

IMPROVEMENT OF SOUNDEX ALGORITHM FOR INDIAN LANGUAGE BASED ON PHONETIC MATCHING

Rima Shah¹

¹Department of Computer Science And Engineering, Parul Institute of Engineering and Technology, Gujarat Technological University, Gujarat, Vadodara

Dheeraj Kumar Singh²

²Department of Information and Technology , Parul Institute of Engineering and Technology, Gujarat Technological University, Gujarat, Vadodara

ABSTRACT

In a system with a large database, there always has been a problem that names may not be spelled well or might not be spelled in a way that one expected. So, data in the database gets degraded. In this case it is required to search the duplicates and merge them in the single entity. In doing so, one problem is that the way in which the strings would be compared. In such cases rather than looking for exact match, approximate string matching would be appreciable. One of the string matching techniques is Phonetic matching which is used to compare the name based on the pronunciation of the words. The similar sounding words could be retrieved from the large database using different phonetic matching algorithm and best known algorithm is Soundex algorithm. Phonetic matching is needed when many people from different culture come together. They either speak with different pronunciation or their writing habits are different. This scenario is very common in India, as we have many different languages like Hindi, Gujarati, Marathi, Tamil etc. In this research work Soundex algorithm is used for Hindi and Gujarati language and applied on the names along with their variations in order to retrieve the output with minimum false hits.

KEYWORDS

Phonetic matching, SoundEx algorithm, Name variations

1. INTRODUCTION

A large number of researches have already being carried out in a well known area of Information retrieval under data mining. One of such technique of information retrieval is Phonetic matching which is used to compare the name based on the pronunciation of the words. The similar sounding words could be retrieved from the large database. Phonetic matching is needed when many people from different culture come together. They either have different styles of pronunciation or have different writing styles for various languages, but their meaning is same. This scenario is very common in India, as we have many different languages like Hindi, Gujarati, Marathi, Tamil etc. Phonetic comparison algorithms are precisely defined methods for quantifying the similarity between speech forms or segments, words, or even entire languages on the basis of their sounds^[5]. Phonetic matching is used to identify the strings that sounds similar, meaning that the string having the similar pronunciations regardless of their spelling. For example if a data operator is given a name for finding out the information from the database for the same.

The operator guesses the possible spelling of the given name or spelling provided to operator may be incorrect. In this case Phonetic matching plays an important role in Information retrieval means searching the data in a database for retrieval. Phonetic matching is used to evaluate the similarity of pronunciations of pairs of strings ^[9]. It focuses on the pronunciation of the word independent the spelling of words. The most common issue with name matching is Name Variations and name variations are like Spelling variations, Phonetic variations, Double names, Double first names ^[4].

The strings can be spelled using different writing styles, but they can be matched phonetically. All the strings represent the same keyword, only way of writing is different ^[2]. Since in rural areas, the word may be spelled or pronounce either wrongly or differently ^[2]. We are able to retrieve the data using phonetic matching. There is no need of exact string matches. There are many techniques had been proposed in order to find the phonetic matching of strings like Soundex, Kstring and Q-gram, Metaphone coding, Daitch Mokotoff, Edit Distance etc ^[4]. Each technique generates a code for the strings and matches them through edit distances ^[4]. In this paper we proposed an approach, which provides a simple and efficient way of matching the strings. Our scheme will work on SoundEx Algorithm for Indian languages especially Hindi, Gujarati. It has been found that most of the rural people don't either spell correctly or make correct pronunciation of the keywords properly ^[2].

The main objectives of the proposed system are, first, to convert the entered strings into its equivalent phonetic forms by applying phonetic rules for each language. Second, is to retrieve the similar sounding names with minimum false hits for both Hindi and Gujarati Language.

2. PHONETIC MATCHING APPROACH

Phonetic Matching performs the matching operation based on the pronunciation of words^[9]. To understand the working of matching operation we will discuss the example of large database that consists of the names Stefan, Steph, Stephen, Steve, Steven, Stove, and Stuffin ^[3]. Suppose that we want to search for the name Stephen ^[3]. The matches that the search finds are called the positives, and those names that it rejects are called the negatives ^[3]. Those positives that are relevant are called true positives, and the others are false positives ^[3].

