Design and Development of Soundex for Assamese Language

Dhrubajyoti Baruah
Assistant Professor,
Dept. of Computer Application,
Jorhat Engineering College,
Jorhat, Assam, India.

Anjana Kakoti Mahanta, Ph.D
Professor,
Computer Science Dept.
Gauhati University,
Guwahati, Assam, India.

ABSTRACT

Differently written words may have similar pronunciations. In computerized textual analysis for English vocabularies, a code mapping is available, accessible in terms of a database function named ‘Soundex’. Depending on pronunciation, codes are generated which are utilized for word suggestion in searching, similarity detection, text mining etc. However, this Soundex is not multilingual and supports English language only. This paper describes designing and development of Soundex for Assamese language.

General Terms

Data mining, Soundex.

Keywords

Text Mining, Soundex, Similarity, Plagiarism.

1. INTRODUCTION

The Soundex algorithm for English is based on the phonetic classification of human articulation (bilabial, labiodental, dental, alveolar, velar, and glottal), which in turn are based on where we put our lips and tongue to make sounds. It was originally proposed for dealing with the problem of having different spelling variations of the same name (e.g., John vs. Jon), and since then it has been applied in several database applications for indexing texts, for instance, it was used in the U.S. census [1]. The Soundex code for a word consists of a letter followed by three numbers. The letter is the first letter of the word and the numbers encode the remaining consonants. Similar sounding consonants share the same number so, for example, the bilabial B, F, P and V are all encoded as 1. Vowels can affect the coding, but are never coded directly unless they appear at the start of the name. The algorithm is shown below:

Step 1. Retain the first letter of the string

Step 2. Remove all occurrences of the following

letters, unless it is the first letter: a, e, i, o, u, w,h,y

Step 3. Assign numbers to the remaining letters (after

the first) as follows:

Table 1. English alphabet grouping

Letter	Number
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M,N	5
R	6

Step 4. If two or more letters with the same number were adjacent in the original name (before step 1), or adjacent except for any intervening h and w, then omit all but the first.

Step 5. Return the first four bytes padded with 0.

As an illustration, for the string ‘America’, soundex code is ‘A562’. If someone, by mistake, writes the string as ‘Amereca’, the same code ‘A562’ is generated by soundex. On this basis we can suggest for correct word, find similarity and can work for plagiarism detection. The algorithm is built for English alphabets only and does not support non-roman scripts.

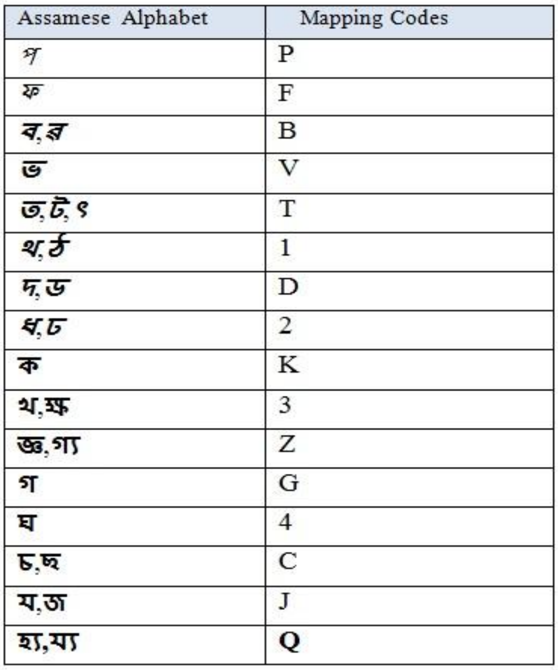
2. RELATED WORKS

Various works related to soundex are carried out for different languages. For instance, set of phonetic matching functions are developed for Japanese language as a joint research of Gunma University, Japan and RMIT university, Australia[2]. Researchers of University of Science and Technology, Algeria have developed soundex algorithm for classical Arabic language[3].India, which has 22 constitution- recognized languages, is also focusing on computational phonetic research. For example, researchers of NMIMS University of Mumbai have developed phonetic similarity finding techniques for Hindi and Marathi languages[4]. Soundex algorithm for Odia language is developed by a research group of Indian Institute of Information Technology, Bhubaneswar.[5] This Odia soundex is coded on human articulation which is also the base of classical English soundex algorithm. Researchers of Gujarata Technological University have worked on soundex algorithm for Hindi and Gujarati language and applied on names along with their variations in order to retrieve the output with minimum false hits.[6]. H.A. Shedeed of A.S. University of Egypt has worked on soundex algorithm to implement in computer based short answer type examination system. His methodology is based on applying the Soundex phonetic algorithm on the answer’s word for English or Arabic language to facilitate a computer based intelligent marking method. The student who responds with the correct spelling answer’s word takes the total point of the question while the student who responds with the correct sounding but not correct spelling word may take points less than or equal to the total points according to the considered subject and the instructor’s opinion[7]

