Myanmar Words Spelling Checking Using Levenshtein Distance Algorithm

Nwe Zin Oo, Tin Myat Htwe
University of Computer Studies Yangon
nwezinoo1@gmail.com, tinmyathtwe@gmail.com

Abstract

Natural Language Processing (NLP) is one of the most important research area carried out in the world of Artificial Intelligence (AI). Spelling Checking, Machine Translation, Automatic Text Summarization, Information Extraction and Automatic Text Categorization Information Retrieval are included in NLP. A Myanmar Spelling Checker is an essential component of many of the common desktop applications such word processors as well as the more exotic applications. In this system, it proposes the process of checking the spelling of a Myanmar input word and suggestion list if it is misspelled Myanmar word. This system is intended to develop a Myanmar Language Spell Checker (or spell check) by using Levenshtein Distance Algorithm. Moreover, Dynamic Threshold Algorithm and Transformation Algorithm are used in Myanmar Spelling Checking. This system uses Zawgyi Myanmar Font and implements using Java Language and MySQL Server. This thesis intends to check the spelling for Animals and Plants, and add correct Myanmar words to the Dictionary. Each Myanmar input word is compared against a dictionary of correctly spelt Myanmar words. This system improves the quality of suggestions for missed spelt Myanmar words.

1. Introduction

Natural Language Processing offers possible solutions to the problem of communication between humans and computers. Knowledge representation of words of a language is one of the significant issues in each system related to NLP. Myanmar Language is similar to other Asian Language including Indian, Chinese, Japanese and Thai language. Myanmar Script has been a majority language of Myanmar over 1000 years old. The letters of the alphabet used in Myanmar script are derived from the *Brahmi* script which has flourished in the Indian subcontinent between 5th Century B.C and 3rd Century A.D. The Myanmar Script is used to write Burmese Language. In addition, Burmese language is a syllabic writing system and its Script is written from left to right and there is no space between words. Therefore, in this system, we emphasize spelling checking for only missed spelt Myanmar words, not for Myanmar sentences and Grammar. This system can correct *Typographical errors* and *Cognitive errors* of Myanmar Words related with Animals and Plants. This system employs a Myanmar word to represent each syllable and consists of 33 symbols for consonants. Most spelling checkers allow the user to add custom words to the spelling checker's vocabulary if this word is not contained in the Dictionary. Simple spell checkers operate on individual words by comparing each of them against the contents of a dictionary.

The rest of this paper is organized as follows: Myanmar Language features is described in Section 2, related works are discussed in Section 3. Section 4 presents String Similarity Algorithms and Levenshtein Distance Algorithm with examples. Section 5 explains the system design. Section 6 describes the detail design of system. Finally, the paper is concluded in Section 7.

2. Myanmar Language Features

With the increasingly widespread use of computers and the Internet in Myanmar, there are large amounts of information in Myanmar languages are used various word processing software packages. NLP for Myanmar Language was formed in 1997 to develop technologies to use Myanmar Language on computer for the localization of operating system and software, spelling checking, grammar checking and sorting in Myanmar Language, and machine translation to/from Myanmar Language from/to foreign languages.

Myanmar language belongs to *Tibeto-Burman* language family and derives from *Sino-Tibetan language tree*. Myanmar Language is similar to other Asian Language including Indian, Chinese, Japanese and Thai language. Myanmar Script has been a majority language of Myanmar over 1000 years old. The letters of the alphabet used in Myanmar script are derived from the *Brahmi* script which has flourished in the Indian subcontinent between 5th Century B.C and 3rd Century A.

In Myanmar Language feature, the format of characters sequence [Consonant (ကြွေးနှိပ်), Medials

(ဗျည်းတွဲ) , Vowel (သရ)] is very important for writing Myanmar words. The user should type the upper vowels (“ -ိ ၊ -ီ ၊ -ံ ၊ -် ”)before the lower vowels (“ -ု ၊ -ူ ၊ -ေ ၊ -ော ၊ -ို ”). Therefore, this proposed system can correct **Typographical errors** which generally occur due to user’s mistakes while typing (“ပင်ဂွင်း” as “ပင်းဂွင်း”), **Cognitive errors** which caused by user who do not know how to spell the Myanmar words (“စင်ရော်” as “ဇင်ယော်”) and **Sequence errors** which often caused the wrong format of character sequence (“င-ံ-ုး” as “င-ု-ံး”).

