Nepali Spellchecker 1.1 and the Thesaurus, research and development

Bal Krishna Bal, Basanta Karki, Laxmi Prasad Khatiwada, Prajol Shretha
Madan Puraskar Pustakalaya, Lalitpur, Nepal
bal@mpp.org.np, basanta_karki@mpp.org.np, laxmi@mpp.org.np, prajol@mpp.org.np

Abstract

This paper is a general overview of the technical and linguistic research and development being carried out in terms of developing the Nepali Spellchecker 1.1 and the Thesaurus. The strength of the current Spell-Checker is discussed in the document. A comparative analysis of the previous version of the spellchecker and the current version is also done. Besides, the testing procedures and mechanisms employed for the performance test of the spellchecker also has been covered. Lastly, the limitations of the current spellchecker are put forward and further recommendations presented. Many changes have not come into effect in terms of the Thesaurus save the size of the entries. A brief overview of the Thesaurus development framework is provided.

1. Introduction

Nepali SpellCheckers do not carry a long history. The first spellcheckers appeared in the language pack of MS Windows and in NepaLinux 1.0, both of which got released for public usage in 2005. The SpellChecker that came along with the OpenOffice.Org in NepaLinux 1.0 was very basic. The word coverage of the system was merely a count of approximately 300,000 words. Still, the development of the system was in itself a big revolution in the field of Natural Language Processing (NLP) for Nepali. Keeping in mind that the applications like the spellchecker has abundant usage in the publication houses, it was felt that the existing spellchecker had to be enhanced and made more robust. Hence, Nepali Spellchecker 1.1 is an enhanced form of the previous spellchecker. In the sections that follow, we will be discussing on the details involved in the research and development of the new version of the spellchecker.

2. Research Methodology and Objectives

The data and the information required for the research work have been acquired by consulting a wide number of resources. This includes different publications on the Nepali Grammar, other available sources like the Nepali dictionaries, active brain-storming sessions and consultation with experts. The research was basically for affix rules applicable to certain headwords categorized by POS category and handling exceptional cases. The primary objectives of the research work was to look for maximum number of affix rules applicable to the headwords listed in the dictionary file, thus generating as many correct Nepali words as possible. The Nepali language, being a highly inflectional and an agglutinative language, it is but impracticable from technical and linguistic aspects to list down all the possible words in one file. Rather the word generation approach by undergoing a morphological analysis and applying affix rules to stem words is the most practicable from practical point of view.

3. OO.Org and HunSpell framework

Hunspell is a spell checker and morphological library. It is very much applicable for languages exhibiting rich morphology and using complex scripts . Some of the interfaces that Hunspell can get connected to are Ispell-like terminal interface, Ispell pipe interface, OpenOffice.org UNO module etc.

Hunspell's code base is derived from the OOo MySpell. Some of the main features of Hunspell are as listed below [3] :

1. It has Unicode support for the first 65535 Unicode characters ;
2. Morphological analysis is possible both in custom item and arrangement style ;
3. There are in the maximum 65,535 affix classes and follows a twofold affix stripping , especially for agglutinative languages, like Azeri, Basque, Estonian, Finnish, Hungarian, Nepali, Turkish, etc. ;
4. It has support for complex compounding (for example, Hungarian and German)
5. It has support of language specific algorithms

required, for example, handling Azeri and Turkish dotted i, or German sharp s);

6. It handles conditional affixes, circumfixes, pseudoroots, homonyms etc. ;
7. The License that HunSpell follows is the GPL/LGPL/MPL tri-license

Depending upon the language specific terriotory, Hunspell may be customized by using the concerned locale file. HunSpell requires two files, respectively the dictionary file that contains words for the language and the affix file that has rules associated to the words in the dictionary by using flags serving as pointers[5]. The two files should be located in the folder ~openofficefolder/share/dic/ooo/. Spell checking is done using the affix file, locale and the dictionary file. While the affix file consists of affix rules, the dictionary file consists of root word, for example, “sing”. All the other forms of “sing” like “sings”, “sang” , “sung” etc. would be generated from the root word by applying the affix rules.

The dictionary file should clearly mention the name of the specific locale being used. This helps to avoid the loading of all the locales available. This dictionary file is possible to edit using a simple text editor[3].

The dictionary.lst has the following format :

```
#dictionary.lst for OOo

DICT ne NP ne_NP
THES ne NP th_ne_NP
HYPH en US hyph_en_US

#end of the dictionary.lst
```

The '#' indicates the commented lines in the file. We may see the following fields for each entry in the list [4]:

Field 1: Entry Type

Under this field, there are three different options which are elaborated below:

"DICT" - spellchecking dictionary
"HYPH" - hyphenation dictionary
"THES" - thesaurus files

You may include one or more of these options depending upon the purpose of the usage of the dictionary file.

Field 2: Language code from Locale

Here, you would need to specify the appropriate language, either “en” - for English, “de” - for Dutch, “ne” - for Nepali.

