String Similarity Measures for Burmese (Myanmar Language)

Khaing Hsu Wai
University of Technology Yatanarpon Cyber City (UTYCC)
Myanmar
khainghsuwai@utycc.edu.mm

Ye Kyaw Thu
National Electronics and Computer Technology Center (NECTEC)
Thailand
ka2pluskha2@gmail.com

Hnin Aye Thant
University of Technology Yatanarpon Cyber City (UTYCC)
Myanmar
hninayethant@utycc.edu.mm

Swe Zin Moe
University of Technology Yatanarpon Cyber City (UTYCC)
Myanmar
swezinmoe.1011@gmail.com

Thepchai Supnithi
National Electronics and Computer Technology Center (NECTEC)
PathumThani, Thailand
thepchai.supnithi@nectec.or.th

***Abstract*—Measuring string similarity is useful for a broad range of applications. It plays an important role in machine learning, information retrieval, natural language processing, error encoding, and bioinformatics. Measuring string similarity is a fundamental operation of data science, important for data cleaning and integration. Real-world applications such as spell checking, duplicate finding, searching similar words, and retrieving tasks use string similarity. In this study, string similarity metrics have been calculated for Burmese (Myanmar language). The encoding table for Burmese has been built based on the pronunciation similarity of characters and vowel combination positions with a consonant. According to the table, strings and words are encoded. Similarity distance is measured between the dataset and query words. Previous string similarity approaches are not suitable for fuzzy string matching of tonal-based Burmese. Therefore, three mapping approaches are proposed in this study.**

***Index Terms*—Myanmar character, Burmese, String similarity metrics, Phonetic similarity, Fuzzy string matching**

I. Introduction

Measuring string similarity is a fundamental operation in many applications of machine learning. It is widely studied in natural language processing (NLP). NLP applications such as text-to-speech, machine translation, spell checking, and information retrieval calculate string similarity metrics to find how similar the strings are. In other words, string similarity metrics help to find similar words according to a given query. Languages are interesting, and each language has its own features and writing systems. In the literature, several approaches have been proposed for string similarity. Most of them are character-based metrics and associated with English or European languages. For Burmese (language in Myanmar), we need to consider new approaches together with the existing string similarity metrics. Burmese is a tonal-based language and also a very rich language [14]. It has 33 consonants, and the consonants are combined with vowels and medials to form syllables. In Burmese, not only one character can form a word (e.g., "ကာ", "dance" in English) but also one syllable can form a word (e.g., "ကိုက်", "like" in English). Additionally, there are many phonetically similar sounds of characters and words in Burmese. In our experiment, we proposed three mappings: phonetic mapping, sound mapping, and syllable combination mapping. We introduced a new approach based on the idea of Soundex, the best-known phonetic encoding algorithm, for retrieving phonetically similar words by calculating the string similarity distance. We have collected two datasets: one dataset contains the confusion pairs of words with real spelling mistakes, and another is a manually developed word similarity dataset. We evaluated six measures (cosine distance, Damerau–Levenshtein distance, Hamming distance, Jaccard distance, Jaro–Winkler distance, and Levenshtein distance) on two datasets, with and without the proposed three mappings. According to our results, all three mappings outperformed the existing approaches for retrieving Myanmar words with similar pronunciations.

II. Related Work

To the best of our knowledge, there is only one proposal that measured phonetic similarities of Myanmar Inter-

nationalized Domain Names (IDNs) [1]. To retrieve phonetically similar Myanmar IDNs, IPA (International Phonetic Alphabet)-Soundex functions were used for matching character values based on their phonetic similarities of Burmese. The normalized similarity method is capable of measuring similarity not only in a single language, but also in a cross-language comparison [2].

The Myanmar characters ultimately descend from a Brahmic script, either Kadamba or Pallava [4]. Likewise, most of the major Indian languages such as Devanagari (e.g., Hindi, Marathi, Nepali), Bengali (Bengali and Assamese), Gurmukhi (Punjabi), Gujarati, Oriya, Tamil, Telugu, Kannada, and Malayalam use scripts that are derived from the ancient Brahmi script. They have approximately the same arrangement of the alphabet, are highly phonetic in nature, and a computational phonetic model was proposed for them [3]. It mainly consists of a model of phonology (including some orthographic features) based on a common alphabet of these scripts, numerical values assigned to these features, a stepped distance function (SDF), and an algorithm for aligning strings of feature vectors. The SDF is used to calculate the phonetic and orthographic similarity of two letters.

