Using Product Similarity for Adding Business

Boumedyen A.N. Shannaq[1], Prof. Victor V. Alexandrov[2]

GJCST Classification (FOR)
H.3.3, H.3.1

***Abstract*-Due to increasing attentionto maximize profits, international firms and corporations oversee the importance of typing and registering name of products. It occurs that the product name sometimes misspelled by customers during the product registration. The customer finds it difficult to search for a product on the internet either because the product is not registered or is not documented in the right order. This study highlights the problem and creates alternative ways to retrieve similar product name. To experiment this idea, a collection of English and Arabic product names have been built, along with 93 training queries and 123 test queries. These collected data are used to evaluate a variety of algorithms to measure effectiveness of using information retrieval operation. The new technique LIPNS shows a considerable improvement over existing.**

Keywords -Names Similarity, Arabic names, SoundX, N-gram, and Information Retrieval.

I. INTRODUCTION

Finding regularities in strings is useful in a wide area of applications which involve string manipulations. Such applications add richer data-profiling capabilities to its data quality offerings, to increasing its customer base in Europe [1]. The task of matching entity names has been explored by a number of communities, including statistics, databases, and artificial intelligence. Each community has formulated the problem differently, and different techniques have been proposed. Finding correspondences between elements of data schemas or data instances is required in many applications. This task is often referred to as matching. A comparison shopping website that aggregates product offers from multiple independent online stores. The comparison site developers need to match the product catalogs of each store against their combined catalog. Names and addresses are critical in identifying a person, a company, an organization etc. This information is the primary keys for accessing the information of an individual or a company in many of the database system that exist in the computer world. The variation of names, the variation in the way they are written or spelt creates major problem to name recognition across the globe. Product names have characteristics that make them different to general text. While there is only one correct spelling for many words, there are often several valid spelling variations for product names, for example ‗Tomato', and ‗Tamato'.Names are also heavily influenced by people's cultural backgrounds. These issues make matching of personal names more challenging compared to matching of general text [2].There are many applications of computer-based name-matching algorithms including record linkage and database searching where variations in spelling, caused for example by transcription errors, need to be allowed for. The success of such algorithms is measured by the degree to which they can overcome discrepancies in the spelling of names. Evidently, in some cases it is not easy to determine whether a name variation is a different spelling of the same name or a different name altogether. Most of these variations can be categorized as Spelling variations, Phonetic variations, Double names and Double first names [3].Now, International firms and corporations oversee the importance of different customers represent different levels of profit for the firm especially Gulf Arabic customer's, who don't know very well, how to type correct product name and they have to make their best guess at how to type the product names correctly. Because if they misspelled product name, exact match search will not find product in the DB, subsequently, the customer will not be willing to use this system again. At the same time, the number of customers will stop purchasing products or services from this company. It is an important indication of the growth or decline of a firm's customer base. Product name search in particular, however, to our knowledge has not been studied. In [4] they have discussed the characteristics of personal names and the potential sources of variations and errors in them, and presented an overview of both pattern matching and phonetically encoding based name matching techniques. There Experimental results on different real data sets have shown that there is no single best technique available. The characteristics of the name data to be matched, as well as computational requirements, have to be considered when selecting a name matching technique.However, we have built a collection of test English and Arabic product names and queries with corresponding relevance judgment; developed a new technique Language-Independent Product Name Search (LIPNS), discuss the results of a series of comparison experiments to see which matching Techniques achieve the best matching quality for different name types, and to compare their computational performance with LIPNS, all name matching techniques were implemented and compared with LIPNS. The obtained results show that new LIPNS technique provides an improvement over variant techniques.

