Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 3; January-March, 2015 pp. 191-195

http://www.krishisanskriti.org/acsit.html

State of the Art of Spell Checker Design for Indian Languages: A Survey

Shivani[1] and DharamVeer Sharma[2]

[1]M. Tech, Department of Computer Science, Punjabi University Patiala, Punjab
[2]Department of Computer Science, Punjabi University Patiala, Punjab
E-mail: [1]shivanigoyal95@gmail.com, [2]dveer72@hotmail.com

Abstract—*For most of the common desktop applications, machine translation systems, office Automation systems, Search engine etc. spell checker plays a very important role. Commercially it is mainly concerned with practical issues of fast response time, reduced memory requirements and the user interface. Spell Checker is a basic tool to identify misspelt words in the text and provides suggestions for them from the database. The task of spell checker is vital in providing correct and quality information through text. The Indian languages are diverse and complex. Different Indian languages are written in different scripts (Malay & English are in roman script, Punjabi is in Arabic script, Hindi is in Devanagari script). Some languages contain half characters, conjuncts which increases the language complexity. Therefore, a large number of spell checker systems (likeSUDHAAR[1], Annam[3], Shabdkosh.com etc.) have been developed for different Indian languages. The user has difficulty to understand, compare and select the most appropriate spell checker due to this diversity. Thus, this survey paper provides a brief overview on error detection and error correction techniques and their capabilities and analysis of available spell checkers in Indian languages and predicts the efficiency of the systems through their ranking.*

1. INTRODUCTION

Now days, Computer technology has become a part of human life. Most of computer applications (like word processor, email, blog writing, keyword searching) required a Spell checker to detect incorrect words (due to mistyping or lack of knowledge of language) and to reduce the effort and time of user.

A Spell Checker is a program that checks the spelling of words in a text document and provides suggestion for incorrectly spelled words in a text document. The task of spell checker is vital in providing correct and quality information through text.

Spell Checker follows basic Mechanisms[5]:-

1. Read a word as input from a text document.
2. Preprocess the word.
3. Check the word whether that word is available in database.
4. If it is present then go to next one.
5. If word is not available then spell checker will check the nearest matching pattern with it and add it in the form of suggestions.

A Spell Checker has three component: An error detector that detects misspelled words, a candidate spelling generator that gives spelling suggestions for the detected misspelled words and an error corrector that select the best correct spelling suggestions out of the list of candidate spelling. Dictionary is used as database in every Spell Checker.

The first spell checker was only designed for English and languages similar to English. But, due to advancement in the field of programming, there are many spell checkers for Indian languages are developed besides the complexities found in them. In this survey paper, various techniques of error detection, error correction and available spell checkers in Indian languages are discussed and efficiency of spell checkers is evaluated through their ranks.

Error Analysis

Error is defined as a measure of the estimated difference between the observed and calculated value. Most common error occurs due to spelling or typing mistake. There are two types of errors, Real word error and Non word error.

Further, According to **Damerau[6]** spelling errors are classified as following:-

Typographic Error

These errors occur when the correct spelling of the word is known, but the word is mistyped by mistake. These types of errors are mostly related to the keyboard.

a. Insertion Error: Insertion error occurs due to insertion of at least one extra letter in the desired word. For example: पंखा ->पंख

b. Deletion Error: Deletion error occurs due to deletion of at least one letter from the desired word.

For example: सब->सेब

c. Substitution Error: Substitution error occurs when one or more letters are replaced by some another letter. For example: हंस->हँस, मुम्बई ->मुंबई.

The substitution errors are mostly related to following reason:

• Different Way to Write Same Word. For example: लिये ->लिए, बहुयें ->बहुएँ

• Vowels with similar sounds. For example: ि ->ी, े ->ै, ं ->ँ

• When a consonant set combines with the pancham Varna (ङ, ञ, ण, न, म), Then it can be shown as an anuswar over the consonant preceding it. For example: चञ्चल -> चंचल, कङ्गन ->कंगन.

d. Transposition Error: Transposition error occurs when two adjacent letters are written in swapped way. For example: कलम->कमल

e. Run- on Error [7]: Run-on errors occur when two or more valid words are erroneously typed side by side without a space in the middle of it. For example: इसके ->इसके, उसकी ->उसकी.In the explained examples: इस, के, उस, कीare four different words.

f. Split word Error [7]: These errors occur when there is some additional space is embedded between the parts of the word. For example: इसके ->इसके. In some cases, split word errors may also give rise to real word errors.