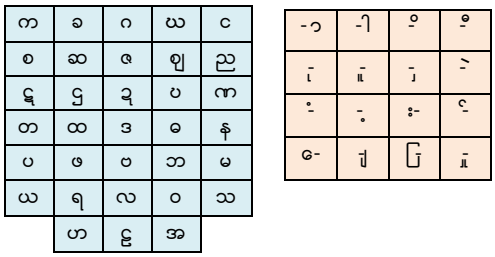

Figure1. Myanmar Characters (Consonants, Medial and Vowels Tables).

3. **Related Works**

Spelling Checking has a long history in Computer Science, and nowadays spell checking system is as an essential part for almost all application software and different people also need different spell checkers in the world. Human language translation for one to another language is a difficult task for natural language has ambiguity and varies according to their own features and nature. In Myanmar, morphological analysis of Myanmar Noun Phrases is often occurred by using Finite State Autonoma and Rule-based morphological analyzer. But, syllabification is also required to tokenize Myanmar Noun Phrases. A comprehensive spelling checker presented a significant challenge in producing suggestions words list for misspelled word when the input word is not matched with the words from dictionary. One such suggestion algorithm is to list those words in the dictionary having Levenshtein Distance (LD) from the original word. **Levenshtein Distance** is named after the Russian scientist **Vladimir Levenshtein** , who devised the algorithm in 1965. The Levenshtein_Distance algorithm has been used in: **Spell checking** , **Speech recognition** , **DNA analysis** , **Plagiarism detection**. There are lots of applications of Levenshtein Distance Algorithm. It is used in Dialectology to estimate the proximity of dialect pronunciations. Moreover, this algorithm is used in biology to find similar sequences of nucleic acids in DNA or amino acids in proteins. Levenshtein Distance can be checked spelling efficiently for English language and non English languages (such as Myanmar). Moreover, the Levenshtein Distance is used for automatically reduction for Medical Name Confusion, analysis for DNA sequences for HIV and checks the spelling for traveling and the name of cities in Myanmar country.

4. **Levenshtein Distance Algorithm**

There are many kinds of String Similarity Algorithms for spelling checking such as Hamming Distance, N-grams, Longest Common Subsequence (LCS) and Levenshtein Distance. Among these algorithms, Levenshtein Distance Algorithm is the best algorithm for two fuzzy strings. In information theory and computer science, the Levenshtein distance is a metric for measuring the amount of difference between two sequences (i.e., the so called edit distance). A generalization of the Levenshtein Distance allows the transposition of two characters as an operation and produces the number of operations need to be transformed from one word to another. Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). It is used in some spell checkers to guess at which word (from a dictionary) is meant when a missed spelt word is encountered and operate Insert, Delete and Substitute transformations. At the end, the bottom-right element of the array contains the answer. The resulted distance is the number of deletions, insertions, or substitutions required to transform s into t.

The greater the Levenshtein distance, the more different the strings are.

Levenshtein Distance Algorithm is shown as follow:

Step	Description
	Set n to be the length of s. Set m to be the length of t.
1.	If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2.	Initialize the first row to 0..n. Initialize the first column to 0..m.
3.	Examine each character of s (i from 1 to n).
4.	Examine each character of t (j from 1 to m).
5.	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6.	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d [i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d [i-1,j-1] + cost.
7.	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Examples

- If s is “ ယူကလစ် ” and t is “ ယူကလစ် ”, then LD(s,t) = 0, because no transformations are needed. The strings are already identical.
- If s is “ ဇင်ရော်” and t is “ စင်ရော်”, then LD(s,t) = 1, because one substitution (change “ဇ” to “စ”) is sufficient to transform s into t.
- If s is “ စပယ်ပန်း” and t is “ စံပယ်ပန်း”, then LD(s,t) = 1, because one insertion(insert “ ံ ” after “ စ ”) is sufficient to transform s into t.
- If s is “ ပင်းဂွင်း ” and t is “ ပင်ဂွင်း ”, then LD(s,t) = 1, because one deletion (delete “ း ” at the fourth of input word) is sufficient to transform s into t.

This section shows how the Levenshtein Distance is computed when the source is “ဇရက်” and the target string is “ဆက်ရက်”.