III. String Similarity Metrics

String similarity determines how similar two strings are. Various studies on string similarity has been carried out for different languages. In the literature, many methods to measure the similarity between strings have been proposed. Each method has its own features useful for NLP. Most similarity metrics are used to reduce minor typing or spelling errors in words or syllables in pronunciation. Based on the properties of operations, string similarity metrics can be divided into several groups.

Edit distance-based metrics estimate the number of operations needed to transform one string to another. A higher number of operations means less similarity between the two strings.

For token-based methods, the expected input is a set of tokens rather than complete strings. The purpose is to find similar tokens in both sets. A higher number of similar tokens means more similarity between the sets. A string can be transformed into a set of tokens by splitting it using a delimiter.

In sequence-based methods, the similarity is a factor of common substrings between the two strings. The algorithms try to find the longest sequence that is present in both strings. The more of these sequences found, the higher is the similarity score.

A. Levenshtein Distance

The Levenshtein distance [5], also known as edit distance, returns the minimum number of edit operations in terms of the number of deletions, insertions, or substitutions required to transform the source string to the target string. A higher number of edit operations means less similarity between two strings. For example, the edit distance between "cat" and "dog" is 3. There are three edit operations needed to transform "cat" into "dog". For Myanmar language, "Fate"-"ကံ"(kan) and "ကန့်"(kan) (exact pronunciation with "ကံ" but different spelling and "kick, lake" in English), two edit operations are required. The Levenshtein distance is perfect for finding similarity of small strings, or for a small string and a big string, where the editing difference is expected to be a small number. The Levenshtein distance is defined recursively, as shown in Eq. (1).

$$dis_{a,b}(i,j) = \begin{cases} 0 & \text{if i=j=0} \\ i & \text{if j=0 and i>0} \\ j & \text{if i=0 and j>0} \\ min = \begin{cases} dis_{a,b}(i-1,j)+1 \\ dis_{a,b}(i,j-1)+1 \quad \text{otherwise} \\ dis_{a,b}(i-1,j-1)+1(a_i \neq a_j) \end{cases} \end{cases} \tag{1}$$

B. Damerau–Levenshtein Distance

The Damerau–Levenshtein distance is an algorithm that is similar to the Levenshtein distance; however, it additionally counts a transposition between adjacent characters as an edit operation [6]. For example, to transform string "CA" to string "ABC", the Levenshtein distance counts three edits, whereas the Damerau–Levenshtein distance is 2. For Burmese, the Levenshtein distance between "ကလေး"("baby") and "ခကလေး"(wrong spelling of "baby") is 3, whereas the Damerau-Levenshtein distance is 2. Variations of this algorithm assign different weights to the edit based on the type of operation, phonetic similarities between the sounds typically represented by relevant characters, and other considerations.

C. Hamming Distance

The Hamming distance between two strings of equal length measures the number of positions with mismatching characters [7]. The Hamming distance only applies to strings of the same length. It is mostly used for error correction in fields such as telecommunication, cryptography, and coding theory. For example, the Hamming distance between "apple" and "grape" is 4, and the distance between "ဘခဖ"("father") and "ဘခဘ"(wrong spelling of "father") is 1.

D. Jaro–Winkler Distance

The Jaro–Winkler distance is another string metric that measures an edit distance between two sequences [8]. The score ranges from 0 to 1, where 0 is "no similarity," and 1 is "exactly the same strings." The Jaro–Winkler distance is used to find duplicates in strings, because the only operation that it considers is to transpose the letters in a string. Eq. (2) describes the Jaro–Winkler distance d_j of two given strings s_1 and s_2, where m is the number

of matching characters, and t is half of the number of transpositions.

$$d_j = \begin{cases} 0 & \text{if m=0} \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{2}$$

E. Cosine Similarity

The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them [9]. By calculating the cosine angle between the two vectors, we can decide if the vectors are pointing to the same direction or not. Two vectors with the same orientation have a cosine similarity of 1, which means that the two strings are equal. For two strings “ဇနီးမောင်နှံ”(“husband and wife”) and “ကလေး”(“baby”), the cosine similarity is 0, but for “ဇနီးမောင်နှံ”(“husband and wife”) and “စနီးမောင်နှံ”(wrong spelling of “husband and wife”), the similarity distance is 0.75, which is nearly 1. Eq. (3) shows the formula of cosine similarity.

$$similarity(A, B) = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3}$$

F. Jaccard Similarity

The Jaccard similarity measures similarities between sets [10]. It is defined as the size of the intersection divided by the size of the union of two sets. For example, for sets $A = \{1, 2, 3\}$ and $B = \{1, 2, 4, 5\}$, the Jaccard similarity is 0.4. The Jaccard similarity is calculated according to the following equation.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \tag{4}$$

G. Soundex Algorithm

The Soundex algorithm is a phonetic algorithm [11]. It is based on how close two words are depending on their pronunciation. For example, the code for “Flower” and the code for “Flour” is ‘F460” according to the Soundex encoding table, because they have the same pronunciation. Based on the idea of the Soundex algorithm, we propose three mappings for Burmese. All mappings aim to find words based on their phonetic similarity.

