Proceedings of the
International Conference on Mechanical Engineering and Renewable Energy 2011
(ICMERE2011) 22- 24 December 2011, Chittagong, Bangladesh

ICMERE2011-PI-129

DEVELOPING A BANGLA SPELL CHECKER WITH POSSIBLE SUGGESTIONS

Sujan Chowdhury[1], **Kaushik Deb**[1], **A.H.M. Ashfak Habib[1], and Alok Kumar Chowdhury**[2]

[1]Faculty of CSE, Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh
[2]Faculty of CSE, Premier University, Chittagong, Bangladesh

Corresponding Author: Dr. Kaushik Deb, E-mail: debkaushik99@cuet.ac.bd

Abstract*- Spell checker is so important for finding out the misspelled/error word from a document, so that the document becomes error free. In this respect, we propose a framework for detecting the misspelled words and also give suitable suggestions for misspelled words. Our propose method with a Graphical User Interface via an editor. The proposed framework presents a new algorithm based on the existing algorithms to generate suggestions for miss-spell words along with a complete design of implementation of a spell checker which is much efficient for Bangla. The proposed approach consists of three basic parts; initially, we proposed to separate the words when user typed in the editor and check it with the database. And then generate the suggestion for misspelled word using LCS (Longest Common Subsequence) and soundex algorithm. Finally, we rank the suggestion. The proposed method is able to check the error word during typing time or a document which already typed. The error words show in red color and generate suggestion after right click on the error words. To test the proposed framework and some results are presented to prove its effectiveness.*

Keywords: Miss-spelled Words, Bangla, LCS, Soundex, Generate Suggestions.

1. INTRODUCTION

The problem of detecting error in words and automatically correcting them is a great research challenge. Its solution has enormous application potentials in text and code editing, computer aided authoring, optical character recognition (OCR), machine translation (MT), natural language processing (NLP), database retrieval and information retrieval interface, speech recognition, text to speech and speech to text conversion, communication system for the disabled (e.g. blind and deaf). The first issue concerns the error patterns generated by different text generating media such as typewriter and computer keyboard, typesetting and machine printing etc. Dictionary look up is one of the two principal ways of spelling error detection and correction. The proposed framework for bangle spell checker consists of three parts as shown in Figure 1. Initially, we separate the word; we work with cursor position because change in the text will occur around the cursor position. Secondly, when we generate suggestion for misspelled word we use longest common subsequence value between two words, if the length between those two words less than two; we add it to our suggestion list to show that it may be the correct word. In this step the no of suggestion word may become more, so we apply soundex algorithm on those suggest words and generate final suggestion list whose spelling are near to the misspelled words. Finally, we rank the suggestion according to ranking list using edit distance algorithm. Our propose framework shown in Figure 1.

We apply our proposed algorithm on some document, specially on ”maa” written by famous writer Anisul Hoque and on some newspaper and get better result. As there are only few works have been done in the respective field for bangla , so we have got great desire to do something better in the field. And the previous implementation has many limitations. So what we have tried to solve these limitations so that we have got the correct suggestion for the misspelled words and develop a well bangla spell checker with enrich able dictionary. As we see the application of bangla language in so many area and making mistake in word become shameful and may be sometimes harmful. So it will be great impact in the field. The basic idea behind spelling suggestions using phonetic encoding is quite simple:

1. Encode the input word using phonetic coding rules;
2. Look up a phonetically encoded lexicon for words with the same code; and
3. Create an ordered list, i.e., suggestions, from the result using some heuristic.

In this paper, we also introduce a phonetic encoding for Bangla, and then demonstrate how a spelling checker would use it to produce suggestions for misspelled Bangla words.

