A Comparison and Analysis of Name Matching Algorithms

Article · January 2007

CITATIONS
66

READS
8,707

1 author:

Chakkrit Snae
Naresuan University
69 PUBLICATIONS **559** CITATIONS

SEE PROFILE

A Comparison and Analysis of Name Matching Algorithms

Chakkrit Snae

***Abstract*—**Names are important in many societies, even in technologically oriented ones which use e.g. ID systems to identify individual people. Names such as surnames are the most important as they are used in many processes, such as identifying of people and genealogical research. On the other hand variation of names can be a major problem for the identification and search for people, e.g. web search or security reasons. Name matching presumes *a-priori* that the recorded name written in one alphabet reflects the phonetic identity of two samples or some transcription error in copying a previously recorded name. We add to this the lode that the two names imply the same person.

This paper describes name variations and some basic description of various name matching algorithms developed to overcome name variation and to find reasonable variants of names which can be used to further increasing mismatches for record linkage and name search. The implementation contains algorithms for computing a range of *fuzzy* matching based on different types of algorithms, e.g. composite and hybrid methods and allowing us to test and measure algorithms for accuracy. NYSIIS, LIG2 and Phonex have been shown to perform well and provided sufficient flexibility to be included in the linkage/matching process for optimising name searching.

***Keywords*—**Data mining, name matching algorithm, nominal data, searching system.

I. Introduction

THE Internet now provides access to vast volumes of *nominal data* - data associated with names e.g. birth/death records, parish records, text articles, multimedia) as identified by Diaz [1]. Mining these data resources effectively involves *linkage* - how two names are related, e.g. surname and forename similarity, same spatio-temporal location, and legal association [2].

From the technical point of view we want to link and match as many names as possible with the correct individuals. If we deal with individuals of the same name, e.g. John Smith, we have to establish a second identifier at least. This can be – and is in many cases – a temporal element, like the date of birth, which is an individual and unchanging property of the person. Another way to circumvent the problem is to establish numbering systems, like ID numbers. Systems of numbers or other ciphers can be generated within individual organizations. It is not likely that the resulting ID numbers will be the same in different organizations. The numbering may have limitations as well, e.g. the individual health care setting (e.g. within a hospital or district) or, in principle, more widely (e.g. the National Health Service number). In the past, the National Health Service number in England and Wales had serious limitations as a matching variable, and it was not widely used on health-care records. With the allocation of the new ten-digit number throughout the NHS all this has been changed [3].

Although numbering systems are simple to implement they can lead to different errors in recording, transcription, and keying. So we have to take into account methods which reduce these errors and facilitate good quality of data entry and retrieval [4]. One such method uses a checking device such as check-digits [5]. When we are not able to use unique numbers or ciphers, natural matching variables are the person's name, date of birth, sex and perhaps other supplementary variables such as the address with postal code and place of birth, which are used in combination for matching. Recently, it has been suggested that this simple code could be extended for security critical places (e.g. airports, checkpoints etc.) with biometric marker information extracted from person identifier information e.g. fingerprints/iridograms [4].

Names have been persistently problematic for nominal data record linkage processing/searching, which undergo variations such as phonetic and alternate spellings. This problem can be made clear in this way: how do we know that differently spelled or pronounced names belong to the same person? Surnames tend to be much more variable in spelling than other lexical objects. Even the most frequent surnames can have many common alternatives. Other variations are often error-based and can also be easily identified. Variation in names is a source of concern, particularly in societies as culturally diverse as ours, where different naming conventions, different languages and writing systems and creative individual preferences come into contact with one another and is sometimes ascribed based on the conventions.

Name variation is one of the major problems in identifying people, because it is not easy to determine whether a name variation is a different spelling of the same name or a name for a different person. Most of these variations can be mainly categorized as character, spelling, and phonetic variations [4, 6, 7].

Manuscript received November 26, 2006.

C. Snae is with the Naresuan University, Phitsanulok, 65000 Thailand (phone: +66-81-4755113; fax: +66 55 261025; e-mail: chakkrit.snae@gmail.com).