IV. Proposed Mappings

String similarity algorithms have some difficulties with Burmese because it is a tonal-based language and is composed of vowels, consonants, and medials. With Myanmar alphabets, many words have the same pronunciation but different meanings (e.g., “ကံ”, “luck” in English and “ကန်”, “lake” in English). Moreover, some words have similar pronunciations and different meanings (e.g., “ခုနစ်”, “seven” in English and “ခုနှစ်”, “year” in English). To consider phonetically similar words, we propose three mapping tables for Myanmar words.

A. Phonetic Mapping

In our proposed methods, the first mapping is the phonetic mapping. Words with the same pronunciation are grouped together. For example, “ကလေး” and “ခလေး” have the same pronunciation. Therefore, “က” (Ka) and “ခ” (Kha) are clustered to “က” (Ka) group. Likewise, other consonants with same pronunciation, such as “ဃ” (Ga) and “ဃ” (Gha), “ပ” (Pa) and “ဖ” (Pha), “ဗ” (Ba) and “ဘ” (Bha) are put together as groups, respectively, and some diacritics, such as “ွ” (Wa Hswe) and “ှ” (Ha Hto), tone marks such as “့” (Aukmyit), “္” (Myanmar sign Virama) are considered to be removed. Mapped characters are using both Myanmar and English alphabets for simple reading and an easier practical implementation. The details of the phonetic mapping table are shown in Table I.

TABLE I: Phonetic Mapping

Char	Mapped Char	Char	Mapped Char
က ခ	က	ွ ှ	(delete)
ဂ ဃ	ဂ	ဣ ဤ ၏ ိ ီ ည်	i
စ ဆ	စ	က် ပ် တ်	d
ဇ ဈ	ဇ	န် မ် ံ	n
ဋ တ	တ	ဲ ရ်	e
ဌ ထ	ထ	ဥ ဦ ု ူ	u
ဍ ဎ	ဍ	ာ ါ	r
ဏ န	န	ဧ ေ	a
ဒ ဓ	ဒ	့ း	(delete)
ပ ဖ	ပ	္	(delete)
ဗ ဘ	ဘ	ဩ ဪ ဩ ဪ	o
ယ ရ	ရ	၎င်း ၎	၎
လ ဠ	လ	၊ ။	s
သ ဿ	သ	င်္ င် င ဉ်	in
ျ ြ	y	?!.*-=#”<>[],+-	s

B. Sound Mapping

The second mapping is the sound mapping. This mapping is similar to the phonetic mapping, but the main difference is in processing Myanmar consonants. As the name of the sound mapping suggests, consonants that have the same movements of mouth, lips, and tongue, are grouped. For example, “က ခ ဂ ဃ င ဟ အ” (Ka Kha Ga Gha Nga Ha A) are clustered to “က” (Ka) group, “ည ဉ” (NyaGyi NyaLay) are clustered to “ည” (Nya) group, “ပ ဖ ဘ ဗ မ” (Pa Pha Ba Bha Ma) are clustered to “ပ” (Pa) group, “ယ ရ” (YaPetLet YaGauk) are clustered to “ရ” (Ya) group. The details of the sound mapping are shown in Table II.

C. Vowel Position Mapping

Myanmar writing system or word formation largely depends on the combination of left, right, upper, and lower characters to a consonant (i.e., consonant clusters or syllable). Here, left, right, upper, and lower characters mean dependent vowels, directives, and subscript consonants that are always written with a consonant [12] according to their written positions.