About[1]- Information Systems DepartmentUniversity of Nizwa , Sultanate of Oman Email: boumedyen@unizwa.edu.om
Email: boumedians@yahoo.com
About[2]- Academy of Natural Sciences, Russia Doc.Nat.Sci., St.Petersburg Institute for SPIIRAS, St. Petersburg, Russia E-mail: alexandr@iias.spb.su

II. Statement of problem

Using product name to retrieve information makes these systems susceptible to problem arising from typographical errors. These exact match search approach will not find instances of misspelled product name or those product names that have more than one accepted spelling. The importance of such name-based search algorithms has resulted in improved name matching algorithms for English that make use of phonetic information. But these language-dependent techniques have not been extended to other languages such as Arabic, Chinese, and Indian etc. In the existing Soundex name search algorithm, the name search is limited to only English. But, the n-gram matching algorithm is not limited to any language and to our knowledge it is not applied for Arabic language name search. The limitation of this work in this area is partly due to the lack of standardized test data. Hence, we developed a collection of 17,265 Arabic and English product names and personal names, along with 93 training queries and 123 test queries. We use this collection to evaluate Soundex, n-gram, including new LIPNS method proposed to calculate the effectiveness of this standard information retrieval measures.

1) Selecting Algorithms

Numerous name search algorithms for Latin-based languages exist that effectively find relevant identification information, that use the phonetic features of names have been researched thoroughly for English, while string similarity techniques have garnered interest because of their language-independent methodology. Identify matching systems frequently employ name search algorithms to effectively locate relevant information about a given product name. In this study, we select the best suited algorithm for name search matching and the comparison is done with LIPNS technique . In [4] and [5] A comparison of various name matching algorithms through the R&W database is described in the Figure 2.1.

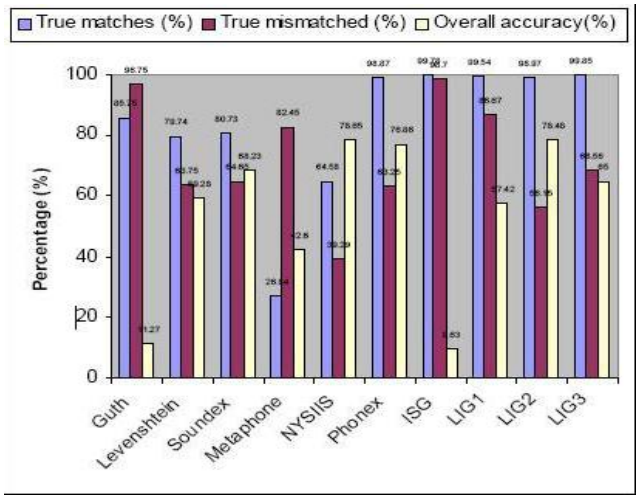


Figure 2.1 A comparison of various name matching algorithms [5].

As shown in Figure 2.1 above the algorithms accuracies are good fro Phonex, Soudex and LIG2.

Table 2.1 Average f-measure values (best results shown boldface and worst results underlined) [4].

	Given names
Soundex	.342
Phonex	.423
NYSIIS	.339
DMetaphone	.275
FuzSoundex	.327
Leven dist	.658
Dam-L dist	.659
Bag dist	.597
SWater dist	.889
LCS-2	.915
Skip grams	.844
1-grams	.839
2-grams	.885
3-grams	.783
Pos 1-grams	.890
Pos 2-grams	.880
Pos 3-grams	.768
LCS-3	.909
Compr BZ2	.458
Compr ZLib	.532
Jaro	.853
SAPS dist	.656
SortWink	.803
PermWink	.888
Editex	.631
Winkler	.891
SAPS dist	.656

is measured. In order to choose best suited algorithm for this purpose, we must define how the performance

III. Performance measure

Many new time warping techniques have been developed to improve its computation efficiency. Examples include Index-Based [6], Filter-Based [7], Wavelet [8], Dynamic Time Warping [9], Accounting Causal Relationships [10] and Dynamic Indexing Technique [11] approaches. It is necessary to evaluate algorithms according to the quality of the results of that search using information retrieval measures. In general, Recall and Precision are often used as retrieval effectiveness criteria. According to [12], high recall means retrieving as many relevant items as possible, while high precision means retrieving as few irrelevant items as possible. More specifically, recall is the proportion of relevant matches actually retrieved, and precision is the proportion of retrieved matches which are relevant. A match is relevant if it is judged based on the user interest.

Moreover, having 100 percent precision and 100 percent recall is essential, but it is a challenge. In order to achieve better performance it is necessary to get as maximum precision and recall as possible.