2. PHONETICALLY SIMILAR CHARACTER ERROR

Phonetic error occurs when the correct spelling of the word is known but the word is mistyped by mistake due to same pronunciation. It can be categorized into following types:

- Class 1:ज->झ, ब->व, न->ण, ग ->घ
- Class 2: फ़ ->फ, ज़->ज, ग़ ->ग, ड़ ->ड
- Class 3:चञ्चल -> चंचल, पण्डित->पंडित.
- Class 4: ु-> ू,ं->ँ, े->ै, ि->ी.

3. SPELL CHECKING STRATEGY

A. Error Detection

Error detection is a process of detecting the misspelt words in the text with help of database. There are two efficiency techniques for detection such type of errors.

1. N-gram Analysis Techniques[8]: N-gram is a method to find incorrectly spelled words in a mass of text. N-gram is a set of consecutive characters taken from a string with a length of where n is a positive integer. N-gram tables can take on a variety of forms. The easiest is a *binary bi-gram* array which is 2D arrays whose elements represent all possible two letter combinations of the alphabet. The value of each character in the array is set to either 0 or 1 depending on whether that bi-gram occurs in at least the word in a predefined lexicon or dictionary .A *binary tri-gram* array would have three dimensions. The above arrays are **non-positional** binary n-gram arrays because they do not represent the position of the n-gram within a word. The most of the structures of the dictionary can be stored by a set of **positional** binary n-gram array. For example, in a positional binary tri-gram array the element at position a, b & c would have value 1 if only if there exists at least one word in the dictionary with the letters l, m and n in positions a, b and c. The trade-off for representing more of the structure of the dictionary is the increase in storage space required for the complete set of positional arrays. Any word may be checked for errors by simply looking up its corresponding entries in binary n-gram arrays to make sure they are all 1's(true).

2. Dictionary Lookup: A dictionary is a list of words that are assumed to be correct. Dictionaries may be represented in many forms, each with their own characteristics like speed and storage requirements. The most common method of detecting errors in a text is simply to look up every word in a dictionary. The drawbacks of this method are to keep a dictionary up to date, and sufficiently extensive to cover all the words in a text. At the same time, response time also increases with the increase in size of dictionary. Dictionary lookup and construction techniques must be according to the purpose of the dictionary. Too small a dictionary may give the user too many false rejections of valid words, too large it may accept a high number of valid low-frequency words. The most common used technique to gain random and fast access to a dictionary is **Hash Table.** To lookup Input string, one has to compute its hash address and retrieve the word stored at that address in the pre constructed hash table. If the word is different stored at the hash address from the Input string or null, a misspelling is indicated. For store a word in the dictionary first calculate each hash function for the word and set the vector entries

corresponding to the calculated values to 1(true). For find if a word belongs to the dictionary or not, we calculate the hash values for that word and look in the vector. If all entries corresponding to the values are 1(true), then the word belongs to the dictionary, otherwise not. The main disadvantage is the need to devise a clever hash function that avoids collisions without large hash table.

B. Error Correction

Error correction consists of two steps: the generation of candidate corrections and the ranking of candidate corrections. The candidate generation process usually makes use of a precompiled table of legal n-grams to locate one or more potential correction terms. The ranking process is used to invoke some lexical similarity measure between the misspelled word and the candidates or a probabilistic estimate of the likelihood of the correction to rank order the candidates. There are following techniques used for error correction:-

1. Morphological Analysis: The morphological analyzer processes the word and delivers the root word, along with the list of possible valid suffixes and prefixes. Out of these affixes, the ones that closely match the affixes of the misspelled word are selected and by attaching these affixes to the root word a list of suggestions are arrived.

2. Soundex Method: The soundex code of the misspelled word is generated, according to the devised coding scheme. An approximate match is performed with the Soundex codes of all the valid words present in the dictionary. A list comprising of words Soundex codes of which closely match those of misspelled word is built.