TABLE II: Sound Mapping

Char	Mapped Char	Char	Mapped Char
က ခ ဂ ဃ င ဟ အ	က	ွ ှ	(delete)
ည ဉ	ည	္	(delete)
စ ဆ ဇ ဈ	စ	က် ပ် တ်	d
ဋ ဌ ဍ ဏ ဎ တ ထ ဒ ဓ န	တ	န် မ် ံ	n
ပ ဖ ဗ ဘ မ	ပ	ဲ ရ်	e
ယ ရ	ရ	ဥ ဦ ု ူ	u
လ ဠ	လ	ာ ါ	r
သ ဿ	သ	ဧ ေ	a
ျ ြ	y	့ း	(delete)
၊ ။	s	ဩ ဪ ဩ ဪ	o
၎င်း ၎	၎	ဣ ဤ ၏ ိ ီ ည်	i
င်္ င် င ဉ်	in	?!.*-=#"< >[],+-	s

The third proposed mapping is based on the syllable formation in Burmese, we call it the vowel position mapping. Thus, the vowels written on the left side of the consonant are under the left (l) group, the right-side vowels are under the (r) group, the upper vowels are under the (u) group, the lower vowels are under the (d) group. If we represent the core concept of the vowel position mapping with Python programming, the code for building a dictionary variable named "map3_dict" will be as follows:

```
map3_dict = [
('[က-အ]', 'c'),
('ျ |ြ', 'y'),
('ေ', 'l'),
('ိ |ီ |ဲ |ံ ', 'u'),
('ွ |ှ |ု |ူ , 'd'),
('ာ |ါ |့ |း ', 'r'),
]
```

Here, "c" is used for consonants, "y" for medial characters "ျ" and "ြ", "l" for the "left", "u" for "upper", "d" for "down" or "lower", and "r" for "right"-side characters. The details of the vowel position mapping are shown in Table III. This mapping is designed for retrieving Myanmar words that have a similar vowel combination structure.

TABLE III: Vowel Position Mapping

Char	Mapped Char	Char	Mapped Char
a-z A-Z	F	က–အ	c
ျ ြ	y	္	p
ေ	l	ာ ါ ့ း	r
ိ ီ ဲ ံ	u	'ွ ှ ု ူ	d
်	k	၊ ။	s
ဣဤဥဦဧဩဪဿ၌၍၏	i	?!.*-=#"<>[],+-	$
၀–၉	n	0-9	D

V. Experiments

We compare 6 similarity measures on our three mappings. They are Levenshtein, Hamming, Jaro–Winkler, Damerau–Levenshtein, cosine, and Jaccard similarities. We conduct two experiments with two datasets that we have collected.

A. Datasets

We have collected two datasets: *Spelling Mistake Confusion Pairs* and *Word Similarity Dataset*.

1) Spelling Mistake Confusion Pairs: The dataset of spelling mistake confusion pairs was developed based on real-world spelling errors. Mainly, we collected general-domain text, especially from Myanmar news and social media websites, such as BBC (British Broadcasting Corporation) Myanmar, VOA (Voice of America) Myanmar, Facebook, and emails during March 2018 and July 2019. The dataset contains 2,381 pairs (i.e., 4762 words). Some examples of confusion pairs are as follows:

1) ကိုကိုကြီး – ကိုကိုကြီး
2) ကောင်းကောင်း် – ကောင်းကောင်း
3) ကောင်းကျပါတယ် – ကောင်းကြပါတယ်
4) ခွင့်မလွတ်ပါနဲ့ – ခွင့်မလွှတ်ပါနဲ့
5) ငါ့စီ – ငါ့ဆီ
6) စီးပွားေ၇း – စီးပွားရေး
7) စွဲချက်တင်နိ်ုင်သောကြောင့် – စွဲချက်တင်နိုင်သောကြောင့်
8) တောင်ပန်အပ်ပါတယ် – တောင်းပန်အပ်ပါတယ်
9) တိုင်ပြည်ချစ်စိတ် – တိုင်းပြည်ချစ်စိတ်
10) ဒေါအောင်ဆနး်စုကြည် – ဒေါအောင်ဆန်းစုကြည်
11) နက်နက်ရှုိင်းရှုိင်း – နက်နက်ရှိုင်းရှိုင်း
12) ပြသနာတက်မှာဆိုးပြီး – ပြဿနာတက်မှာစိုးပြီး
13) ၂၀၁၂၀ – ၂၀၂၀
14) ဝူးရှူး – ဝူရှူး
15) အဆောက်အဦ – အဆောက်အဦး

During the dataset collection, we found that some of the spelling mistakes are caused by encoding conversion between partial Unicode named "Zawgyi" and other Unicode fonts such as "Myanmar3" and "Padauk" (e.g., ကိုကိုကြီး – ကိုကိုကြီး, တနလာၤနေ့ – တနင်္လာနေ့, နိုင်ငံရေး။် – နိုင်ငံရေး၏). Moreover, the spelling mistakes based on pronunciation similarity (e.g., ကျေးပွန်းစွား – ကျေပွန်စွာ, ငါ့စီ – ငါ့ဆီ, ပြသနာတက်မှာဆိုးပြီး – ပြဿနာတက်မှာစိုးပြီး) and shape similarity (i.e., glyph) of Myanmar characters are also found (e.g., စီးပွားေ၇း – စီးပွားရေး, ဝူးရှူး – ဝူရှူး, အဆောက်အဦ – အဆောက်အဦး). All the confusion pairs generally have one-to-one relationship between misspelled and correct words; thus, we assumed it is very useful for evaluating on our three mappings. However, this dataset has few homophones and rhyme words; therefore, it is not suitable for measuring pronunciation similarity.