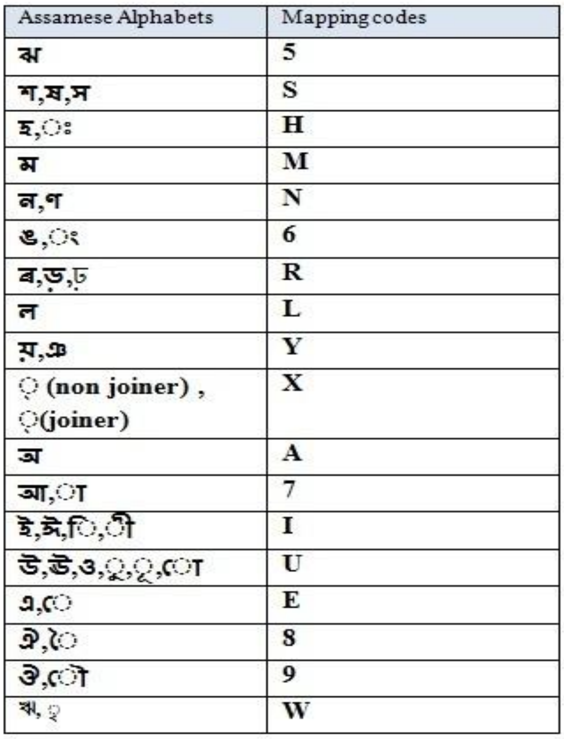
3. OUR APPROACH

Following list is prepared for categorizing Assamese alphabets which also includes some conjuncts . The list is prepared on the basis of similarity in pronunciation, not on the basis of human articulation.

Table 2:Assamese Alphabet Grouping



Assamese Alphabet	Mapping Codes
প	P
ফ	F
ব,ৱ	B
ভ	V
ত,ট,ৎ	T
থ,ঠ	1
দ,ড	D
ধ,ঢ	2
ক	K
খ,ক্ষ	3
জ্ঞ,গ্য	Z
গ	G
ঘ	4
চ,ছ	C
য,জ	J
হ্য,য্য	Q



Assamese Alphabets	Mapping codes
ঝ	5
শ,ষ,স	S
হ,ঃ	H
ম	M
ন,ণ	N
ঙ,ং	6
ৰ,ড়,ঢ়	R
ল	L
য়,ঞ	Y
্ (non joiner) , ্(joiner)	X
অ	A
আ,া	7
ই,ঈ,ি,ী	I
উ,ঊ,ও,ু,ূ,ো	U
এ,ে	E
ঐ,ৈ	8
ঔ,ৌ	9
ঋ, ৃ	W

Instead of considering conventional categorization of bilabial, labiodentals or dental, graphs generated by recording software system are analysed to determine similarity in pronunciation. For instance, graphs corresponding to চ and ছ **are shown below:**

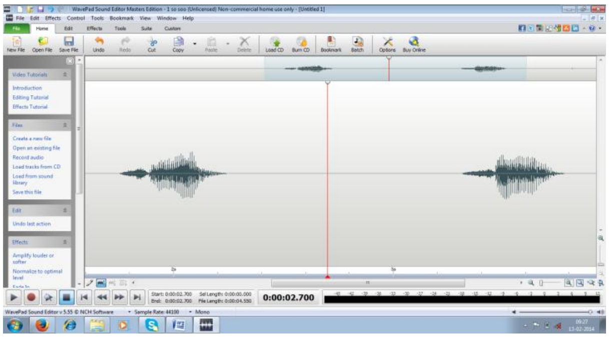


Figure 1: Graphs generated for চ and ছ

Although জ্ঞ and গ্য **are not single alphabets but conjuncts, these are placed on a group of the list due to similarity in pronunciation. Same is the fact for** হ্য and য্য**. Singular** হ **and** য **do not occupy the same group in the list but their conjunction with 'jyo-kaar' give rise to similarity in pronunciation. Hence it is decided to create a group for these two conjuncts. On the other hand, alphabets like** ন and ণ **share the same group and hence for their conjuncts like** ন্য,ণ্য **we need not to create any separate group.**

