Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(5): 135-138

Research Article ISSN: 2394 - 658X

Luitspell: Development of an Assamese Language Spell Checker for Open Office Writer

Kishore Kashyap, Hirakjyoti Sarma and Shikhar Kumar Sarma

Department of Information Technology, Gauhati University, Guwahati, India
hirak.gu@gmail.com

ABSTRACT

This work primarily aim on different aspects of designing a spell checker for the Assamese language and integrating it as an add-on into Open Office Writer. Besides emphasizing on error detection and suggestion generation the programming model and challenges of developing the add-on for the aforementioned application has also been discussed. The system also takes care of the morphological aspects of the words. The algorithm generates a set of suggestion for misspelled words with the dictionary words by minimum edit distance techniques. Other techniques for suggestion generation such as soundex scheme are not explored. The system after integrating with Open Office Writer can detect spelling errors of Assamese words and if found wrong, it will suggest correct alternatives. This work mainly emphasizes on (1) Precision (2) Recall and (3) Well behavior in Open Office environment. For error detection and suggestion generation the results are given in terms of precision and recall.

Key words: Spell Checker, Assamese language, minimum edit distance, software integration

---

INTRODUCTION

In computing, a spell checker is an application program that flags words in a document that may not be spelled correctly. A typical spell checking application presents a list of alternatives for each misspelled word encountered in a document. Some spell checkers allow the user to add correct words to the dictionary to increase the vocabulary. Developing a spell checker in Assamese is not a straight forward task. The Assamese language is a linguistically rich and complex Indo-Aryan language. While checking for a spelling for a particular Assamese word we have to look for its spelling rules, word structure format etc. Moreover, it should also check for valid Assamese characters in the word. The Assamese Unicode range is U+0980 to U+09FF. The Assamese language has significant similarities with the Bengali language in the representation of the character set of the language while linguistically it has significant differences.

- Our strategy for the spell checking can be briefly outlined as below:
- Take a word from the text or document (token generation).
- Detect if the word is misspelled
- If misspelled, analyse it morphologically.
- Generate a suggestion list.

To achieve the above four steps we have decided the following components to be combined for developing the spell checker.

- Token generator
- A lightweight morphological analyser
- The standard dictionary vocabulary set, grammatically annotated
- Rule base
- The main spell checker engine

Another aspect of our work is to investigate the integration of our system into a larger system (in our case Open Office Writer) to examine its behaviour in real case scenario.

A BRIEF OVERVIEW OF ASSAMESE LANGUAGE

The Assamese language is cordially associated with the most important Indo-European Language Group. We have to study the Indo-European Language group to find the origin of Assamese language though it seems that it is

made up of the Prota-Astrolied and Sino-Tibetan Language Group. Ascoli divided the Indo-European Language group into two main group viz. Satam and Centum. Indo-Aryan Languages are derived from the Indi-Iranian group which is one of the four sub-division of Satam. Assamese Language has also come through the three stages [(1) Old Indo-Aryan 1500BC-600BC→ (2) Middle Indo-Aryan 600BC-1000AD→ (3) New Indo-Aryan 1000AD-till now] of Indo-Aryan Language as the other Modern Indian Languages. The Indo-Aryan Languages, viz. Assamese, Bangla, Oriya etc., are derived from Avahattha through Magadhi Apravhransa. The earliest evidence of Assamese dates back to the literature of the Charyapadas, written by a few Buddhist scholars. The Assamese language present in the Charyapadas, reflects the initial stages of the evaluation of the Assamese language. There are eight vowel phonemes and twenty-one consonant phonemes including two semi-vowels in Standard Colloquial Assamese.

CHARACTERISTICS OF ASSAMESE LANGUAGE

Morphological Characteristics

The Assamese language has many special morphological characteristics. Out of which few are outlined below:

1. Numbers are not grammatically marked in Assamese. There are two types of Numbers in Assamese viz., Singular and Plural.
2. Gender is also not grammatically marked in Assamese. Linguistically, there are three types of Genders in this Language: Masculine, Feminine and Common. Common gender indicates neuter gender forms.
3. Kinship nouns are inflected according to the person's i.e., second person and third person. This peculiarity is absent in other NIA languages.
4. As a suffix rich language Assamese possesses many derivational, Inflectional Suffixes as well as many prefixes like other NIA languages.
5. Assamese possesses variation of enclitic definitive i.e., classifiers which are not found in other NIA languages except Bangla and Oriya.
6. There are seven types of Cases in Assamese language, Nominative, Accusative, Instrumental, Dative, Ablative, Genitive, and Locative.

Syntactical Characteristics

The Assamese language has many special syntactical characteristics. Out of which few are outlined below:

1. The general syntactic structure of Assamese language is Subject+Object+Verb (SOV).
2. Syntactically, the Assamese sentence structure is mainly divided into three types - Simple, Complex and Compound.
3. Assamese sentence structure is flexible. Depending on the context or mood of the speaker it might vary.
4. Assamese sentence structure is of different kinds. Very short sentences are found frequently. Sometimes long expressions are made by the addition of the indeclinable.
5. In Assamese, there are subject-verb agreements. The verbs in Assamese agree with the subjects in person. There is no agreement in number or gender like some other languages, viz., English or Spanish etc.
6. Verb less sentences are also common in Assamese language.
7. Idiomatic expressions are also found in Assamese.