2) Similar Pronunciation Dataset: We developed the similar pronunciation dataset to evaluate similarity scores provided by our three mappings. Based on the correct

Myanmar word, we manually added one homophone and three more rhyme words, such as “Hat:Bat”, “Fun:Sun”, “Honey:Money”. For example, the first column word မြူးတူး(“festivity” in English) is the correct word, the second column မြူးထူး is the homophone word, and the other following columns ဂျူးဖူး, ကူးလူး, and ပြူးတူး are three rhyme words of the first column word (see Table IV). We collected 200 pairs for the similar pronunciation dataset, with 1,000 words in total.

TABLE IV: Examples from the Similar Pronunciation Dataset

Correct Word	Homophone	Rhyme1	Rhyme2	Rhyme3
မြူးတူး	မြူးထူး	ဂျူးဖူး	ကူးလူး	ပြူးတူး
ပြဿနာ	ပြဿနာ	ရှုစမ်း	ကြာပန်း	ကျင်နာ
တချို့	တစ်ချို့	အချို့	သချို့	နှစ်ချို့
ကြေးမြို	ကျေးမြို	ခွေးမြို	ကြေးမို့	ချေးသီး
ဂဃနဏ	ဂဂနန	ခခဃဃ	မမထထ	ခခရရ
လက်ရွေးစင်	လက်ရွေးစဉ်	လက်ယွေးစင်	ရက်ရွေးစင်	လက်ရွေးဇင်

Examples for how our three proposed mappings work can be seen as the following table.

TABLE V: Examples of Three Proposed Mappings

Phonetic Mapping	Sound Mapping	Vowel Position Mapping
ပစ္စည်း –> ပစၥi	ပစ္စည်း–> ပစၥi	ပစ္စည်း –> ccpcckr
ပစ်စည်း –> ပစၥi	ပစ်စည်း –> ပစၥi	ပစ်စည်း –> cckcckr

B. Evaluation

For the evaluation, we measured string similarity on each pair from both original datasets: “Spelling Mistake Confusion Pairs” and “Similar Pronunciation Dataset”. Next, we encoded or converted the original data into our 3 mappings and measured string similarity again. Finally, we counted the correct words or similar words based on the three thresholds “<=1”, “<=2”, and “<=3” for “Levenshtein, Damerau–Levenshtein, and Hamming distance measures” and “>=0.9”, “>=0.7”, and “>=0.5” for “Jaro–Winkler, cosine, and Jaccard distance measures”.

VI. Results and Discussion

TABLE VI: String similarity distances for the word “လက်ရွေးစင်” (“selection”) in English

Word - Similar Word	Levenshtein	Pronunciation	Sound	Vowel
လက်ရွေးစင် လက်ရွေးစဉ်	1	0	1	0
လက်ရွေးစင် လက်ယွေးစင်	1	0	0	0
လက်ရွေးစင် ရက်ရွေးစင်	1	1	1	0
လက်ရွေးစင် လက်ရွေးဇင်	1	1	0	0

The number of correct words found for six similarity measures on the “Spelling Mistake Confusion Pairs dataset” is shown in Figure 1. According to these experimental results, our phonetic mapping gave a better word correction rate than four existing distance measures (Levenshtein, Damerau–Levenshtein, Hamming, and Jaccard) for threshold <=1 or >=0.9. Similarly, the sound mapping also achieved higher or comparable results, except for the Jaro–Winkler and cosine similarity. On the other hand, the vowel position mapping approach obtained the lowest correction rate for all thresholds.

