International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:12, 2008

Sounds Alike Name Matching for Myanmar Language

Yuzana, Khin Marlar Tun

*Abstract*__Personal name matching system is the core of essential task in national citizen database, text and web mining, information retrieval, online library system, e-commerce and record linkage system. It has necessitated to the all embracing research in the vicinity of name matching. Traditional name matching methods are suitable for English and other Latin based language. Asian languages which have no word boundary such as Myanmar language still requires sounds alike matching system in Unicode based application. Hence we proposed matching algorithm to get analogous sounds alike (phonetic) pattern that is convenient for Myanmar character spelling. According to the nature of Myanmar character, we consider for word boundary fragmentation, collation of character. Thus we use pattern conversion algorithm which fabricates words in pattern with fragmented and collated. We create the Myanmar sounds alike phonetic group to help in the phonetic matching. The experimental results show that fragmentation accuracy in 99.32% and processing time in 1.72 ms.

*Keywords*__ natural language processing, name matching, phonetic matching

I. INTRODUCTION

PERSONAL name matching is of enormous substance for all applications that convenient with personal names. The problem with personal names is that they are not only one of its kind and sometimes even for single name many variations exist. This leads to the fact that databases on the one hand may have several entries for the same person and on the other hand have one entry for many different persons. Finding and matching personal names is at the core of an increasing number of applications: from text and Web mining, search engines, to information extraction, reduplication and data linkage systems. Variations and errors in names make exact string matching problematic, and approximate matching techniques have to be applied. When compared to general text, however, personal names have different characteristics that need to be considered [2].

Name matching can be defined as the process of determining whether two name strings are instances of the same name. As name variations and errors are quite common, exact name string comparison will not result in good matching quality. Generally, a normalized similarity measure between 1.0 (two names are identical) and 0.0 (two names are totally different) is used. The two main approaches for matching names are phonetic encoding and pattern matching. There are many methods of string search. The approaches to measuring word similarity can be divided into two groups, the orthographic and phonetic approaches [5].

The orthographic (look alike) approaches disregard the fact that alphabetic symbols express actual sounds, employing a binary identity function on the level of character comparison. The Edit distance measures count the number of steps required to transform one. Eg. Levenshtein distance measure. Some researcher classified character level errors as: typographical errors (correct spelling known), cognitive errors (lack of knowledge or misconceptions), and phonetic errors (similar sounding spelling).

The phonetic approaches (sounds alike) on the other hand, attempt to take advantage of the phonetic characteristics of individual sounds in order to estimate their similarity. Phonetic matching is a key component to approximate name-based searching such as Soundex method. It is measure for determining how the target set of attribute value is close in meaning to the query. Similarity measures can be based on tables, functions or taxonomical enumeration simple scalar value closeness. String similarity measures estimate the similarity between two strings based on the number of characters they have in common. Phonetic matching helps get the most out of database, by giving the power to perform 'approximate' searches, as opposed to exact matches. Approximate searches allow users to find the data they want, as opposed to limiting them to exactly what they asked for. Specifically, phonetic matching attempts to break a word down into its component sounds. This helps retrieve not only the exact matches, but variations such as Smith and Smyth. A Phonetic algorithm is an algorithm for indexing of words by their pronunciation. Most phonetic algorithms were developed for use with the English language; consequently applying the rules to word in other languages might not give a meaningful result [5]. In this paper we introduce background knowledge of our mother language and Phonetic nature of Myanmar language in section 2. Some related works compare to ours are described in section 3 and propose method, algorithm and in section 4. In section 5 are depicted the experimental results and conclusion in section 6.

II. MYANMAR LANGUAGE

The Myanmar language belongs to the Sino-Tibetan family of languages of which the Tibeto-Myanmar (Tibeto-Burman) subfamily forms a part. It has been classified by linguists as a monosyllabic or isolating language with agglutinative features. It is a tonal and analytic language. The language utilized the Burmese script which derives from the mon scripts and ultimately from the brahmi script. It is written from left to right and no spaces between words, although informal writing often contains spaces after each clause. It is syllabic alphabet and written in circular shape. It has sentence boundary mark. According to traditional tones on grammar, the Myanmar

language is said to have basically 33 consonants, viramas, independent vowels and dependent vowels altogether about 20 vowels and 4 medials. As we all know that language is a speech sound. So we can say that language is sounds, systematic, arbitrary and creative. The linguistic have spoken language and written language. So we found that some countries have different style in spoken language and written language. However, depending on the individual, the written marks may vary even for the same sound.