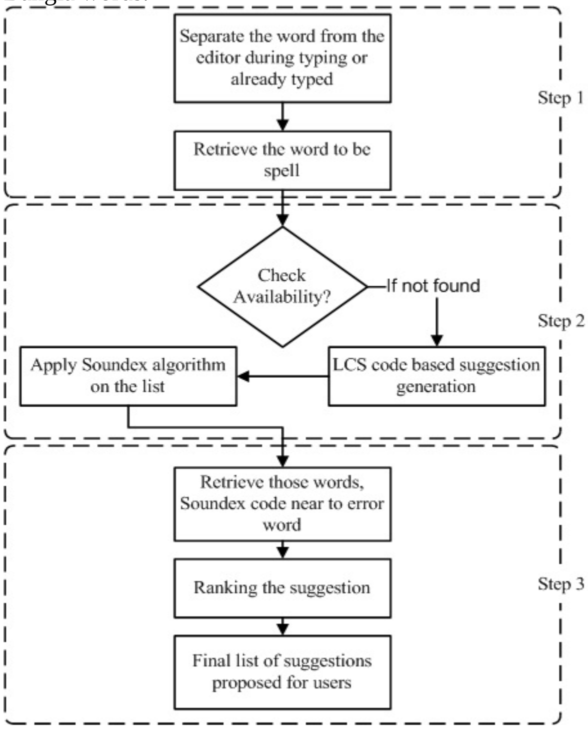

Separate the word from the editor during typing or already typed
Retrieve the word to be spell
Step 1
Check Availability?
If not found
LCS code based suggestion generation
Apply Soundex algorithm on the list
Step 2
Retrieve those words, Soundex code near to error word
Ranking the suggestion
Final list of suggestions proposed for users
Step 3

Fig. 1: Proposed framework for Bangla Spell Checker

2. BACKGROUND AND PRESENT STATE OF THE PROBLRM

Bangla, also known as Bengali, is the language of approximately 210 million people, the majority of whom live in Bangladesh and in the Indian state of West Bengal, making it the 4th most widely spoken language in the world. Detecting a misspelled word for a language is trivial for typographical errors and the cognitive phonetic errors. But cognitive homonym errors, which are real word errors1, cannot be detected easily. We need to consider the context of a word to detect a misspelled word in this case. For Bangla, approximate string matching algorithms [1] and a direct dictionary look up method [2] has been used so far for the detection of typographical errors and cognitive phonetic errors. In our spelling checker, we used the direct dictionary look up method for detecting a misspelled word. But none of these methods, including our method, deals with homonym errors. Some work in bangla for Indian written language done earlier [3]. In Bangladesh some natural language processing centre (NLP) deal with bangla perform some phonetic encoding and try to minimize the error word in bangla [4]. They propose a spell checker [5] according to their phonetic rules which cannot generate suggestions in runtime and their generated suggestion is not so efficient in best of our knowledge. At present total bangla spell checker is under development. It will detect error not only word but also try to detect the sentence error in bangla in text document future. There are also some work existing on spell checker Phonetic Spelling and Heuristic Search [6]. All of the spell checker [7] existing can't generate run time suggestion like Microsoft office and the generation of Suggestion is not so efficient.

3. TYPES OF ERROR

There are so many possibilities of error human [8] may perform during typing, it may be cause due to not careful and it also may be occur if he or she do not know the correct spelling of word. In ACM survey K.Kukich [9] breaks down human typing errors into two classes.

I. Typographical error
II. Cognitive error

- Typographical error

Generally occur due to people's mistakes while typing.

e.g. misspelling 'বানান' as 'বানন'.

- Cognitive errors

Caused by writers who do not know how to spell the word

e.g. misspelling 'ভুল' as 'ভূল'

Cognitive error can be classified as two types:

a) Phonetic error
b) Homonym error

- Phonetic errors

Substituting a phonetically equivalent sequence of letters

e.g., misspelling 'চল' as 'ছল'

- Homonym errors

Accidentally produce a real word.

e.g., misspelling 'আমার' as 'আমরা'

4. PROPOSED FRAMEWORK

The existing technique sometimes fails to detect error of some word that is phonetically correct but actually not exists. So the suggestion generation is not possible in this case LCS technique is efficient for generating suggestion.

4.1 Longest Common Subsequence

The longest common subsequence (LCS) [10] problem is to find the longest subsequence common to all sequences in a set of sequences.

