Global Journal of Enterprise Information System

DOI: 10.18311/gjeis/2016/7658

Need for Customized Soundex based Algorithm on Indian Names for Phonetic Matching

G. Christopher Jaisunder[1*], Israr Ahmed[2] and R. K. Mishra[1]

[1]National Informatics Centre, Department of Electronics and Information Technology ((DeitY), Ministry of Communication and Information Technology, Government of India, New Delhi - 110003, Delhi, India; Christopher@nic.in, r.k.mishra@nic.in

[2]Department of Computer Science, Jamia Millia Islamia, New Delhi - 110025, Delhi, India; israr_ahmad@rediffmail.com

Abstract

In any digitization program, the reproduction of the handwritten demographic data is a challenging job particularly for the records of previous decades. Nowadays, the requirement of the digitization of the individual's past records becomes very much essential. In the areas like financial inclusion, border security, driving license, passport issuance, weapon license, banking sectors, health care and social welfare benefits, the individual's earlier case history is a mandatory part of the decision making process. Documents are scanned and stored in a systematic method; each and every scanned document is tagged with a proper key. Documents are retrieved with the help of assigned key, for the purpose of data entry through the software program/ package. Here comes the difficulty that the data, particularly the critical personal data like name and father name etc., may not be legible for the reading purpose and the data entry operators type the characters as per their understanding. The chances of error is of high order in name variations in terms of duplicate characters, abbreviations, omissions, ignoring space between names and wrong spelling. Now the challenge is that, result of data retrieval over these key fields may not be proper because of the wrong data entry. We need to explore the opportunities and challenges for defining the effective strategies to execute this job without compromising the quality and quantity of the matches. In this scenario, we need to have an appropriate string matching algorithm with the phonetic matching. The algorithm is to be defined according to the nature, type and region of the data domain so that the search shall be phonetic based rather than simple string comparison. In this paper, I have tried to explain the need for customized soundex based algorithm on phonetic matching over the misspelt, incomplete, repetitive and partial prevalent data.

Keywords: Component, Demographic Data, False Positive, False Negative, Phonetic Matching, Soundex Based Algorithm

(Date of Acceptance: 05-April-2016**; Plagiarism Check Date:** 12-April-2016**; Peer Reviewed by Three editors blindly:** 26-April-2016**; Reviewer's Comment send to author:** 03-May-2016**; Comment Incorporated and Revert by Author:** 09-June-2016**; Send for CRC:** 19-June-2016**)**

1. Introduction

Lot many researchers have already worked on the Information retrieval depends upon the various requirement. One such requirement is the phonetic matching of the names with the names in the database by comparing the way of pronunciation of words. In Indian perspective, the names in the database shall belong to various states of republic of India. In the practical scenario, globally these individual names shall be heterogeneous in nature having wide range of varieties; but, locally homogeneous in nature in respect of the common names, spellings and pronouncing method. The earlier decade data collected from the handwritten files through the manual data entry process may not be of high level of accuracy. The document may not be legible for reading the personal data and the data entry operator could have entered the data as per the pronunciation best known to him. He could have guessed the possible spelling of the name during the data entry process. These issues need to be addressed as part of the retrieval system while utilizing this type of data in any decision making system. Phonetic algorithm matching is basically to compare the names for similar sounding names irrespective of the spelling of the name in the database. Hence for retrieval purpose, phonetic matching is highly required for reading the personal data from the database. Soundex based techniques play an important role in retrieval of names from the database. Phonetic matching is used to evaluate similarity of the names without looking into to the actual spelling of comparing the name by character to character. The most common issue with name matching is the name variations of categories largely of Phonetics, little bit of partial Name, abbreviation, Regional & uncommon names. In fact, the comparison is being carried out on the soundex coded string and not on the actual name.

*Author for correspondence

The paper is organized as follows. Section I gives the introduction of the subject matter of this paper. Section II gives the introduction to the phonetic matching. Section III gives the understanding of the concepts of soundex algorithm. Section IV gives the details of the proposed algorithm for the Indian names for the purpose of storage and retrieval. Section V shows the experiments and performance with proposed algorithm. And, the last section VI concludes the paper and followed by references.

2. Phonetic Matching

In every part of life, the phonetic matching plays a very important role. Phonetic matching can be defined as a process of identifying a set of strings those is most likely to be similar in sound to a given keyword. The strings can be spelled using different writing styles but they can be matched phonetically[1]. A phonetic algorithm is an algorithm for indexing of words by their pronunciation. They are necessarily complex algorithms with many rules and exceptions just because of the English spelling and pronunciation is complicated by historical changes in pronunciation and words borrowed from many languages. Phonetic matching is used to identify strings that may be of similar pronunciation, regardless of their actual spelling[2]. 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]. There is no single best technique available. Objective of selecting a suitable technique is to reduce the false positive and false negative cases[8].

