Research Report on The Standalone Nepali Spell Checker

Bal Krishna Bal, Prajwal Rupakheti
Madan Puraskar Pustakalaya
Lalitpur, PatanDhoka
Nepal

1, prajwalrupakheti@gmail.com

May, 2009

TABLE OF CONTENTS

Abstract

The report discusses the design and development issues of The Standalone Nepali Spell Checker application. We also talk about the methods used for implementation and present the current results of the system.

Introduction

Spell Checkers have been quite appealing among the general public in Nepal and other South Asian countries in the recent days. It could have been partly because of the fact that the languages in this region attained computational processing capabilities only after the advent of the Unicode encoding system [1]. Madan Puraskar Pustakalaya (MPP) already released at least four different versions of the Nepali Spell Checker incorporated with the localized OpenOffice.org Writer application [2, 3]. The latest version of the Nepali Spell Checker came out with the Nepali localized OpenOffice.org Writer 2.4, which was released in May 2008. This version of the Spell Checker has the word coverage of 6 million Nepali words and hence is quite robust in terms of performance. However, it still requires that the end user installs the OpenOffice.org Writer application in his/her machine. This poses several potential problems on the user's end – space, version compatibility issues etc.

The "Standalone Nepali Spell Checker" is aimed towards freeing the user from the need of installing OpenOffice.org Writer application as a prerequisite for using the Spell Checking utility for Nepali. However, we still follow the HunSpell framework by adopting the HunSpell engine and the two resource files for Spell Checking – the .dict and .affix files, customized for the Nepali language. This preserves the original robustness of the OpenOffice.org Writer based Nepali Spell Checker in the current Standalone version, yet at the same time provides flexibility and comfort of using an easy to use, simple and light weight text editor. The Standalone Nepali Spell Checker currently works just on the Windows platform.

Methods

For the development of The Standalone Nepali Spell Checker, we have employed the HunSpell engine [4] which is a Spell Checker and morphological library. This Spell Checker engine was initially developed for the Hungarian Spell Checker but now has the capabilities of processing theoretically all languages with Unicode support. The HunSpell framework comprises the HunSpell engine and two resource files – the .dict and the .affix files. NHunSpell[1], which is a C# library, is additionally used that acts as an intermediate interface between the HunSpell engine and the resource files. The Graphical User Interface (GUI) of the text editor for the Nepali Standalone Spell Checker is developed using the Microsoft C#.Net 2008. As far as the Spell Checking mechanism itself is concerned, the HunSpell engine basically performs a lookup in the

[1] http://www. sourceforge.net/projects/nhunspell

.dict file for a word and at the same time also works on forming derivational words out of the head words in the .dict file and the corresponding rules in the .affix file. If a matching word does not result out of the above process, it infers that the given word is typed incorrectly and hence it generates a list of possible suggestions for the word by using Levenshtein Edit Distance Technique [6].

In Fig.1 below, we present the system architecture of The Standalone Nepali Spell Checker.

