SEQUENCE CLUSTERING ALGORITHM FOR SPELL CHECKING AND SPELL SUGGESTION IN TAMIL LANGUAGE

Dr. J. Indumathi[1], Anish A[2]
Department of Information Science and Technology,
College of Engineering, Guindy, Anna University, Chennai,
TamilNadu, India.
[1]indumathi@annauniv.edu, [2]msgtoanish@gmail.com

ABSTRACT

Spell checking aids the user to identify the mistakes in the spelling and also suggests the user with the intended spelling for the misspelled word. This paper proposes a sequence clustering algorithm for spell checking in Tamil language. Even though there are many algorithms for spell checking in other major languages especially English, the lack of an effectual algorithms in Tamil language impedes the development of the language technologies and its applications. The proposed algorithm reduces the number of distance between the misspelled word and the dataset or word in the dictionary thereby making the algorithm faster and determines the intended word suggestions.

KEYWORDS: canti mistake, sequence clustering, N-gram, spell check

1. INTRODUCTION

The spell checkers are an application programs that flags words in a document that may be misspelled or misplaced. Though there are several spell checking and suggesting application are available for languages like English, no fully functional application is available for the Tamil language. The existing systems either find the misspelled words from an existing list of words in the datasets or the omission of a required letter or inclusion of an inappropriate letter between two adjoined words (*Canti* mistake) [S. Jananie, et.al, (2014)]. Further, several issues have been also identified in these systems. Basically the miss-spelled word can be classified into two types, they are non-word error and real word error. The non-word error is nothing but the mis-spelled word and the real word error is nothing but the word, which is misplaced in the sentence.

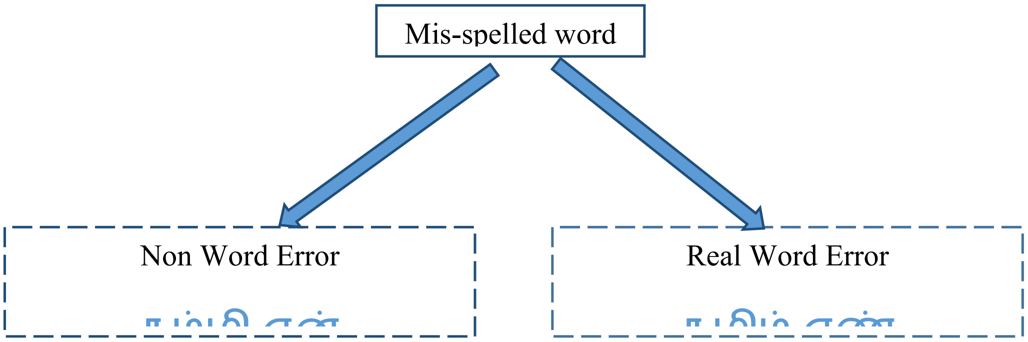

Figure 1. TYPES OF MISSPELLED WORD

About 80% of all misspelled Tamil words (non-word errors) in human typewritten text are due to single-error misspellings.

