If found record े as the suffix. अँगारे=अँगार +े विकासे=विका स+े 2) Exceptions hold the following: काले=कालो+ ए तीते=तीतो+ए

The rules are placed in two separate files one for the prefix and the other for the suffix. This allows better handling of affixes. As we already know that the removal of affixes for irregular pattern is difficult, we

have tried to firstly implement simple rules for regular patterns. To be able to formulate the rules and apply them, the irregular affixes will have to be studied in more detail. For regular affixes, we have formulated the rule with a header line which includes the rule number for indexing, the type of affix, the number of sub rules, the morpheme as affix, and the respective grammatical category it represents. All of these fields are space delimited. After the header line, sub rules are present. A snippet of the rule file is given below:

suffix rule file:

1 SFX 1 हरू HRU

हरू .

2 SFX 1 ले PLE

ले .

Abbreviations:

SFX - Suffix

PLE – Ergative

HRU – Plural Marker

The sub rules are simple. The first field indicates what is to be deleted and the second field indicates what is to be appended. The two fields are delimited by space.

7. Morphological Analyser

As we have already mentioned that for the morphological analyzer a stemmer should be priorly present. The MA that we are currently developing uses the core engine of the stemmer and adds few functionality to it so that it not only strips off the morphemes but also keeps track of which bound morphemes are present and which grammatical category they belong to. The grammatical category are present in the rule files as mentioned above. The work flow of the core engine of the MA is presented in the flowchart given below:

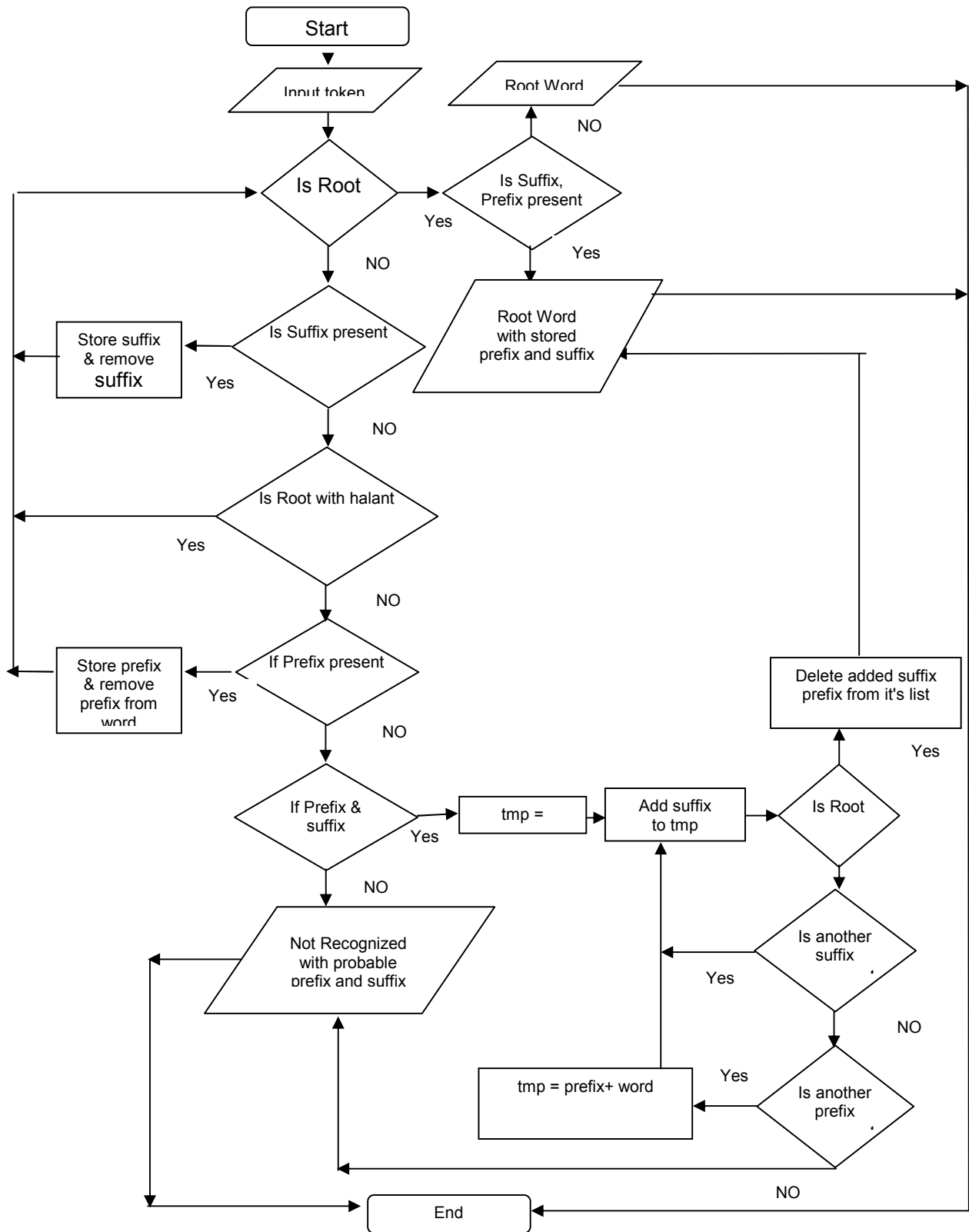


Fig.1. Flowchart of the MA and Stemmer

8. Output of the Nepali Morphological Analyser

The Nepali MA has three outputs.

- 1) Input word is a root, POS information of the root word.
- 2) Root form of the word and the affixes attached, POS information of the root word and the affixes.
- 3) The root of the input word not found by the system, affixes present, POS information of the affixes found.
- 4) The input word is not recognized by the system.

9. Cases

- 1) The input is a free morpheme, for eg., कलम(root, NN);
- 2) The input word consists of free morpheme and one or more suffixes, for eg., शहरीया=शहर(root,NN)+इया(SUFF);
- 3) The input word consists of a free morpheme and one or more prefixes, for eg., उपशीर्षक=उप(PREF)+शीर्षक(root, NN);
- 4) The input word consists of a free morpheme and one or more prefixes and one or more suffixes, for eg., विदेशी=वि(PREF)+देश(root,NN)+ई(SUFF);
- 5) The input word is not recognized by the MA but consists of morphemes, for eg. ककहरू=कक(not recognized) + हरू(HRU).

The output will be as follows with the respective input:
input:

उपअधपक्काहरू उपनगरपालिकाहरू।

output:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<text>
```

```
<s>
```

```
<w
```

```
morph="उपअधपक्का(root)+हरू(HRU)">उपअधपक्काह
```

```
रू</w>
```

```
<sp> </sp>
```

```
<w
```

```
morph="उप(PREF)+नगर(root)+पालिका(SUFF)+हरू(HRU)">उपनगरपालिकाहरू</w>
```

```
<p>|</p>
```

```
</text>
```

Abbreviations:

SUFF - suffix

PREF – prefix

HRU - plural marker

10. Conclusion

The system described is a simplified version of the stemmer and a morphological analyzer for Nepali. Currently, the system does not handle words formed as a result of the combination of two free morphemes. The database of word breaking rules is also of negligible size right now. The current algorithm of the core engine of the system also requires further optimization. However, the first prototype of the MA has been developed and ready for preliminary testing.

Several technical and linguistic challenges are but natural to be posed during the development of the system owing to the rich morphology of the Nepali language. We expect to add features like complexity handling in due course of the research and development.

11. References

[1] Jurafsky, D. and Martin, J. *Speech and Language Processing, An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, University of Colorado Boulder Fifth Indian Reprint, 2005

[2] Kinoshita, J. , Nascimento, S. and Menezes, C. , *A Portuguese Grammar Checker Based on CETENPHOLA Corpus*

[3] Mathew, D. *A Course in Nepali*, RatnaPustak Bhandar, 1998

[4] Acharya, J. *A Descriptive Grammar of Nepali and an Analyzed Corpus*, Georgetown University Press, 1991

[5] *Bigyan Nepali Sabdakosh*, Royal Nepal Academy, Nepal, 2002.

[6] Pokharel, M., *Nepali Vakya Vyakaran*, Ekta Books Distributors Pvt. Ltd., Thapathali, Kathmandu, 2054 B.S

[7] Adhikari, H. R., Bhandar, B.P and Bhotahiti, *Samasamayik Nepali Vyakaran*, Kathmandu, Third Edition 2062 B.S

[8] Luitel, K.P, Luitel, L., Bhandar, B.P and Bhotahiti, Kathmandu, Second Edition 2058 B.S

[9] <http://comp.lancs.ac.uk/computing/research/stemming/Links>

[10] “Hunspell”
<http://sourceforge.net/projects/hunspell>

[11] Aronoff, M. and Fudeman, K., *What is Morphology?* Blackwell Publishing, Incorporated (October 1, 2004)