In Assamese script typing, ্(joiner) is used to join two singular alphabets that forms conjuncts. ্ **(non joiner) or 'holonto' is used to indicate to shorten the pronunciation of former alphabet. Neither** ্ (joiner) nor ্**(non joiner or holonto) has it's own pronunciation. But both play the same role in pronunciation of the associated word. Hence these are allotted into the same group of the listing.**

On the basis of the grouping, the following PL/SQL program is developed to generates soundex codes:

Table 3: PL/SQL code

```
create or replace function axom_soundex(word nvarchar2)
return varchar2
is
i number(4):=1; --Loop counter
j number(4):=1; --Loop Counter
len number(4); --for storing length of the given Assamese
word
letter nvarchar2(1); --for processing single letter
zukta nvarchar2(4); --for processing special zuktakhyar
scode nvarchar2(23); --final soundex code goes here
axom_word nvarchar2(15);
--Begining:defining array for storing soundex code
type kodes IS VARRAY(23) OF varchar2(3);
sounds kodes:=kodes();
--Ending:defining array for storing soundex code
begin
scode:=''; --initialize scode with empty string
--Begining:Initialization of array for storing soundex code
FOR i in 1 .. 23 LOOP
    sounds.extend();
   sounds(i):='0';
  END LOOP;
--Ending:Initialization of array for storing soundex code
  axom_word:=unistr(word); --input is coming in form of
Hexa code,converted into char
i:=1; --Initialization Loop counter
select length(axom_word) into len from dual; --storing length
```

```
of assamese axom_word in variable 'len'

while(i<=len)
loop
select substr(axom_word,i,1) into letter from dual;
--Storing single character in variable 'letter'
select substr(axom_word,i,3) into zukta from dual;
 --Storing three character in variable 'zukta'
--beginning of checking special zuktakhyar
if(zukta='হ্য' OR zukta='য্য') THEN
sounds(j) := 'Q';
i:=i+3;
ELSIF (zukta='ক্ষ') THEN
sounds(j) := '3';
i:=i+3;
ELSIF (zukta='জ্ঞ' OR zukta='গ্য') THEN
sounds(j) := 'Z';
i:=i+3;
--end of checking special zuktakhyar
ELSE
--beginning of checking single letter
CASE
   WHEN (letter='প') THEN sounds(j) := 'P';
  WHEN letter = 'ফ' THEN sounds(j) := 'F';
   WHEN (letter = 'ব' OR letter='ৱ') THEN sounds(j) := 'B';
   WHEN (letter = 'ভ') THEN sounds(j) := 'V';
   WHEN (letter = 'ত' OR letter='ট' OR letter='ৎ') THEN
sounds(j) := 'T';
   WHEN (letter = 'থ' OR letter='ঠ') THEN sounds(j) := '1';
   WHEN (letter = 'দ' OR letter='ড') THEN sounds(j) := 'D';
WHEN (letter = 'ধ' OR letter='ঢ') THEN sounds(j) := '2';
   WHEN (letter = 'ক') THEN sounds(j) := 'K';
   WHEN (letter = 'খ') THEN sounds(j) := '3';
   WHEN (letter = 'গ') THEN sounds(j) := 'G';
   WHEN (letter = 'ঘ') THEN sounds(j) := '4';
   WHEN (letter = 'চ' OR letter='ছ')  THEN sounds(j) := 'C';
   WHEN (letter = 'য' OR letter='জ')  THEN sounds(j) := 'J';
   WHEN (letter = 'ঝ')  THEN sounds(j) := '5';
 WHEN (letter = 'শ' OR letter='ষ' OR letter='স')  THEN
sounds(j) := 'S';
   WHEN (letter = 'হ' OR letter='ঃ')  THEN sounds(j) := 'H';
   WHEN (letter = 'ম')  THEN sounds(j) := 'M';
   WHEN (letter = 'ন' OR letter='ণ')  THEN sounds(j) := 'N';
   WHEN (letter = 'ঙ' OR letter='ং')  THEN sounds(j) := '6';
   WHEN (letter = 'ৰ' OR letter='ড়' or letter = 'ঢ়')  THEN
sounds(j) := 'R';
   WHEN (letter = 'ল')  THEN sounds(j) := 'L';
   WHEN (letter = 'য়' OR letter='ঞ')  THEN sounds(j) := 'Y';
   WHEN (letter = '্' OR letter = '')  THEN sounds(j) := 'X';
   WHEN (letter='অ') THEN sounds(j):='A';
   WHEN (letter = 'ঋ' OR letter='ৃ') THEN sounds(j):='W';
   WHEN (letter='া') THEN sounds(j) := '7';
   WHEN  (letter='আ') THEN sounds(j) := '7';
   WHEN (letter='ই' OR letter='ি' OR letter='ঈ' OR
letter='ী') THEN sounds(j) := 'I';
   WHEN (letter='উ' OR letter='ু' OR letter='ঊ' OR letter='ূ'
OR letter='ও' OR letter='ো') THEN sounds(j) := 'U';
WHEN (letter='এ' OR letter='ে') THEN sounds(j) := 'E';
  WHEN (letter='ঐ' OR letter='ৈ') THEN sounds(j) := '8';
  WHEN (letter='ঔ' OR letter='ৌ') THEN sounds(j) := '9';
  ELSE sounds(j) := '0';
  END CASE;
i:=i+1;
  --end of checking single letter
END IF;
j:=j+1;
end loop;
FOR i in 1 .. 23 LOOP
      scode:=scode || sounds(i); --join each individual code in
array 'sounds'
   END LOOP;
scode:=REPLACE(scode, 'XX', 'X');  --make joiner and
hasanta equal
   scode:=TRIM(TRAILING '0' FROM scode);
 --remove unwanted zeros
return scode;
end;
```