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.

3. Soundex Algorithm

Searching names in large database have always been a problem. The solution to the problem was given by Robert Russell in1912 as he proposed the soundex algorithm[4]. 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[5,6]. One solution is to say that two names are approximate matches if they sound the same. Here, the question is, whether we could build the right algorithm with the sound principles that can be extended to reduce the error rate[7]. 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[2]. 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.

Outline of Soundex Algorithm[2]

- Retain the first letter of the string
- Change all occurrences of the following letters to zero: a, e, h, i, o, u, w, y
- 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
- Remove all pairs of digits which occur beside each other from the string that resulted after the previous step.
- Remove all the zeros from string that results from the previous step.
- Return the at most four characters, right-padding with zeroes if there are fewer than four.

Taking an example we will see how soundex algorithm works. Example-"SMITH" will code to "S5030" which will then reduce to "S530"by computing the steps of soundex algorithm.

4. Proposed Algorithm

The design of this proposed algorithm is to help the Indian names matching retrieval system that sounds similar irrespective of their spelling. In the proposed algorithm, the soundex based algorithm is customised to create the coded string in a specialized fashion. For the purpose of this study, only single component name has been taken for the analysis purpose. Each and every name shall go through the customised process (Figure 1) of cleaning,

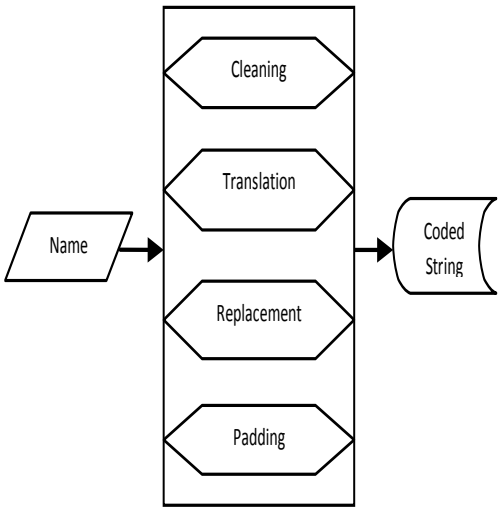

Figure 1. Customised Process.

translation, replacement and padding and the coded string is stored in the database accordingly. During the name searching operation, the search name is codified in the same fashion and the coded string is compared with the coded string stored in the database. Outcome of this process shall include false positive and false negative cases in the result set that needs to be narrowed down.

Cleaning:

- In this module, the character in the name or surname is being scanned one by one.
- Only the alphabetical characters are selected for forming the coded string.
- All the other characters including space, numerals and special characters are removed.

Translation:

- The result string of cleaning module is taken as the input for this module.
- The first character of the string is always retained as it is unless otherwise the letter 'E'.
- In case if the first character is 'E', then it is translated to 'I'.
- If the character is 'V', then it is translated to 'W'.
- If the character is 'J' then it is translated to 'Z'.
- If the character is 'Q' than it is translated to 'K'.
- The characters 'A','Y','I','U','E','O' are dropped.

Replacement:

- The result string of translation module is taken as the input for this module.
- Replace the characters 'PH' with the character 'F'.
- Replace the characters 'TH' with the character 'T'.
- Replace the characters 'DH' with the character 'D'.
- Replace the characters 'SH' with the character 'S'.
- Replace the characters 'CK' with the character 'K'.
- Replace the characters 'GH' with the character 'G'.
- Replace the characters 'KH' with the character 'K'.
- Replace the characters 'CH' with the character 'C'.

Padding:

- The result string of the replacement module is the input to this module.
- The string is truncated to a maximum of 4 characters if the string is more than four characters.
- If the string is less than four characters, then the character zero '0' is padded on the right.
- The result string is the proposed coded string.

5. Experiments with Proposed Algorithm

The implementation of this algorithm is proposed in simple steps and without much complication. The algorithm shall be implemented by using any programming language depends upon the convenience of the developer who wants to use it. Since the proposed algorithm is for the large database search, any database programming language shall be suitable for the implementation. I used oracle pl/ sql database programming language for the purpose of testing and analysis. One component Indian names are tested in the generic soundex algorithm as well as in the proposed algorithm to witness the performance of the proposed algorithm over the generic soundex algorithm. The result of the proposed algorithm is very much appreciable with minimal false positives/ negatives for the names, irrespective of the way it is being spelled. For the testing purpose, several names from various states have been considered to have the effect of the proposed algorithm in the process of name search. The effect shall be analyzed in quantitative as well as qualitative dimensions, i.e., the number of matches and relevancy of the matches.

