Effect of Component Based Search for Phonetic Matching on Indian Names

Article · July 2017

CITATIONS
0

READS
13

3 authors, including:

George Christopher Jaisunder
Mewar University
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Israr Ahmad
Jamia Millia Islamia
16 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Effect of Component Based Search for Phonetic Matching on Indian Names

G. Christopher Jaisunder[a], Israr Ahmad[b] , Rajesh Kumar Mishra[c]
[a] NIC, Ministry Of Electronics & Information Technology, GOI, New Delhi, India
[b] Department Of Computer Science, Jamia Millia Islamia, Delhi, India
[c] NIC, Ministry Of Electronics & Information Technology, GOI, New Delhi, India

Abstract

In any digitization program, search over the demographic data is a challenging job particularly over the incomplete and incorrect personal data. Nowadays, the requirement of digitization of the individual's past records become very much essential and hence the search over the record keys too. Obviously, the past records do not have the biometric information of the individuals as part of the personal data. Hence, during digitization, the personal data; particularly the individual's name needs to be recorded with accuracy so as to retrieve the correct data and to enable the individual to avail the services that includes area like financial inclusion, security identification, driving license, passport issuance, weapon license, banking sectors, health care and social welfare benefits. As part of the digitization, the documents are retrieved for data entry and the operators type the characters as per their understanding and the chances of error is of high order particularly in the name's spelling by means of duplicate characters, abbreviations, omissions, ignoring space between names and wrong spelling and order of the components. The need for the search over demographic data, particularly on names shall be categorized into two search process namely verification and identification. The name verification process shall confirm the person's identity with a crosscheck on the other parameters while the identification process shall report the possible availability of the same or similar name(s) being searched in the whole system. While the spelling mistakes shall be managed to some extent by using the soundex based algorithm, the challenge remains same for the data retrieval over individual's name components recorded in wrong order. 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 component selection over the search name with an appropriate customized phonetic matching. The selection of the number of components shall be defined according to the nature, type and region of the individual so that the search shall be meaningful rather than simple string comparison. In this paper, we have tried to explain the effect of component based search algorithm on phonetic matching over the misspelt, incomplete, repetitive and partial prevalent data.

KEYWORDS-Verification, identification, coded string, demographic data, name component, phonetic matching, soundex based algorithm, false positive, false negative

I. INTRODUCTION

Many researchers have already worked on the Information retrieval depends upon the various requirements. One such requirement is the "component based phonetic matching" of the names with the names in the database by comparing the way of

pronunciation of words. In Indian perspective as far as the legacy data is concerned, there is no concept of the name components representing first name, middle name and last name. Only the very recent web enabled self service portals enable the individuals to enrol the first name, last name and middle name optionally as part of the registration process. The legacy data; particularly the earlier decade data collected from the handwritten files through the manual data entry process do not have this identification. The data entry operator could have guessed the possible spelling and order of the name during data entry process. These issues need to be addressed as part of the retrieval system while searching the names within the decision making context.

It is basically to compare the names for similar sounding components irrespective of the spelling of the search name in the database. Hence for retrieval purpose, the permutation of all the components is highly required for reading the personal data from the database. Permutation of all the components based techniques play an important role in retrieval of names from the database. It is used to evaluate similarity of the names without looking into the actual order or completeness of the search name. The most common issue is the additional/ missing space in the name, partial name, abbreviation and uncommon names. In fact, the comparison is being carried out on the soundex coded selected components of the search name and not on the actual name.

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 component matching. Section III gives the understanding of the concepts of nPr search. 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 followed by references.

II. COMPONENT MATCHING

In every part of life, the name matching plays a very important role both in verification and identification process. In verification process, we may not have much trouble as the input name is being collected from the concerned individual. Also, we shall assume that the individual may not give his name wrongly as the verification is for his own benefit. At the same time, in the identification process, the input is not collected directly from the concerned individual but through different agencies. We may not know the source of the search name as the same might have been collected directly from the individual as part of verification process or through some other security agencies for want of identification. Whatever may be the case, the search name is divided into ‘n’ number of components on the basis of the default delimiter ‘space’. All such components are stored in the database as part of knowledge base.

