Architectural and System Design of the Nepali Grammar Checker

Bal Krishna Bal, Prajol Shrestha
Madan Puraskar Pustakalaya, PatanDhoka, Nepal
bal@mpp.org.np, prajol@mpp.org.np

Abstract

This paper describes the architectural and system design of the Nepali Grammar Checker, which is in due course of research and development. The development follows a modular approach with the Grammar Checker consisting of independent modules. These modules then in turn serve as a pipeline for the over all integrated system. The Grammar Checker aims to check the grammatical errors such as nominal and verbal agreement, parts of speech inflections, phrase and clause structure and the different categories of sentence patterns for Nepali.

1. Introduction

After some preliminary work on the Nepali Spell Checker, the Nepal Component of the PAN Localization Project has started the research and development of the Nepali Grammar Checker. Currently, the Natural Language Processing Team has been involved in the research work for the design and development of the Nepali Grammar Checker. Since it is a purely research and development oriented work, changes in the system design are quite natural at certain phases leading to an ultimate prototype of the Grammar Checker. At the current state, we propose a tentative architectural and system design of the Nepali Grammar Checker. However, at some later time, the design may be subject to changes, as more research findings get revealed.

2. Architecture

The high level system architecture of the Nepali Grammar Checker is as shown in Figure 1. As seen from the figure, the Nepali Grammar Checker has the following components:

1. Tokenizer Module;
2. Morphological Analyser Module;
3. Parts of Speech (POS) Tagger Module;
4. Chunker and Parser Module;
5. Grammatical Relation Finder;
6. Syntax checker Module;
7. Suggestions creating Module.

A brief description on each of the above follows.

2.1. Tokenizer Module

The tokenizer module splits the input text (paragraphs) from an input file into sentences. The tokenized sentences are are further tokenized into words, one word per line.

2.2. Morphological Analyzer Module

The "Morphological Analyser" module analyzes the tokenized words thus returning either the input words as root or breaking down the input words into its constituent morphs, root word and associated affixes. The word breaking is done on the basis of word breaking rules available for Nepali. In addition to this, this module also performs the first level Parts of Speech (POS) tagging of the morphs as per the information gathered from the free morpheme and affix lists. The term first level POS tagging implies that further processing would need to be carried out for hundred percent correct POS tagging. For instance, with cases whereby, a particular free morpheme takes more than one POS, the module assigns more than one POS. The assignment of a particular POS out of the several possible, i.e. POS disambiguity is handled by the POS Tagger module. Cases of particular words not being able to be POS tagged by the morphological analyser module would similarly be handled by the POS tagger module.

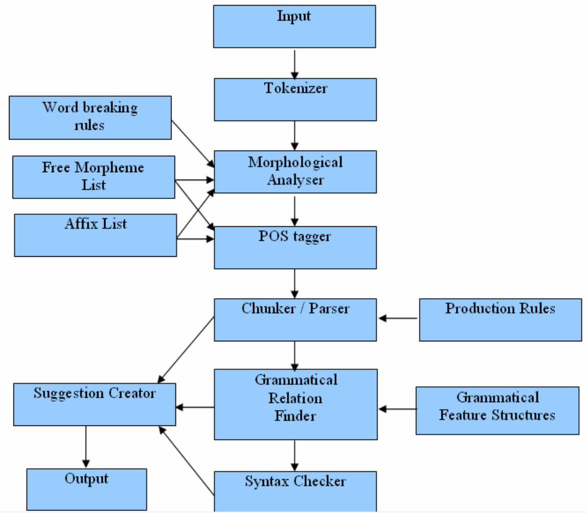

Fig. 1. High Level Architecture Diagram of the Nepali Grammar Checker

3. Parts of Speech (POS) Tagger Module

The module "POS Tagger Module" POS tags the untagged and POS tag undetermined tokenised words. As mentioned earlier, the morphological analyser module itself would do the first level POS tagging. Hence, the basic works of this particular module would be the following:

i) Assigning a particular POS category to a word out of the several possible thus resolving POS ambiguity;
ii) Assigning POS category to the derived or inflected words by collecting the required information from the Morphological Analyzer module;
iii) Assigning POS category to unknown words;

As with the case of the previous module, this module also relies heavily on the rules for POS tagging a particular word. This basically involves contextual rules, such as adjective preceeds a noun.

4. Chunker and Parser Module

The module "Chunker and Parser Module" identifies the chunks or phrases from the POS tagged words in the sentence of the input text file. For this, apart from the two files, the free morpheme list and the affix list, we would need a set of context-free grammar rules or productions.

