String Similarity Measures for Burmese (Myanmar Language)

Khaing Hsu Wai  
UTYCC, Myanmar  
khainghsuwai@utycc.edu.mm

Ye Kyaw Thu  
NECTEC, Thailand  
ka2pluskha2@gmail.com

Hnin Aye Thant  
UTYCC, Myanmar  
hninayethant@utycc.edu.mm

Swe Zin Moe  
UTYCC, Myanmar  
swezinnmo.1011@gmail.com

Thepchai Supnithi  
NECTEC, 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.

1 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 (Tun, 1990). 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., “က”, “dance” in English) but also one syllable can form a word (e.g., “ြုကိက်”, “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.

2 Related Work

To the best of our knowledge, there is only one proposal that measured phonetic similarities of Myanmar Internationalized Domain Names (IDNs) (Ohnmar Htun, 2910). 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 (Htun et al., 2011).

The Myanmar characters ultimately descend from a Brahmic script, either Kadamba or Pallava (Wikipedia, 2019). Likewise, most of the major Indian languages such as Devanagari (e.g., Hindi, Marathi, Nepali), Bengali (Bengali and Assamese), Gurumuki (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 (Singh, 2006). 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.

3 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.

3.1 Levenshttein Distance

The Levenshtein distance (Levenshtein, 1966), 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”-“ကေလ်”(kan) and “ကန်”(kan) (exact pronunciation with “ကေလ်” 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).

\[
\text{dis}_{a,b}(i, j) = \begin{cases} 
0 & \text{if } i=j=0 \\
i & \text{if } j=0 \text{ and } i>0 \\
j & \text{if } i=0 \text{ and } j>0 \\
\min \left( \text{dis}_{a,b}(i-1, j) + 1, \text{dis}_{a,b}(i, j-1) + 1, \text{dis}_{a,b}(i-1, j-1) + 1(a_i \neq a_j) \right) & \text{otherwise}
\end{cases}
\]

3.2 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 (Damerau, 1964). 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 “ကေလ်”(“baby”) and “ကန်”(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.

### 3.3 Hamming Distance

The Hamming distance between two strings of equal length measures the number of positions with mismatching characters (Hamming, 1950). 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 “apple” and “grape” for error correction in fields such as telecommunication, cryptography, and coding theory.

### 3.4 Jaro–Winkler Distance

The Jaro–Winkler distance is another string metric that measures an edit distance between two sequences (Jaro, 1989). 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.

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

### 3.5 Cosine Similarity

The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them (Singhal, 2001). 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 “ခေလး” (“husband and wife”) and “ဇနီးေမာင်နှံ” (“baby”), the cosine similarity is 0.4. The Jaccard similarity is calculated according to the following equation.

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

### 3.6 Jaccard Similarity

The Jaccard similarity measures similarities between sets (Jaccard, 1901). 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.

### 3.7 Soundex Algorithm

The Soundex algorithm is a phonetic algorithm (Odell, 1956). 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.

### 4 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., “ချ” “luck” in English and “ကို,” “lake” in English). Moreover, some words have similar pronunciations and different meanings (e.g., “ကချ,” “seven” in English and “ကုချ,” “year” in English). To consider phonetically similar words, we propose three mapping tables for Myanmar words.

### 4.1 Phonetic Mapping

In our proposed methods, the first mapping is the phonetic mapping. Words with the same pronunciation are grouped together. For example, “ကာလး” and “ကာလး” have the same...
pronunciation. Therefore, “ဗ” (Ka) and “ဝ” (Kha) are clustered to “ဗ” (Ka) group. Likewise, other consonants with same pronunciation, such as “ဗ” (Ga) and “မ” (Gha), “ပ” (Pa) and “ဖ” (Pha), “မ” (Ba) and “ဖ” (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 1.