Field 3: Country Code from Locale

Here you would need to specify your country code name, for instance, “US” for United States, “GB” for Great Britain, “NP” for Nepal and so on.

Field 4: Root name of file(s) for field 1

Under this field, a combination of the language code and the country code joined by the underscore symbol is sufficient in the case of dictionaries, for instance ne_NP for Nepali language and Nepal. However, for the thesaurus and the hyphenation, you would also need to additionally add the “hyph” or “th” before the language and country code, for instance, “hyph_ne_NP” and “th_ne_NP”.

4. Results

In the results section, the findings of the study and the current status of the spellchecker would be summarized. This includes the analysis of the performance of the new version of the application.

4.1. Word Formation and Coverage

The dictionary file in the given spellchecker consists of approximately 39,869 headwords. These headwords generate more words getting associated with one or more affixes. The given spellchecker is expected to generate as much as 9,30,000 words, which means it can spell-check and provide the same amount of suggestions. Below, we provide an overview of the approach and statistics of the dictionary and the affix rules development.

4.2. Dictionary and affix rules development

In order to populate the dictionary file, the Nepali words have been categorized under the general parts of speech category, viz., verbs, noun, pronoun, adjective, adverb, conjunction, interjection, particle etc. As far as practicable, the dictionary has been attempted to populate with stem words or root words. Compound words are placed in the dictionary file only in case the rules are difficult to apply for generating particular compound words. Such words constitute some 2,500.

Given below is a statistical report of the composition of the dictionary and the possible word generation applying affix rules.

4.3. Verbs

We have tried to exhaustively include the stem forms of the Nepali verbs. In doing so, we, however, also have included some compound verb forms. Pure stem forms of the Nepali verbs constitute some 5,000 words. With the application of affix rules to these stem verb forms, it is expected to generate as much as 6, 78,400 possible verbal forms.

For instance, we have placed अचेट्, a stem form of the verb in the dictionary file to which we apply the suffix आइ, which is listed in the affix file. The resulting word is verbal form is अँचेटाइ. Furthermore, the creation and application of the rules have been done in such a manner that whenever possible and required, the breeding patterns resulting into the formation of some other additional words is highly considered. This, for example, applies to नगर्, a negation of गर्. This negation can then be propagated to other possible word generation like नगरेको, नगरून् etc. Similarly, गर् breeds into गरेको, गरून् etc. Our study has revealed that the maximum possible words in the Nepali language relate to verbs.

4.4. Nouns

The spellchecker has some 17,000 noun headwords. Our attempt was not to include proper nouns and concentrate more on common nouns. However, some concession has been made in including names of zones, districts, village development committees, some common proper names etc. This has been done with a view to give some localization color to the application. Nepali nouns in combination with postpositions form several compound words. Hence the spellchecker has tried to capture such words by including the possible postpositions that combine with nouns in the affix list.
Examples serve the following:

किताबहरूको, किताबहरूमा.

In the above, while किताब is a noun, -हरू indicate the plural form, -को and -मा are postpositions.

4.5. Pronouns

The pronouns constitute some 400 words in the given spellchecker. These are not only the pure pronoun headwords but are also compound pronouns. Usually these are the words forming as a result of the combination of the pronouns with the postpositions.

4.6. Adjectives

There are around 11,000 adjectives in the spellchecker. However, the pure adjective headwords are just 709 in number. The two very common rules applied are the following:

-आ suffix for plural

-ई suffix for the female gender

For eg.,

राम्रा कपडाहरू

राम्री केटी

4.7. Adverbs

The pure adverbs constitute some 2040 words in the dictionary file.

4.8. Conjunctions

A total of 40 conjunctions are listed in the dictionary file.

4.9. Interjections

The pure interjections are 150 in number.

4.10. Nuance particles

A total of 30 nuance particles are listed in the dictionary file.

4.11. Postpositions

There are around 80 postpositions and their derived forms in the doctionary file.

4.12. Comparison of the Nepali Spellchecker 1.0 and the Nepali Spellchecker 1.1

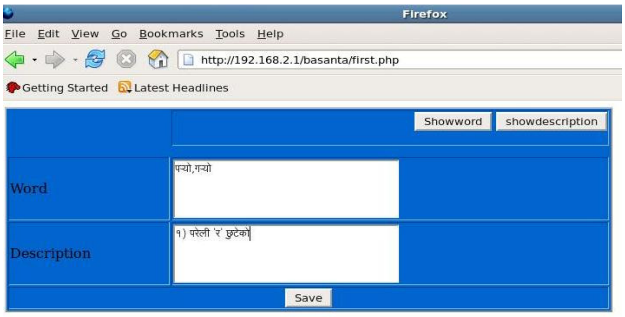
Since both of the Spellcheckers have been developed under the HunSpell framework, the development methodology is the same. However, in the new version, we have tried to be more scientific and systematic in terms of the affix rules development and even in terms of housekeeping the entries in the dictionary and the affix rule files. In the past, we had focused more on postpositional word forms and affix rules applicable in the formation of individual compound words. With the new version, we have tried to group not just individual words but words of different parts of speech category but to which the same set of affix rules are applicable. In terms of spell checking, the previous version could check about 300,000 Nepali words, while the current one checks around 930,000 Nepali words.