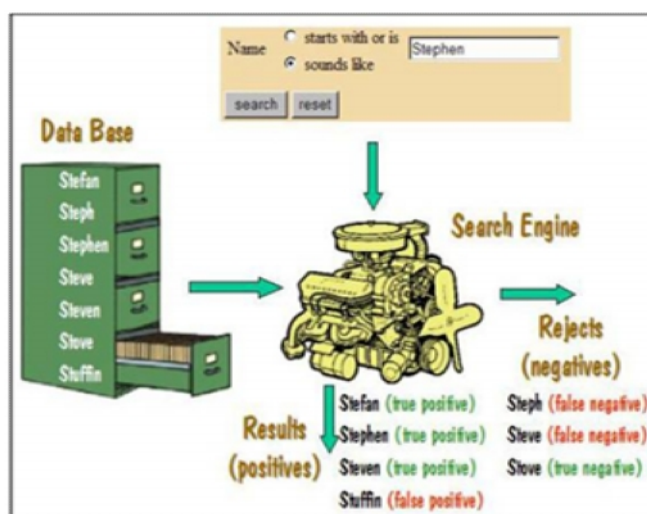


Figure 1: True and False Negative ^[3]

As an example, let us assume that the matches found when searching for Stephen in the above database are Stefan, Stephen, Steven, and Stuffin^[3]. The first three are probably relevant, and are names that we would have wanted to see. So these are the true positives^[3]. Stuffin, however, is probably not relevant – it is a false positive^[3]. The names that were rejected are Steph, Steve, and Stove^[3]. Of those, Stove is probably not one that we would have wanted. So it is a true negative^[3]. But Steph and Steve are ones that we would probably be interested in^[3]. They are false negatives. A large number of researches are already being carried out in a well known area of Information retrieval under data mining. One of such technique of information retrieval is Phonetic matching which is used to compare the name based on the pronunciation of the words. The similar sounding words could be retrieved from the large database. For this, many name matching algorithms are used like SoundEx algorithm, Edit Distance algorithm, K-String and Q gram algorithm, Guth algorithm, Daitch Mokotoff algorithm, Metaphone coding algorithm.

A. Soundex Algorithm:

Searching names in large database have always been a problem. The solution to the problem was given by Robert Russell in 1912 as he proposed SoundEx algorithm^[10]. The names might be misspelled in a large database or might not be spelled as one expected. In this case rather than looking for exact matching, searching for approximate matching will be significant^[6, 7]. One solution is to say that two names are approximate matches if they sound the same. And a method known as SoundEx was developed to determine if two names sound similar^[8]. SoundEx is the best-known phonetic matching scheme. Developed by Odell and Russell, and patented in 1918, SoundEx uses codes based on the sound of each letter to translate a string into a canonical form of at most four characters, preserving the first letter^[1]. SoundEx is a system whereby values are assigned to names in such a manner that similar-sounding names get the same value. These values are known as SoundEx encodings. A search application based on SoundEx will not search for a name directly but rather will search for the SoundEx encoding. Based on the SoundEx encoding the similar sounding names would be retrieved from the large database.

1. Retain the first letter of the string
2. Change all occurrences of the following letters to zero: a, e, h, i, o, u, w, y
3. Assign numbers to the remaining letters (after the first) as follows:
 - b, f, p, v = 1
 - c, g, j, k, q, s, x, z = 2
 - d, t = 3
 - l = 4
 - m, n = 5
 - r = 6
4. Remove all pairs of digits which occur beside each other from the string that resulted after the previous step.
5. Remove all the zeros from string that results from the previous step.
6. Return the first four characters, right-padding with zeroes if there are fewer than four.

Figure 2: Outline of Soundex Algorithm^[1]

Taking an example we will see how SoundEx algorithm works. Example-"SMITH" will code to "S5030" which will then reduce to "5300" by computing the steps of SoundEx algorithm.

3. PROPOSED APPROACH

The main idea of proposed approach is to use the Soundex algorithm for Indian Language, which has been used for European, German, Shri-Lankan etc. language till today. Using Soundex algorithm we can retrieve the names and surnames that sounds similar, independent of their spelling. The result may be with minimal false hits which will show the accuracy of the algorithm. Soundex algorithm works fine, but only for English. But more than spelling, in India, we have another issue to be addressed: Words getting transliterated among Indian Languages. For example: In railway reservation chart, name of the person will be written in English as well as in Hindi. If the person is from Kerala and the name is transliterated to some other language. The only thing remain same is its pronunciation. So it will be great if we can search on this data based on pronunciation. As shown in the figure 2 the proposed System consists of Parsing, Phonetic Equivalent, and Comparison Module and Output module. Two strings will be given as input by user. The strings may be in English, Hindi or in Gujarati Language. The inputted strings will be then passed to parsing module for the further procedure. Parsing algorithms specify how to recognize the strings of a language and assign each string one (or more) syntactic analyses. Here parsing will be analyzing the string which is given as input and after analyzing the string it will pass it to the phonetic Equivalent module. The language of the string will be identified based on the first character of string.