Steps 1 and 2

		ဇ	ရ	က	်
	0	1	2	3	4
ဆ	1				
က	2				
်	3				
ရ	4				
က	5				
်	6				

Steps 3 to 6 When i=1

		ဇ	ရ	က	်
	0	1	2	3	4
ဆ	1	1	2		
က	2	2			
်	3	3			
ရ	4	4			
က	5	5			
်	6	6			

Steps 3 to 6 When i= 2

		ဇ	ရ	က	်
	0	1	2	3	4
ဆ	1	1	2	3	
က	2	2	1	2	
်	3	3	2		
ရ	4	4	3		
က	5	5	4		
်	6	6	5		

Steps 3 to 6 When i= 3

		ဇ	ရ	က	်
	0	1	2	3	4
ဆ	1	1	2	3	4
က	2	2	1	2	3
်	3	3	2	1	2
ရ	4	4	3	2	
က	5	5	4	3	
်	6	6	5	4	

Steps 3 to 6 When i= 4

		ဇ	ရ	က	်
	0	1	2	3	4
ဆ	1	1	2	3	4
က	2	2	1	2	3
်	3	3	2	1	2
ရ	4	4	3	2	1
က	5	5	4	3	2
်	6	6	5	4	3

The Levenshtein Distance of these two words is in the lower right hand corner of the matrix, (i.e. LD = 3) that shows how many changes is needed. This corresponds to our intuitive realization that “ ဇရက် ” can be transformed into “ ဆက်ရက် ”

- by substitution of “ ဆ ” in the place of “ ဇ ” (Substitution).

- by addition of “ က ” at the second of input word (Addition).

- by addition of “ ် ” at the third of input word (Addition).

5. System Design

There are three main components in this system, Levenshtein Distance gives the distance between two strings and Dynamic Threshold lists the most similarity Myanmar words from the Dictionary. When we transform from the input word to the destination word we uses Transformation Algorithm for messaging what kind of transformation is needed and the position of these transformations are needed to take place. And, the user also can add new Myanmar Words to the Dictionary.

The overview of system flow is showed as follow:

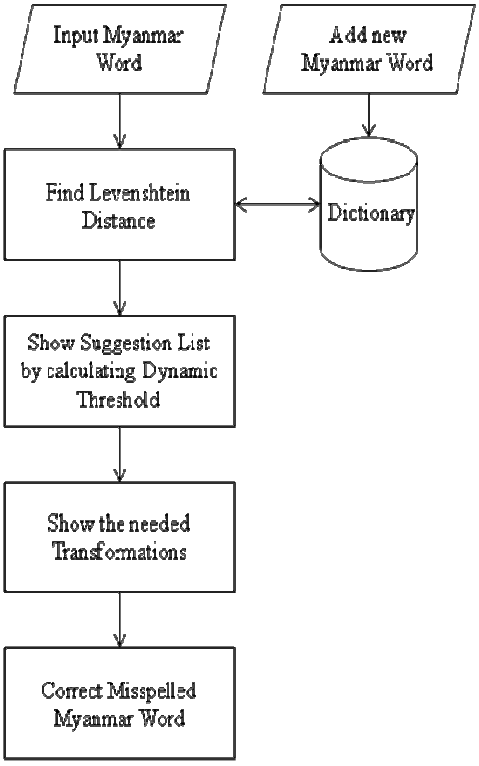

Figure 2. Overview of the system.

5.1 Dynamic Threshold Algorithm

The Levenshtein distance value above threshold is not considered to be aligned and showed in the suggestion list box. The system only shows suggested words which have Levenshtein distance is equal or less than dynamic threshold, are considered to get more similar Myanmar words. The threshold value in this system is not predefined number, it calculates dynamically based on the length of input word and destination word.

5.2 Transformation Algorithm

After the system checks how many operations are needed to transform from one word to another, the system should message the position of these transformations to take place. This transformation algorithm helps for the operations of Insert, Delete and/or Substitute from the input word to the user selected word.

6. **Detail Design**

In the proposed system, the user can choose one of the suggested words and check the needed transform operations from original word to another. When the user chooses one word from the list of suggested words, the system will show the Levenshtein Distance of the original word and the user selected word. The user can choose one of the correctly spelt Myanmar words and the proposed system automatically changes that missed spelt Myanmar word into the correctly spelt Myanmar word. If the input word is correctly spelt but it against with the sequences of Myanmar Word Spelling, then the system will show that word is not suitable with Myanmar Word Spelling Rules. The user can know about the Myanmar Word Spelling Rules from Myanmar Word Spelling Rules Page. Moreover, the user can add the new other Myanmar words into the dictionary. The system does not allow the insertion of the identical Myanmar words into the dictionary. This proposed system uses the MySQL Server and Query Browser to store the correctly spelled Myanmar words with 3 tables (tblanimal, tblplant , tblother).

The detailed design of the Myanmar Spelling Checking System is as shown in figure 3.