IV. Rule-Based Algorithm

Rule-based algorithms attempt to represent knowledge of common spelling error patterns in the form of various rules for how to transform a misspelled word into a valid one. According to [13], correction candidates are generated by applying all possible rules on the misspelled word and retaining the valid dictionary entries which produces this result and it can be ranked. Frequently, a numerical score is assigned to each candidate, based on the probability of having particular error corrected by the corresponding rule, which means a closer match.

V. Soundex

Soundex, presented in [14], was invented by odell and Russell in 1918 and used by the U.S. Census to match American English names. Soundex translates a name into a four-character code based on the sound of each letter. The first letter of the name is kept constant, while the rest of the letters are coded into digits.

VI. Phonex

Phonex [15] is a variation of Soundex that tries to improve the encoding quality by pre-processing names according to their English pronunciation before the encoding. All trailing ‗s' are removed and various rules are applied to the leading part of a name (for example ‗kn' is replaced with ‗n', and ‗wr' with ‗r'). As in the Soundex algorithm, the leading letter of the transformed name string is kept and the remainder is encoded with numbers (again removing zeros and duplicate numbers). The final Phonex code consists of one letter followed by three numbers.

VII. N-grams

Use an inverted index of n-grams to avoid going through the entire list during search. First, all names in the list are given a unique number,. then for every possible n-gram (with an alphabet of L letters, there are Ln possible n-grams), a list of all the numbers of names containing that n-gram is constructed. In order to find close matches to a specific name, the union of all lists with names having an n-gram in common with that name is taken. Answers can be stored in a heap, sorted after n-gram distance, and the answers with too large distance can be skipped to save sorting time [16].

VIII. Longest common sub-string (LCS)

This algorithm [24] repeatedly finds and removes the longest common sub-string in the two strings compared, up to minimum lengths (normally set to 2 or 3). This algorithmic suitable for compound names that have words (like given- and surname) swapped

IX. Prior studies

Since 1918 several Researches proposes different way to develop English Soundex algorithm, such as Phonix, N-gram, Edit-distance algorithms. At present there are a number of name-matching algorithms employing different degrees of complexity to overcome name variations. Algorithms that use the phonetic features of names have been developed for English, Soundex, presented in [17], was invented by Odell and Russell in 1918 and used by the U.S. Census to match American English names. The Soundex algorithm is designed primarily for English names and is a phonetically based name matching method. Soundex method is more accurate than just relying on character similarities between the names, the Soundex algorithm is not ideal, when comparing first names, different codes could be given for abbreviated forms of a name for example ‗Tom' , ‗Thos', and ‗Thomas' would be classed as different names. Algorithms such as Soundex, Phonix, and Metaphone are all designed for English names. Non-English Phonetic Algorithms are dealing with other languages. Soundex method for French language developed by [18] based on the Russell Soundex method but is adapted for the French language and classifies each name as a three-letter code. Like the Russell Soundx Coding Technique, the names Mireille, Marielle and Merilda which are all given the code MRL. Recent work on improving Soundex focuses primarily on improving performance by manipulating names prior to encoding, or altering Soundex codes after encoding. Examples include Celko's [19], Code-Shifting [20], Hodge's Phonetex [21] and Editex [22]. Holmes showed that for English, n-gram techniques are less effective than Soundex-based techniques. The explanation provided is that since n-grams are unaware of the phonetic information Soundex uses, they are not able to recognize the phonetic equivalence of various characters. Even so, Hodge notes that n-gram techniques are better equipped to handle insertion and deletion errors. An alternative approach is the use of string distance measures and n-grams. N-gram techniques are language-independent, differing significantly from Soundex in that they do not rely on phonetic similarity. Similarity, as identified by n-grams, is based purely on spelling rather than phonetic information. The two most commonly used values of n are bigrams and trigrams. Edit distances are used to determine the similarity of words after phonetic encoding has been completed. It is significant to note that, unlike other ranking measures, edit distances are not calculated in linear time, given two names, of length p and q, their edit distance would be computed in (pq Θ) and must be computed at run-time, while other techniques such as Soundex and n-grams allow retrieval systems to store encoded names and simply use them at run-time. Personal name matching is very challenging, and more research into the characteristics of both name data and matching techniques has to be conducted in order to better understand why certain techniques perform better than others, and which techniques are most suitable for what type of data. More detailed analysis into the types and distributions of

errors is needed to better understand how certain types of errors influence the performance of matching techniques.

