Designing hybrid approach Spell checker for Oriya

Asst. Prof. Hari Hara Padhy
Department of Computer Science and Engineering
Gandhi Institute of Technology and Management (GITAM), Bhubaneswar, Odisha, India

Prof. Dr. Sanghamitra Mohanty
Department of Computer Science and Application
North Orissa University, Baripada ,Odissa ,India

Abstract- This paper Illustrates on different aspects of designing Spell checkers in Indian languages and a hybrid approach to the design of Oriya language Spell checker. Oriya is a morphologically rich language in which most of the morphemes coordinate with the root words in the form of suffixes. Spelling errors like substitution, transposition, insertion and deletion of characters can be handled using different approaches like Minimum edit distance classifiers & N-gram approach and Morphological analysers. To build an efficient Oriya spellchecker for generating high percentage of suggestion accuracy the technique combined with Edit distance, N-gram approach, and Morphological analyser is the best. The algorithm generates appropriate suggestion set for misspelled words by matching with the dictionary words. The algorithm implemented in such a way that searching complexity is minimum. It considers word length of misspelled words and accordingly it minimizes searching of characters from the dictionary. We carried out several experiments to compare different approaches for generating percentage of suggestion accuracy from given misspelled words.

Keywords – Morpheme, Minimum edit distance classifier, Searching complexity, N-gram approach, Morphological analyser.

I. INTRODUCTION

Spell checking is a well researched area in NLP, which deals with detection and automatic correction of spelling errors in an electronic text document. A Spell checker is a tool that will check the spelling of words in a document, validate them and in case the checker finds error, list out the correct spelling in the form of suggestions. Any word processor should have a Spell checker associated with it as the user may commit mistakes during typing. The Spell checker detects the mistakes and prompts the user with a set of suggestions, which will aid the correction of the misspelled word. Oriya language is a complex language. Here the suffixes, postpositions and case endings agglutinate with the verbs, nouns, adverbs or pronouns. Also one or more suffixes can combine with the base word. Hence Morphological analysis of the input word is a must for Oriya Spell checker. For error detection and suggestion generation we have to recombine words split using the Morphological analyser. Hence a Morphological Generator also forms part of a Spell checker.

All the misspelled words are classified into word level and sentence level error classes. Finally, the error patterns are formed at the character level. The detection of real word error needs higher level knowledge compared to the detection of nonword error. In fact, detection of real word error is a problem that needs NLP tools to solve. Quite often, it is not possible to separate the problem of real error detection from that of correction. Even for nonword errors, correction is a nontrivial task. Several approaches based on Minimum edit distance, Similarity key, rules, N-grams, Probability and Neural nets are proposed to accomplish the task [1-8, 11, 12]. According to Damerau, non-word errors can be classified into four major types. They are substitution error, deletion error, insertion error, and transposition error. Most of the misspellings take place by slips or omissions, like slips of matras (matra or phala) .Complete omission of vowel diacritical markers or some part of the diacritical markers. Wrong use of vowel diacritical markers is also noticed. Wrong uses of characters, which are phonetically similar to the correct ones, have also been observed. In the case of compound consonants (called yuktAksar in Oriya), mistakes takes place are due to ignorance. We have observed a great deal of confusion in the uses of long and short vowels. Aspirated and unaspirated consonants are also noted to be confounded. A great deal of confusion has also been seen to occur in the use of dental and cerebral nasal consonant. As a result there is a chance of committing mistakes if proper spelling

rules are not remembered. Similar confusion is seen in the case of three sibilant sounds in Oriya (palatal, dental and cerebral). Utterance of these characters are hardly differentiable. Our observation also reveals that the three trill consonants (liquid r, cerebral R and aspirate Rh) are well confounded. From our observation it is clear that most of the misspellings are due to

(i) phonetic similarity of Oriya characters,
(ii) the difference between the graphemic representation and phonetic utterances, and
(iii) lack of proper knowledge of spelling rules.

To the best of our knowledge no other work reported the detection of error position and error type in a misspelled string combine different approaches. From this standpoint our effort is a pioneering one. In typography most of the errors are seen to occur at word level and they are mostly non word errors. In some cases, sentence level errors such as real-word errors, grammatical errors, run on errors, split word errors etc. are also observed.