A. Place and Manner of Articulation in Myanmar Phonetic

Place of articulation is the point where active and passive articulators meet in the production of speech sounds. Manner of articulation is the way in which active and passive articulators are held apart after being held together. It involves how they are held apart – slowly or suddenly, whether the air-stream escapes through the mouth in a puff or with friction, etc. According to the survey made by the International Phonetic Association, there are eleven places of articulation shared by all languages of the world. International Phonetic Alphabet, IPA is a system of phonetic notation based on the Latin alphabet, devised by the International Phonetic Association [9] as a standardized representation of the sounds of spoken language. It is used by linguists, speech pathologists and therapists, foreign language teachers, singers, actors, lexicographers, and translators. There are only seven types of consonant in terms of their places of articulation in the Myanmar language. Similarly, there are only seven manners of articulation in the Myanmar sound system. Table – I shows the places of and manners of articulation in it.

TABLE I. THE PLACES AND MANNERS OF ARTICULATION IN MYANMAR LANGUAGE

PA --→ MA	BI	LD	DE	AL	PA	AP	PL	VE	GL
P	p, ph, b			t, th, d				k, kh, g	
S									ʔ
A					tʃ, dʒ	tɕ, tɕ, dʑ			
N	m, hm			n, hn			ɲ, hɲ	ŋ, hŋ	
L				l, hl					
F		f, v	θ, ð	s, sh z	ʃ, ʒ	ɕ			h
V	w			r			j		

PA : Place of Articulation
BI : Bilabial
LD : Labiodental
DE : Dental
AL : Alveolar
PA : Palato-Alveolar
AP : Alveolo-Palatal
PL : Palatal
VE : Velar
GL : Glottal

MA : Manner of Articulation
P : Plosive
S : Stop
A : Affricate
N : Nasal
L : Lateral
F : Fricative
V : Vowel Glide / Semi-Vowel, Semi-Consonant

III. RELATED WORK

A. Phonetic encoding

In many computer applications involving the recording and processing of personal data there is a need to allow for variations in surname spelling, caused for example by transcription errors. A number of algorithms have been developed for name matching, i.e. which attempt to identify name spelling variations, one of the best known of which is the Soundex algorithm. Common to all phonetic encoding techniques is that they convert a name string into a code according to how a name is pronounced (i.e. the way a name is spoken). Naturally, this process is language dependent. Most techniques have been developed mainly with English in mind. *Soundex* is the best known phonetic encoding algorithm. It keeps the first letter and converts the rest into numbers according to an encoding table. The idea of indexing information is how it sounds, rather than alphabetically was born. It has become known as soundexing. Soundex algorithm steps are initially retaining the first letter of the string in step 1. Step 2 removes all occurrences of (a, e, i, o, u, h, w, y) unless they are first letter. Step 3 is assigning the remaining letter (b, f, v, p) for 1, (c, g, j, k, q, s, x, z) for 2, (d, t) for 3, (l) for 4, (m, n) for 5 and (r) for 6 respectively. If two or more letters with the same number were adjacent in the original name (before step 1), then omit all but the first. Step 4 fill with 0 if there are < 3 digits or otherwise drop the rightmost digits. Step 5 is returning the first four characters. Thus we conclude that Smith=S530 and Smyth=S530 is same name [5]. *Phonex* [1] is a variation of Soundex that aims to improve the encoding quality by pre-processing names according to their English pronunciation. In paper [1] describes a comparative analysis of a number of these algorithms and, based on an analysis of their comparative strengths and weaknesses, proposes a new and an improved name matching algorithm, which call the Phonex algorithm. The analysis takes advantage of the recent creation of a large list of "equivalent surnames", published in the book Family History Knowledge UK. *Phonix* goes a step further than Phonex and applies more than one hundred transformation rules on groups of letters [10]. Some of these rules are limited to the beginning of a name, some to the end, others to the middle, and some will be applied anywhere. *NYSIIS (New York State Identification Intelligence System)*is based on transformation rules similar to Phonex and Phonix, but it returns a code only made of letters. *Double-Metaphone* [3] aims to better account for non-English words, like European and Asian names. The algorithm contains many rules that take the position within a name, as well as previous

and following letters, into account. Similar as NYSIIS, it returns a code only made of letters. *Fuzzy Soundex* is based on q-gram substitutions [2] and combines elements from other phonetic encoding algorithms. Similar to Phonix, it has transformation rules that are limited to the beginning or end of a name, or that are applicable anywhere. When matching names, phonetic encoding can be used as a filtering step (called *blocking* in data linkage), i.e. only names having the same phonetic code will be compared using a computationally more expensive pattern matching algorithm. Alternatively, exact string comparison of the phonetic encodings can be used.