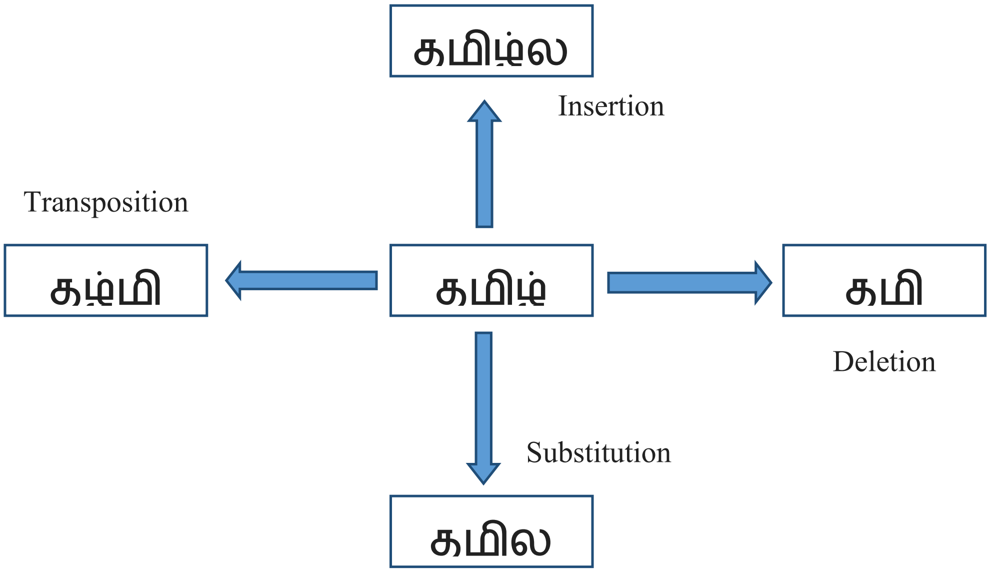

Figure 2. TYPES OF NON-WORD ERROR

According to the proposed approach, each word is checked whether it exists in the dictionary using a sequence clustering algorithm. If it does not exist, then the *n-gram* based technique is used to generate possible suggestions for the given word. And required rules are written to get the appropriate suggestions by considering *Canti* check as well to identify the appropriate joining letter of two adjoined words [S. Jananie, et.al, (2014)].

A list of 250,000 unique and error-free words are included in the dictionary. These words have been collected from various sources [Gupta, et. al., (2012)]. It is very difficult to gather all the words in Tamil language. Therefore, add to dictionary option has been introduced to collect new words from users and add to the existing dictionary after the moderation.

To reduce the search space, the dictionary has been divided into different files based on the first letter of the word. Due to the complex nature of Tamil script compared to English, stacks and lists have been used during the processing of words. These rules have been written in such a way that it can be extended further in future. All these processing is being done without Romanizing the Tamil text, while in most of the other approaches Tamil language is processed in Romanized form.

The proposed system gives better accuracy than the existing systems; 85.77% accuracy was noted when considering the suggestions generation. This result had been calculated by analyzing the suggestions generated by the system for the words that are not in the dictionary. Hence the proposed approach, which has dictionary check with sequence clustering algorithm, suggestions generation with *n-grams* is a complete solution for Tamil spell checking.

1.1 SPELL CHECKING

The task of identifying and flagging incorrectly spelled words in a document written in a natural language.

1.2 SPELL SUGGESTION

The process of suggesting the user to the misspelled words with the most likely intended ones.

2. N-GRAMS TECHNIQUE

N-gram technique is a method to find incorrectly spelled words in a sentence. Instead of comparing each entire word in a text to a dictionary, just n-grams are controlled. A check is done by using an n-dimensional matrix where real n-gram frequencies are stored. If a non-existent or rare n-gram is found the word is flagged as a misspelling, otherwise it will correct the spelling.

An n-gram is a set of consecutive characters taken from a string with a length ‘n’. If n is set to one then the term used is a unigram, if n is two then the term is a Bigram, if n is three then the term is trigram[Mishra, et. al., (2013)]. The n-gram algorithm was developed as one of the benefits is that it allows strings that have differing prefixes to match and the algorithm is also tolerant of misspellings. Each string that is involved in the comparison process is split up into sets of adjacent n-grams. The n-grams algorithms have the major advantage that they require no knowledge of the language that it is used with and so it is often called language independent or a neutral string matching algorithm [Hasan Muaidi, et. al, (2012)].

N-gram analysis is used in spell-checker after compiled a table of n-gram binary values or frequency counts from large corpora, for comparative purposes to check if each n-gram in an input string is likely to be valid in the language. Consider the two strings, share the more similar they are ‘சந்தோஷம்’, ‘சந்தேகம்’.

N-Gram Similarity of ‘சந்தோஷம்’

Word	சந்தோஷம்				
Letter Unigrams	ச	ந்	தோ	ஷ	ம்
Letter Bi-grams	சந்	ந்தோ	தோஷ	ஷம்	
Letter Tri-grams	சந்தோ	ந்தோஷ	தோஷம்		

N-Gram Similarity of ‘சந்தேகம்’.

Word	சந்தேகம்

Letter Unigrams	ச	ந்	தே	க	ம்

Letter Bi-grams	சந்	ந்தே	தேக	கம்

Letter Tri-grams	சந்தே	ந்தேக	தேகம்

To measure the N-gram similarity coefficient, we calculate the union and common term of the two strings.

Union	சந்	ந்தோ	தோஷ	ந்தே	தேக	கம்	ஷம்

Common	சந்

Similarity coefficient δ = |common N-grams| / |Total N-grams|

δ =1/7=0.14

N-gram similarity measure works best for insertion and deletion errors, well for substitution errors, but very poor for transposition errors.

