ISSN:0975-9646

Rule Based Phonetic Search for Slavic Surnames

Janki B. Pardeshi[#1], B .R. Nandwalkar[*2]

[#1]*P.G Student, Dept. of Computer Science and Engineering,*
Late G.N. Sapkal College of Engineering, Anjaneri, Nashik, India
[*2]*Assistant Professor, Dept. of Computer Science and Engineering,*
Late G.N. Sapkal College of Engineering, Anjaneri, Nashik, India

***Abstract*—for indentifying a person, surname is natural identifier. There are different applications in Natural Language Processing among which we are considering searching of surname for databases of communications service providers, person registries, social networks or genealogy as a matching string of surname. This paper is about solution to searching algorithms for these databases. This paper defeats the issue of phonetic algorithm for Slovak and (regional) neighboring languages (Czech, Polish, Ukrainian, Russian, German, Hungarian, Jewish) surnames. This solution provides high precision and recall for searching surnames in these languages, with the help of phonetic search algorithm. The algorithm provides method for conducting search to find information related to imperfectly identified surname.**

Keywords— Phonetic Algorithms, Rules, Natural Language Processing

I. INTRODUCTION

To recognize a person surname assumes an essential part, Searching data identified with uncertainly distinguished surname might be of an extraordinary significance that it requires exceptional exploration for directing such a search. Case in point instance emergency calls where distinguishing a person is of crucial importance. The surname might be given defectively. Subsequently, the user is not certain of the surname or spelling of the surname. This may be because of errors presented by the operator while making the content for a search query input.

There are algorithms available to solve such problems but not all algorithms can solve all types of errors. Individual algorithms primarily different language group to which they are intended for or there might be typographical errors that they are able to identify. Input query errors based on phonetic pronunciation of a word in a specific language can be reduced by phonetic matching approach. In this approach the search matches strings that have similar pronunciation despite of actual spelling. The phonetic algorithm should be able to find misspelled name from the database. This paper contains phonetic matching of words such as surnames that are in text form. The search query could be in the form of a string i.e. a surname, which can be undertaken to find against words that contains surnames stored in data. The initial proposed idea for the research was designated in the few recent years. The above research does not include identification of words such as surnames in a text format. Every search algorithm emphasizes to remove all errors that are introduced by typing that causes same pronunciation of letters or words such as syllables that are in the surnames which gives efficient approach of identifying surnames with canonical indexed forms.

Many search algorithms are available for English language and those supports partially for other languages, but these are not intended for Slavic or other territory languages which are the target of our research.

The proposed idea not only aims to emphasis on phonetics that are of Slovak language but also covers certain rules for areas of languages that are based in East Europe.

II. PREVIOUS WORK

2.1 The traditional search approach

The approach was to find the exact match when the records present in search result that are to be matched the input string. In partial or pattern matching approaches there might be a query which contains uncertain text that can be must with characters or a query that can be modeled to exactly match the expected text, For example, Regular expression. Another approach is based on comparatively matching certain group of words regardless of their position. There are several approaches which can be used to eliminate that are introduced by typing. In such approaches minimum count of transformation is calculated to translate the first word to the second. Other algorithms based on grammatical approach that includes stemming, matching synonym that can be employed for Slovak language in [2] as well.

The last approach is finding phonetic similarity that could be represented by languages that depends on algorithms of string matching. Personal Name Recognizing Strategy (PNRS) [3].

There are several algorithms that are hybrid matching algorithms that are made of several other algorithms and research [4].

2.2 Elimination of Typing Errors

The Damerau-Levenshtein metric infers the similarity of strings as for mistakes which emerges when composing the word. The mistakes are gathered into four classifications:

1. Extra insertion of a letter
2. Cancellation of letter
3. One or more letter substitution
4. Neighboring letter transposition

2.3 Phonetic Approach:

Phonetic algorithms are intended to remove errors in the search query that emerge due to phonetic pronunciation of word in different languages. In most of the existing phonetic algorithms a word is transformed into canonical form. The canonical form is nothing but a string that is created by applying rules of transformation defined by a particular algorithm to the original word.

Transformation steps:
Input: Surname, Canonical form length
Output: Canonical form
1) Begin
2) Transform surname to Capitalized;
3) Transliterate surname;
4) Find a rule for the surname last letter;
5) Append first letter code to the canonical form
6) Append before vowel code to the canonical form
Otherwise append code to the canonical form;
7) Append padding;
8) End

III. PROPOSED WORK

In Proposed system the algorithm is designed for names originating in Slovakia and the regions in central and eastern European countries or for Slavic and languages that are morphologically rich. The algorithm is aim to solve problem of typing errors and inaccuracy.

Slavic languages have some differences compared to English. These languages use diacritical characters. All well-known phonetic algorithms, which we had the opportunity to test, cannot transform diacritical characters. To adapt them for our tests we used transliteration, which maps diacritical letters to a basic alphabet letter.
Following are the principles:

3.1 The Rule for Each Alphabet

Each essential letter set is characterized a rule where rule contains alternatives groups of letters that starts with a letter to which the rule is defined. Every alternative have its own phonetic codes. The rules are assigned phonetic codes. The rules for the code in the transformation are defined using the character position in the word followed by the letter. These rules contain codes for every first letter of the word and for every letter that that takes after a vowel and for every last letter in the word, also phonetic codes for various cases.