B. For Arabic language

Researchers observed in [8] that the need for effective identity matching systems has led to extensive research in the area of name search. For the most part, such work has been limited to English and other Latin-based languages. Consequently, algorithms such as Soundex and *n*-gram matching are of limited utility for languages such as Arabic, which has a vastly different morphology that relies heavily on phonetic information. The dearth of work in this field is partly due to the lack of standardized test data. Consequently, they built a collection of 7,939 Arabic names, along with 50 training queries and 111 test queries. We use this collection to evaluate a variety of algorithms, including a derivative of Soundex tailored to Arabic (ASOUNDEX), measuring effectiveness using standard information retrieval measures. Our results show an improvement of 70% over existing approaches [8].

C. For Indian language

The multitude of Indian languages and dialects are written using 9 scripts. While each of these scripts has been encoded separately in the Unicode scheme, applications supporting Indian languages are yet to be found on a number of standard platforms. One primary reason could be the fact that rendering and processing in general, of Indian languages is complex and mandates distinctly different techniques. Orthography follows a phonetically driven basis of composition of "phonetic units" to form complex glyphs. While the character set is compact, authentic rendering implies a generative mechanism that can produce glyphs corresponding to all possible character sequences. Complex as it may seem, clear rules can be defined based on a canonical treatise by Panini, the ancient grammarian. The rules establish a perfect correspondence between phonemes constituting a syllable and its graphical form. And such rules can be defined for each of the Indic scripts. Decomposing text using this phonemic basis, followed by phoneme based computations provides a single unified technique for rendering Indic scripts. In fact, it is well suited even for other processing tasks such as sorting, searching, speech synthesis, speech recognition, transliteration, etc. In this paper, historical basis and complexity in the processing of Indic scripts are first presented. That is followed by the detailed explanation of phonemic scheme, its ancient historical background, applicability to Unicode and ISCII, and a unified technique for Indian language processing tasks, with specific examples from a shaping/rendering engine using OpenType fonts. Finally, experience from varied applications is briefly discussed in paper [6].

D. For Bangla language

Researchers presented in [4] a phonetic encoding for Bangla that can be used by spelling checkers to provide better suggestions for misspelled words. The encoding is based on the Soundex algorithm, modified to match Bangla phonetics. They start by analyzing Soundex encoding scheme when applied to Bangla. Next propose a new encoding that handles the case of Bangla words, including those containing conjuncts. They conclude with a demonstration of a prototype spelling checker that uses this phonetic encoding to offer suggestions for a set of misspelled Bangla words.

Phonetic matching is used to identify strings that are likely to have similar pronunciation, regardless of their spelling. The well-known phonetic matching techniques such as Soundex, although simple to compute, are in practice less effective than non phonetic measures such as edit distances. In paper [4] the authors analyze the problem of phonetic matching to explain why Soundex-like techniques are unlikely to work well, and proposed a more sophisticated technique for phonetic matching based on information derived from a pronunciation dictionary. This technique is not yet efficient enough for a production system, but is more effective than Soundex and Phonix.

IV. PROPOSED METHOD

Exact match does not guarantee the reliable result. Therefore many researchers attempt to achieve sounds a like matching method with their contribution. Hence, in English language they have been developed for phonetic encoding techniques such as soundex, phonex, phonix etc. We can see that not only for English but also in European languages like Germanic, Slavic, and French etc. they implemented phonetic or orthographic encoding for their languages since last decades. And some developed countries like Japan and Korea they reached to some extend in linguistic processing. Nowadays Unicode emerged for some languages as Arabic languages and Asian languages. These countries have been developed their languages processing in strong momentum. Our country Myanmar is one of the Asian countries and our mother languages have no white spaces or delimiters to segment the syllables. And we have lack of support for searching and sorting in DBMS. As a consequences, phonetic matching, sounds a like searching is still necessary in Unicode based application. We proposed sounds a like consonant group for Myanmar language in table II.

