Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Assamese Spell Checker Design and Implementation

Biswajit Sarma[1], Debashree Goswami[2] and Geetoshree Goswami[3]
[1]*Department of Computer Science and Engineering, Jorhat Engineering College*
[2]*State Bank of India, Gitanagar Branch, Guwahati*
[3]*Maharishi Vidya Mandir Senior Secondary School, Silpukhuri, Guwahati*

Abstract—Spell checkers form a vital ingredient of text processors, character recognition system, dictionary search engines, language processing software and similar tools. Though considerable work has been done in the area for English and related languages,the Indian language scenario presents a relatively more complex and uphill task. This paper describes strategies involved in the implementation of a spell checker for Assamese, the official language of the North Eastern Indian state of Assam. It deals with the development of spell-checker in Assamese language one of the most popular language in Indian subcontinent. The problems of compound words is also carefully tackled. In this paper we describe the insertion and searching of Assamese document. Compare the methodologies with existing solutions available in the literature and then propose solutions for each step. Finally, we conclude by showing the performance and evaluation of our proposed solution.
Keywords— Spell checker; Natural Language processing; Assamese Language; Searching; Suggestion .

I. INTRODUCTION

A Spell Checker has become a vital ingredient of high technology applications such as speech recognition and generation, character or text recognition systems and pen-based interfaces. Building a Spell Checker calls for an exhaustive study of several aspects of the language in question the construction of a morphological analyzer, a detailed dictionary, rules for resolving inflexions and other language dependent anomalies, to name a few. The main task of a spell checker is to automatic detection and correction of word error appearing in a text document. It looks simple in principle, but difficult in practice specially for Indian languages. The word-error can belong to one of the two distinct categories, namely, non word error and real word error. Let a string of characters separated by spaces or punctuation marks be called a candidate string. A candidate string is a valid word if it has a meaning. Else, it is a non word. By real word error we mean a valid but not the intended word in the sentence, thus making the sentence syntactically or semantically incorrect. In both cases the problem is to detect the erroneous word and either suggest correct alternatives or automatically replace it by the appropriate word. Another important issue is the computerized dictionary which concerns the size of the dictionary, the problem of inflection and creative morphology, the dictionary file structure, dictionary partitioning, word access techniques and so on. Dictionary look up is one of the two principal ways of spelling error detection and correction. Irrespective of the technique used, one of the aims of a spell-checker is to provide a small set of correct alternatives for an erroneous string, which includes the intended word. If the number of correct alternatives becomes one then the correction can be done automatically. Even if the number is small, it is manually convenient to choose the intended word from this small subset.

II. PROBLEM DEFINATION

A typical spell checking application, presents a list of alternatives for each misspelled word encountered in a document. The user in turn either selects one of these words, or decides to retain and treat the current word as a valid one. Some checkers also allow the user to add words to the lexicon of correct words thereby enhancing the vocabulary. Spell checking techniques can be broadly classified into three categories.

2.1. Non-word error detection

This involves detection of non-words[1], i.e. words not present in the lexicon of valid words, or misspellings. The most commonly used techniques to detect such errors are n-gram analysis and dictionary look-up. The latter employs efficient dictionary lookup/pattern-matching algorithms (such as hashing techniques,tries, finite state automate, frequency ordered binary search trees, etc.),dictionary partitioning schemes and morphological processing techniques,whereas the former often makes use of frequency counts or probabilities of occurrence of N-grams in a large corpus of text.

2.2. Isolated-word error correction

In this category, correction[3], usually in the form of suggestion generation is performed without taking into account the textual or linguistic context in which the words appear. Minimum edit distance techniques, Similarity key technique,Rule-based methods, N-gram Probabilistic and Neural Network techniques fall into this category.

2.3. Context-dependent word correction

Context-dependent word correction[1] takes care of real-word errors (errors that result in another valid word) and non-word errors, which have more than one potential correction. Traditional Natural Language Processing (NLP) and Statistical Language Processing (SLM) form the two main approaches being explored.

III. ASSAMESE - The Script and the Language

Basically an Indo-Aryan language, Assamese has derived its phonetic character set and behavior from Sanskrit. It is written using the Assamese script, which is similar to Bangla. Assamese is written from left to right and top to bottom,in the same manner as English. A large number of ligatures are possible since potentially all the consonants can combine with one another. Vowels can either be independent or dependent upon a consonant or a consonant cluster. The Assamese alphabet has consonant letters, independent vowel letters, dependent vowel signs(matras), punctuation and numerals. The Assamese alphabet is almost identical to the Bengali alphabet except for the letter r [ra]in Assamese,which is used in place of r [ra] in Bengali, and the letter [wa] which is used only in Assamese.