A. Character Variation

The problem is created by capitalization, punctuation, spacing, qualifiers and abbreviations (Branting, 2001) can be shown as follows:

- Capitalization, e.g. brown and Brown; SMITH and Smith
- Punctuation, e.g. WILL SMITH and WILL-SMITH; SMIT and S.M.I.T
- Spacing, e.g. YOUNGSMITH and YOUNG SMITH
- Qualifiers, e.g. WILL SMITH and WILL SMITH YOUNG
- Abbreviations, e.g. ROB and ROBBIN; BOB and BOBBY

B. Spelling Variations

These variations rely on the assumption that the source and target names are strings which differ because of errors or transcription differences (e.g. different pronunciation). Spelling error patterns can be taken into consideration and single-error misspellings (mistyping) can be categorized as follows [8]: (1) insertion, e.g. MCMANUS as MACMANUS; (2) deletion or omission, e.g. ROBBIN as ROBIN; (3) substitution, SMYTH as SMITH; (4) transposition, e.g. BREADLEY and BRAEDLEY. Generally such variations do not affect the phonetic structure of the name but still cause problems in matching names.

C. Phonetic Variations

The phonemes suffer one or more modification, with the result that the structure of the name is substantially altered where the phonemes of the name are modified. For example, the nickname Pooh, as it is spelled in English, would be spelled in German as Puh. Where the phonemes of the name are modified, e.g. through mishearing, the structure of the name may be substantially altered. MAXIME and MAXIMIEN are related names but their phonetic structure is very different. Indeed, phonetic variations in first names can be very large as illustrated by ADELINE and its shortened form LINE.

From searching on the Internet for some personal names, e.g. in Thailand, we have found many variants of them which refer to the same names. Table I shows the results of variants of the personal name called "Somchai" using Google search in many search processes.

TABLE I
VARIANTS OF "SOMCHAI" USING GOOGLE SEARCH

Personal name	Variation	Number of results
Somchai	Somchai	200,000
	Som Chai	149,000
	Somchay	1,640
	Somshai	231
	Somchia	77
	Somchair	48
	Somchaiy	35
	Somcai	17

As can be seen from Table I, personal name like Somchai, Som Chai, and Somchay are the most similar but names like Somchair, Somchaiy, and Somcai exhibit the least similarity. With the help of name matching methods we find reasonable alternatives of the original name, e.g. Somchai. Then all alternative names of Somchai can be used in one single search process which covers all variants at once.

In this paper we present variations of names and some basic characteristics and description of name matching algorithms to overcome name variations. Here we present four different types of name matching procedures which are based on probabilistic phonetic and sound variation recognition, composite and hybrid algorithms which can deal with multicultural names as well. Result of name matching comparison can be measured and analyzed. Finally, the conclusions of our study and further work which has to be performed are discussed.

II. CURRENT MATCHING ALGORITHMS

The objective of the nominal data record linkage process is to determine whether two or more records refer to the same (or similar) person, object or event. Several methods are used in practice to complete the linkage process. In all nominal data studies methods must exist to overcome the problems of name variation identified earlier. Several researchers [6, 9, 10, 11] have proposed composite and hybrid (based on alternative types of variation such a spelling or phonetics) methods to overcome name variation, and most hybrid methods are language specific with highly evolved software for parsing and linking the names, times, and spatial variables used in matching. This process is usually referred to "name matching" or "name linkage", can be identified using standard components such as last names (surnames) and can be processed in different types of name matching method (e.g. NYSIIS [11], Guth [12], Levenshtein [13], Soundex [14], Metaphone [15], , and Phonex [16]) and. An initial literature search suggested that hybrid approaches using a combination of hybrid and composite (involving discrete and probability measured) algorithms best overcome general name variations in conventional linkage studies [17,18]. For computer based studies it seemed likely that similar approaches would be appropriate.

Many techniques have been used to cope with the important problem of matching variant names. However, most of these techniques were developed for general word matching and as a result they are not optimized for personal names matching. Spelling as well as phonetic variations combined with cultural aspects are the more challenging problems for automated multicultural name matching systems.

It is noted that probabilistic matching is the recommended strategy for computerised record linkage [18]. It is preferred because probability levels can be set to reflect accommodate weights associated with identifier values and coding errors thus maximising the available information in the nominal data record linkage and including multiple dimensions.