METHODOLOGY FOR DEVELOPING THE SPELL CHECKER

Error Detection

Our main approach for error detection is checking edit distance against dictionary words. For this purpose Levenshtein distance algorithm is used. Levenshtein algorithm efficiently calculates the edit cost of two strings. But one drawback of Levenshtein algorithm is that it checks edit distance for characters at any index of the string. The word is first stemmed and root word is checked. Error detection block diagram is shown in Fig. 1 whereas Fig.2 depicts suggestion block of our system.

Tokenization

As the sole purpose of a spell checker is to detect and correct misspelled words, every word of a text or document must be extracted, tested and validated. This process of extraction of words one by one is called token generation. In this phase, nothing more is done except forming the words taking the characters sequentially till the delimiters or space is found. The task of generating proper tokens of a language like Assamese is not very straight forward due to its rich morphological characteristics.

Stemming

Due to morphological richness of the Assamese language one morphological analyser (MA) is required to extract all the inflectional forms of the words got from the token generator. The morphological analyser we have used for our work is not an MA in true sense rather we can call it a light weight MA.

Standard Dictionary Vocabulary Set

This is the grammatically annotated dictionary of Assamese words classified using Assamese morphological rules.

Spell Checker Engine
This is the main component of the spell checker which will take tokens from the token generator and check whether it is a correct word or a misspelled one by testing it with proper algorithm and the standard dictionary vocabulary set.

Minimum Edit Distance Checker
This module deals with the checking minimum edit distance of the misspelled word with the stored dictionary root words. With a given threshold value the words are chosen for suggestion.

Word Re-Constructor
This module is required to reconstruct the stemmed word as we only check the root words for errors eliminating the affixes, during suggestion generation phase the stemmed affixes with the correct root word is to be reconstructed again.

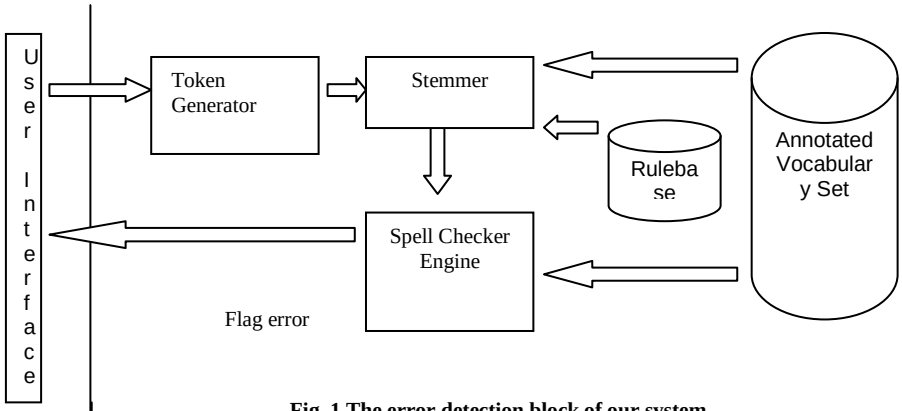

Fig. 1 The error detection block of our system

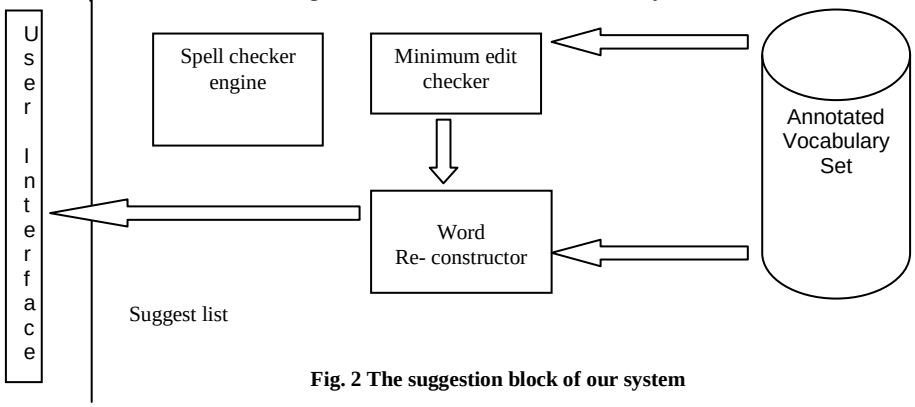

Fig. 2 The suggestion block of our system

DEVELOPMENT OF 'LUITSPELL' SPELL CHECKER ADD-ON AND CHALLENGES

Open Office is complex software with rich facilities. But it is very interesting that despite the complex behaviour of the software it is rather easy to develop add-ons for it. However, one is at the risk of crashing the system with a little bug in the user code. The problem we encountered during the development work is as such:

- The first problem the dictionary format
- Second problem we faced was whether explicit user threads should be invoked?
- Fourth problem is user interaction design

Attacking the Problems and resolving them
The first problem is the dictionary format. The dictionary we are using for this experiment contains root word of around 16,000 words. The dictionary size is very small. It contains same word more than once as a single word can carry different meanings. It does not follow any convention for POS or affixes. It is just a simple text file with one word per line format. So during the execution time the words are stored in memory as linked lists.