II. PROPOSED ALGORITHM

A. Minimum edit distance algorithm –

EDIT DISTANCE (S1,S2)

```
1int m[i ,j]=0
2 for i ←1 to |s1|
3 do m[i ,0]=i
4 for j← 1 to |s2|
5 do m[0 , j]=j
6 for i ←1 to |s1|
7 do for j ←1 to |s2|
8   do m[i , j]=min{m[i-1, j-1]+if (s1[i]=s2[j]) then 0 else 1  fi,
9                   m[i-1,j]+1,
10                  m[i ,j-1]+1 }
11 return m[|s1|, |s2|]
```

The time complexity between two strings s1 and s2 is o(|m| × |n|), where |m| denotes the length of a string s1 and |n| denotes the length of s2.The idea is to use use the dynamic programming algorithm ,where the characters in s1 and s2 are given in array form. The algorithm fills the (integer) entries in a matrix whose dimensions equal the lengths of the two strings whose edit distances is being computed. The (i,j) entry of the matrix will hold (after the algorithm is executed) the edit distance between the strings consisting of the first i characters of s1 and the first j characters of s2. The edit distance calculation is explained in Figure 1, Where the three quantities whose minimum is taken correspond to substituting a character in s1,inserting a character in s1 and inserting a character in s2.The typical cell [i, j] has four entries formatted as a 2×2 cell. The lower right entry in each cell is the min of the other three, corresponding to the three entries m[i-1, j-1] +0 or 1 depending on ,whether s1[i]=s2[j], m[i-1, j]+1 and m[i ,j-1]+1.

		ଅ	ର	ଟ
	0	1	2	3
ଅ	1	0	1	2
ର	2	1	0	1
କ	3	2	1	1

Figure1. Minimum Edit distance calculation between two words ‘ଅରଟ’ and ‘ ଅରକ’

N-gram approach

The idea of using N-grams in language processing was discussed first by Shannon[9].After this initial work, the idea of using N-grams has been applied to many problems such as word prediction, spelling correction ,speech recognition, translated word correction and string searching. One main advantage of the N-gram method is that it is language independent.

In a spelling correction task , an N-gram is a sequence of N letters in a word or a string. The N-gram model can be used to compute the similarity between two strings by counting the number of similar N-grams they share. The more similar N-grams between two strings exist the more similar they are. Based on this idea the similarity coefficient [10] can be derived. The similarity coefficient δ is defined by the following equation.

$$\delta_n (a, b) = \frac{|\alpha \wedge \beta|}{|\alpha \vee \beta|} \quad (1)$$

Where α and β are the N-gram sets for two words a and b to be compared. $|\alpha \wedge \beta|$ denotes the number of similar N-grams in α and β and $|\alpha \vee \beta|$ denotes the number of unique N-grams in the union of α and β .

B. System design and implementation

The design of Spell checker is provided in Figure-2. The idea of this proposed design is how the input text is processed to find potential spelling errors. For Spell checking, the input text is compared with dictionary. The input texts are taken in N-gram approach. Using the techniques of Edit distance and rule based morphological analyser searching is performed and suggestion is generated for misspelled words. All these activities have been implemented in java platform with database as text file with Oriya text as Unicode and we have used 'utkal' Font for Oriya script in java.