For instance,

NP -> (MOD) (ADJ) N
ADJP -> (MOD) ADJ
AdvP-> (MOD) ADV
PoP-> (NP) PostPOSN
VP-> (ADvP) (NP) (NP) (PoP) (AdvP) VB
S-> NP VP

Abbreviations:
NP – Noun Phrase
MOD – Modifier
ADJ – Adjective
N – Noun
ADJP – Adjective Phrase
AdvP – Adverb Phrase
ADV - Adverb
PoP – Postpositional Phrase
PostPOSN – Postposition
AdvP – Adverbial Phrase
VB-Verb
S – Sentence
NP – Noun Phrase
VP – Verb Phrase

The right hand side of these rules represent the terminals whereas the left hand side are the non-terminals. The terminals delimited by the parenthesis on the right hand side of the rules are optional.

The chunker module would consult the production rules and the POS tags of the words as tagged by the POS tagger module to label the phrases in the sentence or identify the chunks in the sentences. For instance, a look up that results in the finding of the words in the sentence in the following POS category order sequence (MOD) (ADJ) N would result into being labelled as a Noun Phrase (NP) thus reducing to the non-terminal in the rewrite rules. The chunker module would consult the production rules and the POS tags of the words as tagged by the POS tagger module to label the phrases in the sentence or identify the chunks in the sentences. For instance, a look up that results in the finding of the words in the sentence in the following POS category order sequence (MOD) (ADJ) N would result into being labelled as a Noun Phrase thus reducing to the non-terminal in the rewrite rules. The POS tags enclosed in parenthesis indicate optionality in the occurrence.

5. Grammatical Relation Finder Module

This module assigns the grammatical roles like SUBJECT, OBJECT and VERB to the identified phrases or chunks from the previous module. The assignment of grammatical roles is based on the agreement phenomenon in the Nepali language. Nepali language has the following agreement patterns:

1) SUBJECT-VERB agreement;
2) Agreement between Modifier and Head in the Noun Phrase.

For instance,

In the sentence,
म घर जान्छु ।

The Noun Phrase (NP), म is in agreement with the verb (जान्छु) and hence म is the SUBJECT, जान्छु is the VERB and the remaining word in the sentence घर is the OBJECT.

In order to realize this module, agreement feature structure grammar need to be added to each of the entries in the free morpheme list and the affix list. A tentative format of the agreement feature structure grammar to be added would be as shown below:

यो | PDM | AGREEMENT NUMBER SG |

यी | PDM | AGREEMENT NUMBER PL |

मान्छे | NN | AGREEMENT NUMBER SG |

म | PFS | AGREEMENT PERSON FIRST NUMBER SG VERB INTRANSITIVE |

हामी | PFP | AGREEMENT PERSON FIRST NUMBER PL VERB INTRANSITIVE |

In the cases above, the POS tag of the lexical entries, which appear right after the first pipe symbol (PDM for यो, for instance) would have been assigned right in the implementation phase of the Morphological Analyzer. Hence in this phase, i.e. the Grammatical Relation Finder Module, the feature structures like AGREEMENT that primarily concerns NUMBER, PERSON, GENDER, VERB (TRANSITIVE/INTRANSITIVE) and their respective values need to be added both to the entries of the free morpheme list and the affix list. Besides, some more context free rules with feature structure grammar constraints need to be formulated. An example of such a rule is given below:

<NP HEAD AGREEMENT> = <VP HEAD AGREEMENT>

6. Syntax checker module

The module "Syntax checker module" checks whether the SUBJECT OBJECT VERB (SOV) pattern is observed in the input sentence or not. Besides, this module is also responsible for checking agreement between words in the sentence locally, i.e. in between neighbouring words. This would thus handle situations like whether the adjective actually preceeds a noun or not.

7. Suggestions creating module

The module "Suggestions creating module" consults the modules 4, 5 and 6, respectively, the Chunker and Parser Module, Grammatical Relation Finder Module and the Syntax Checker Module and suggests a different sentence structure in case it finds that the input sentence is syntactically incorrect.

8. Conclusion

Out of the modules proposed in the architectural and system design of the Nepali Grammar Checker, the research and development of the first two modules, namely the tokenizer and the morphological analyzer has been partially completed. A prototype of the two modules is also ready and in the process of being tested and added complexity handling features. As mentioned earlier, the proposed architectural and system design of the Nepali Grammar Checker is subjective to changes as further findings of the research work come out in future.

9. 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.*