For the general case of an arbitrary number of input sequences, the problem is NP-hard. When the number of sequences is constant, the problem is solvable in polynomial time by dynamic programming (see Solution below). Assume you have N sequences of lengths. A naive search would test each of the subsequences of the first sequence to determine whether they are also subsequences of the remaining sequences; each subsequence may be tested in time linear in the lengths of the remaining sequences, so the time for this algorithm would be

$$O\left(N \prod_{i=1}^{N} n_i \right)$$

For the case of two sequences of n and m elements, the running time of the dynamic programming approach is O (n × m). For an arbitrary number of input sequences, the dynamic programming approach gives a solution in there exist methods with lower complexity, which often depend on the length of the LCS, the size of the alphabet, or both.

4.2 LCS function

Let two sequences be defined as follows: X = (x1, x2...xm) and Y = (y1, y2...yn). The prefixes of X are X1, 2,...m; the prefixes of Y are Y1, 2,...n. Let LCS (X_i, Y_j) represent the set of longest common subsequence of prefixes X_i and Y_j. This set of sequences is given by the following.

$$LCS(X_i,Y_j)=\begin{cases}\phi & if\ i=0\ or\ j=0\\ (LCS(\mathrm{X}_{i-1},\mathrm{Y}_{j-1}),\mathrm{x}_i) & if\ x_i=y_j\\ longest(\mathrm{LCS}(\mathrm{X}_i,\mathrm{Y}_{j-1}),\mathrm{LCS}(\mathrm{X}_{i-1},\mathrm{Y}_j)) & if\ x_i\neq y_j\end{cases}$$

To find the longest subsequences common to Xi and Yj, compare the elements xi and yi. If they are equal, then the sequence LCS (Xi-1, Yj-1) is extended by that element, xi. If they are not equal, then the longer of the two sequences, LCS (Xi, Yj-1), and LCS (Xi-1, Yj), is retained. (If they are both the same length, but not identical, then both are retained.) Notice that the subscripts are reduced by 1 in these formulas. That can result in a subscript of 0. Since the sequence elements are defined to start at 1, it was necessary to add the requirement that the LCS is empty when a subscript is zero.

4.3 LCS Example

A brute-force approach to solving the LCS problem is to enumerate all subsequence of X and check each subsequence to see if it is also a subsequence of Y, keeping track of the longest subsequence found. Each subsequence of X correspond to the subset of the indices {1, 2, 3…..m}of X.

For example if a user type the word “**computer”** as “**comuter“**

If we apply LCS technique how many word are matched sequentially so that we can understand what was the actual word that is helpful to show the correct suggestion.

The calculation the LCS of a row of the LCS table requires only the solutions to the current row and the previous row.

4.4 Computing the length of the LCS

The following pseudo code finds the set of longest common subsequence length between two strings with dynamic programming.

```
function LCSLength(X[1..m], Y[1..n])
    C = array(0..m, 0..n)
    for i := 0..m
       C[i,0] = 0
    for j := 0..n
       C[0,j] = 0
    for i := 1..m
       for j := 1..n
         if X[i] = Y[j]
            C[i,j] := C[i-1,j-1] + 1
         else:
            C[i,j] := max(C[i,j-1], C[i-1,j])
    return C[m,n]
```

Fig. 2: Pseudo code for compute the LCS length.

4.5 Soundex Algorithm

Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English. The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling. Soundex is the most widely known of all phonetic algorithms and is often used (incorrectly) as a synonym for "phonetic algorithm". Improvements to Soundex are the basis for many modern phonetic algorithms.

The Soundex [11] coding rules as follows:

1. Retain the first letter of the string. The first letter of the name is the letter of the Soundex code, and is not coded to a number.
2. Remove all occurrences of the following letters, unless it is the first letter: a, e, h, i, o, u, w, y.
3. Assign numbers to the remaining letters as follows (after the first letter):
    - b, f, p, v => 1
    - c, g, j, k, q, s, x, z => 2
    - d, t => 3
    - l => 4
    - m, n => 5
    - r => 6
    - h, w are not coded
4. Two adjacent letters with the same number are coded as a single number. Letters with the same number separated by an h or w are also coded as a single number.
5. Continue until you have one letter and three numbers. If you run out of letters, fill in 0s until there are three numbers.