3.1. Grammatical features

(1) Personal markers used in various kinship terms in connoting the age and rank of both the speaker and listener form a unique feature of the languages spoken in North-Eastern India such as Assamese, Bodo, Karbi, and Mising. For instance, in Assamese, in the phrase, □□□□□ □□□□□□□ [Phonetic pronunciation: [tumar deutar] meaning your father, ra is the personal deictic or marker.

(2) The process of negation of verbs in Assamese is another feature, which clearly demarcates it from the rest of its sisters in new Indo-Aryan languages and other Dravidian languages. In Assamese □ [n] is attached to the verb followed by a vowel, which is the exact copy of the vowel of the first syllable of the verb, as in □□□□□□ [nalage] do not want' (1st, 2nd, 3rd person) □□□□□□ [nilikho] will not write' (1st person).The various negative markers in Assamese are: □□ [no] □□ [na] □□ [ni] □□ [nu] □□ [ne].

(3) The use of plural suffixes is another feature of Assamese. For instance, the entire bound forms such as □□□ [xokol] □□□ [bur] □□ [hot] □□□ [zak] etc..denote plurality and are suffixed to a noun or a pronoun.

(4) The extensive use of classifiers is another feature of Assamese. For almost everything or every shape the language uses a different classifier. Table 1 lists a few of such classifiers.

Table 1:

Classifier	Follows
□□ [zan]	A definite noun (masculine gender)
□□□[zani]	A definite noun (feminine gender)
□□ [to] □□ [khan]	A common noun (usually an neuter)

(5) The classifiers are also combined with all types of nouns and numerals occurring in the language resulting in the following type of grammatical constructions. □ [e] + □□ [zan] + □□□□□ [manuh] (numeral +classifier + noun) □□□□□ [manuh] + □ [e] + □□ [zan] (noun + numeral+ classifier)

3.2.Morphological Analysis in the Indian Context

Indian languages, like many other languages of the world have a relatively free word order. They also have a rich system of case endings and post-positions (collectively called vibhakti). The majority of grammar frameworks are designed for English and other positional languages. As far as morphological processing of Indian languages is concerned, the team at the Department of Computer Science and Engineering, Indian Institute of Technology Kanpur has already initiated some work. They have adopted the Paninian Grammar approach, for morphological processing of languages such as Hindi, Telegu, Kannada, Marathi,Bengali and Punjabi.