3.2 Length of the canonical form:

Length of the canonical form get vary depend on string which has been passed as an input. Every character in the canonical form is produced by is created by changing the input string from start of a word took after till the end regardless of the last character of the canonical form.

3.3The Last Syllable Transformation

In this algorithm the last letters are transformed. The algorithm implies a specific rule in view of the last letter of the data word. If there are ending alternatives for the last word the algorithm tries to locate the longest one. The relevant options are expected to coordinate the letters toward the end of every word and phonetic code in added toward the end of canonical form once the change is done. This transformation proceeds with the procedure right from the earliest starting point towards every unprocessed part of the input string.

3.4 Proposed algorithm

Input: Surname
Output: Canonical Form
Begin:
Transform Surname to upper case
Transliterate surname
Find a rule for surname last letter
If (letter is diacritical)
{
Append prefix to letter
}
If (ending alternative exists)
Use ending alternative code as the last of canonical form
Else use rule last letter code as the last of canonical form
While (unprocessed letters exists)
If (current letter is diacritical)
then append prefix
If (alternative exists)
then use alternative to fetch code
Else use rule to fetch code
If(processing first letter)
Then append first letter code to the canonical form
Else if(vowel is next)
Append before vowel code to the canonical form
Else append code to the canonical form
End while
End

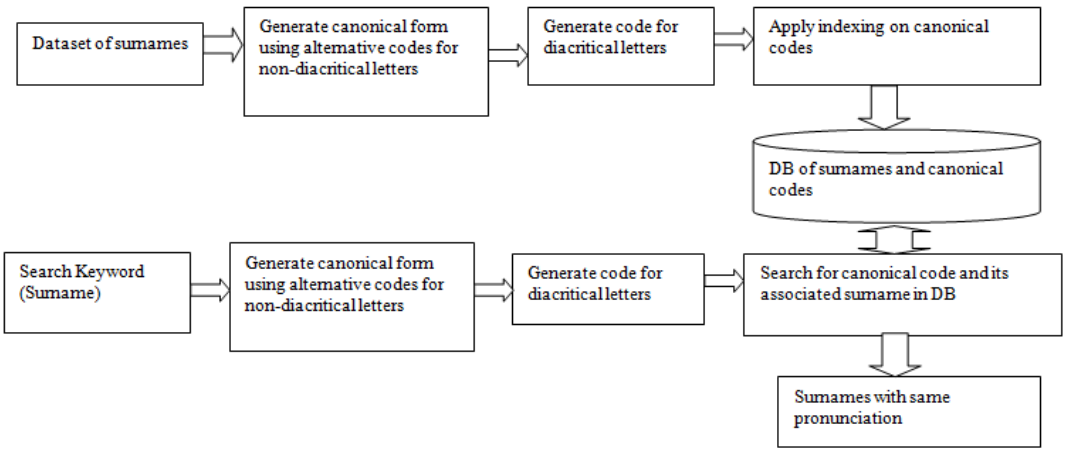

Fig 1: Proposed system Architecture

IV. RESULT AND ANALYSIS

By using phonetic algorithm system will generate different set of canonical form for the string to be search. A hit list contains "good" and "bad" documents. The quality of the search engine is evaluated by the proportion of good documents, their position and the amount of missing "good" documents. The effort made to increase a proportion of "good" hits, placing them at the top of the hit list and to reduce the amount of the missed "good" hits. Despite the principle of phonetic algorithms retrieval differs from the search engines, because they take in account the word phonetics; the ordering and scoring of the retrieved hits by the relevance is just as important when showing them to an end user.

Hits Distance Threshold: To assess "how good" the hit is we used the metric Damerau–Levenshtein distance. With a calculated distance we were able to perform the hit list ordering or use it for algorithms scaling (to filter the retrieved hits by the maximum distance). For our testing of the average precision and precision recall graphs we used the hit list ordered by the Damerau–Levenshtein distance.

Precision and Recall are the most common metrics. Precision is the level of usability and Recall is the level of completeness of the records in a hit list. Precision is defined by expression (1), recall by expression (2),

Precision= relevant hits in hit list/retrieved hit (1)

Recall = relevant hits in hit list/Relevant Documents in collection (2)

We used Precision-recall graphs to test Existing algorithm and Proposed algorithm. We have chosen to present the result of the most adequate candidates, considering good results in all the mentioned metrics. In the figures below we showed the result of the algorithms tested on the test set of surnames. The graph in shows the degradation of precision against percentage recalls. The graph shows a trade-off between precision and recall. Trying to increase recall typically introduces more bad hits into the hit list, thereby reducing precision. Trying to increase precision typically reduces recall by removing some good hits from the hit list.