Using this algorithm, both "Robert" and "Rupert" return the same string "R163" while "Rubin" yields "R150". "Ashcraft" and "Ashcroft" both yield "A261".

5. IMPLEMENTION PROCESS

The proposed implementation process follows the following steps. Firstly it detects the misspelled words and then generates suggestion for the misspelled words and gives option to the user to change or replace the word with the correct one

.

5.1 Detect Misspelled Words

When we start to find the error word we deal with the position of the cursor because any change in the text will occur around the cursor, so it is a good policy to work with the cursor position. Here we deal with some change in cursor position which causes the errors are describe details and the method to detect the error. Find out the cursor position in rich textbox is as follows:

int cursor_position = richTextBox1.SelectionStart;

The cusrsor_position returns an integer value within the

richtextbox where the cusrsor now locate.
Now we deal with the change at cursor position

i) If the word change in the last position
Example: এখন একটা দরকারি কথা বলা প্রয়োজ ←
Here the word typed at last position is a wrong word, so we have to understand the change,this is done by check from the last position to backward until a space,new line or some other condition not satisfied.

ii) If the word change in the middle position
Example: এখন একটা দরকাি। কথা বলা প্রয়োজন
⇧

Here the word changed at middle position is a wrong word, so we have to understand the change,this is done by check from the cursor position to backward until a space,new line or some other condition not satisfied and forward until another criteria fulfill.

iii) If a new character insert in the front or delete from the front of word
Example:
এখন একটা দরকারি ।বকথা বলা প্রয়োজন (insert in the front)
এখন একটা দরকারি ।থা বলা প্রয়োজন (delete from the front)

iv) If two string any how added and change the meaning of the correct word, though previously both are correct
Example:
এখন একটা দরকারি কথাবলা (two word added) প্রয়োজন

v) If two string any how detached and change the meaning of the correct word, though previously it was correct
Example:
এখন একটা দরকারি কথাবলা প্রয়ো জন (detached)

The type of error discussed can be detected by the following flowchart shown in Figure 3.

5.2 Generate Suggestion for Misspelled Words

When we understand the error word then our task is to generate the suggestion for the misspelled word, this is done by press right click on the error word . A list of suggestion is shown according to their edit distance, the user choose the appropiate word that will be best match or so many option he may add it into dictionary because no name is entered into database, but if he or she think that his name should store on his pc for further use so it may be stored.The user have also option to ignore it. Now we see how we get the right click position or how the process understand the type of click which is shown in Figure 4.

5.3 Ranking the Suggestion

Once we have a list of suggestions, we need some way to rank them by the most likely to be the correct spelling. My research into the best way to go about this turned up the Edit Distance Algorithm. The edit distance is defined as the smallest number of insertions, deletions, and substitutions required changing one string into another. Here we consider the error length may be greater or less then the actual word, so when add the suggestion we compare the length that will help us to show the correct suggestion.