4. PERFORMANCE OF THE PROGRAM AND EXPERIMENTAL RESULTS

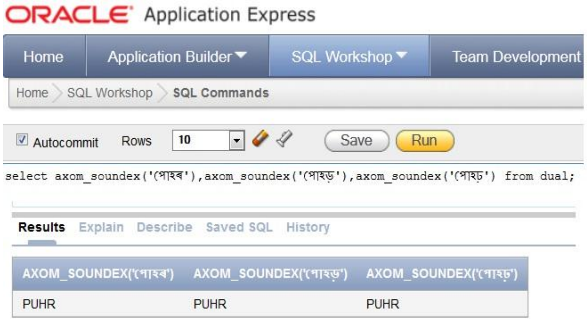
Our developed Assamese soundex is tested for several words and satisfactory results are found. Let us consider two words ‘নতুন’ and ‘নতুণ’. The first is correctly written while second is having spelling mistake, without any diversion from pronunciation. A call to our function axom_soundex in SQL: -

Select axom_soundex('নতুন'),axom_soundex('নতুণ') from dual; yields ‘NTUN’ and ‘NTUN’. Generation of same code exhibits that these two words have same pronunciation. Some other examples are listed below:

Table 4: example

Serial No	Words	Axom_soundex code
Sl 1	পোহৰ,পোহড়,পোহঢ়	PUHR,PUHR,PUHR
Sl 2	দলং, দলঙ	DL6, DL6
Sl 3	সহ্য,সয্য	SQ, SQ
Sl 4	ক্ষুৰধাৰ,খুৰধাৰ	3UR27R, 3UR27R
Sl 5	জ্ঞান,গ্যান	Z7N, Z7N
Sl 6	কষ্ট, কস্ত	KSXT, KSXT
Sl 7	চহৰীয়া,ছহৰীয়া	CHRIY7, CHRIY7

In serial no 1, last alphabets of the three words are listed in the same category represented by code R. Initial alphabets being the same, our soundex generates the same code representing the fact that these have similar pronunciations. Same fact is applicable for generation of codes in serial no 2. Snapshot of experiment done using oracle Application express is shown below:



```
select axom_soundex('পোহৰ'),axom_soundex('পোহড়'),axom_soundex('পোহঢ়') from dual;
```

AXOM_SOUNDEX('পোহৰ')	AXOM_SOUNDEX('পোহড়')	AXOM_SOUNDEX('পোহঢ়')
PUHR	PUHR	PUHR

Figure2: Experimental Result

In serial number 3, second alphabets of the two words 'হ' and 'য' do not belong to same category as their pronunciations are not similar. But the combinations, 'হ্য' and 'য্য' results the same pronunciation. This is tackled by creating a separate group indicated by code Q. Alphabet

' ক্ষ', that appears in serial no 4 , has no Unicode slot. For generation of soundex code, we were to consider 'ক্ষ' as composition of ক, joiner(্) and ষ. Serial no 5 depicts similar code generation between conjuncts and singulars. Serial no 6 depicts similar code generation between two conjuncts. In serial no 7, although the first alphabet is different , same code is generated proving similarity in pronunciation. English soundex is not able to do it; if the first alphabet is different, Englsih soundex generates different codes. For example, 'psychology' and 'sychology' get different codes 'P224' and 'S242' respectively, although they are similar in pronunciation.