E.g. If the String "સૈનિક" is given as input, language will be identified as Gujarati language from the first character 'સ'.

E.g. If the string "नूपुर" is given as input, language will be identified as Hindi language from the first character 'न'.

The main goal of phonetic equivalent string module is to generate the soundex code for the input string. The string from the parser will be then extracted in phonetic equivalent string module. Once the language of the string is identified in previous module, the soundex code for respective language will be generated. The soundex code of the two inputted strings will be compared in the next module that is comparison module.

E.g. If we take example of Two Gujarati strings "સૈનિક" "સૈનીક"
Then,

- 1) First 'સ' will be identified then it will generate soundex code S.
- 2) 'ૈ' will be identified then it will generate soundex code D.
- 3) 'ન' will be identified then it will generate soundex code L.
- 4) 'િ' & 'ી' both will be identified as B
- 5) 'ક' will be identified then it will generate soundex code F.

E.g. If we take example of two Hindi strings "नूपुर" and "नपुर"

Then,

- 1) First 'न' will be identified then it will generate soundex code L.
- 2) 'ु' & 'ू' will be identified then it will generate soundex code C.
- 3) 'प' will be identified then it will generate soundex code M.
- 4) 'र' will be identified then it will generate soundex code P.

The comparison module will compare the two input strings. Based on the comparison it will generate the result in the form of, if two strings are similar sounding or not. If the two strings are similar sounding and even though the result is given as false, we consider it as false hit. To reduce the false hit ratio string comparison technique will be applied.

If the two strings are similar sounding and even though the result is given as false, then string comparison technique will be applied. Based on this process the length of the both strings will be identified. Then the number of similar characters will be counted and the number of dissimilar character will be counted. If the numbers of dissimilar characters are minimum then we can consider it as similar sounding strings. Based on the whole process false hit ratio can be reduced.

E.g. if two strings "सैनिक" and "सैनीक" are taken as input then they will be encoded as 'SDLBF'. As both strings get same encoding the result is given as both strings sounds alike. This we can consider as true positive.

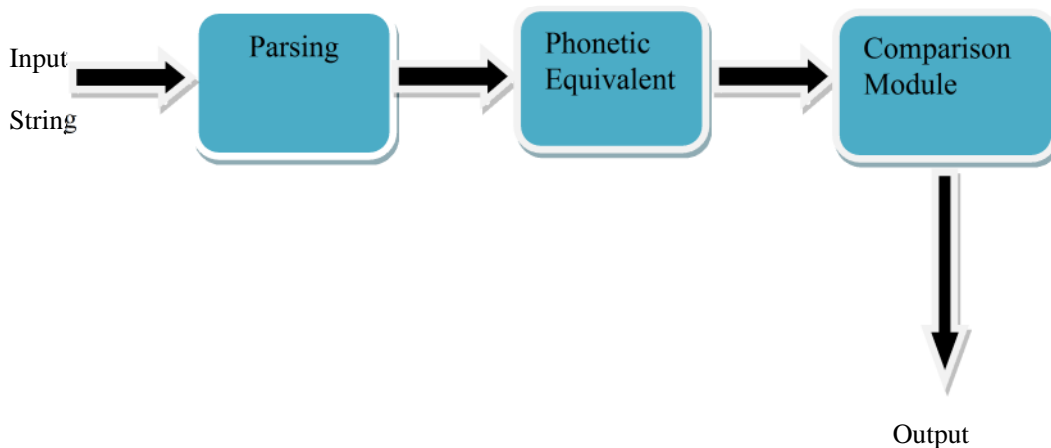


Figure 3: System Design

E.g. for the two Gujarati words " " & "ચન્" "

It is similar sounding but it comes up with soundex code HNKP & HLKP respectively. Now even though both are similar it is giving the false positive.

E.g. if two strings "कुंदन" and "कुन्दन"

It is similar sounding but it comes up with soundex code FCNKL & FCL0KL respectively. Now even though both are similar it is giving the false positive.