For example, while searching a name for want of verification in the database, the matches found may be of our interest as each and every match names shall have the complete name with all the components. The other parameters are verified to have a cross check and ensure the higher level of true positive and lower level of false positive. This system shall work well in case of name verification process. At the same time, for name identification process, we need to go for the partial names also to

have a meaningful decision making process. Since, the name is to be searched in the whole database for the availability of full/ partial names irrespective of the order of the components.

In large databases with diverse sources of names, the name conventions may also need to be handled such as the use of patronymic, house name, village name and surname in any order. The search name key is to be made with the components of the search name that is being searched in the database irrespective of its order. During this process, we shall have the full set of true positives, false positives, true negatives and false negatives. True positives/ negatives are the cases with clear matches and false positives/ negatives are the cases that we are very much interested upon.

III. nPr SEARCH

Any given name shall be divided into many components depends upon its length. We have tried a concept of permutation based search to study the effect by using the permutation of all the components in the given search name. Permutation is the arrangements where the order is important and repetitions or recurrence is not allowed. In statistics, nPr means the number of different permutations of n distinct objects taken r at a time. The mathematical formula for nPr is n!/(n-r)!; where factorial (!) is the result of multiplying consecutive integers start from 1 to the given number.

During the name searching operation, the search key is defined from the search name. Let us assume that the search name consists of ‘n’ number of components and try to search the name in the database with ‘r’ number of components taken at a time. Taking an example of ‘ASHOK KUMAR SHARMA’ as the search name, the components are ‘ASHOK’, ‘KUMAR’ and ‘SHARMA’. i.e., if we take ‘r’ as 1, then the search is made with all single components and the number of such search keys will be 3P1 = 3!/(3-1)! = (3x2x1)/(2x1) = 6/2 = 3. i.e., ‘ASHOK’, ‘KUMAR’ and ‘SHARMA’. If we take ‘r’ as 2, then the search is made with all two components and the number of such search keys will be 3P2 = 3!/(2-1)! = (3x2x1)/(1) = 6/1 = 6. i.e., ‘ASHOK KUMAR’, ‘ASHOK SHARMA’, ‘KUMAR ASHOK’, ‘SHARMA ASHOK’, ‘KUMAR SHARMA’ and ‘SHARMA KUMAR’. If we take ‘r’ as 3, then the search is made with all three components and the number of such search keys will be 3P3 = 3!/(3-3)! = (3x2x1)/(1) = 6 = 6. i.e., ‘ASHOK KUMAR SHARMA’, ‘ASHOK SHARMA KUMAR’, ‘KUMAR ASHOK SHARMA’, ‘KUMAR SHARMA ASHOK’, ‘SHARMA ASHOK KUMAR’ and ‘SHARMA KUMAR ASHOK’. This completes the permutations key to be matched when we consider a three component name. Similarly, we shall have nPr key sets to search the name in the database where ‘n’ is the number of components taken ‘r’ components at any point in time. For example, if the search name is of ‘eight’ components taken ‘two’ components at any point in time, then we shall have ‘fifty six’ search keys.

IV. PROPOSED ALGORITHM

Researchers use different algorithms in the search engine to search a name in large databases. The design of this proposed algorithm is to help the Indian names matching retrieval system that sounds similar names irrespective of their spelling. The proposed algorithm is component based ‘nPr’ search with the customized soundex in

the component level for want of more number of approximate matches. Possible solution to search names in the database was given by Robert C Russell who had developed the soundex algorithm and patented in 1918 [1]. There is no single best technique available. Objective of selecting a suitable technique is to reduce the false positive and false negative cases [6]. Rather than looking for exact matching, searching for approximate matching will be significant [2]. One solution is to say that two names are approximate matches if they sound the same. The soundex algorithm shall be used in a decision support system to search the approximate match names. The soundex codes are the coded string in a specialized fashion. Here, the question is, whether we could build the right algorithm with the sound principles that can be extended to reduce the error rate [3].

Soundex is the best-known phonetic matching scheme. 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 [4]. The strings can be spelled using different writing styles but they can be matched phonetically [7]. 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.