3. Edit Distance: Edit distance is a simple technique. It is defined as the minimum number of editing operations (i.e. insertions, deletions and substitutions) required to transform one string into another. In other word, by applying the four editing operations, which commonly generate typographic errors i.e. addition, deletion, substitution and transposition of letters, another list of suggestions are arrived. Edit distance is useful for correcting errors resulting from keyboard input, since these are often of the same kind as the allowed edit operations. It is not quite as good for phonetic spelling error correction, especially when the difference between spelling and pronunciation is big as in English or French.

4. Rule Based Technique: Rule-based techniques are algorithms or heuristic programs that represent knowledge of common spelling error patterns in the form of rules for converting misspellings into valid words. The candidate generation process consists of applying all applicable rules to a misspelled string and retaining every valid lexicon word those results. Ranking is frequently done by assigning a numerical score to each candidate based on a predefined estimate of the probability of having made the particular error that the invoked rule corrected.

5. N-gram Based Techniques: N-gram analysis has already been described earlier in error detection module. Letter n-grams, including trigrams, bigrams, and/or unigrams, have been used in a variety of ways in text recognition and spelling correction techniques. They have been used in OCR correctors to capture the lexical syntax of a dictionary and to suggest legal corrections.

4. ANALYSIS OF AVAILABLE SPELL CHECKERS IN INDIAN LANGUAGES

The Indian languages are diverse and complex. Some languages contain half characters, conjuncts which increases the language complexity. Different Indian languages are written in different scripts (like Malay & English are in roman script, Punjabi &Urdu are in Arabic script, Hindi is in Devanagari script). Therefore, a large number of spell checker systems have been developed for different Indian languages using above techniques. This section provides the analysis of available spell checkers in Indian languages and predicts the efficiency of the systems through their ranking.

A. Sanskrit Spell Checker

Sanskrit called the mother of all Indian languages. All the Indian languages are expected to be derived from Sanskrit language. Sanskrit is free ordering language (or syntax free language) &there is no ambiguity in the form of the words even if the order changes. A Morphological based spell checker has been designed for Sanskrit. An algorithm has been designed to check the validity of a word. If the word is not found in the vocabulary, then the word is scanned from right to left to identify a valid suffix string such that it occurs in at least one rule. If such a rule is not found then the word is rejected as invalid and the system proceeds further. After this, at rear end of the word a transformation is carried out to obtain a pruned word. If it is found in the vocabulary or any other rule is applicable to the word obtained then the word is valid. But if no rule is applicable then the word is declared as invalid and a suggestion list is generated. The Spellchecker is implemented in Java. For display, the documents are converted into Unicode. The morphological analysis of a word serves as a foundation for POS- tagging. Similarly, it is being used in stemming for searching root words in Sanskrit Wordnet.

B. NAAVI (Oriya Spell Checker)

It is online spell checker. It deals with the error detection and either automatic or manual correction for the words that have been misspelled. Some algorithms have been developed in order to find the most accurate and appropriate results. The searching techniques that have been employed are very fast in

processing such that it processes 170000 Oriya words for each misspelled word. The words in the dictionary are stored according to the word length for effective searching. It also takes help of the Oriya Morphological Analyzer for ascertaining the mistakes of inflection, derived and compounding words. Oriya Spell Checker is being successfully running embedded in a word processor. Using the similar technique a Hindi Spell Checker with 270000 words and an English Spell Checker with 20000 words has also been designed for word processor. This software is designed using Java and Java Swing for both the Windows-98/2000/NT and the Linux O/S.

C. Malayalam Spell Checker

In Malayalam, a large number of words can be derived from a root word, a purely dictionary based approach for Spell Checking is not practical. Hence a 'Rule cum Dictionary' based approach is followed. The grammatical behavior of the language, the formation of words with multiple suffixes and the preparation of the language module are dealt with in detail. The different modules in the Spell Checker Engine viz. Morphological Analyzer, Morphological Generator, Error detection and suggestion generation is done. It splits the input word into root word, suffixes, post positions *etc.* and checks the validity of each using the rule database. Finally it will check the dictionary to find whether the root word is present in the dictionary. If anything goes wrong in this checking it is detected as an error and the error word is reprocessed to get 3 to 4 valid words, which are displayed as suggestion. This spell checker is a subsystem developed by CDAC, Tiruvananthpuram, which could be integrated in larger applications like Microsoft word or any word processor as a macro. While running as a macro in word, it functions as an offline spell checker in the sense that one can use this software with a previously typed text file only. Both off line and online checking are possible when it is integrated with the text editor. It generates suggestions for wrongly spelt words.