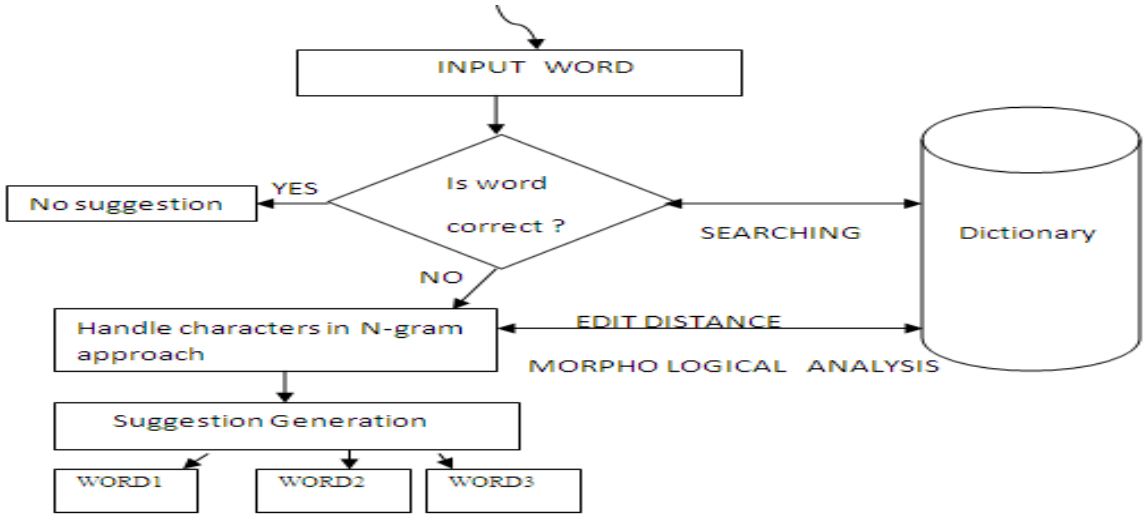

Figure2. Oriya Spell checker system design

III. Experiment and Result

In this paper we propose a technique of error detection that can pinpoint the error position in a big majority of cases and thus reduce the number of correct alternatives to a large extent. The approach is based on matching the

string using Edit distance concept and N-gram concept combined the techniques of morphological analysers . For correct spelling of every word we have referred to the ***Sukhabodha Oriya Abhidhan*** by Debendra Kumar dash. Three experiments were carried out by taking Edit distance approach and combination of Edit distance ,N-gram approach and Morphological analysers. The results is represented in Table-1.

Comparison of Suggestion accuracy (%)

Experiment No.	Edit Distance(E.D.) (% of Suggestion accuracy)	E.D.+N-gram +Morphological (% of Suggestion accuracy)
01.	86.4	93.64
02.	83.19	94.61
03.	85.18	95.28

Table- 1

Table-1 shows the comparison of suggestion accuracy (in percentage) of three experiments carried out by taking alone Edit Distance and combination of Edit Distance, N-gram approach and Morphological analysers.

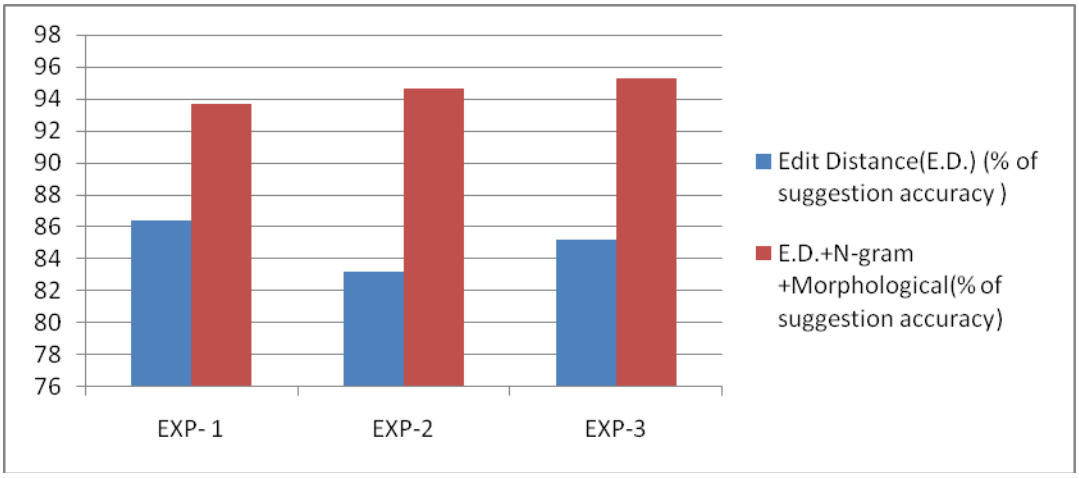

Figure 3 Comparison of percentage of suggestion accuracy of Edit distance approach and combination of Edit distance, N-gram approach and Morphological analysers.