Outline of customized soundex algorithm [5] is; 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’, it is translated to ‘K’. Characters ‘A’,’Y’,’I’,’U’,’E’ &’O’ are dropped. Now, replace ‘PH’ with ‘F’; ‘TH’ with ‘T’; ‘DH’ with ‘D’; ‘SH’ with ‘S’; ‘CK’ with ‘K’; ‘GH’ with ‘G’; ‘KH’ with ‘K’; ‘CH’ with ‘C’. 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.

Now, the search name is divided into ‘n’ number of components. The permutations of set of all the component form search keys by applying the ‘nPr’ algorithm with a customized soundex codes generated for each of the search keys. For the purpose of this study, we have considered maximum of five component name for the analysis purpose. The search key 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 the false positive and false negative cases in the result set. The effect of component based search matching is being experimented to analyze the false positives/ negatives in the search name process.

V. EXPERIMENTS WITH PROPOSED ALGORITHM

The implementation of this algorithm is proposed in very simple steps and without much complication. The algorithm shall be implemented by using any programming language depends upon the need and convenience of the developer who wants to experiment or use it. Since the proposed algorithm is for a large database search, any database programming language shall be suitable for the implementation.

We have used Oracle pl/sql database programming language to test and analyze the effect of the proposed algorithm. The given name shall have 'n' number of components and out of which the search key shall have 'r' number of components selected in the real time scenario. For the experimental purpose a sample of 572 cases picked up to work on 57203 Indian names from various states to have the effect of the proposed algorithm on the Indian names. The number of components in the sample varies from one component to five. The maximum value of 'n' is taken as 5 and the value of 'r' varies from 2 to 4.

For example, if the number of components in the search name is 5, then the search is made with all permutations of 4, 3 & 2 components. If the number of components in the search name is 4, then the search is made with all permutations of 4 & 3. If the number of components in the search name is 3, then the search is made with all permutations of 3 & 2. If the number of components in the search name is 2, then the search is made with all permutations of 2. One component search name is done with the one component directly. The number of sample name components and percentage that are having component based matches are represented in Figure 1.

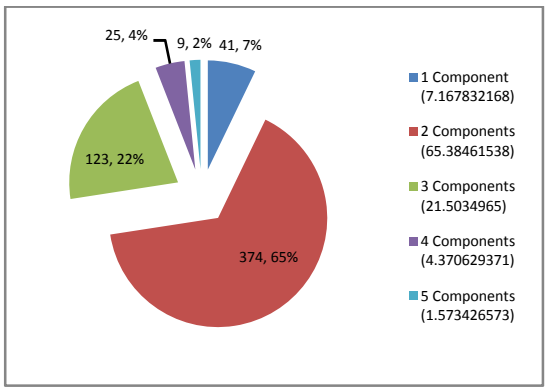

Figure 1 : Match Components

In general, it is observed that the simple name search and the proposed algorithm performed equally in 61.71% (353 out of 572) of the cases. The effect shall be analyzed in quantitative as well as qualitative dimensions i.e., the number of matches and the relevancy of the matches.

To analyze the quantity of the sample having matches, each and every sample name had been tested against the names in the database. The simple name search brought 353 matches and the component based search brought 464 matches that include all true positives, true negatives, false positives and false negatives. And, there is no match for 219 cases through simple name search and 108 cases through the component based search. That means, the component based search brings more alternatives of approximate names than the simple name search. The quantitative performance and search effect of the proposed algorithm in respect of the approximate name matches is given in Figure 2.

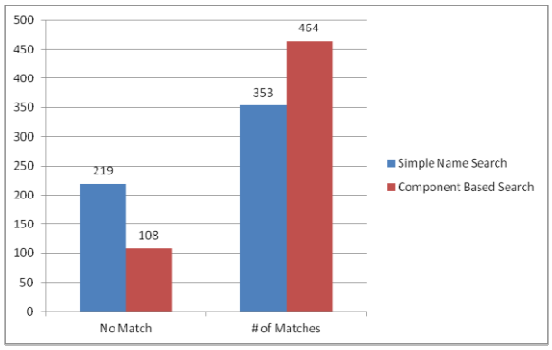

Figure 2 : Search Effect

To analyze the quality of the match, we have calculated the priority of the match score. i.e., if the complete name is matching with all the components in order, then the score starts with a value 500. If the component of the search name is matching with the first component of the match name, then a score of (160 - position) is added; score for second component match is (140 – position); third (120 – position); fourth (100 – position); fifth (80 – position); and so on where 'position' is the integer representing match component position in the search name. Some of the examples are illustrated in the Match Table I.