IV. Morphological Analysis of Assamese used for spell checking

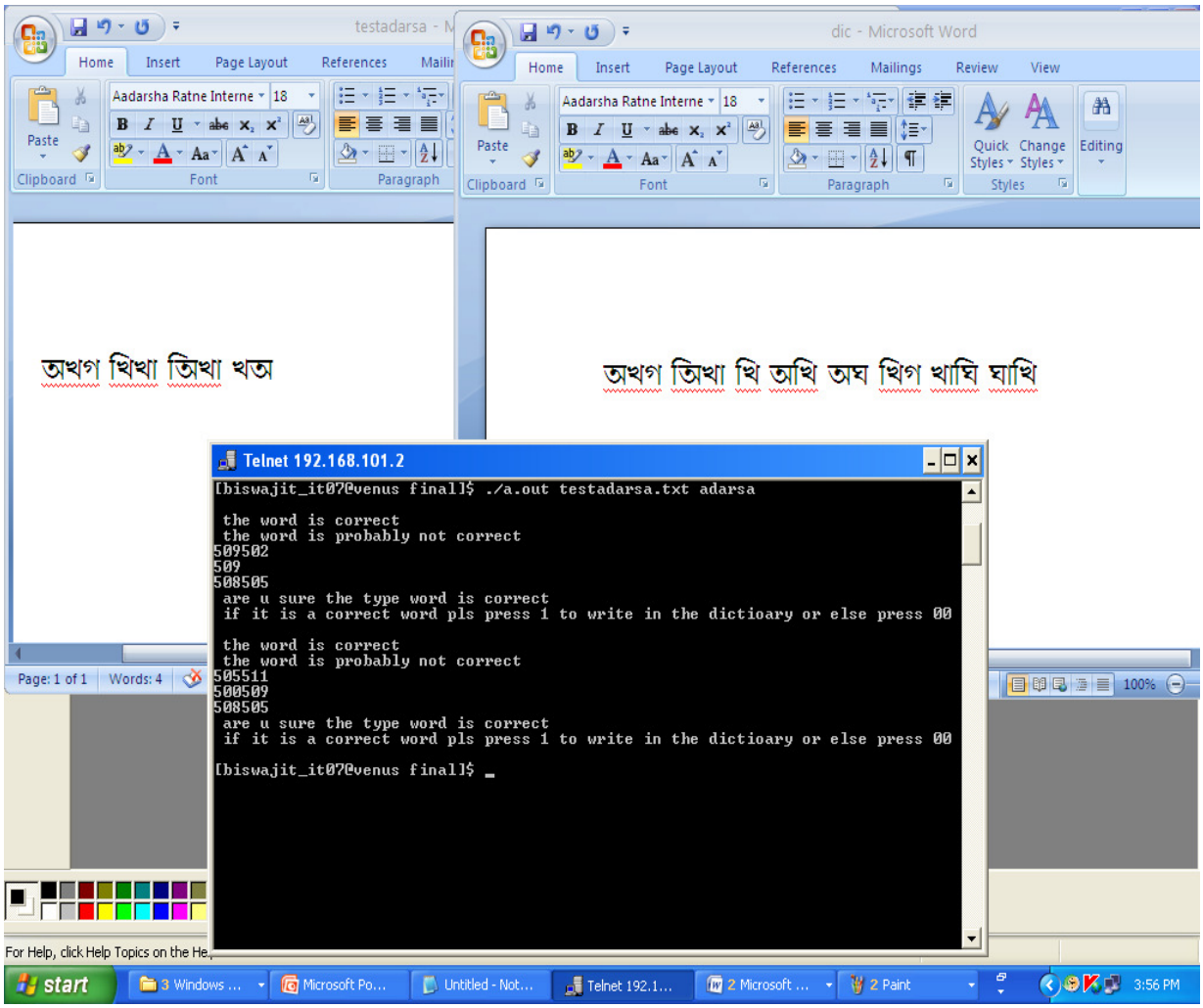
A system of Suffix stripping has been used for the development of a Morphological Analyzer for Assamese. It is based on two kinds of knowledge. The dictionary contains knowledge either about a stem or a word chosen to be the reference form. The dictionary also contains information about the syntactic category and the grammatical features of the word. All the possible inflections[2] that can occur are added to the knowledge base. The algorithm for stripping is more complex than that for a Finite State Automaton. At the top level, it consists of finding a rule to be triggered, matching the suffix given in the rule with the suffix of the input word, and then substituting the new suffix to give a base word to look up in the dictionary. The same technique is again applied for the prefixes to derive the root word. For example: □□□□□□□□□ [soawalibor] meaning 'the girls', the morphological analyzer returns the root word □□□□□□ [soali] meaning 'girl'. Similarly for the word □□□ [bola] meaning 'let us go' the root word □□ [bol] is returned.

4.1. Dictionary building and searching

Another related work is how to construct the Dictionary. The Dictionary file is initially not in the main memory. Initially when the program runs the content of the Dictionary file builds a binary search tree according to the value used in the encoding of words. It makes the searching little easier and the searching time complexity improved.

4.2. New word entry in dictionary and Suggestions Generation for incorrect word

Two important works are New word entry in the dictionary and suggestion generation. For new word entry if a word is not found in the database and if the user is sure that the word is correct then he is allowed to enter the new word in the dictionary .The new entry in the dictionary is possible from any font which are used by the user and the available in the transliteration. This entry will place



the word in proper position in the binary tree in the memory. In next time when ever the word again encounter then the spell checker must recognize the word as a valid word. Suggestion of similar valid word must be produced to the user if the spell checker encountered an invalid word. This generation of similar word is based on the similarity matching algorithm.

REFERENCES

[1] Monisha Das, S. Borgohain, Juli Gogoi. Design and Implementation of a Spell Checker for Assamese, In Proceedings of the Language Engineering Conference, 2002
[2] Bidyut Baran Chaudhuri.Towards Indian Language Spell-checker Design, In Proceedings of the Language Engineering Conference
[3] Bhupesh Bansal, Monojit Choudhury, Pradipta Ranjan Ray, Sudeshna Sarkar and Anupam Basu. Isolated-word Error Correction for Partially Phonemic Languages using Phonetic Cues
[4] PRAHALLAD Lavanya, PRAHALLAD Kishore, GANAPATHIRAJU Madhavi. A simple approach for building ransliteration editors for Indian languages