X. Methodology

We developed new name matching algorithm, LIPNS, and evaluated it against n-grams and SOUNDEX techniques.
The algorithms experimented are as briefly outlined below:

- Prior Work
  - Soundex
  - Phonex
  - N-gram
  - Longest common sub-string(LCS)

- Our Algorithm
  1) LIPNS

1) Soundex

Soundex is perhaps the best known and most cited of the similarity key algorithms. Soundex translates a name into a four-character code based on the sound of each letter. The first letter of the name is kept constant, while the rest of the letters are coded into digits according to Table X.1.

Table X.1 soundex phonetic codes

Letters	Code
a ,e, h, i, o, u, w, y	0
b, f, p, v	1
c, g ,j, k ,q, s, x, z	2
d, t	3
L	4
m, n	5
R	6

Letters with the same Soundex digit as their preceding letter are ignored. After coding the entire name, all zeros are eliminated. Finally, the code is truncated or padded with zeros to one initial letter and three digits. As an example Appel → A1104 →A104→A14→A140, Tufaha →T01000→T010→T1→T100.
The encoding algorithm is very fast in practice after the calculation of the code it can be used to quickly lookup possible matches in the name list indexed by Soundex codes. The Soundex algorithm is rather crude and can sometimes go very wrong. Two names with different initial letters will never have the same Soundex code, even though they have the same pronunciation (e.g. Kamel→K540 and camel →C540). The algorithm is designed for English, but even with common English names, it fails easily.

2) Phonex

The Phonex Algorithm was first published by Lawrence which is also a phonetic based name matching algorithm. Metaphone algorithm converts a word to any of the combination of the 16 consonant letters. The Conversion rule of Phonex algorithm is like Soundex ignores vowels after the first letter and duplicate letters are not added to the code. It‟s more accurate compared to soundex in certain cases (ex: Bonner and Baymore gives the metaphone codes of BNR and BMR respectively while the Soundex gives the same code which is B560 [25].

3) N-gram

There are several different n-gram similarity measures. A simple measure given by [23] is the count of the total number of n-grams two words have in common, gram-count = |N1 \ N2|, where N1 and N2 are the sets of n-grams of the two words. Another measure used by[19], is n-gram distance function, gram-dist = |N1| + |N2| − 2 |N1 \ N2|,|N1| and |N2| denote the number of n-grams in the two words and can be calculated from the length of the words. “Salad” has 4,bigrams, and “Salata” has 5 bigrams. They share three bigrams. The n-gram distance between them is thus 4 + 5 − 2 * 3 = 3, similar example to Arabic product name
“شرطة” has 3 bigrams and “شرطا” has 5 bigrams The n-gram distance between them is thus 3 + 5 − 2* 0 = 8 .The similarity measures presented above do not take into account the ordering of letters within words. The two most commonly used values of n are 2 and 3 (bigrams and trigrams).

4) Longest common sub-string(LCS)

This algorithm is based on a subroutine computing implicitly the longest common subsequence (LCS) between the text and every substrings of the pattern. This subroutine can be used to compute the length of the LCS between a compressed text and an uncompressed pattern in time O(mn1..5); the same problem with a compressed pattern is known to be NP-hard[26]. For example, the two name strings „gail west‟ and „vest abigail‟ have a longest common sub-string „gail‟. After it is removed, the two new strings are „west‟ and „vest abi‟. In the second iteration the sub-string „est‟ is removed, leaving „w‟ and „v abi‟. The total length of the common sub-strings is now 7. If the minimum common length would be set to 1, then the common white space character would be counted towards the total common sub-strings length as well. A similarity measure can be calculated by dividing the total length of the common sub-strings by the minimum, maximum or average lengths of the two original strings similar to Smith-Waterman above). As shown with the example, this algorithms suitable for compound names that have words (like given- and surname) swapped. The time complexity of the algorithm, which is based on a dynamic programming approach [11], is O(|s1|×|s2|) using O(min(|s1|, |s2|)) space [4].

5) language-independent product name search (LIPNS)

The LIPNS technique was developed to satisfy the following requirements:

- Product name or any name with small differences should be recognized as being similar.
- If one product name is just a random anagram of the characters contained in the other, then it should (usually) be recognized as dissimilar.