MATCH TABLE I

# of Components match over total # of components	# of Match Cases	Match Name Order with Example	Match Value
5 / 5	5	In the Given Order (RAM NAIK KAILASH PATI SINHA)	1085
4 / 4	14	In the Given Order (SUKHJINDER PAL SINGH BHATTI)	1010
3 / 3	119	In the Given Order (D D RATHOR)	914
2 / 2	17305	In the Given Order (GURMEET SINGH)	797
1 / 1	2147	In the Given Order (SURENDER)	659

3 / 4	3	First, Second & Third Components (SUKHJINDER PAL SINGH BHATTI)	414
3 / 4	3	First, Second & Fourth Components (LT COL SANDEEP SINGH)	393
3 / 4	4	First, Third & Fourth Components (SRI KRISHAN KUMAR PANDEY)	372
3 / 4	1	Second, Third & Fourth Components (GOURU VENKATA RAMI REDDY)	354
2 / 3	4303	First & Second Component (AJAY KUMAR GUPTA)	297
2 / 2	611	First & Third Component (JITENDER PAL SINGH)	296
2 / 2	11	First & Fourth Component (MAJOR RAHUL PRATAP SINGH)	295
2 / 3	603	First & Second Component (AMAR BAHADUR SINGH)	277
2 / 3	286	First & Third Component (MAHENDAR PAL SINGH)	276
2 / 3	11	First & Fourth Component (RANA KUNWAR PRATAP SINGH)	275
2 / 3	371	First & Second Component (ROSHAN SINGH NEGI)	257
2 / 3	67	First & Third Component (SURESH KUMAR SHARMA)	256

2 / 3	**13**	**First & Fourth Component (RAM RAY SINGH GURJAR)**	**255**

The effect of the proposed algorithm shall be appreciated with result having meaningful false positives/ negatives for the names, irrespective of the order of the components that is being followed and the customized soundex algorithm that is being adopted. Also, it is observed that the two component matches gives the maximum quantity as well as the quality of the matches when compared to the simple name search. At the same time, in case of one component match, more false positive cases are reported compared to the simple name search because of the soundex algorithm being used that identify the ' RAM' and ' RAHIM' as matches.

VI. CONCLUSION

There are many methods of name search being followed for different applications. 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 search key of phonetic codes for the significant improvement in the results on the search of Indian names in terms of accuracy than the conventional name search algorithm. The effect of the proposed algorithm plays a major role in defining the soundex key components depending upon the nature and quality of data. Advantage of the proposed component based search 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 is a clear direction in the preparation of soundex key components on the Indian names by taking care of the regional aspect. Efforts have been made to achieve the objective of analyzing the false positives and false negatives while balancing the number of alternatives during the name search process over the conventional search 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.

REFERENCES

[1] Russell R.C., Patent Numbers 1261167, U.S. Patent Office, Washington, D.C., April 1918.

[2] Hall P.A.V., and Dowling G.R., “Approximate String matching”, Computing Surveys, 12(4), pp.381-402, 1980.

[3] Christian P., “Soundex - Can it be improved?”, Computers in Genealogy, 6(5), pp.1-8, March 1998.

[4] Zobel J., and Dart P., “Phonetic String Matching: Lessons from Information Retrieval”, SIGIR '96, pp.166-172, August 1996.

[5] Jaisunder G. C., Ahmed I., and Mishra R. K., "Need for Customized Soundex based Algorithm on Indian Names for Phonetic Matching", Global Journal of Enterprise Information System, 8(2), pp. 30-35, 2016.

[6] Mishra R K., "Information Technology as Management Tool for Process Re-Engineering and Preventing Forgery of Indian Documents", Jamia Millia Islamia, Central University, March 2010.

[7] Chaware S., and Rao S., "Analysis of Phonetic Matching Approaches for Indic Languages", International Journal of Advanced Research in Computer and Communication Engineering, 1(2), April 2012.