For the experimental purpose, a sample of 448 cases picked up to work on 4480 names in the database. Each and every sample name has been tested against the names in the database. The soundex algorithm brought 2300 matches and the proposed algorithm brought 1289 matches that include all true positives, true negatives, false positives and false negatives. The overall performance of the proposed algorithm in respect of the matches is given in figure 2.

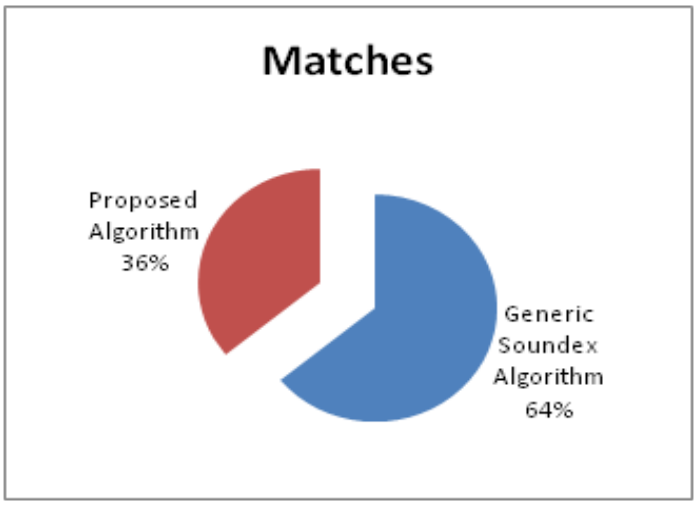

Figure 2. Performance Chart.

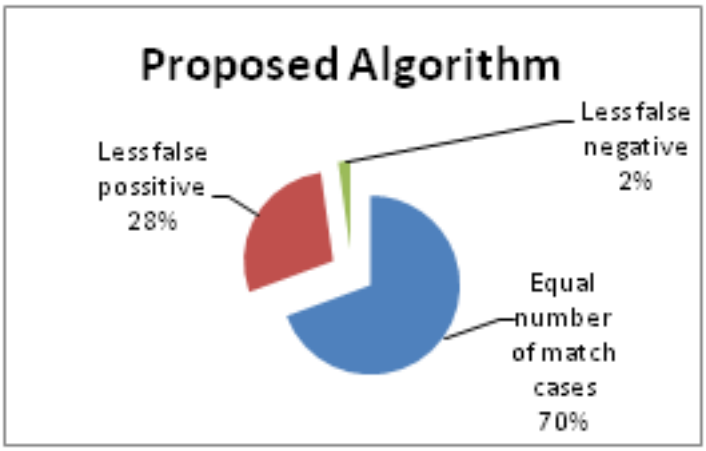

Figure 3. Performance Chart.

It is observed that, the soundex algorithm and the proposed algorithm performed equally in 69.42% cases. For example, the name 'PRAKASH' has exactly the same number of matches in both the algorithms. But, at the same time, we could see the avoidable false positive cases in 28.35% cases. For example, in the generic soundex algorithm, the name 'SANJAY' has the matches "SANGHA, SANJAY, SINGH, SUMESH, SINGHI, SANGE, SANGYU, SINGKO, SAMYIK and SING". The names like 'SINGH, SUMESH, SINGHI, SANGE, SANGYU, SINGKO, SAMYIK and SING' are not having any relevancy. The proposed algorithm matches with only name called ' SANJAY'. And, of course we could see 2.23% false negative cases that are missing in the soundex algorithm. One such example of false negative is that, the name 'CHAND'. The names like 'CHANDA, CHANDRAN, CHANDRA, CHANDILA, CHANDER, CHANDELA, CHANDAR and CHANDRASEKAR' are the names that we could have been interested in. The same has been illustrated in figure 3.

Also, it is observed that the quality and quantity of the matches were ensured in the proposed algorithm and some examples are illustrated in table 1.