Finally the output module displays the result in the form of if two strings are similar sounding strings or not.

4. MODIFIED SOUNDEX ALGORITHM FOR INDIAN LANGUAGE

The modified algorithm takes two strings as input to match the two words. Based on the first character of Word, the language of word will be identified. The language may be Hindi, Gujarati or English. The modified soundex Algorithm will generate the Soundex Code of entered words and compare the two Soundex code. If the Soundex code generated is matched, then result generated as the words entered as input are similar sounding names. If two words are similar sounding name and even if the result is given as false then by applying comparison technique the total number of characters the two words will be counted. If total numbers of dissimilar characters are less than two words are to be considered as similar sounding names.

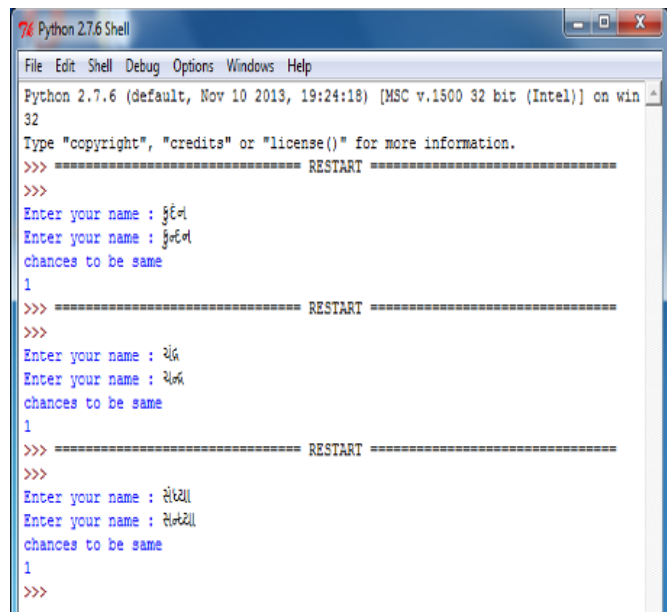
- 1) Get the input String. Two words to match
- 2) Find the language of both words using the first character
- 3) Each character will be converted in to lower character
- 4) Remember the first character
- 5) Calculate soundex equivalent code of word1 and word2.
- 6) Compare the two soundex code
- 7) If match found then generate result.
- 8) Else Apply string matching technique
 - I. Count the total number of character
 - II. If it is same then compare each character and count the number of similar character.
 - III. If frequency of counter is total_lenght-1 or total_lenght-2 then we can conclude that both words are similar. That means total false comparison should be one or two max.
- 9) Generate result

Table 1: Matching Status of Strings for Three different Language Using SoundEx Algorithm

Strings Taken As Input			Matching Status of Strings		
English	Hindi	Gujarati	English	Hindi	Gujarati
Sainik & Saineek	सानक & सैनीक	સૈનિક & સૈનીક	Yes	Yes	Yes
Sandhya & Sanadhya	सध्या & सन्ध्या	& સન્ધ્યા	Yes	Yes	Yes
Kundan & Kunadan	कुंदन & कुन्दन	& કુન્દન	Yes	Yes	Yes
Sishir & Shisir	शशार & सशार	&	Yes	Yes	Yes
Nupoor & Noopur	नुपूर & नूपुर	નુપૂર & નૂપુર	Yes	Yes	Yes

As described earlier Original Soundex Algorithm works well but for English Language. But for Indian Language especially for Hindi and Gujarati we found number of variations in writing the word. Also the modified Soundex algorithm matches the words like “ ” & “સજ્ય” written in Gujarati language and returns the result as true, which was earlier retrieved as false hit. By applying modified Soundex algorithm the word written in Hindi and Gujarati with different variations can be retrieved as true positive which increases the efficiency of Soundex algorithm. We have tried different names and applied the Soundex algorithm on them.