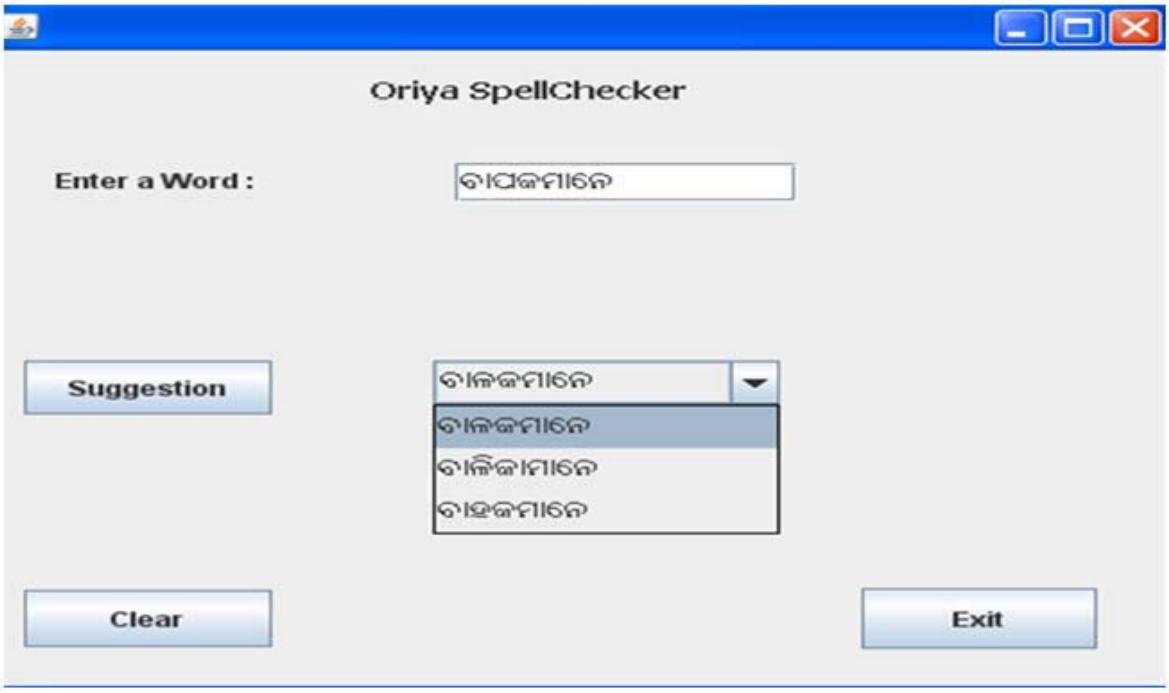

Figure 4. Snapshot of Oriya Spell checker

IV.CONCLUSION

Automatic spelling correction with high correction accuracy still remains a challenge. There is a considerable degree of difficulty in suggesting the correctly spelled words without taking the context into consideration. This article presented design and implementation details of the Spell checking system for Oriya. This Spell checking system is capable of detecting various spelling errors in formal Oriya texts. To the best of our knowledge, this is the first such system for Oriya and other Indian languages. We hope that this research work will attempt to narrow down the gap that exists between Oriya and other natural languages in the Natural Language Processing field.

REFERENCES

[1] R.C. Angell, G.E. Freund and P. Willet, (1983) "Automatic spelling corection using a trigram similarity measure", *Information Processing and Management.* 19: 255-261.

[2] V. Cherkassky and N. Vassilas (1989) "Back-propagation networks for spelling correction". *Neural Network.* 1(3): 166-173.

[3] K.W. Church and W.A. Gale (1991) "Probability scoring for spelling correction". Statistical Computing. 1(1): 93-103.

[4] F.J. Damerau (1964) "A technique for computer detection and correction of spelling errors". *Commun. ACM.* 7(3): 171-176.

[5] R.E. Gorin (1971) "SPELL: A spelling checking and correction program", *Onlinedocumentation for the DEC-10 computer.*

[6] S. Kahan, T. Pavlidis and H.S. Baird (1987) "On the recognition of characters of any font size", *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-9. 9: 174-287.

[7] K. Kukich (1992) "Techniques for automatically correcting words in text". *ACM Computing Surveys.* 24(4): 377-439

[8] V.I. Levenshtein (1966) "Binary codes capable of correcting deletions, insertions and reversals". *Sov. Phys. Dokl.,* 10: 707-710.

[9] C.E. Shannon, “prediction and Entropy of original printed Englllish” Bell sys.Tec.J.(30):50-64,1951

[10] U.pfeifer,”Retrieval Effectiveness of proper original Name search methods”, information processing and management,32(6):667-679,1996

[11] W. B. Cavnar and J. M. Trenkle “N-gram –based Text categorization,” proceeding of the symposium on Document Analysis and Information Retrieval ,University of Nevada, Los vegas,1994

[12] P. Willett “Document Retrieval Experiments using Indexing vocabularies of varying size II .Hashing, Truncation – Diagram and Trigram Encoding of Index Terms” J.Doc . 35,296,1979