- Language independence- the LIPNS technique should work not only in English, but also in many different languages.

The similarity between two product names is calculated in four steps:

1-Separate product names into letters .

2- Create a matrix by assigning first product name as a row of letters, and second product name as a column of letters.

3- Computing the similarity between letters by assigning one for similar letters and zero for dissimilar letters.

4- Computing the similarity between two product names by using the following formula: Ss(R,C) = 1 - (SumD / L)

- Ss is similarity score
- R is the letters set for the first product name (Row)
- C is the letters set for the second product name (Column)
- SumD is the summation of the ones lies on diagonal matrix.
- L is the length of product name in the DB

Assume that all relation scores are in the {0, 1} range, which means that if the score gets a minimum value (equal to 0) then the two product names are absolutely similar. To obtain effective results, the user has to just increase/decrease the Score value Estimator, which was estimated at (0.25), this score value was obtained through repeated trials and strenuous efforts based on the user"s terminal benefit and satisfaction as main consideration. For example, the LIPNS for "Salata" and "Salad" are shown below according to

Table X.2.

Table X.2 Similarity Matrix for „Tomato" and „Tamato"

	T	o	m	a	t	o
T	1	0	0	0	1	0
a	0	0	0	1	0	0
m	0	0	1	0	0	0
a	0	0	0	1	0	0
t	1	0	0	0	1	
o	0	1	0	0	0	1

Ss = 1 – (5 / 6)

Ss= 0.16

Since the Ss result is less than 0.25 , „Tomato" and „Tamato" are considered to be similar. In order to improve the performance we modified LIPNS steps as follow:

M.1- Compare two names before step 1, if two names are similar then stop and exit (matching).

M.2- If M.1 not matching then go to step one and two (LIPNS).

M.3- Delete not matching letter from both names and stop matching

M.4 -Go back to M.1 and continues.

If number of letter elimination is more than two, both names are not similar and stop matching. For example, the MLPINS similarity for "Tomato "and "Tamato "are performed as follow.

If "Tomato " = "Tamato then They are Similar , stop MLIPNS.Else Build matrix

	T	o	m	a	t	o
T	1					
a		D(stop)				
m						
a						
t						
o						

If "Tmato " = "Tmato then Similar and stop MLIPNS.Else Continue matrix building and starting from next letter where you stop before (in this case m)

Etc...

XI. Result

The results show that R-precision is superior to average precision for the Arabic product name search task since the number of relevant result is too small for interpolation. However, weak ordering in the result sets demands that R-precision be calculated over a number of random permutations of the results. The LIPNS, Soundex and N-gram were tested over queries and the effectiveness of the result is shown in Table XI .1.

Table XI .1 Comparison of various name matching algorithms with LIPNS

Technique	Average precision	R-Precision
Soundex	0.39089	0.2451
Bigrams	0.3339	0.1823
Trigram	0.13772	0.01151
LIPNS(Ss=0.25)	0.7579	0.5970
LIPNS(Ss=0.3)	0.3925	0.2591

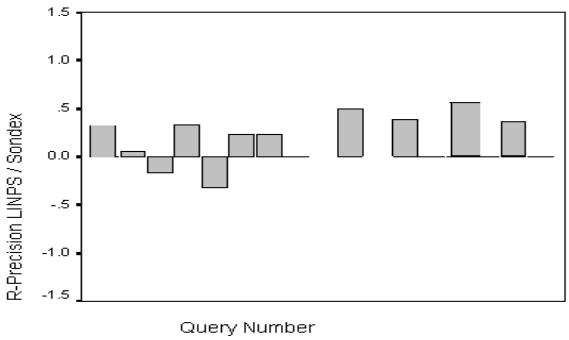
Table XI.2 Comparison of various name matching algorithms with MLIPNS

Technique	Average precision	R-Precision
Phonex	0.69	0.49
LCS	0.73	0.52
MLIPNS	0.81	0.95