TABLE VII: String similarity distances for the word “လွင့်စဉ်” (“scatter” in English)

Word - Similar Word	Levenshtein	Pronunciation	Sound	Vowel
လွင့်စဉ် လွင့်စင်	1	0	1	0
လွင့်စဉ် လွှင့်စင်	N/A	0	1	1
လွင့်စဉ် လွင့်ဇင်	N/A	1	1	0
လွင့်စဉ် လွန့်စင်	N/A	1	1	0

TABLE VIII: String similarity distances for the word “အကဲခတ်” (“to assess” in English)

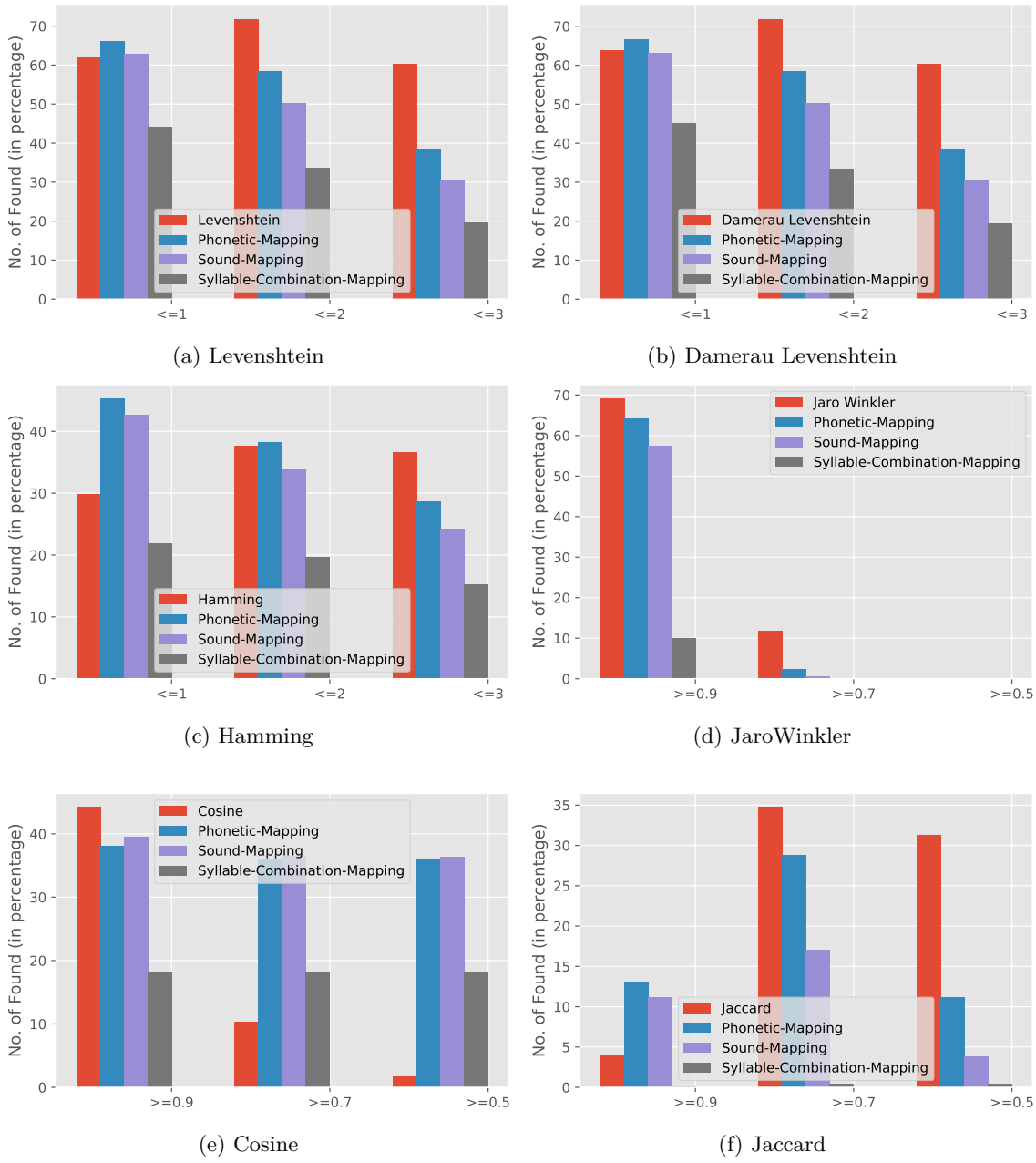
Word - Similar Word	Cosine	Pronunciation	Sound	Vowel
အကဲခတ် အကဲခပ်	N/A	1.0	1.0	1.0
အကဲခတ် အကဲဆတ်	N/A	N/A	N/A	1.0
အကဲခတ် အမြဲတက်	N/A	N/A	N/A	N/A
အကဲခတ် မဆဲတတ်	N/A	N/A	N/A	1.0

For thresholds “<=2” and “<=3” (“>=0.7”, “>=0.5” for Jaro–Winkler and cosine similarity), generally, all proposed mappings are lower than raw Myanmar text input. However, we found that the phonetic mapping and sound mapping matched more correct words from the “Spelling Mistake Confusion Pairs” dataset for Hamming and cosine similarities.