The difficulty of the name matching task and the requirements for an effective algorithm to perform this task, both depend on the type and degree of name variations which occur. More recently published name matching techniques are either of the composite or hybrid form [18] and several novel hybrid algorithms (e.g. LIG2, and LIG3) have been developed for specific purposes. All the name matching algorithms encountered in the literature and presented in this paper are based on alphabetic and/or phonetic similarity and/or name transformations (e.g. forename abbreviations) but may use a variety of distance and other metrics for representing the match. From an initial search of the literature, we distinguished four types of algorithms and implemented them using the C programming language:

1) spelling/string analysis based algorithms (e.g. Guth and Levenshtein),
2) phonetic/sound based algorithms (e.g. Soundex, Metaphone, NYSIIS, and Phonex),
3) composite algorithms (spelling or sound, e.g. ISG),
4) hybrid algorithms (spelling and sound, e.g. LIG algorithms).

- *Guth algorithm*. This type name is based on the approach due to Guth [12]. The method is left to right sequence driven, and is essentially alphabetic but is independent of language and ethnic issues. It is straightforward to code, is portable, and gives reliable results. It is, however, weak when comparing short names.
- *Levenshtein algorithm*. These are strictly alphabetic techniques based on edit distance metrics first fully described by Levenshtein [13]. Edit distance is defined for strings of arbitrary length and counts differences between strings in terms of the number of character insertions and deletions needed to convert one into the other, the minimum edit distance is then the similarity.
- *Soundex algorithm*. The method implemented here is due to Odel and Russell [14]. Soundex is a commonly used technique and has been modified for languages other than English [11,19].
- *Metaphone algorithm*. This type name is taken from Binstock and Rex [15] although many variants exist. The method implemented assumes English phonetics but works equally well for forenames and surnames.
- *NYSIIS algorithm* is an alphabetic algorithm which is easy to implement and which yields canonical index code similar to Soundex. However, NYSIIS differs from Soundex in that it retains information about the position of vowels in the encoded word by converting all vowels to the letter A [19]. The NYSIIS method returns a purely alphabetic code. NYSIIS has been modified and used successfully for an extensive series of record linkage studies and also in the pre-processing step of a generalised, iterative, record linkage system [11].
- *Phonex algorithm* is a combination of the two methods, Soundex and Metaphone. The method was proved to give a good overall performance when applied to names in the English language [16].
- *ISG algorithm*. These are hybrid techniques combining alphabetic and phonetic approaches. The similarity comparison is based on the Guth method. The method implemented is due to Bouchard and Pouyez [6]. Bouchard [10] explains that the approach seeks to overcome phonetic variations between names.
- *LIG algorithms* (e.g. LIG1, LIG2, and LIG3) are hybrid algorithms which combine phonetic and spelling based approaches using similarity measure as probability which described by Snae [18]. The algorithms are a combination of three name matching methods: Levenshtein, Index of Similarity Group (called ISG), and Guth. The LIG algorithms have the best performance in term of producing most accurate true matches, overcoming name variations, and increasing the hit rate. They have proved to be more accurate than other methods in the literature which provide phonetic tuning to address multi-cultural names without depending on the language [18].

III. Method Comparison and Accuracy Measurement

The success of name matching algorithms is measured by the degree to which they can overcome discrepancies in the spelling of surnames. In many cases it is not easy to determine whether a name variation is a different spelling of the same name or a different name altogether.

To overcome this, and to provide a body of "difficult cases" that might be more typical in other languages we constructed a test dataset of some 11,369 key base-names *from the Dictionary of English Surnames* edited by Reaney and Wilson (abbreviated here as R&W) [20] and their various phonetic and spelling variations (can be used as a ground truth to provide list of related names called *R&W group* and to measure the accuracy of each name matching algorithm). We used this dataset as standard for a more critical comparison of accuracy and performance. This dataset for example contains the base-name **Tennyson** and variations**, Tenneson, Tennison, Tenison**.

To calibrate each method with the R&W list several inter-method comparing techniques were investigated that measure the performance (or efficiency), of the methods.