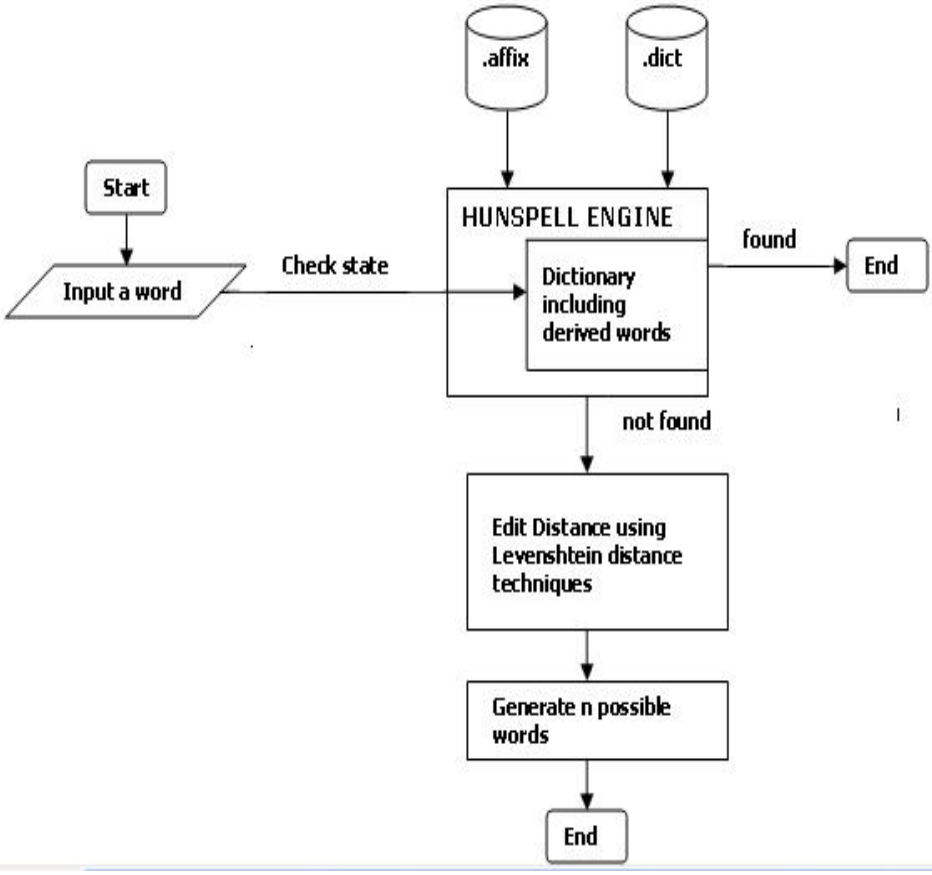

Fig.1 High Level System Architecture of The Standalone Nepali Spell Checker

Results

As mentioned earlier, the Standalone Nepal Spell Checker has the same robustness as that of the OpenOffice.org Writer version of the Nepali Spell Checker. Hence the word coverage of 6 million Nepali words in terms of word coverage and the capabilities of suggesting options are also applicable for this version of the Nepali Spell Checker. Below, we present a few screenshots of the current system.

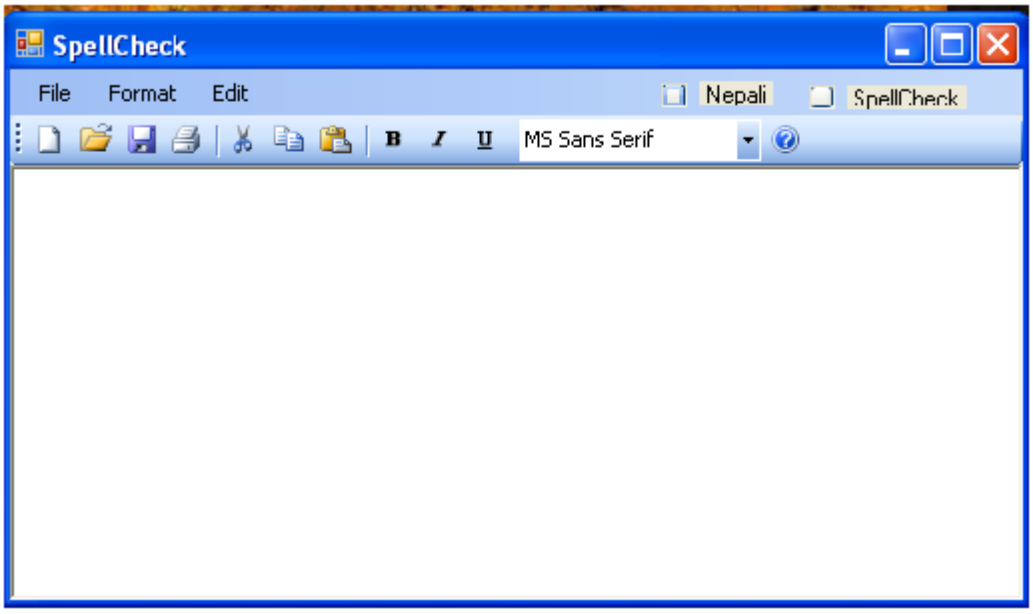

Fig.2. User Interface of "The Standalone Nepali Spell Checker"

The User Interface of "The Standalone Nepali Spell Checker" can be switched from English to Nepali and vice versa by just clicking on the check box on the upper right for language.