According to these experimental results, our new two mappings (phonetic and sound mappings) are applicable for string similarity measurement on spelling mistake confusion words. Moreover, based on the current results for thresholds “<=2” and “<=3” (or “>=0.7” and “>=0.5”), we clearly found that the vowel position mapping is able to retrieve approximately 50% of the correct words for Levenshtein, Damerau–Levenshtein, Hamming, and cosine similarities.

The results of retrieving similar pronunciation words, such as homophones and rhyme words, with six similarity measures on the “Similar Pronunciation Dataset” is shown in Figure 2. As we expected, two of our proposed mappings, phonetic mapping and sound mapping, achieved the highest number of found errors for all thresholds of Levenshtein, Damerau–Levenshtein, Hamming, Jaro–Winkler, cosine, and Jaccard similarities. Additionally, the vowel position mapping also obtained the highest or comparable results for existing five distance measures, except for the Jaccard distance measure.

We did a detailed analysis on distance values, and we found that our proposed three mappings have a zero distance value (i.e., no distance value) for some similarly pronounced words. For example, the string similarity distances for the word လက်ရွေးစင် and similar pronunciation



(a) Levenshtein (b) Damerau Levenshtein

(c) Hamming (d) JaroWinkler

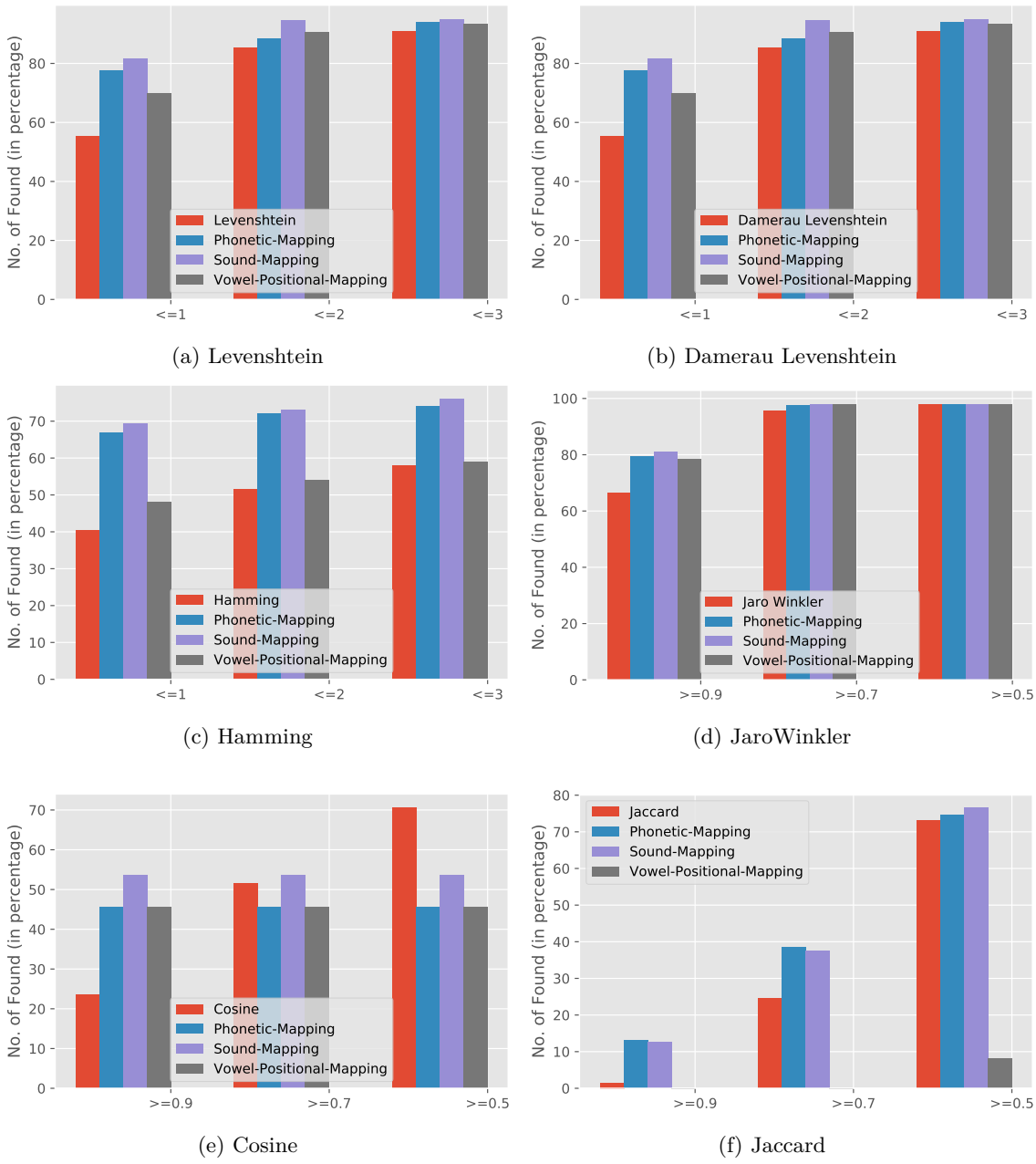
(e) Cosine (f) Jaccard

Fig. 1: Results with the spelling-mistake confusion dataset

and rhyme words လက်ရွေးစဉ်, လက်ယွေးစင်, ရက်ရွေးစင် and လက်ရွေးဇင် for Levenshtein and our three mappings for the threshold "<=1" are shown in Table VI. Moreover, our three mappings retrieved similar words well, compared with inputting raw Myanmar text. For example, although Levenshtein distance (for the threshold "<=1") retrieved only one similar word of လွှင့်စဉ် ("scatter" in English), our three mappings were able to retrieve three more similar words လွှင့်စင်, လွှင့်ဇင် and လွန့်စင် (see Table VII). One more example of cosine and our three mappings' string similarity distances of the word အကဲခတ် ("to assess" in English) (for threshold ">=0.9") can be seen in Table VIII. Here, "N\A" means "Not Applicable", and the expression is not contained in the threshold distance.

VII. Conclusion

In this paper, we have presented the first study of the string similarity measurement based on the pronunciation similarities for Burmese. We proposed three new mappings (phonetic mapping, sound mapping, and vowel position Mapping) and proved a better retrieving of similarly pronounced words, homophones, and rhyme words. Moreover,



(a) Levenshtein