Accuracy measurement is the way to measure how efficient each name matching algorithm is. From R&W group and ground truth, to calibrate any matching method **M**, the b stacks of *k* R&W surnames which M produces for each of the *b* R&W basenames were set up. These stacks were each compared with the *v* R&W variants for each basename and the number of exact surname matches *a* was calculated. From these, the number of true matches was produced as:

$$\frac{\sum_{1}^{b}\left\{\frac{a}{v} * 100\right\}}{b}$$

And the number of true mismatches as:

$$\frac{\sum_{1}^{b}\left\{\frac{k-a}{k} * 100\right\}}{b}$$

Here every R&W basename was compared (using M) with every R&W surnames. Each of the resulting b stacks was then compared with the appropriate R&W basename/variant group and the number *a* of exactly matched surnames produced as before. From these, the number of true matches and true mismatches were then calculated using the same formulae.

The accuracy of M was then calculated as suggested by Lait and Randell [1998] as:

$$\frac{a}{k} * 100$$

Finally, the elapsed execution time for each calibration was captured. The results of these calibrations are presented in Fig. 1.

There is a tradeoff between good and poor results. For any method M if *a* is high or is equal to *v*, i.e. every surname in a stack matched all the R&W variants, M is classified as a very good method. Conversely, if *a* is low, M is a poor method. Thus, the most desirable method is one that optimises this tradeoff, i.e. making true matches or *a* as high as possible.

IV. Results and Analysis

The accuracy of each name-matching method was determined by calibrating the number of true matches, true mismatches and overall accuracy (Section III). These measures provide three complementary measures of performance (e.g. good or poor method), which represents how efficient each name matching algorithm is. The following examples of calibrations are presented using the methods described in Section III.

The column names *b*, *k*, *a*, and *v* are defined as follows: *b* is the size of the stacks of surnames, *k* is number of the surnames, which each name matching method produces for each of the basenames *(b)*, *v* is number of the R&W variants for each basename, and *a* is the number of exact surname matches. True matches, true mismatched and overall accuracy are the calibrations described in Section III.

TABLE II
Example of Calculation and Comparison of NYSIIS using R&W Dataset

Basename (*b*)	*k*	*a*	*k-a*	*v*	True matches (%)	True mismatched (%)	Overall accuracy (%)
Brown	3	2	1	5	(2/5)*100=40	(1/3)*100=33	2/3*100=67
Jones	8	4	4	10	(4/10*100=30	(4/8)*100=50	(4/8)*100=50
Smith	6	4	2	5	(4/5)*100=60	(2/6)*100 =33	(4/6)*100 =67
Williams	3	2	1	5	(2/5)*100=20	(1/3)*100=33	(2/3)*100=67
…					110/4 = 43%	207/4=36%	251/4=63%

"Time elapsed during execution" calculates the difference between the start and the finish of each name matching algorithm including read/write data and return output on screen, thus this gives us the execution time and presents the speed of each method. The overall accuracy and execution time of each name matching method is presented in Fig. 1 and Fig. 2 respectively. The algorithm execution times are highly dependent on the processor speed and configuration of the machine the program was executed on. For this investigation an AMD-PC compatible 686X, running at 120MHz, was used and the algorithm executed as a single task, since multi-tasking is likely to affect algorithm execution times when other processes use the CPU.

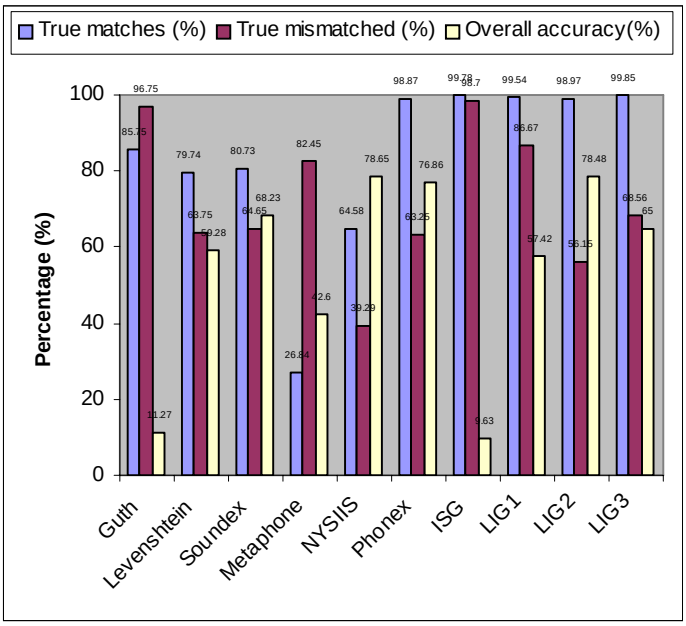

Fig. 1 Comparison of 10 types of matching algorithm using R&W dataset

The results from Fig. 1 suggest that the more name matches, the more true mismatches and the less accurate output will be due to some name matches are not in the list of R&W dataset.