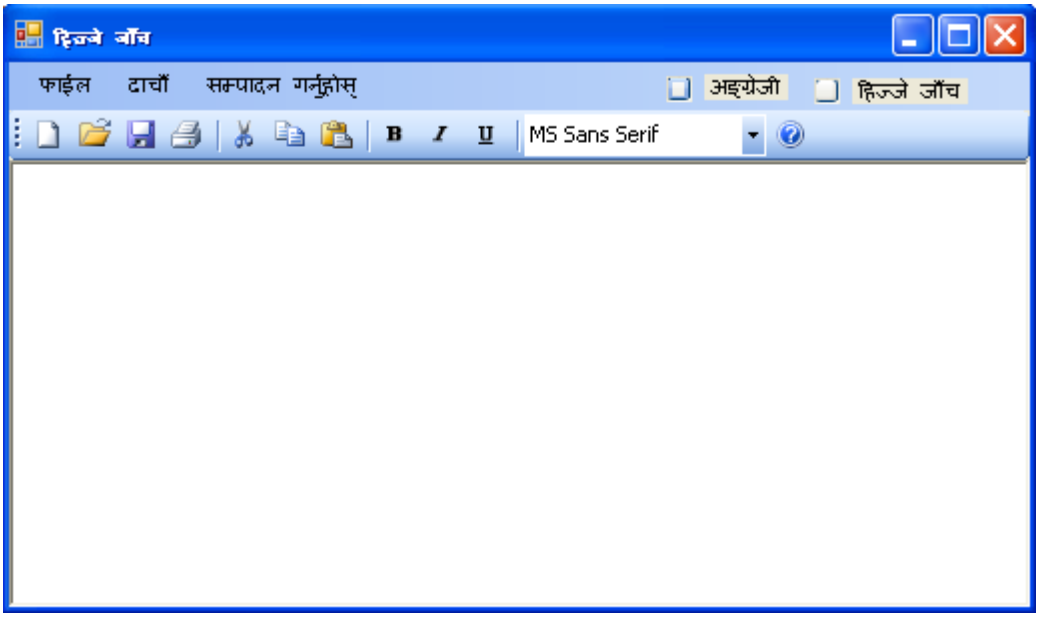

Fig.3. Nepali User Interface of "The Standalone Nepali Spell Checker"

For spell checking, one has to check on the "Spell Check" box located on the upper right corner. After checking the box, the application will start checking the incorrect words are marked in red text as shown below in Fig.4. The incorrect word when right clicked would be provided a number of suggestions that the user may select from the suggestion list. Correspondingly, the selected word would replace the incorrect word and the color would also

be changed into black, i.e. the default color of the text. There is also a provision of ignoring the marking of a word as incorrect. To ignore the incorrect marking, one may simply select "□□□□ □□□□□" from the list which is "Ignore".