5. ASSAMESE SCRIPT AND UNICODE

Assamese language is the native language spoken by more than 1.3 crore people as per 2001 Govt. of India census. Worldwide estimate of Assamese speaking people in 2007 is 15.4 million.[8] Assamese language is written using Assamese scripts that has a glorious historical base. The Umachal rock inscription of the 5th century evidences the first use of Assamese script in the region. Rock and copper plate inscriptions from then onwards, and *Xaansi* bark manuscripts right up to the 18th–19th centuries show a steady development of the Assamese script. The script could be said to develop proto-Assamese shapes by the 13th century. [9]

Unicode, the global consortium of the computing industry standard, has not allotted any separate slot for Assamese alphabets, and due to similarity with Bengali scripts, their alphabets are being used of late. 'ৰ ' ,' ৱ ' etc. are unique alphabets of Assamese scripts. In unicode, these are listed with Bangla script as special addition. Sorting of filenames written in Assamese, is hence not possible. Due to improper presentation of Assamese alphabets in unicode, software programming also becomes chaotic. For instance, in the PL/SQL program shown above, processing of Assamese alphabet ' ক্ষ ' was a problem. 'ক্ষ' is an independent alphbet of Assamese script whose singular exsistance is not available in Bangla. Unicode has not given any extra slot of hexadecimal code to this alphabet. Inspite of being a singular alphabet, we were to treat it as conjucts of ক, joiner(্) and ষ. Hence, in programming, we were to treat this alphabet as conjuct. Such improper unicode slot allotment of Assamese alphabets will create further problem in Assamese language related research and development. For example , bubble sort or any other sorting technique will not work for Assamese alphabets which is fairly possible for Bangla alphabets.

6. CONCLUSION AND FUTURE SCOPE:

Algorithm of Soundex for Assamese language is implemented in PL/SQL programming and hence callable from any front-end platform. Although the algorithm works fine for most of the cases, it is open for modification to tackle any unseen drawback. Unicode consortium may be requested to consider the different issues of Assamese scripts discussed throughout this paper. Developed program may be utilized in different linguistics and computational tasks ranging from word suggestion to plagiarism detection[10].Oracle corporation may consider to include this function in its future releases with native language support. In future, the developed soundex tool may also be used in the research and development of computerized short answer type examination system in Assamese language.

7. REFERENCES

[1] M. Alejandro Reyes-Barragán, Luis Villaseñor Pineda, Manuel Montes-y-Gómez, "A Soundex-based Approach for Spoken Document Retrieval", 7th Mexican International Conference on Artificial Intelligence, Atizapán de Zaragoza, Mexico, October 27-31, 2008 Proceedings: pp 204-211, 2008.

[2] Michiko Yasukawa, J. Shane Culpeppery, Falk Scholery, "Phonetic Matching in Japanese", SIGIR 2012 Workshop on Open Source Information Retrieval.August 16, 2012, Portland, Oregon, USA.

[3] Nedjma Djouhra Ousidhoum, Asma Bensalah, and Nacéra Bensaou, "A New Classical Arabic Soundex Algorithm", Proc. of Int. Conf. on Advances in Communication and Information Technology 2012, ACEEE.

[4] Sandeep Chaware, Srikantha Rao, "Analysis of Phonetic Matching Approaches for Indic Languages ", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2012.

[5] Rakesh Chandra Balabantaray, Bibhuprasad Sahoo, Sanjaya Kumar Lenka, Deepak Kumar Sahoo, Monalisa Swain, IIIT Bhubaneswar, "An Automatic Approximate Matching Technique Based on Phonetic Encoding for Odia Query", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012.

[6] R Shah, D Singh, Gujarat Technological University, "Improvement Of Soundex Algorithm For Indian Language Based On Phonetic Matching" International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.4, No.3, June 2014

[7] Howida AbdelFattah Shedeed,Faculty of computers and Information Sciences, Ain Shams University Cairo, Egypt, "A New Intelligent Methodology for Computer based Assessment of Short Answer Question based on a new Enhanced Soundex phonetic Algorithm for Arabic Language", International Journal of Computer Applications, November,2011.

[8] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

[9] http://en.wikipedia.org/wiki/Assamese_alphabet

[10] Dhrubajyoti Baruah, Anjana Kakoti Mahanta, "A New Similarity Measure with Length Factor for Plagiarism Detection", International Journal of Computer Applications (0975 – 8887) Volume 72– No.14, May 2013