6. Conclusion

There are many method of creating phonetic codes for Indian names. However, performance depends on naming conventions, which depends upon part of the globe, of the subject. This paper proposes the effect of proposed algorithm for creating the phonetic codes for the significant improvement in the results on the search of Indian names in terms of accuracy than the generic soundex algorithm. The effect of the proposed algorithm plays a major role in defining the soundex components depending upon the nature

Table 1. Match Table

S.No.	Name	Generic Soundex Algorithm	Proposed algorithm
1	SANJAY	SANGHA, SANJAY, SINGH, SUMESH, SINGHI, SANGE, SANGYU, SINGKO, SAMYIK, SING	SANJAY
2	AJAY	AHUJA, AS, AKSHAY, AJAY, A.K., AKHA, ASI, ASH, A.C, AJA, A.S	AJAY, AJA
3	GOEL	GOEL, GULI, GOAL, GOYAL, GIALO, GOLO,	GOEL, GULI, GOAL, GOYAL, GIALO, GOLO
4	SARA	SURYA, SHRI, SORA, SARA, SHER, SRO, SRI, SIROHI	SURYA, SORA, SARA, SRO, SRI
5	KUMAR	KUMAR34, KUMAR, KAMAR, KIMAR, KEMAR, KUMAR12	KUMAR, KAMAR, KIMAR, KEMAR
6	LAL	LOLY, LALI, LAL	LOLY, LALI, LAL
7	SINGH	SANGHA, SANJAY, SINGH, SUMESH, SINGHI, SANGE, SANGYU, SINGKO, SAMYIK, SING	SANGHA, SINGH, SINGHI, SANGE, SANGYU, SING
8	RAJ	RAGHU, RISHI, RS, REGHU, RAJU, RAJA, ROJO, RAJ, REGO	RAJU, RAJA, ROJO, RAJ
9	PRAKASH	PARKESH, PRAKESH, PRAKASH, PARKASH	PARKESH, PRAKESH, PRAKASH, PARKASH
10	MANJU	MANOJ, MANAKI, MANESH, MANJU, MINGKI, MAYANK, MAYING, MANISH, MANGKHYA, MINGE	MANOJ, MANJU
11	TEJ	TAGI, TEJI, TAYAGI, TACH, TECHI, TEK, TAKU, TAGIA, TOK, TOCHA, TYAGI, TAKA, TAK, TEJ, TAKO, TAJA, TASO, TAJO, TAGE	TEJI, TEJ, TAJA, TAJO
12	NARENDER	NARENDER, NARENDAR, NARENDRA	NARENDER, NARENDAR, NARENDRA
13	CHAND	CHANDA, CHANADO, CHAND	CHANDA, CHANDRAN, CHANADO, CHANDRA, CHANDILA, CHANDER, CHANDELA, CHANDER34, CHNADILA, CHAND, CHANDAR, CHANDRASEKAR
14	RAMESH	RAHAMAJ, RANKA, RINKU, RAMNIWAS, RAMESH, RAMJI, RIANG, RANJI	RAMESH
15	KAPIL	KAPIL	KAPIL

& quality of data. Advantage of the proposed algorithm is that the similar sounding names shall be picked up from very large database of personal data irrespective of the data entry method being followed. Also, this proposed algorithm a clear direction in the preparation of soundex coded string on the Indian names by taking care of the regional aspect. Efforts have been made to achieve the objective of minimizing the false positives and false negatives while balancing the number of alternatives during the name search process. The proposed algorithm works far better than the generic soundex algorithm on Indian names. This work shall further be improved in respect of the customization to meet the requirement of the search over the digital libraries.

7. References

1. Chaware S, Rao S. Analysis of Phonetic Matching Approaches for Indic Languages. International Journal of Advanced Research in Computer and Communication Engineering. 2012 Apr; 1(2).
2. Zobel J, Dart P. Phonetic String Matching: Lessons from Information Retrieval. 1996.
3. Beider A, Morse SP. Phonetic Matching: A Better Soundex. 2010 Mar.
4. Beider A, Morse SP. Phonetic Matching: An Alternative to Soundex with Fewer False Hits. 2008.
5. Hall PAV, Dowling GR. Approximate String Comparison. Computing Surveys. 1980; 12:381–402.
6. Hall PAV, Dowling GR. Approximate String matching. Computing Surveys. 1980; 12(4):381–402.
7. Christian P. Soundex - can it be improved? 1998 Mar.
8. Mishra RK. Information Technology as Management Tool for Process Re-Engineering and Preventing Forgery of Indian Documents. Jamia Millia Islamia, Central University. 2010 Mar.

Citation:

G. Christopher Jaisunder, Israr Ahmed and R. K. Mishra
"Need for Customized Soundex based Algorithm on Indian Names for Phonetic Matching"
Global Journal of Enterprise Information System. Volume-8, Issue-2, April-June, 2016. (http://informaticsjournals.com/index.php/gjeis)

Conflict of Interest:

Author of a Paper had no conflict neither financially nor academically.