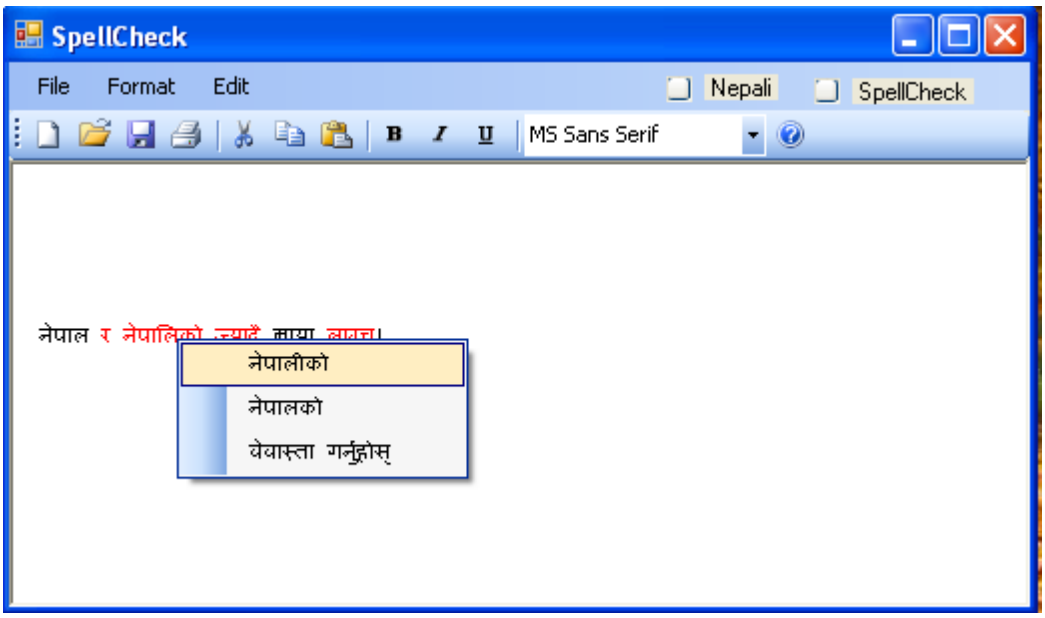

Fig.4. Marking of incorrect words and the suggestion list

In Fig.5 below, we present a corrected version of the text subjected to Spell Checking in Fig. 4 above.

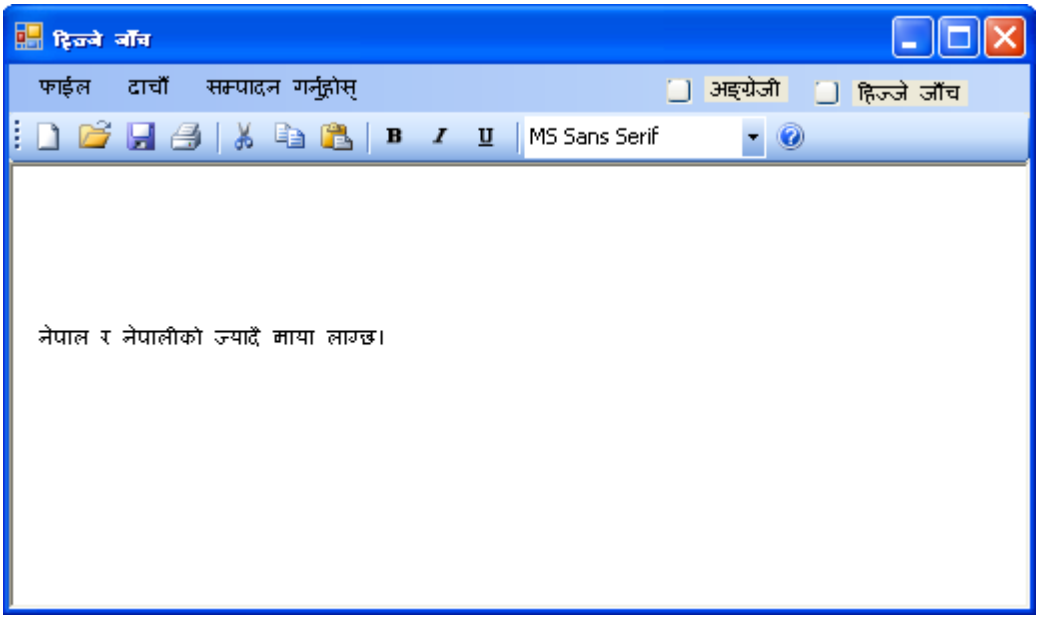

Fig.5. Spell checked and corrected text.

The editor, though, simple has almost all the features that most text editors have. New files can be created using the editor itself and saved accordingly.

Conclusion

In this paper, we discussed the design and development of The Nepali Standalone Spell Checker. We also talked about the methods adopted for the implementation and presented the current results of the system. Future work could be collecting feedbacks from the end users and refining the system accordingly.

Acknowledgment

The PAN L10n works have been carried out with the aid of a grant from the International Development Research Centre, Ottawa, Canada, administered through the Center for Research in Urdu Language Processing (CRLUP), National University of Computing and Emerging Sciences, Lahore, Pakistan (NUCES).

Our special thanks go to the engineering students from Kantipur Engineering College and Gandaki College of Engineering and Sciences, respectively, Ms. Anu Sharma, Ms, Miru Poudel, Ms. Preeti Amatya and Mr. Sagun Shrestha for helping us with the project as part of their internship.

References

[1] http://unicode.org

[2] B. K. Bal and P. Shrestha, "Nepali Spellchecker," PAN Localization Working Papers 2004-2007.

[3] B. K. Bal, B. Karki, and L. Khatiwada, "Nepali Spellchecker 1.1 and the Thesaurus, Research and Development," PAN Localization Working Papers 2004-2007.

[4] http://en.wikipedia.org/wiki/Hunspell

[5] http://en.wikipedia.org/wiki/Levenshtein_distance