### Table 1: Phonetic Mapping

| Char | Mapped Char | Char | Mapped Char |
|------|-------------|------|-------------|
| က    | က             | ခ    | က               |
| ဂ    | ဂ             | ဃ    | ဂ               |
| င    | င             | ဟ    | ဟ               |
| ပ    | ပ             | ဖ    | ဖ               |
| ဗ    | ဗ             | ဘ    | ဘ               |
| ယ    | ယ             | ရ    | ရ               |
| လ    | လ             | သ    | သ               |
| ◌ျ | ျ            | ြ◌ | ြ◌           |

4.2 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, “ဗ” (Ka Kha Ga Gha Nga Ha A) are clustered to “ဗ” (Ka) group, “ဗ” (NyaGyi NyaLay) are clustered to “ဗ” (Nya) group, “ဗ” (Pa Ba Bha Ma) are clustered to “ဗ” (Pa) group, “ဗ” (YaPetLet YaGauk) are clustered to “ဗ” (Ya) group. The details of the sound mapping are shown in Table 2.

### Table 2: Sound Mapping

| Char | Mapped Char | Char | Mapped Char |
|------|-------------|------|-------------|
| က    | က             | ခ    | က               |
| ဂ    | ဂ             | ဃ    | ဂ               |
| င    | င             | ဟ    | ဟ               |
| ပ    | ပ             | ဖ    | ဖ               |
| ဗ    | ဗ             | ဘ    | ဘ               |
| ယ    | ယ             | ရ    | ရ               |
| လ    | လ             | သ    | သ               |
| ◌ျ | ျ            | ြ◌ | ြ◌           |

4.3 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 (Thu and Urano, 2007) according to their written positions.

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:

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

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 3. This mapping is designed for re-
trieving Myanmar words that have a similar vowel combination structure.

| Char | Mapped Char | Char | Mapped Char |
|------|-------------|------|-------------|
| a-z  | A-Z         | က-အ | F            |
| y    | o           | ြ    | p            |
| l    | i           | ု    | r            |
| u    | ျ           | ူ    | s            |
| i    | ြ           | ဳ    | n            |
| n    | 0-9         | d    | 0.9          |
| D    |             |      |              |

Table 3: Vowel Position Mapping

5 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.

5.1 Datasets

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

5.1.1 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:

- ကို ကို ြဂာီး - ကို ကို ြကီး
- ေကာင်းေကာငျ - ေကာင်းေကာင်း
- ေကာင်းကျပါတယ် - ေကာင်းြကပါတယ်
- ခွင့် မလွတ်ပါနဲ့ - ခွင့် မလွှတ်ပါနဲ့
- ငါ့စီ - ငါ့ဆီ
- စီးပွားေ◌၇◌း - စီးပွားေရး
- စွဲချက်တင်နိုြု င်ေသာေြကာင့် - စွဲချက်တင်နိုင်ေသာေြကာင့်
- ေတာင်ပန်အပ်ပါတယ် - ေတာင်းပန်အပ်ပါတယ်
- တို င်ြပည်ချစ်စိတ် - တို င်းြပည်ချစ်စိတ်
- ေဒါ်ေအာင်ဆန်းစု ြကည်-ေဒါ်ေအာင်ဆန်းစု ြကည်
- နက်နက်ရိူင်းရိူင်း - နက်နက်ရှိင်းရှိင်း
- ြပီးနှာပါမှာဆိုးြပီး - ြပဿနာတက်မှာစိုးြပီး
- ၂ဝ၁၂ဝ - ၂ဝ၂ဝ
- ၀◌ူးရှူး - ဝူရှူး
- အေဆာက်အဉီ - အေဆာက်အဦး

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., ကို ကို ြဂာီး - ကို ကို ြကီး, တနလာၤေန့ - တနလုါးြန့်ေန့).

Moreover, the spelling mistakes based on pronunciation similarity (e.g., ေကျးပွန်းစွာ - ေကျပွန်စွာ, ငါ့စီ - ငါ့ဆီ, စီးပွားေ◌၇◌း - စီးပွားေရး) and shape similarity (i.e., glyph) of Myanmar characters are also found (e.g., ြမူးတူး - ြမူးထူး, ပူးတူး - ပူးတူး, တိုြပည်ချစ်စိတ် - တိုြပည်ချစ်စိတ်). 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.