4.13. Complexities

The basic complexity was related to rule making and generalization of the rules for as much parts of speech category words as possible. The main challenge was in finding a midway to accomodate all the exceptions for a certain rule set. It was noted that often during the development and parallely testing process, OpenOffice.Org 2.0 got slowed down. This problem was soon fixed by removing useless free spaces in the dictionary file.

4.14. Performance testing

Assuming that all Unicoded Nepali text that we collected from different sources were spelt correctly, our main basis for performance testing of the spellchecker was trying to get as less wavy red lines as possible. On every test session, the words marked with wavy red line were collected and tried to fix it in the consecutive tests. For the purpose of maintaining a checklist, a simple web interface was devised in PHP. Through the interface missing words and suggestions could be suggested. A screenshot of the web interface is shown below in figure 1.


Firefox
File Edit View Go Bookmarks Tools Help
http://192.168.2.1/basanta/first.php
Getting Started
Latest Headlines
Showword
showdescription
Save

Word: पऱ्यो,गऱ्यो

Description: १) परेली 'र' छुटेको

Figure 1. Web interface used for forming checklist

4.15. Limitations of the Nepali Spellchecker 1.1

Some of the limitations of the Nepali Spellchecker 1.1 are listed below:

1. Eyelash ऱ instances as in पऱ्यो, गऱ्यो etc.

   This was chiefly because the input of such words was done by the input method which did not address the Zero-Width Joiner (ZWJ) and Zero-Width- Non-Joiner (ZWNJ) issue. In the future versions, since we would be using the rectified input method, such issues would be automatically addressed.

2. Noun and its adjective derivatives and vice versa E.g. दया (noun) and its adjective derivative दयालु

3. Other Verb Derivatives Eg. चिसिनु from adjective चिसो

4. Prefixes Eg. उप + हार = उपहार

The current spellchecker does not include words formed as a result of the prefixation. This issue would be addressed in future versions.

5. Compound Words Eg. महत्वपूर्ण Compound word handling is yet to be addressed. Principally two nouns could be joined to form compound words but often the resulting words turn out to be insensible.

6. Emphatic words e.g. पढेको पढ्यै

4.16. Thesaurus enhancements

The modality of thesaurus development has been the same as in the earlier version except that the size of the thesaurus in terms of words has increased from 600 to 5,500. As in the previous version, we have followed the MyThes framework developed by Kevin Hendricks.

This thesaurus has the facility to provide a words meaning and synonym but not it's antonym, which a thesaurus should give if thesaurus as such is taken in the strictest sense. MyThes was made specially to provide OOo with a thesaurus. It is the first thesaurus for OOo and is still being used with some enhancements from the OOo community. Originally, it did not support UTF-8 encoding, which was a big setback for countries lacking their own 8-bit ASCII character set. Recently, László Németh, the creator of Hunspell, provided a patch for MyThes to support Unicode. This patch could be patched to MyThes if versions of OOo older than 2.0.2 are present. In versions OOo 2.0.2 and above, the patch will have been automatically integrated into OOo. The creation of this patch has been a milestone in context of Internationalization (I18n) of MyThes, because non-latin languages now can be integrated into the thesaurus of OOo.

4.17. Thesaurus implementation in Ooo

MyThes is a very simple thesaurus. This thesaurus does not only provide information on synonyms, but also meanings and part-of-speech of a word. The main features of MyThes are as listed below:

- It is written in C++ to make it easier to interface with Pspell, OpenOffice, AbiWord, etc.
- It is stateless, as no static variables are used and should be completely reentrant with no if defs.
- It compiles with -ansi, -pedantic, and -Wall with no warnings, which makes it quite portable.
- It uses a simple perl program to read the structured text file and generate the index file which contains the index needed for binary searching.
- It is very simple with “lots” of comments. The main "smarts" are in the structure of the text file that makes up the thesaurus data.
- It comes with a ready-to-go structured thesaurus data file for en_US extracted from the WordNet-2.0 data.
- The source code has a BSD license (and no advertising clause).

5. Conclusion

The Nepali Spellchecker and Thesaurus have undergone significant change if not made a giant leap while coming from the previous version to the current one. The wide word coverage justified by the performance testing of the current version reveals this fact. However, there is still lot of things to be improved in the current spellchecker to make it of the industrial strength. We hope to rectify the limitations in our future releases

6. References

[1] “OO Lingucomponent project:” http://lingucomponent.openoffice.org/

[2] http://en.wikipedia.org/wiki/MySpell

[3] http://hunspell.sourceforge.net/

[4] http://elle.epfl.ch/article.php3?id_article=63

[5] http://hpux.tn.tudelft.nl/hppd/hpux/Text/ispell-3.2.06/man.html

[6] Nepali Team - Madan Purskar Pustakalya; "*PAN Localization Guide to Open Source Localization*” 2006, Printworks Pakistan