The result is retrieved in the form of if two strings are similar Sounding or not. Some of the names are given with matching status of the strings in table above.



```
Python 2.7.6 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter your name : કુંદન
Enter your name : કુંદન
chances to be same
1
>>> ===== RESTART =====
>>>
Enter your name : ચંદ
Enter your name : ચંદ
chances to be same
1
>>> ===== RESTART =====
>>>
Enter your name : સંદી
Enter your name : સંદી
chances to be same
1
>>>
```

Figure 4: Outputs Obtained by modified Algorithm

As shown in the above figure we can say that modified soundex algorithm gives result efficiently while dealing with words like “ ” & “સંદી”, “ ” & “કુંદન”, “ ” & “ચંદ” written in Gujarati language. The same is possible for Hindi language. The words like सध्या & सन्ध्या, कुंदन & कुन्दन, "चद्र" & "चन्द्र" written in Hindi language can be retrieved as similar sounding names. As described earlier the same names were retrieved as false hit. But by applying modified Soundex algorithm the same has been retrieved as True Positive both in Hindi and Gujarati language. The output has been shown in figure below.

```

Python 2.7.6 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter your name : संख्या
Enter your name : सन्ख्या
chances to be same
1
>>> ===== RESTART =====
>>>
Enter your name : कंटल
Enter your name : कन्दल
chances to be same
1
>>> ===== RESTART =====
>>>
Enter your name : चंद
Enter your name : चन्द
chances to be same
1
>>> |
    
```

Figure 5: Outputs Obtained by modified Algorithm

5. IMPLEMENTATION

The implementation of algorithm was carried out in Python programming Language. The code of the algorithm is organized to optimize the matching process by avoiding unnecessary execution of loops, recursion and redundant comparison of strings. In addition, the code conforms to a comprehensive coding standard enforcing best practices and avoiding pitfalls. Functionality is provided to easily apply the algorithm on a Hindi, Gujarati and English words written with different possible variations of same name, meaning that the name will be same but the way of writing them would be different. The outputs provide the comparison of pairs of words and names in the form if they are similar sounding names or not, as identified by the algorithm.

For measuring the accuracy of the algorithm we used a set of 100 words written in Hindi, Gujarati and English Language. All entries were taken as distinct names along with their different possible variations. Then the experiment was carried out on the words and retrieved the result in the form if two words are similar sounding or not. Besides of these 100 names algorithm takes any word as input written in Hindi, Gujarati or English language As shown in fig.4 and fig.5 we have tried some names with half consonants written in Gujarati and Hindi language. The modified soundex algorithm for Indian language especially Hindi and Gujarati Language gives the 100% accuracy for the word written with the half consonants as depicted in fig.3 and figure 4. The original soundex algorithm works well but only for English language.

6. CONCLUSION AND FUTURE WORK

The phonetic matching is very important and complex technique, but needed in every part of the life. This paper proposes the modified Soundex algorithm for identifying the pairs of strings that sounds similar written in Hindi, Gujarati and English Experiments on a number of pairs of words shows that the Modified Soundex algorithm produce better results in terms of accuracy than the

original Soundex algorithm. Directions for future research include refining the algorithm to make use of the pronunciations provided in an online or offline dictionary to perform phonetic matching.

ACKNOWLEDGEMENTS

I am very grateful to **Dr. Vilin P. Parekh**, Principal of Parul Institute of Engineering and technology for his continuous help and support.

I also extend my thanks to Head of Department **Prof. Gordhan B. Jethava** for providing facilities for caring out the dissertation work.

I heartily thank **Mr. Dheeraj Kumar Singh (Dissertation guide)** for giving me such a chance to undertake dissertation under the subject of Improvement of Soundex Algorithm for Indian Language Based on Phonetic Matching.

I am also very thankful to my husband **Mr. Anant B Patel** for all his support and help. At each and every step he encourages me for this research work.

REFERENCES

- [1] Hettiarachchi, G.P., Attygalle, D., "SPARCL: An improved approach for matching sinhalese words and names in record clustering and linkage", IEEE, Colombo, 2012.
- [2] 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
- [3] Beider,A, Stephen P. Morse, Phonetic Matching: A Better Soundex, March, 2010.
- [4] Name and Address Matching Strategy, White Paper, 2007 December.
- [5] Brett Kessler, Phonetic Comparison Algorithms, Transaction of Philological Society Volume 103:2 243- 260, 2005.
- [6] Hall, P. A. V., and Dowling, G. R. (1980), Approximate String Comparison, Computing Surveys, 12, 381-402.
- [7] P.A.V. Hall, G.R. Dowling, Approximate string matching. Computing Surveys, 12(4):381{402, 1980.
- [8] Peter Christian, SoundEx - can it be improved? March, 1998.
- [9] Justin Zobel, Philip Dart, Phonetic String Matching: Lessons from Information Retrieval.
- [10] Beider,A, Stephen P. Morse, PhoneticMatching:An Alternative to SoundEx With Fewer False Hits.