The second problem was encountered when we tried to automatically suggest correct alternatives for a misspelled word. Later, we decided to leave it and adopted a way which facilitates us to suggest correct words after explicit event dispatch.

The third problem was the most difficult one to tackle with. The problem was that user has to select a text range to check spelling. If multiple words are found wrong then they were shown in separate windows one window for each wrong word. Now the problem is solved by accumulating all the suggestions into a single data structure and then showing them to the user.

EXPERIMENTAL RESULTS

We considered 3 documents containing Assamese words and tried to calculate the precision and recall in comparison to hunspell. The results are tabulated below:

Precision

Our system is showing good result as it is well modelled with Assamese language rules.

Table - 1 Experimental Result

Spell Checker Engine	Doc1(1000 words)	Doc2 (5000)	Doc3 (10000)
Luitspell	0.58	0.63	0.69
Hunspell	0.50	0.62	0.70

System Demo Screen Shots

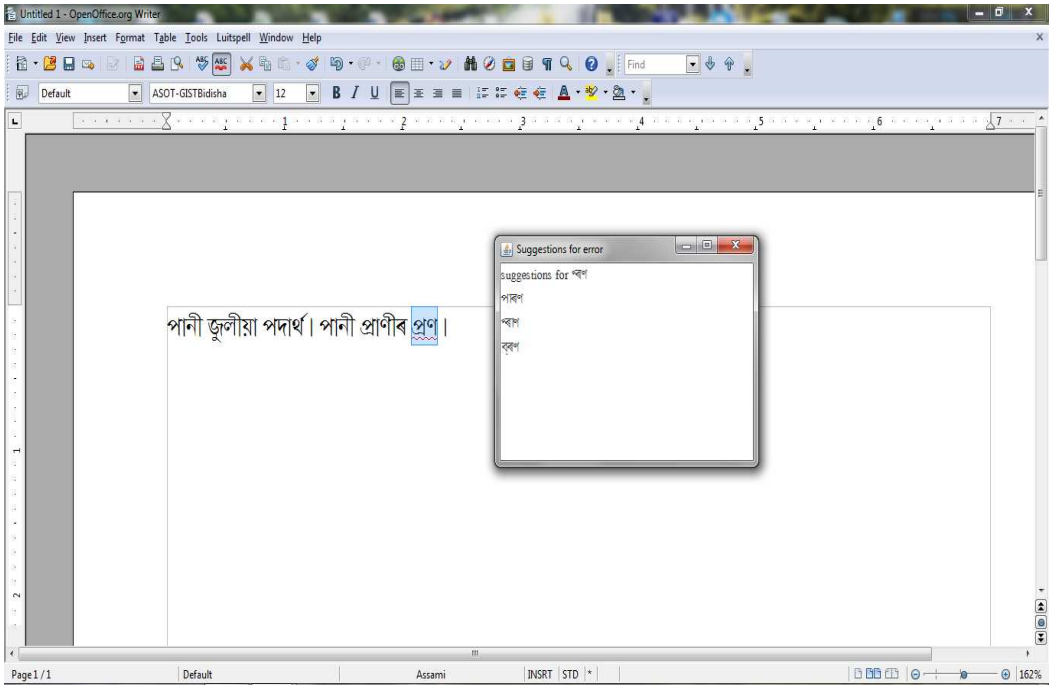

Fig. 3 Luitspell is detecting a misspelled Assamese word and suggesting correct alternatives

CONCLUSION

A dictionary based approach for 'spell checker' for Assamese has proposed. Evaluation of the test sets have shown the proposed approach perform equivalent to hunspell but some fallback in suggestion generation. There remains a possibility for extension of this work. In this work we generate the suggestion words based on the misspelled word and not the context of the sentence. One can improve the suggestion words considering the sentence context. One can provide option to the user to add suffixes in the suffix file and as well as in the suffix tree. Much work to do is in the user interaction part. Some improvement is necessary for suggestion generation. Moreover, the edit distance technique need to be improved.

REFERENCES

[1] Karen Kukich, Techniques for Automatically Correcting Words in Text, *ACM Computing Surveys*, **1992,** 24 (4), 377 - 439**.**

[2] GS Lehal, Design and Implementation of Punjabi Spell Checker, *International Journal of Systemics, Cybernetics and Informatics*, **2007**, 70-75.

[3] N UzZaman and M Khan, A Comprehensive Bangla Spelling Checker, *Proceeding of International Conference on Computer Processing on Bangla (ICCPB-2006)*, Dhaka, Bangladesh, **2006.**

[4] Editha D Dimalen, Davis Muhajereen and D Dimalen , An Open Office Spelling and Grammar Checker Add-in using an Open Source External Engine as Resource Manager and Parser, *Proceedings of the 4th National Natural Language Processing Research Symposium (NNLPRS)*, Manila

[5] *http://wiki.services.Open Office.org/wiki/Documentation/DevGuide*