2.1 N-GRAM GENERATING ALGORITHM

```
functionget_n_grams(word, n) returns n_grams_list
        l ← length(word) -n
        n_grams_list← empty()
        forifrom0 tol do
n_grams_list← append( substring (word, i, n) )
```

3. SEQUENCE CLUSTERING ALGORITHM

Sequence clustering is a fundamental research topic and could be applied in various fields, such as data mining and multimedia information retrieval. Suggested by [T. W. Liao, (2005)], a generic approach of sequence clustering consists of two major parts. The first part is the clustering algorithm, which involves the process of gradually grouping together similar sequences. The second part is the calculation of similarity, which quantifies the degree of similarity between sequences by calculating the distance separating them. As an example, the Euclidean distance is one of most common distance measures for sampled data sequences, as well as the dynamic time warping (DTW) distance for string-like sequences.

From another point of view, sequence similarities can be further divided into whole similarities and partial similarities. The whole similarity refers to the similarity that appears throughout the entire sequence, which may be the trend of the sequence itself. The partial similarity indicates the similarity which exists between subsequences within the sequence. The sequence length can be either equal or variable while the sequence labels can be of a single label or multi labels. Many-faceted properties of sequence clustering are summarized in Table 1. There are four cases in the Table 1. In the case 2, since we choose the Euclidean distance as the similarity measure between sequences, it cannot be directly applied for two sequences of unequal length. Note that, if the editing distance is chosen for string-like sequences, the case 2 will be reasonable and usually known as global alignment. Regarding to the case 3, the case 3 can be considered as a special case of case 4. For practical reasons, we focus on the case 1 and case 2, i.e., sequence of equal-length with whole similarity, and sequence of variable-length with partial similarity. In the former case, named the single-label clustering, a sequence is to be assigned only one label which indicates a certain kind of trend in sequence. In the latter case, named the multi-label clustering, a sequence could be assigned several labels as long as subsequences meet criteria indicating by the partial similarity measurement [Xu, et. al., (2005)].

In our work, we introduce a basic approach to solve the single-label clustering problem. Then, the approach will be extended to solve the multi-label clustering problem.

Table 1: Properties of sequence clustering

sequence length similarity	Equal-length	Variable-length
Whole similarity	Case 1: Single-label	Case 2
Partial similarity	Case 3	Case 4: Multi-label

4. CONCLUSION

The method we introduced in this paper reduces the number of distances to be calculated without removing a single word from the dictionary. This makes the algorithm faster than other approaches and presents a satisfactory success rate of 85.77% in a challenging dataset. The success rate is 11.18% higher than the baseline for this task. Therefore, a spell checker with a small dictionary would be very likely to raise false alarms over correctly spelt rare words. As previously mentioned, the corpus contained the attempts of very poor spellers and therefore misspelled words were often very far from their targets. Another shortcoming of the corpus is the fact that it is organized as a simple list of words without context, making it difficult to refine calculations specifically for real-word errors.

REFERENCES

- **[Hasan Muaidi, et. al., (2012)]** Muaidi, Hasan, and Rasha Al-Tarawneh. "Towards Arabic Spell-Checker Based on N-Grams Scores." *International Journal of Computer Applications* 53, no. 3, pp. 12-16, 2012.

- **[Gupta, et. al., (2012)]** Gupta, Neha, and Pratistha Mathur. "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, 2012.

- **[Mishra, et. al., (2013)]** Mishra, Ritika, and Navjot Kaur. "A Survey of Spelling Error Detection and Correction Techniques." *International Journal of Computer Trends and Technology,* vol. 4, issue3. pp. 372-374, 2013.

- **[Xu, et. al., (2005)]** Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." *Neural Networks, IEEE Transactions on* 16, no. 3, pp. 645-678, 2005.

- **[S. Jananie, et.al, (2014)]** S. Jananie and K. Sarveswaran, " Hybrid Approach For Spell Checking Of Tamil Language", *Proceedings of the Peradeniya Univ. International Research Sessions*, Sri Lanka, vol. 18, 2014.

- **[T. W. Liao, (2005)]** Warren Liao, T. "Clustering of time series data—a survey." *Pattern recognition* 38, no. 11, pp. 1857-1874, 2005.