(b) Damerau Levenshtein

(c) Hamming

(d) JaroWinkler

(e) Cosine

(f) Jaccard

Fig. 2: Results with the similar pronunciation dataset

the phonetic mapping and sound mapping are applicable for spelling correction by string similarity measurement of Burmese under the threshold "<=1". In the future work, we plan to expand the two datasets and conduct string similarity experiments to confirm our current mapping tables.

References

[1] Ohnmar Htun, Shigeki Kodama, Yoshiki Mikami, "Measuring Phonetic Similarities in Myanmar IDNs", 2010.

[2] Shigeaki Kodama, Yoshiki Mikami, Cross-language Phonetic Similarity Measure on Terms Appeared in Asian Languages, International Journal of Intelligent Information Processing Volume 2, Number 2, June 2011

[3] Anil Kumar Singh, "A Computational Phonetic Model for Indian Language Scripts", Proceedings of Constraints on Spelling Changes: Fifth International Workshop on Writing Systems, 2006

[4] Burmese Language Wikipedia Page: https://en.wikipedia.org/wiki/Burmese_language

[5] Levenshtein, V. I., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", Soviet Physics Doklady, Vol. 10, p.707, 02/1966

[6] Damerau, Fred J., "A technique for computer detection and correction of spelling errors", Communications of the ACM, 7 (3): 171-176, March, 1964

[7] Hamming, R. W, "Error detecting and error correcting codes". The Bell System Technical Journal. 29 (2): 147-160, April 1950

[8] Matthew A. Jaro, Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida, Journal of the American Statistical Association, 84(406):414-420, June 1989.

[9] Singhal, Amit, "Modern Information Retrieval: A Brief Overview", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35-43., 2001

[10] Jaccard, P., "Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines", Bulletin de la Societe Vaudoise des Sciences Naturelles, 241-272, 1901

[11] Odell, Margaret King , "The profit in records management Systems", New York, 20: 20, 1956

[12] Ye Kyaw Thu and Yoshiyori Urano, "Positional Mapping: Keyboard Mapping Based on Characters Writing Positions for Mobile Devices", Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 07, 110-117, 2007

[13] Thein Tun, "Acoustic phonetics and the phonology of the myanmar language", School of Human Communication Sciences, La Trobe University, Melbourne, Australia, 2007.

[14] Thein Tun, "The domain of tones in burmese", SST 1990 Proceedings, pp. 406–411, 1990.

[15] Jelly fish Documentation URL: https://buildmedia.readthedocs.org/media/pdf/jellyfish/latest/jellyfish.pdf

[16] Jelly fish python library for doing approximate and phonetic matching of strings: https://pypi.org/project/jellyfish/