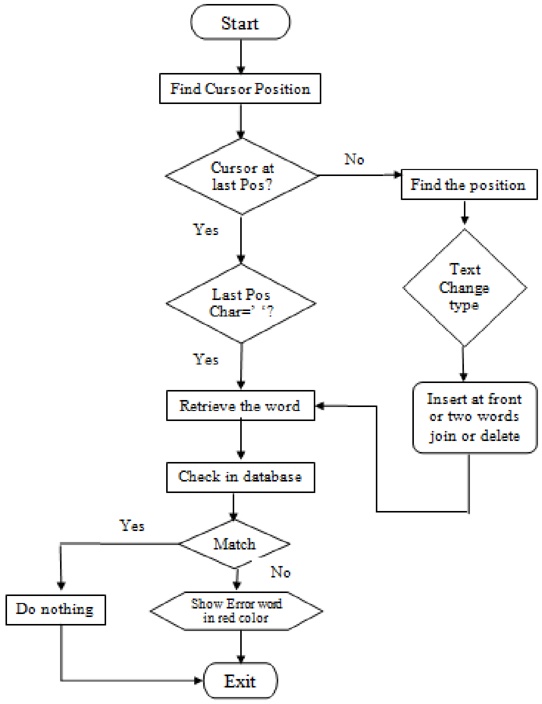


Fig. 3: Flow chart for detect misspelled words

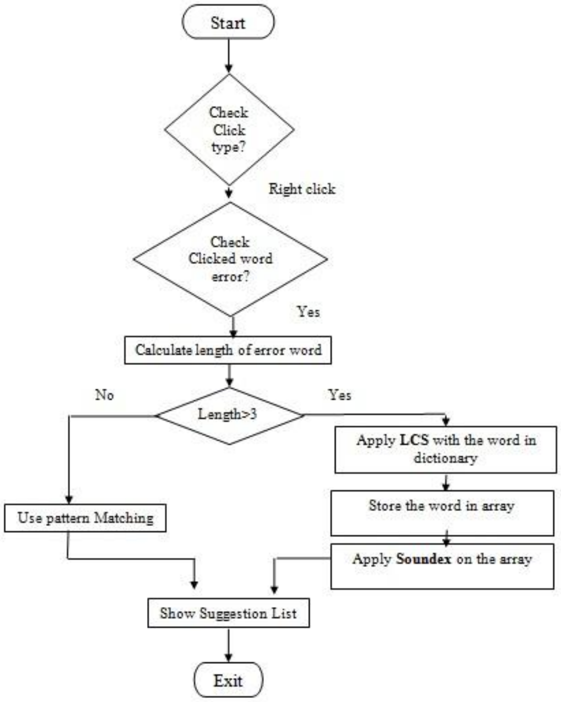

Start
Check Click type?
Right click
Check Clicked word error?
Yes
Calculate length of error word
No
Yes
Length>3
Apply LCS with the word in dictionary
Store the word in array
Use pattern Matching
Apply Soundex on the array
Show Suggestion List
Exit

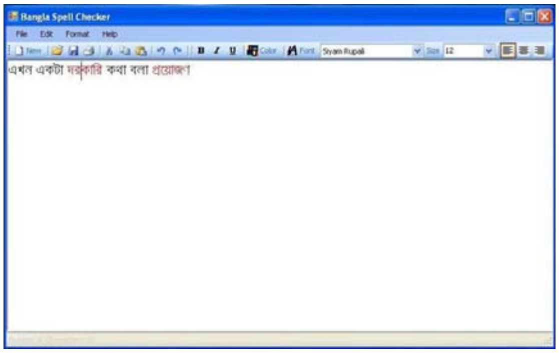
Fig. 4: Flow chart for generating suggestion list

6. EXPERIMENTAL RESULTS AND CONCLUSION

Here we demonstrate some experimental result that we apply on our own editor. The Figure 5 shows the detection of misspelled word during typing. Whereas Figure 6 shows the suggestion list of the misspelled word ‘দরকারি’, when right click is pressed on that misspelled word ‘দরকারি’. Our generated suggestion shows all the possible suggestions for the misspelled word ‘দরকারি’. The list shows that our generated suggestion is almost near to the misspelled word ‘দরকারি’। Similarly the Figure 7 shows the another suggestion list for the misspleed words ‘প্রয়োজন’ . it also shows the resluts near the the misspelled word ‘প্রয়োজন’. We apply it in so many text documnet and get better result hence prove the efficiency of our proposed framework.

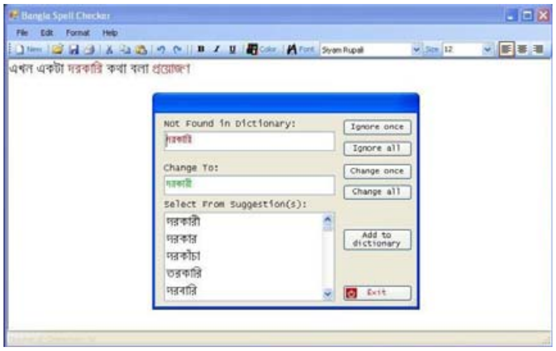
In Conclusion, our goal is to develop a spell checker for bangla which we feel badly inneeded for our daily life. So we try to overcome the commom error and genrate the possible suggestions for those misspelled words and hopefully try to provide better suggestions.