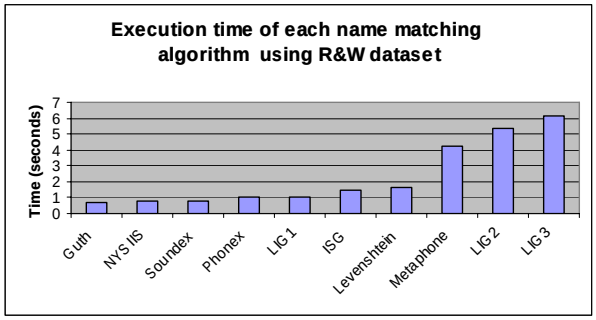

Fig. 2 Execution time of each name matching using R&W dataset

In Fig. 1 NYSIIS, LIG2, and Phonex had the highest overall accuracy measures of any algorithms on the *British Isles VRI Companion* and R&W datasets (e.g. 78%, 78%, and 76% respectively in R&W dataset). They also had the lowest true mismatches respectively (39%, 56, and 63%).

From name matching results shown here, it can be concluded that advantages and disadvantages are discovered in terms of true matches, true mismatches and overall accuracy with all the methods examined (and their myriad variants). The Phonex method seemed to be appropriate for use, however; clearly any method, which is used, should be able to overcome missed nominal data linkage conditions. The analysis performed here suggests that hybrid methods (e.g. LIG2 and LIG3) are best in these situations although for the obvious reason that a pronounciation bias is unlikely to be reflected in a spelling bias.

Phonex reduces substantially the number of false positive matches in successive pruning operations, which are the key to the program. However, LIG algorithms (LIG1, LIG2, LIG3), based on the composite hybrid index of similarity and Phonex, based on the combination of the Soundex and Metaphone, are essential for success due to its ability, to provide an initial high number of positive matches.

The spelling/string analysis based methods, Guth is (not surprisingly) less accurate overall than the phonetic/sound based algorithms Phonex and Soundex. A closer examination of the results shows that this is because the number of false positives and false negatives produced by the alphabetic methods are significantly higher than those produced by the phonetic based algorithms. The Phonex algorithm produces the largest number of true positive matches, but the false positive and false negative matches figure is not substantially larger than for the other methods resulting in this technique providing the best overall results. NYSIIS is not as good as Phonex even though they have lowest false negative but the true positive matches are not very high.

The ISG and Guth methods do not perform well. Here, matches are identified by letter position in the surname matching the target order or sequence of letters. Conceptually, this should be identical to matching for sound similarity. However scoring sound similarities prove to be problematical especially when these similarities have to be combined. Furthermore, phoneme similarity coding rules must be tuned (e.g. to allow *ph* and *f* to be more or less significant). This does imply that the method is not easily standardised in the general case.

The ISG algorithm, despite using phoneme based comparisons, avoids the need for much tuning and consequently gives better true matches results. The Guth method finds more matches when applied to short names (which in this application is necessarily a disadvantage). The interplay of these effects and different sample sets results in marginally different performances. However, in all the cases we have examined, the results are essentially similar for these two methods.

The additional complexity of the Metaphone algorithm does not result in significant performance gains over the simpler Soundex algorithm. However, the Metaphone method is more robust. When names with arbitrary additional consonants are seeded the other methods produced fluctuating performance statistics. In contrast Metaphone always produced the same rather dull performance. This robustness is attributed to the lower number of character pair comparisons that the Metaphone algorithm requires to determine a match.

The Soundex and Metaphone algorithms are highly susceptible to missing matches where a name pair starts with a different letter (e.g. KAVANAGH, CAVANAGH and HAVANAH). This is especially true with names beginning with vowels - where typically many equivalents exist (e.g. EWELL, ULE, YOUL, WHEWELL, HEWEL), which should result in matches being reported. In contrast, spelling/string analysis based algorithms are more robust with regard to initial letter differences.