TABLE II. SOUNDS ALIKE CONSONANT GROUP

Group no	Phonetic sound	Characters
G001	(P,VE)	ဝ,ဃ
G002	(F,AL)	ဈ,ဆ
G003	(F,AL)	ဇ,ဈ
G004	(P,AL)	ဋ, တ

G005	(P,AL)	ဌ,ထ
G006	(P,AL)	ဦ,ပ,ဒ,ဓ
G007	(N,AL)	ဏ,န
G008	(P,BI)	ဗ,ဘ
G009	(V,P) (V,AL)	ယ,ရ
G010	(L,AL)	လ,ဠ
G011	(F,DE)	သ,ဿ
G012	Mixture	က,ခ,ပ,ဖ,မ,ဝ,ဟ,အ

TABLE III. ORDINARY VOWEL

	Low		Mid		High
1	အိ (í)	2	အီ (_ɪ)	3	အီး (ì)
4	အေ (é)	5	အေ့ (_e)	6	အေး (è)
7	အဲ့ (ɛ́)	8	အယ် (_ɛ)	9	အဲ (ɛ̀)
10	အ (á)	11	အာ (_a)	12	အား (à)
14	အော့ (ʹ ပ)	15	အော် (_ ပ)	16	အော (̀ပ)
17	အို့ (ó)	18	အို (_o)	19	အိုး (ò)
20	အု့ (ú)	21	အု (_u)	22	အူး (ù)

TABLE IV. NASALIZED VOWEL

23	အင့် (́ĩ)	24	အင် (_ĩ)	25	အင်း (̀ĩ)
26	အန့် (́ã)	27	အန် (_ã)	28	အန်း (̀ã)
29	အွန့် (́ũ)	30	အွန် (_ũ)	31	အွန်း (̀ũ)

TABLE V. COMBINED NASALIZED VOWEL

32	အိန့် (́eĩ)	33	အိန် (_eĩ)	34	အိန်း (̀eĩ)
35	အုန့် (́oũ)	36	အုန် (_oũ)	37	အုန်း (̀oũ)
38	အိုင့် (́aĩ)	39	အိုင် (_aĩ)	40	အိုင်း (̀aĩ)
41	အောင့် (́aũ)	42	အောင် (_aũ)	43	အောင်း (̀aũ)

TABLE VI. STOP VOWEL

44	အစ် (̀i ̆)	45	အက် (̀ɛ ̆)	46	အတ် (̀a ̆)
47	အုတ် (uʔ)	48	အိတ် (̀eɪ ̆)	49	အုတ် (̀ou ̆)
50	အိုက် (̀aɪ ̆)	51	အောက် (̀au ̆)	52	အ (မတြာဝက်) (ə)

In the sounds a like matching system there are 2 phases. First is the preprocessing and second is matching. The consonant combines with vowel and sometime it include medial to form the complete syllables in Myanmar language. In the preprocessing step we build the consonant groups which consist of 12 groups as in table. These consonants combine with medial and vowels group. The vowel group which consist of 21 ordinary vowel groups, 9 nasalize vowel groups, 12 combined nasalize vowel groups, 9 stop vowel groups shown in table III, IV, V and VI. After that we use pattern conversion algorithm stated in Fig 1. The inputs are the number of possible syllables in 672 syllables to generate pattern with their respective string pattern.

Phase 1: Pre Processing

Phase 1(a). Pre-define Case 1

Pattern conversion Algorithm

Input: Unicode Myanmar character
Output : String pattern
Begin
1. Accept input in Unicode character
For i=1 to end of string
Begin
2. Generate string pattern by using the specified code table
3. Fragment the string to form separate syllables
End;
4. Alter the string code to form collated string fragments
5. Return collated string
End;

Fig 1. Pattern conversion algorithm

We use pattern conversion algorithm because we want to store data into string pattern that is affective in pattern matching process. This algorithm works for Myanmar character into patterns and these patterns are being into segment the word boundary at the same time collated string. After that we select pattern into their phonetic group.

Phase: 1(b). Pre-define Case 2

And the admin can easily insert employee profile data into database. In this case names are inserted into Unicode. These name syllables are using pattern conversion algorithm as described in Fig 1. These syllables are segmented and collated and grouped into phonetic syllables. The result patterns are stored in word string pattern (W pattern) and grouped in phonetic group string pattern (Pg pattern) in the database.

Phase 2: Matching

In the matching phase, the user find with (name) match key. The match key uses pattern conversion algorithm in Fig 1 to segment and collate the syllables. We arrange for search key in string pattern to effectively and efficiently match in Unicode database application. The user can choose with 8 options to search the name. The detail step can be seen in Fig 2. Case 0 to 3 is ordinary name matching and case 4 to 7 is sounds a like (phonetic) name matching.