Spell checker is an essential tool for any language. Currently in case of English, spell checker can check both misspelled words and as well as grammatical errors. In best of our knowledge still there is no spell checker which can check grammatical errors in bangla sentence. In near future we hope we will try to develop an editor which will check grammertical errors in bangla sentence according to reknowned bangla grammer book. Subsequently we hope we will develop a bangla dictionary with numerous words.



এখন একটা দরকারি কথা বলা প্রয়োজণ

Fig. 5: Error occurred during typing shown in red color



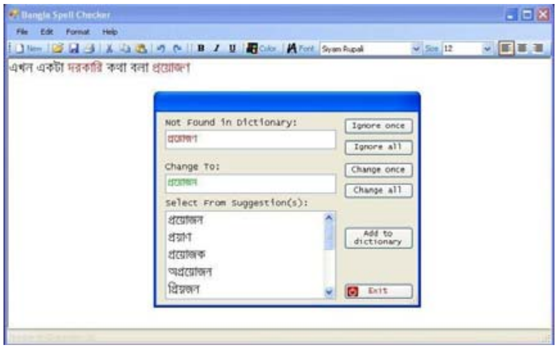
এখন একটা দরকারি কথা বলা প্রয়োজণ

Not Found in Dictionary: দরকারি

Change To: দরকারী

Select From Suggestion(s):
দরকারী
দরকার
দরকাচা
তরকারি
দরবারি

Fig. 6: Generate suggestion for error word ‘দরকারি’ on right click on the word



এখন একটা দরকারি কথা বলা প্রয়োজণ

Not Found in Dictionary: প্রয়োজণ

Change To: প্রয়োজন

Select From Suggestion(s):
প্রয়োজন
প্রয়াণ
প্রয়োজক
অপ্রয়োজন
প্রিয়জন

Fig. 7: Generate suggestion for error word ‘প্রয়োজণ ’on right click on the word

7. REFERENCES

[1] Arif Billah Al-Mahmud Abdullah and Ashfaq Rahman, “A Different Approach in Spell Checking for South Asian Languages”, *Proc. 2nd International Conference on Information Technology for Applications (ICITA)*, China, 2004.

[2] B. B. Chaudhuri, “Reversed word dictionary and phonetically similar word grouping based spell-checker to Bangla text”, *Proc. LESAL Workshop*, Mumbai, 2001.

[3] Aniruddh Bhatt, Monojit Choudhury, Sudeshna Sarkar and Anupam Basu, “Exploring the Limits of Spellcheckers: A comparative Study in Bengali and English”, *Proc. The Second Symposium on Indian Morphology, Phonology and Language Engineering (SIMPLE'05).* Published by CIIL Mysore, pp. 60 - 65, Kharagpur, INDIA, February 2005.

[4] Naushad UzZaman and Mumit Khan, “A Double Metaphone Encoding for Bangla and its Application in Spelling Checker”, *Proc. 2005 IEEE Natural Language Processing and Knowledge Engineering*, Wuhan, China, October, 2005.

[5] Naushad UzZaman and Mumit Khan, “A Bangla Phonetic Encoding for Better Spelling Suggestion”, *Proc. 7th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, December, 2004.

[6] F.J. Damerau, “A technique for computer detection and correction of spelling errors”, *communication of ACM*, vol. 7, issue. 3, pp. 171-176, 1964.

[7] Arif Billah Al-Mahmud Abdullah and Ashfaq Rahman, “Spell Checker for Bangla Language: An Implementation Perspective”, *Proc. 6th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, 2003.

[8] P. Kundu and B.B. Chaudhuri, “Error Pattern in Bangla Text", *International Journal of Dravidian Linguistics*, vol. 28, issue. 2, 1999.

[9] Karen Kukich, “Techniques for automatically correcting words in text”, *ACM Computing Surveys*, vol. 24, issue. 4, pp. 377 - 439, 1992.

[10] The longest common subsequence algorithm, available online at http://www.ics.uci.edu/~eppstein/161/960229.html

[11] The soundex algorithm, available online at http://www.sound-ex.com/soundex_method.htm