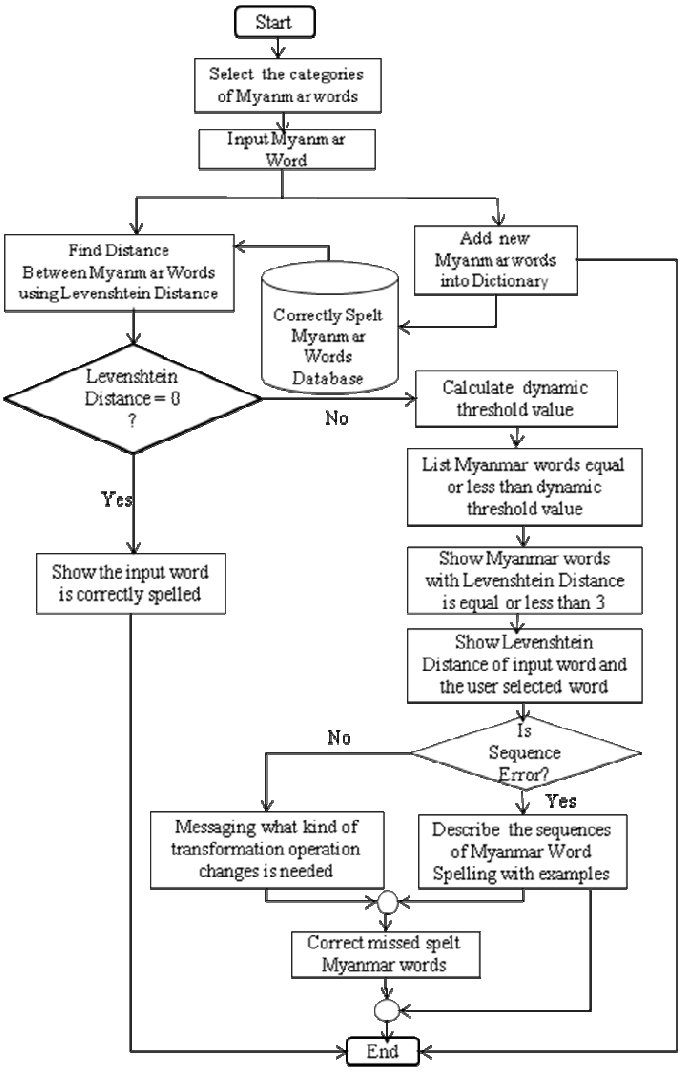

Figure 3. Detail Design of the system.

The home page of the proposed system is shown in figure 4.

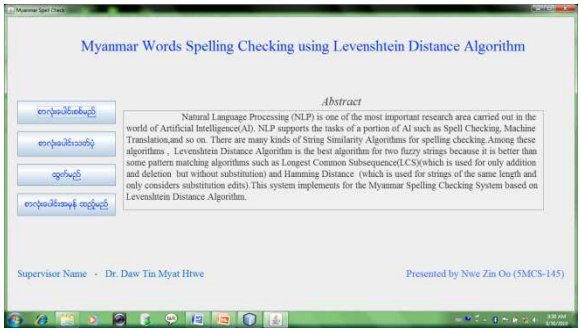

Figure 4. Main Interface of the system.

When the user chooses one word from the list of suggested words, the system shows Levenshtein Distance of the original word and the user selected word. And the system shows that what kind of transformation is needed to be changed from one word to the user selected word as shown in figure 5. When the user chooses the " ပြင်ပါ " button, the proposed system automatically corrects that missed spelt Myanmar word into the correctly spelt Myanmar word. Moreover, the user can add the new correctly spelt Myanmar words into the Dictionary.

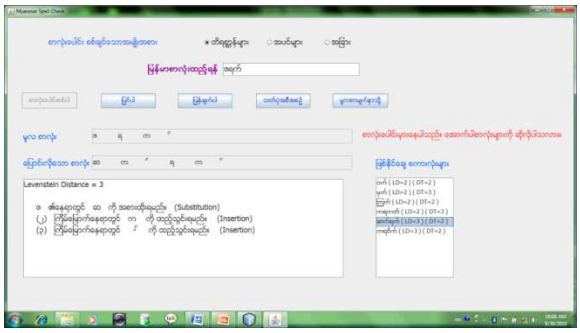

Figure 5. Spelling Checking Page of the System.

6.1 Analysis

This system is emphasized on the Spelling Checking of Myanmar Words consulting with Plants or Animals using Levenshtein Distance Algorithm. This system can check any words of Myanmar words but it can support only three transformation operations for correcting the input word to the destination word because the greater the Levenshtein distance, the more different the strings are. Therefore, dynamic threshold algorithm is used to get more similar Myanmar words dynamically from the dictionary.