5.1.2 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 ြမူးတူး (“festivity” in English) is the correct word, the second column ြမူးထူး is the homophone word, and the other following columns ြမူးထူး, ြဖူးထူး, and ြမူးထူး are three rhyme words of the first column word (see Table 4). We collected 200 pairs for the similar pronunciation dataset, with 1,000 words in total.
Examples for how our three proposed mappings work can be seen as the following table.

| Phonetic Mapping | Sound Mapping | Vowel Position Mapping |
|------------------|---------------|------------------------|
| လက်ေရွးစင် | လက်ေရွးစဥ် | လက်ေယွးစင် |
| လက်ေရွးစဥ် | လက်ေယွးစင် | ရက်ေရွးစင် |
| လက်ေရွးဇင် | လက်ေရွးဇင် | လက်ေရွးဇင် |

Table 5: Examples of Three Proposed Mappings

5.2 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”.

6 Results and Discussion

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.

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 လက်ေရွးစင် and similar pronunciation and rhyme words လက်ေရွးစိုး, လက်ေရွးစိုး, လက်ေရွးစိုး and လက်ေရွးစိုး for Levenshtein and our three mappings for the threshold “<=1” are shown in Table 6. 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 လက်ေရွးစိုး (“scatter” in English), our three mappings were able to retrieve three more similar words လက်ေရွးစိုး, လက်ေရွးစိုး and လက်ေရွးစိုး (see Table 7). One more example of cosine and our three mappings’ string
similarity distances of the word ကဲခတ် (“to assess” in English) (for threshold “>=0.9”) can be seen in Table 8. Here, “N\A” means “Not Applicable”, and the expression is not contained in the threshold distance.

| Word - Similar Word | Levenshtein | Pronunciation | Sound | Vowel |
|---------------------|-------------|---------------|-------|-------|
| လက်ေရွးစင် | 1 | 0 | 1 | 0 |
| လက်ေယွးစင် | 1 | 0 | 0 | 0 |
| ရက်ေရွးစင် | 1 | 1 | 1 | 0 |
| လက်ေရွးဇင် | 1 | 1 | 0 | 0 |

Table 6: String similarity distances for the word “ကဲခတ်” (“selection”) in English

| Word - Similar Word | Levenshtein | Pronunciation | Sound | Vowel |
|---------------------|-------------|---------------|-------|-------|
| စဥ် | N/A | 0 | 1 | 1 |
| ဇင် | N/A | 1 | 1 | 0 |
| စင် | N/A | 1 | 1 | 0 |

Table 7: String similarity distances for the word “စဥ်” (“scatter” in English)

7 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 re-
Table 8: String similarity distances for the word “အကဲခတ်” (“to assess” in English)

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

experiments to confirm our current mapping tables.

References

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Comm. ACM*, 7(3):171–176.

R. W. Hamming. 1950. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29:147–160.

Ohnmar Htun, Shigeaki Kodama, and Yoshiki Mikami. 2011. Cross-language phonetic similarity measure on terms appeared in asian languages. *International Journal of Intelligent Information Processing*, Volume 2:9–21.
Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des dranes et dans quelques régions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:241–72.

Matthew A. Jaro. 1989. 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.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10:707.

Margaret King Odell. 1956. The profit in records management systems. page 20: 20, New York.

Yoshiki Mikami Ohnmar Htun, Shigeki Kodama. 2910. Measuring phonetic similarities in myanmar idns. March 2010, pages Page–129–135.

Anil Kumar Singh. 2006. A computational phonetic model for indian language scripts.

Amit Singhal. 2001. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43.

Ye Kyaw Thu and Yoshiyori Urano. 2007. Positional mapping: keyboard mapping based on characters writing positions for mobile devices. In *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 2007, Nagoya, Aichi, Japan, November 12-15, 2007*, pages 110–117.

Thein Tun. 1990. The domain of tones in burmese. pages pp. 406–411.

Wikipedia. 2019. *Burmese language wikipedia page*. [Online; accessed 18-July-2019].