To confirm the significance of these results ,we collected the results from all five of the scenarios described above and tested their composite significance, by using the R-precision measures for several queries to compare the retrieval history of two algorithms as follows. RPLipns/Soundex(i)=RPLIPNS (i)– RPSoundex(i). A value of RPLipns/Soundex(i) equal to 0 indicates that both algorithms have equivalent performance (in term of R-precision) for the i-th query. A positive value of

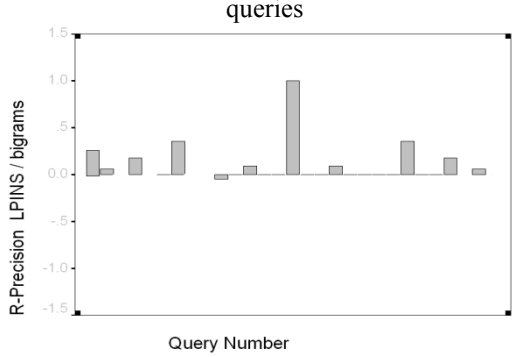
RPLipns/Soundex(i) indicates a better performance by algorithem LINPS (for the i-th query) while a negative value indicates a better retrieval performance by algorithm Soundex. For instance, the difference RPLipns/Soundex(i)= 0.5970 - 0.2451 is 0.3519, this indicates a better performance by algorithem LIPNS. Figure XI.1 illustrates the RPLipns/Soundex(i) values (labeled R-Precision LIPNS / Soundex) for two hypothetical retrieval algorithms over eleven sample queries. The algorithm LIPNS is superior for nine queries while the algorithm Soundex performs better for the two other queries.

Figure XI.1 Precision histogram for eleven hypothetical queries



The difference RPLipns/bigrams(i)= 0.5970 - 0.1823 is 0,4147, this indicates a better performance by algorithm LIPNS. Figure XI.2 illustrates the RPLipns/bigrams (i) values (labeled R-Precision LIPNS / bigrams) for two hypothetical retrieval algorithms over eleven sample queries. The algorithm LIPNS is superior for ten queries while the algorithm bigrams performs better for the one other query.

Figure XI.2 Precision histogram for eleven hypothetical queries



In other hand, trigram doesn‘t perform better for any queries, compared to LIPNS algorithm. From these result, we may draw the fact that LIPNS provides a statistical significant improvement over both Soundx and bigrams in the general situation, also obvious is the fact that our LIPNS and MPLINS techniques provide an improvement in performance that is significant even at the 99% level. The result obtained through this new method is purely independent of languages and different type of characters in Arabic, Latin, Indian, Russian etc. LIPNS and MLIPNS is an innovative method of its kind which is totally independent of all the world languages and a globally long awaited concept. The outcome of this new method clearly shows that this method should be superior to any other earlier methods available.

XII. Conclusions

The new method developed is purely independent of languages and different type of characters in Arabic, Latin, Indian, Russian etc. has been experimented. The SOUNDX test works only for English characters and names, whereas the proposed method has proved that it is not restricted to English language only and outperform over the other methods. It also compares LPINS method with n-gram method and proved that newly proposed method is superior to n-gram method as well. The specialty of LPINS method is that it works with all languages as well and it is an efficient method to implement for any related application. As per the formula derived here in this study, the researcher or user can control and modify the score value which will affect the R-precision and Average precision value. This is something new and does not exist in other methods according to our knowledge. The new formula found in this study can be controlled and easily modified to adjust the score values as it gives a user friendly environment. For example, to obtain effective results, the user has to increase/decrease the Score value Estimator, which was estimated here is at 0.25. This score value was obtained though repeated trials and strenuous efforts based on the user‘s terminal benefit and satisfaction. The score value has been selected to be main criteria to measure the similarity between the product names. Even though, this new method performs better, this can be applied with different applications such as name verification in customer data bases. Execution times are very important for a matching process to take place and show the results especially when the matching is attached to a sequence business process, in order to achieve these objectives we develop MLIPNS technique. Hence an appropriate decision on the algorithm to be used should be decided based on the algorithm accuracy, quality of data and. Execution time of the algorithm. The appropriate algorithm can be decided by optimizing these three factors based on the Business requirements.