D. Annam (Tamil Spell Checker)

Tamil Spell checker helps the user to identify most of mistyping error. It is available both offline and online. The task implemented in Tamil Spell checker are Case marker, postposition checking for nouns, Adjective checking for nouns, Case ending and PNG marker checking for verbs, Adverb checking, and Adjacent key errors checking. The applications of the Tamil Spell checker are Word processors, search engines, information filtering and extraction systems, and machine translation systems. The modules extract the root word from the given word (noun/verb) with the help of Morphological Analyzer and the root word is checked in the dictionary and if found, the word is termed as correct word. Otherwise, the correction process is activated. The correction process includes error handling and suggestion generation modules. After finding the type of error, the right form of suffixes; nouns or verbs are given as input to suggestion generation module. With the help of Morphological Generator, the correct word is generated. This module also handles the operations like select, change or ignores the suggested word and adding the word to the dictionary.

E. Akshara (Telugu Spell Checker)

A pure corpus based statistical stemming algorithm has been developed for Telugu. The performance of this stemmer for the spell checking application has been studied in various combinations with dictionary and morphology based approaches. Large scale spelling error data has been obtained from 10 Million word Telugu corpus. The raw corpus as it was typed has been compared with the final version after three levels of proof reading and certification by qualified and experienced proof readers. A number of tools have been developed to prepare such a data. Since words are large and complex and hence too numerous in Telugu and proper morphological analysis is difficult, it would be useful to perform studies at lower levels of linguistic units. The syllable level is a natural choice since writing in Indian languages is primarily syllabic in nature. N-gram models have been built at syllable level.

F. Assamese Spell Checker

The Spell Checker exists in the form of separate modules for error detection and correction, as well as a stand-alone system in which all the spell checking routines have been integrated. The *non-words* are detected by looking up text words in a dictionary of valid words. The dictionary used is actually a word list of around 72000 Assamese words. A hash table has been used as a lexical lookup data structure. A Soundex encoding scheme for Assamese has been designed based on the encoding scheme for English which comprises of a set of rules for encoding words and 14 numerical codes. The Soundex code of the misspell word is computed, & the dictionary is searched for words, which have similar codes. For ranking the suggestions, the technique of Levenshtein edit distance technique is used.

G. Marathi Spell Checker

In this ongoing project, a standalone spell checker is being built for Marathi. From the Central Institute of Indian Language (CIIL) corpus 13000 distinct words approximately have been listed. Similarly different Marathi texts are being used to build up the Dictionary. Morphological Analyzing is also being carried out on the words listed in the dictionary. For example, an automatic grouping algorithm identified 3,975 groups out of 12,886 distinct words. First word is usually the root word. Thus, there are approximately 4000 root words from Marathi corpus. A manual proof reading will be done on these results. Further enhancements of derivational morphology will help in increasing the vocabulary. Besides enhancing word lists and rules, enhancements for representing rules for ordering of multiple

suffixes in all part of speech categories are required. More elaborate orthographic rules need to be incorporated. Morphology based spellchecker may be extended to include further syntactic and semantic analysis. Besides spellchecking, the morphology based analysis is currently being used in a few applications at the Center for Indian Languages. A motivation behind the stand-alone spellchecker is that it can be used without an editor through a packaged interface, or it can be integrated with other compatible applications such as OCR.

Table 1: Rank of Marathi Spell Checker

Name of Spell checker	Khandbahale.com
Global rank	320,723
India rank	30,873
Daily page viewer per visitor	2.50

H. Bangla Spell checker

The first Bangla Spell Checker was designed by BidyutBaranChaudhuri. The technique works in two stages. The first stage takes care of phonetic similarity error. For that the phonetically similar characters are mapped into single units of character code. A phonetically similar but wrongly spelt word can be easily corrected using this dictionary. The second stage takes care of errors other than phonetic similarity. It works in both online and offline mode. The spell checker was embedded in a word processor. If there is only single error in the misspelled word then the most appropriate suggestion is found in the top four words of the suggestion list. But suggestions cannot be given on some inflected words. It also has a special feature to add words in the dictionary against which spellings are checked. The basic purpose of the spell checker is to detect the erroneous word and either suggests correct alternatives or automatically replace it by the appropriate word.