The LIG algorithms have the best performance in term of producing most accurate true matches. They have proved to be more accurate than other methods in the literature. However, LIG1 and LIG2 are unable to deal satisfactorily with similarities where one element of a name pair is an abbreviated shorter form e.g. BRAM and BRAMBERLEY or WILD and WILDSMITH. Both methods come up with the low indexes of similarity scores which are usually below our cut-off level (LIG1 and LIG2 < 0.5). These low indexes of similarity scores will almost certainly result in missed nominal data linkages. To overcome abbreviated name problems an arbitrary truncation of the longer name is used to match the length of the shorter. This can be achieved in a number of ways and LIG3 reflects the most effective.

In terms of execution time (see Fig. 3) it was discovered that the Guth and NYSIIS methods were the most efficient. This is because the Guth algorithms do not need to perform name encoding (which the phonetic methods do), and because the basic algorithms of Guth and NYSIIS are simpler and shorter than either ISG or LIG3.

V. Conclusion

We have implemented (and parameterised) most of the more common name matching algorithms found in the literature. We note that hybrid name matching algorithms is a recommended strategy for computerised record linkage and

name search. It is the preferred method because probability calculation can be refined in various respects to accommodate weights associated with identifier values and coding errors thus maximising the available information in the nominal data and including multiple dimensions.

Results of method comparison and accuracy measurement suggest that it is difficult to single out one method, which has the highest match accuracy and is the best single method for nominal data linkage. Some methods may be inadequate for linkage purposes although they do provide a good starting point for further work. The hybrid and composite methods seem to achieve our objective of a matching procedure which produces all possible matches while Phonex reproduces human levels of accuracy.

We concluded that we could discover advantages and disadvantages in terms of accuracy with all the methods examined (and their myriad variants).Clearly any method which is used should be able to work under multi-ethnic conditions. Our work suggests that methods based on distance measures are best in these situations - for the obvious reason that a pronounciation bias is unlikely to be reflected in a spelling bias.

The choice of a name-matching method would seem to depend on the specific application and the intended use of the results. For example, to overcome phonetic variations, Phonex seems the most appropriated method. If an algorithm was required to find as many possible matches in a data sample with a given name then the Phonex, and LIG algorithms are likely to be best as they achieved more true matches and a high overall accuracy than the other methods tested. This is largely due to the low number of false positive matches, which the technique yields.

Our future intention is to harness hybrid name matching algorithms into a searching system for local organizations which uses onomastic, spatial as well as temporal ontological components, called LOWCOST (Local Organization Search With Consolidated Ontologies for name, Space, and Time) [21]. As names of organizations can be used in many different alternatives as well as in different writing systems LOWCOST comprises a name matching part which leads to a better matching of names in different variations. This will help implementing integrated search tools for the semantic web environment.

REFERENCES

[1] B. M. Diaz, "Nominal data visualisation: The Star-Trek Paradigm," *Computers in Genealogy*, vol. 5, no. 1, 1994, pp. 23-34.

[2] I. Winchester, "The linkage of Historical Records by Man and Computer: Techniques and problems," *Journal of Interdisciplinary History*, vol. 1**,** 1970, pp. 107-124.

[3] L. E. Gill, "OX-LINK: The Oxford Medical Record Linkage System, Complex linkage made easy, Record Linkage Techniques," in: *Proc. of an International Workshop and Exposition*, 1997, pp. 15-33.

[4] C. Snae and M. Brueckner, "Concept and Rule Based Naming System," *The Information Universe: Journal of Issues in Informing Science and Information Technology*, vol. 3, 2006, pp. 619-634.

[5] R. W. Hamming, *Coding and Information Theory*, 2nd Ed. Englewood Cliffs, NJ: Prentice Hall, 1986.

[6] G. Bouchard and C. Pouyez, "Name Variations and Computerised Record Linkage," *Historical Methods*, vol. 13, no. 2, 119-125, 1980.

[7] L. K. Branting, "Name-Matching Algorithms for Legal Case-Management Systems," Refereed article in: *The Journal of Information, Law and Technology (JILT)*, 2002. Available: http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002_1/branting/

[8] D. Jurafsky and J.H. Martin, *Speech and Language Processing*, Prentice Hall, 2000.