This proposed system can correct *Typographical errors*, *Cognitive errors* and *Sequence errors* of Myanmar Words. *Sequence errors* can be caused in writing Myanmar words with wrong format sequence that may be two or three combinations of consonants, medial or vowels. For example, ("ခ-ိ-ု" as "ခ-ု-ိ") (two combinations) and ("ခ-ျ-ိ-ု-း" as "ခ-ိ-ျ-ု-း") or ("ခ-ျ-ိ-ု-း" as "ခ-ျ-ု-ိ-း") (three combinations). This proposed system can check and correct such kinds of *Sequence errors* because Levenshtein Distance Algorithm is two dimensions array string similarity algorithm. The user can check the spelling of Myanmar words not only simple Myanmar words but also (*Pali*) or (*Pat Sint*) words by using this proposed spelling checking system.

If the input Myanmar Word is different from the words in the Dictionary, the system will show the list of suggested most approximate correctly spelled Myanmar words with least Levenshtein Distance in ascending order. The Levenshtein Distance value above dynamic threshold is not considered to be aligned and showed in the suggestion list box. The proposed system only shows the suggested Myanmar words with Levnshtein Distance is equal or less than dynamic threshold, are considered to get more similar Myanmar words. When the user input the Myanmar word which has more than Levenshtein Distance is equal to 3, the system shows only possible similar Myanmar words in the suggestion list, but it cannot be transformed. The user can add new correctly spelt Myanmar words in to the dictionary. If the input word is already existed in the dictionary, the system will show the message that word is already existed. So, the system will not allow having identical words in the dictionary. Moreover, the system can check the Myanmar Spelling rule for Myanmar words. When the user input the wrong format of Myanmar character sequences, the system shows the correct sequences for Myanmar word which is matched with Myanmar Spelling rules.

7. Conclusion

This system can correct *Typographical errors*, *Cognitive errors* and *Sequence errors* of Myanmar Words. The user can easily know how transformations are needed and where these transformations are needed to take place. The user can also add custom Myanmar Words to the spelling checker's vocabulary. This system can check any words of Myanmar words but can correct only three transformation operations for changing the input Myanmar word to the destination Myanmar word because if the Levenshtein Distance is more than 3, there may show many irrelevant Myanmar words. In this proposed system, we mainly check the spelling of Myanmar words consulting with Animals and Plants. It is just only for spelling checking with possible suggestions and correcting the erroneous Myanmar words using Levenshtein Distance Algorithm. This system can be extended to correct misspelled Myanmar Sentences. This proposed system is very useful in applications that need to determine dynamically how similar two strings are, such as Myanmar spell checkers.

References

[1] **"မြန်မာစာလုံးပေါင်း သတ်ပုံကျမ်း"** *Department of Myanmar Language commission , ministry of education, Union of Myanmar June 1986.*

[2] **“မြန်မာစာ မြန်မာစကား”** *Department of Myanmar Language commission , ministry of education, Union of Myanmar June 2007.*

[3] *Win Ko, Implementation of Spell-Checking System by using Levenshtein Distance Algorithm, in proceeding of Third International Conference on Computer Application, UCSM.*

[4] *Khaing Su Yee, Detecting the behaviors of HIV DNA sequences using Levenshtein Distance Algorithm, UCSY.*

[5] *Su Sandar, Khaing Moe San, Myanmar Unicode Spelling Checker(MUSC) based on Natural Language Processing, in proceeding of Third International Conference on Computer Application, UCSM.*

[6] *Ye Sis Min, Khin Aye Than, Automatic Reduction for Medicine Name Confusion by using Orthographic Matching, in proceeding of Third International Conference on Computer Application, UCSY.*

[7] *Ei Ei Han, Nilar Thein, Identification of a Word Boundary of Myanmar Text based on Finite State Autonoma, in proceeding of Third International Conference on Computer Application, UCSY.*

[8] *Wai Wai Hnin , Morphological Analysis of Myanmar Noun Phrases, in proceeding of Third International Conference on Computer Application, UCSY.*

[9] *Naing Lin Oo, N-Gram-Based Spelling Checker for Myanmar Noun Words, UCSY.*

[10] *http://en.wikipedia.org/wiki/Burmese_language*

[11] *http://en.wikipedia.org/wiki/Burmese_alphabet*

[12]*http://en.wikibooks.org/wiki/Algorithm_implementation/Strings/Levenshtein_distance*

[13]*http://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein_distance*

[14] *http://en.wikibooks.org/wiki/Spelling-checker*