Matching algorithm

Procedure MySAKMatch( )
Input : Desire Match character in Unicode
Output: Result in Unicode character
W pattern = word group string pattern
Pg pattern = Phonetic group string pattern
Begin
1.Match key into pattern by using conversion algorithm
2.Get match key option
if option
{
case 0: exact match
if target string pattern ==source W pattern
return pattern;
case 1: starts with
if target string pattern ==1st syllable of source W pattern
return pattern;
case 2: ends with
if target string pattern ==Last syllable of source W pattern
return pattern;
case 3: include in
if target string pattern є source W pattern
return pattern;
case 4: exact match
if target string pattern ==source Pg pattern
return source code;
case 5: starts with
if target string pattern ==1st syllable of source Pg pattern
return pattern;
case 6: ends with
if target string pattern ==Last syllable of Pg pattern
return pattern;
case 7: include in
if target string pattern є source Pg string pattern
return pattern;
}
3. convert return patterns into Characters;
End;

Fig 2. Matching algorithm

V.IMPLEMENTATION AND RESULTS

Matching algorithm implementation can be seen as following steps in Fig 3. In the case of searching a person whose name is Miss Su Yin Yin San (မစုရင်ရင်စမ်း). Mr.(Maung) or (U)can be place infront of name like Mr in English name represent the male gender. (Ma) or (Daw) also represent female as Miss. So we remove this stop word. We also remove Dr., Capt., Maj., Col. etc. So we enter with (စုရင်ရင်စမ်း) Su Yin Yin San. The result come out with one name with 6 varitions of spelling in different consonant, virama, vowel. The step 1 gets syllable segmenting in precision of 99.77%, recall of 98.88% and F-Measure of 99.32%. Processing time of segementing the syllables is one syllable in 1.7 ms. The collation accuracy gets 95.88%. We will attempt to achieve efficient matching accuracy and processing time.

Match key
- စုရင်ရင်စမ်း

Step 1 : Pattern conversion
- |စု|ရင်|ရင်|စမ်း|
- 0600050000| 1c00000500| 1c00000500| 0600001a02

Step 2 : Match
- Chose Case 4,

Step 3 : Return in Myanmar Character
- Results
- ၁.စုရင်ရင်စမ်း
- ၂.စုယဥ်ယဥ်စန်း
- ၃.စုယဥ်ယဥ်ဆန်း
- ၄.ဆုယဥ်ယဥ်ဆန်း
- ၅.ဆုရင်ရင်စဏ်း
- ၆.ဆုရင်ရင်ဆဏ်း

Fig 3. Algorithm implementation

VI. CONCLUSION

Effectual name matching system is neceesary in unicode based application system in our country. We propose for sounds a like name matching method that is compatable for Myanmar language.We believe that our approach is reliable and efficient method for Myanmar character matching. We achieve good accuracy in syllable fragmentation and collation for matching process.

REFERENCES:

[1] A.J Lait and B. Randell. An assessment of name matching algorithms. Technical report, Deptartment of Computer Science, University of Newcastle upon Tyne, 1993.
[2] D. Holmes and C. M. McCabe. Improving precision and recall for soundex retrieval. In *Proceedings of the IEEE International Conference on Information Technology – Coding and Computing (ITCC)*, Las Vegas, 2002.
[3] L. Philips. The double-metaphone search algorithm. *C/C++ User's Journal*, 18(6), 2000.
[4] N.Uzzaman , M.Khan "A Bangla Phonetic Encoding for Better Spelling Suggestions", PAN Localization Project. International Development Research Centre, Ottawa ,Canada.
[5] P. Jokinen, J. Tarhio, and E. Ukkonen. "A comparison of approximate string matching algorithms". *Software Practice and Experience*, 26(12):1439–1458, 1996.
[6] R.K.Joshi, K.Shroff, S.P.Mudur." A phonetic Code Based Scheme for Effective Processing of Indian Languages", 23rdInternationalization and Unicode Conference, Prague, Czech Republic , March 2003.
[7] R. Cilibrasi and P. M. Vit´anyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.
[8] S.U.Aqeel, S.Beitzel, E.Jensen, O.Frieder and D.Grossman. "On the Development of Name Search Techniques for Arabic", Illinois Institute of technology ,Chicago,IL 60616
[9] The International Phonetic Association. University of Glasgow, Glasgow, UK, http://www.arts.gla.ac.uk/IPA/ipa.html
[10] T. Gadd. " PHONIX: The algorithm". *Program: automated Library and information systems*, 24(4):363–366, 1990.