[9] I.P. Fellegi and A. B. Sumter, "A Theory for Record Linkage," *Journal of the American Statistical Association*, vol. 64, pp. 1183-1210, 1969.

[10] G. Bouchard, "The processing of ambiguous links in computerised family reconstruction," *Historical Methods*, vol. 19, no. 1, pp. 9-19, 1986.

[11] D. De Brou and M. Olsen, "The Guth Algorithm and the Nominal Record Linkage of Multi-Ethnic Populations," *Historical Methods*, vol. 19, no. 1, pp. 20-24, 1986.

[12] G. J. A. Guth, "Surname Spellings and Computerised Record Linkage," *Historical Methods. Newsletter*, vol. 10, no. 1, pp. 10-19, 1976.

[13] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Dokl*., vol. 6, pp. 707-710, 1966.

[14] K. M. Odell and R. C. Russell, *Soundex phonetic comparison system* [cf. *U.S. Patents 1261167* (1918), *1435663* (1922)].

[15] A. Binstock and J. Rex, *Practical Algorithms for Programmers.* Addison-Wesley, Reading, Mass., pp. 158-160, 1995.

[16] A. J. Lait and B. Randell, "An Assessment of Name Matching Algorithm," *Society of Indexers Genealogical Group, Newsletter Contents, SIGGNL issues 17*, 1998.

[17] C. Snae and B. M. Diaz, "Name Matching for Linkage Among English Parish Register Records," in Proc. of the *Human and Computer Conf*, pp. 218-224, Japan, 2001.

[18] C. Snae and B. M. Diaz, "An Interface for Mining Genealogical Nominal Data Using the Concept of linkage and a Hybrid Name Matching Algorithm," *Journal of 3D-Forum Society*, vol. 16, no. 1, 2002, pp. 142-147.

[19] L. E. Gill, M. J. Goldacre, H. M. Simmons, G. A. Bettley, and M. Griffith, "Computerised Linkage of Medical Records: Methodological Guidelines," *Journal of Epidemiology and Community Health*, vol. 47, pp. 316-319, 1993.

[20] P. H. Reaney and R. M. Wilson, *A Dictionary of English Surnames*, Oxford University Press, 1997.

[21] C. Snae and M. Brucker, " LOWCOST: Local Organisation Search With Consolidated Ontoplogies for name, Space and Time," in Proc. of the *International Conf. on Software Engineering*, Innsbruck, Austria, February 13 – 15, 2007.

Chakkrit Snae was born in Rayong, Thailand on 8th May 1972. Currently working at Department of Computer Science and Information Technology, Faculty of Science, Naresuan University. Ph.D. in Computer Science, University of Liverpool, Liverpool, England, 2006. M.Sc. in Computer Science, University of Newcastle Upon Tyne, Newcastle, England, 1999. B.Sc. in Mathematic, Naresuan University, Phisanulok, Thailand, 1995.

In 2005 he started lecturing at Department of Computer Science and Information Technology, Naresuan University, Thailand, mainly for **SE, ES, AI,** and **IS.** His teaching concept is how to make SE (Software Engineering) evolving to ES (Expert System) by use of AI (Artificial Intelligence) and IS (Information Systems). He used to work part time as an official Thai website translator for Everton Football Club, England, while he was doing PhD in Liverpool. Since he arrived at Naresuan University he spent most of his time on various research areas. He has implemented several systems, such as **IT-TELLS** (an automated transcription tool for English and Thai writing system), **RESETT** (Rule-based Expert System for English to Thai Transcription), **LOWCOST**: Local Organisation With Consolidated Ontologies for name, Space and Time), **LOBO** (Local Organisation Business Ontologies), **BOOT** (Business and organisation Ontologies for Thailand)**GINO** (Gender Indication using Name Ontologies), **NESTA** (Name Expert System using Thai Astrology), **NESTNOC** (Name Expert System using Thai Naming Ontologies and Clustering)and systems that are related to ontologies, naming system and convention. Currently he is working on **NARESUAN-M**^{2} (Name Recognition Expert System Using Automated Name Matching Methods).