Table 2: Rank of Bangla Spell Checker

Name of Spell checker	Shuddhoshabdo.com
Global rank	19,779,445
India rank	N/A
Daily page viewer per visitor	1

I. Hindi Spell Checker

The important factors in the design of Hindi spell checker are: the dictionary data structure, inclusion of the equivalent English words, features of auto correction in the Hindi word in case of wrong entry by the user. The wrong entry here means spelling mistake of one character missing, matra mistakes (more than one also), mistake in the usage of half consonant.

Table 3: Rank of Hindi Spell Checker

Name of Spell checker	Shabdkosh.com
Global rank	3,497
India rank	296
Daily page viewer per visitor	4.62

J. Akhar (Punjabi Spell Checker)

Akhar is a Bilingual spell checker. It is available offline. It is a language sensitive spell checker i.e., if text is entered in English then English spell checker is invoked or if text is entered in Punjabi then Punjabi Spell Checker is invoked. It is a font independent spell checker and it can work on any popular Punjabi fonts such as, Anantpur Sahib, Amritlipi, Jasmine, Punjabi, Satluj etc. This removes the contrast on the user to type the text in pre-defined font only.

5. CONCLUSION

This paper presents the area of Spell checking Strategy and Analysis of errors. It has surveyed on various errors detection and error correction techniques that are helpful in finding the errors. This paper also mentioned the influence of Indian languages and has surveyed on the available spell checker in different Indian languages. In future, author will design and implement new Spell Checker for Hindi language.

REFERENCES

[1] RupinderdeepKaur and Parteek Bhatia, “Design and Implementation of SUDHAAR-Punjabi Spell Checker,” International Journal of Information and Telecommunication Technology, Vol. 1, Issue 15, pp.0976-5972 May, 2010.

[2] Daniel Jurafsky, James H. Martin, Speech and Language Processing, PEARSON, 2nd ed, September 1999.

[3] T.V. Geeta, RajaniParthearath, “Tamil Spellchecker”, Resource centre for Indian language Technology Solution, Tamil Internet, Chennai, Tamil Nadu, India, 2003.

[4] Neha Gupta, PratisthaMathur, “Spell Checking Techniques in NLP: A Survey,” International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 12, pp. 217-221, December 2012.

[5] Gurpreet Singh Lehal, “Design and Implementation of Punjabi Spell Checker”, International Journal of Systemics, Cybemetics and Infomatics, pp.70-75, Jan. 2007.

[6] F.J. Damerau, "A technique for computer detection and correction of spelling errors", Communication of ACM, pp. 171-176, 1964.

[7] P.Kundra, B.B.Charudhari (1999), “Error pattern in Bangla text”, International Journal of Dravidian Linguistics, 28(2), 1999.

[8] E.M. Riseman, A. R. Hanson, "A Contextual Post Processing System for Error Correction using binary n-grams", IEEE Transactions on Computer, Volume. 23, Issue 5, pp. 480-493, May 1974.

[9] SanghamitraMohanty,“Analysis and Design of Oriya Morphological Analyzer: Some Tests with OriNet”, TDIL Newsletter, March 2004.

[10] NamrataTapaswi, Suresh Jain, VaishaliChourey, “Morphological-based Spellchecker for Sanskrit Sentences”, International Journal of Scientific & Technology Research, Volume 1, Issue 3, pp. 1-4, April 2012.

[11] Monisha Das, S. Borgohain, JuliGogoi, S. B. Nair, "Design and Implementation of a Spell Checker for Assamese", Language Engineering Conference(LEC'02), pp.156, Dec. 2002.

[12] Veena Dixit, Satish Dethe, Rushikesh K. Joshi, “Design and Implementation of a Morphology-based Spellchecker for Marathi”, TDIL Newsletter, 2006.

[13] Santhosh. T. Varghese; R. Ravindra Kumar; K.G.Sulochana, “Malayalam Spell Checker”, International Conference on Universal Knowledge and Language, RCILTS-Malayalam. Goa, pp.5.3.38, November.2002