XIII. References

1) Ted Friedman, Andreas Bitterer, ” Similarity Buys Evoke to Gain Technology Market Presence” , © 2005 Gartner, 20 July 2005 ID Number: G00129921.
2) F. Patman and P. Thompson. Names: A new frontier in text mining. In ISI-2003, Springer LNCS 2665, pages 27–38
3) A. J. Lait1 and B. Randell. ”An Assessment of Name Matching Algorithms” Department of Computing Science University of Newcastle upon Tyne,2005.
4) Peter Christen. A Comparison of Personal Name Matching: Techniques and Practical Issues”, Computer Sciences Laboratory Research School of Information Sciences and Engineering, September 2006.

5) Chakkrit Snae, Acomparison and Analysis of Name Matching Algorithms", International Journal of Applied Science, Engineering and Technology Volume 4 Number 1 2007 ISSN 1307-4318.
6) Kim S W, Park S, and Chu , Efficient processing of similarity search under time warping in sequence databases", Information Systems 29: 405–20, 2004.
7) Kwak T Y and Lee Y J A filtering method for searching similar multi-dimensional sequences under the time-warping distance", Information Systems 28: 791–813, 2003.
8) Chan F K P, Fu A W C, and Yu C ," Wavelets for efficient similarity search of time series With and without time warping", IEEE Transactions on Knowledge and Data Engineering 15: 686–705, 2003.
9) May Yuan, John McIntosh, Assessing Similarity of Geographic Processes and Events", © Blackwell Publishing Ltd., 2005.
10) Ella Mae Matsumura, Sandra C. Vera-Muñoz, ApplyingAccounting Causal Relationships: Experimental Evidence on the Decision Performance Effects of Problem Similarity and Comparison", Wisconsin Alumni Research, 2005.
11) G. Qian, Q. Zhu, Q. Xue and S. Pramanik , A Dynamic Indexing Technique for Multidimensional Non-ordered Discrete Data Spaces", ACM, 2006.
12) Ricardo Baeza-Yates & Berthier Ribeiro-Neto Modern Information Retrieval", Publishers, New York, Addison-Wesley, 1999.
13) P. A. V. Hall and G. R. Dowling. Approximate string matching", ACM Com-putting Surveys, 12(4):381–402, 1980.
14) Camps, R., & Daude, J. Improving the efficacy of approximate personal name matching", 8th International Conference on Applications of Natural Language to Information Systems, 2003.
15) Gadd, T.N. PHONIX: The Algorithm", Program – Electronic Library and Information Systems, 1990.
16) Pfeifer, U., & Poersch, T., & Fuhr, N. Searching Proper Names in Databases". Proceedings of the Conference on Hyper-text – Information Retrieval – Multimedia, Germany, pages 259-275, 1995.
17) Binstock, A., & Rex, Practical Algorithms for Programmer". Reading, MA: Addison-Wesley, 1995.
18) Gerard Bouchard and Christian Pouyez, Name Variations And Computerized Record Linkage", Historical Methods, Vol. 13, No. 2, pp119-125, Spring 1980.
19) Celko, J. Joe Celko's SQL For Smarties: Advanced SQL Programming", 2nd Ed., Burlington, MA: Morgan Kauffman, 1995.
20) Holmes, D., & McCabe, M. Improving Precision and Recall for Soundex Retrieval", Proceedings of the 2002 IEEE International Conference on Information Technology - Coding and Computing, pages 22-28, 2002.
21) Hodge, V., & Austin, J. Technical Report YCS 338", Department of Computer Science, University of York, 2001.
22) Zobel, J., & Dart, P. Phonetic String Matching: Lessons from Information Retrieval", Proceedings of the 19th International Conference on Research and Development in Information Retrieval, pages 166-172, 1996.
23) Pfeifer, U., & Poersch, T., & Fuhr, N." Retrieval Effectiveness of Name Search Methods" Information Processing and Management, 32(6), pages 667-679, 1996.
24) C. Friedman and R. Sideli ,"Tolerating spelling errors during patient validation", Computers and Biomedical Research,25:486–509, 1992.
25) Rupesh Shyamala ,"Name Address Matching Matching: A F Fuzzy perspective using various Name Matching Algorithms", 2006.
26) Alexander Tiskin, Faster subsequencet recognition in compressed strings", Department of Computer Science, The University of Warwick,Coventry CV4 7AL, United Kingdom,2006.