In proposed system we increased precision against recall by handling diacritical letters. We have generated three test datasets of surnames containing 3, 22, 106 surnames for testing as Test1, Test2 and Test3 and got results as follow:

Testing Datasets	Relevant Hits	Retrieved Hits	Relevant Documents
Test1	5	18	5
Test2	44	89	58
Test3	106	1471	114

Table 1: Existing algorithm Results

Testing Datasets	Relevant Hits	Retrieved Hits	Relevant Documents
Test1	3	4	5
Test2	28	40	58
Test3	106	1051	114

Table 2: Proposed Algorithm Results

From these values we have computed precision recall graph by using equation (1) and (2).

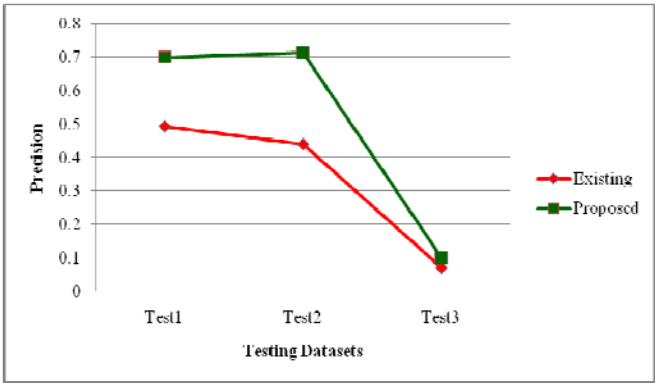

Fig 2: Precision graph for Existing algorithm and Proposed Algorithm.

Fig 3: Recall graph for Existing algorithm and Proposed Algorithm.

As shown in precision graph fig.2 values are obtained from different set of test queries which describes the higher precision than previous system. Values are computed based on 3 test set of queries.

Set of Test 1 consist of 3 queries against training set of 178 records, which gives precision value 0.2 for existing system and 0.6 for proposed system value.

Test 2 consist of 22 set of queries against same records as test 1 which omits values as 0.4 and 0.7 for existing and proposed system respectively.

Test3 consist of 106 set of queries against whole training dataset containing 483243 records and it results in values as 0.7 and 0.1 for existing and proposed system respectively

For recall graph 3 test sets are applied to compute values:
Test 1 taken for 3 queries set and it gives values as 1 for existing system and 0.6 for proposed system.
While conducting test 2 for set of 22 queries on training set of 178 records it gives recall for existing as 0.75 and for proposed it is 0.48.
Test 3 injected 106 set of queries for total dataset of 483243 records it gives recall value 0.92 for existing and 0.9 for proposed system.
The goal is to move the precision curve up and to the right which means that precision is better than existing system. And will give more accurate results.

V. CONCLUSIONS

During literature survey we found many existing phonetic algorithms. For each algorithm we could not find any rule which covers Slovak specific languages with diacritical letters. Therefore it is possible that the proposed algorithm can be the first algorithm that is designed for languages that are Slovak specific. The rule set of this algorithm is rigorously designed and based on common surnames occurring in middle Europe. The proposed algorithm implements an innovative principle of ending syllable transformation; it also covers diacritical letters transformation with better precision of the hit list.

ACKNOWLEDGMENT

I would like to thank my guide, Prof. B.R. Nandwalkar, for his guidance and support. I will forever remain grateful for the constant support and guidance extended by guide, in making this paper. Through our many discussions, he helped me to form and solidify ideas. The invaluable discussions I had with him, the penetrating questions he has put to me and the constant motivation, has all led to the development of this paper.

REFERENCES

[1] Dušan Zahoranský and Ivan Polasek, "Text Search of Surnames in Some Slavic and Other Morphologically Rich Languages Using Rule Based Phonetic Algorithms" IEEE/ACM Transactions On Audio, Speech, And Language Processing, Vol. 23, No. 3, March 2015
[2] T. Kuzár, "Clustering on Social Web," *Inf. Sci. Technol. Bull. ACM Slovakia*, vol. 5, no. 1, pp. 34–42, 2013.
[3] C. Varol and C. Bayrak, "Hybrid Matching Algorithm for Personal Names," *ACM J. Data Inf. Qual. 3*, vol. 4, p. 18, Sep. 2012, Article8.
[4] J. Soo and O. Frieder, "On Foreign Name Search," ECIR, LNCS 5993 pp. 483–494, 2010.
[5] D. Zahoranský and I. Polasek, "Rule Based Phonetic Search Approach for Central Europe, SISY 2010," in *Proc. IEEE Int. Symp. Intell. Syst. Inf.*, Subotica, Serbia, 2010, pp. 71–76.
[6] M. Hosseini *et al.*, "Selecting a Subset of Queries for Acquisition of Further Relevance Judgements," pp. 113–124, 2011, ICTIR, LNCS 6931.
[7] K. Mahesh, *Text Retrieval Quality: A Primer. OracleTechnologyNetwork*,2001[Online].Available:http://oracle.com /technology/products/ text/htdocs/imt_quality.htm, Oracle Corporation. (April 2009). Retrieved April 26, 2009.
[8] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos, "Summarization system evaluation revisited: N-gram graphs