Similarity Based Naming System Using Name Matching Method

Hun Aye
University of Computer Studies (Meiktila), Myanmar
hanayenge@gmail.com

ABSTRACT

Names are the most important as they are used in many process, such as identifying of people. Names are stored within large information system includes government, medical, educational, and even commercial records which are kept about individuals. Each culture has a set of names as well as a rage of permitted variations in its naming system. This paper involves names used for identifying culture, relation, race and nationality of Shan name. Name of Shan are divided into thirty three kinds of culture for different groups. There has been the different of relations name slightly. This paper develops for this different relational name in Shan cultures. This paper discusses the characteristics of personal names and presents potential sources of variations and errors. Names are many elements of personal names which vary in and between the different cultures. This paper uses name matching method of Levenshtein Distance algorithm.

1. INTRODUCTION

Names are important in many societies, even in technologically oriented ones which use ID (Identification) systems or other ways to identify individual people. There are many elements of personal names which vary in and between the different cultures. Names such as personal surnames are the most important in the naming system as they are used in many processes, such as identifying of people, record linkage and for genealogical research as well. On the other hand, variation of names can be seen a major problem for the identification and search for people.

Names have been persistently problematic for the nominal data record linkage processing or searching, which undergo variations such as phonetic and alternate spellings. This problem can be solved 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 based on error 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 are 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. Variations and errors in names make exact string matching problematic, and approximate matching techniques have to be applied. When compared to general text, however, personal names have different characteristics that need to be considered. Names serve for labeling of categories or classes and for individual items. They are properties of individual which are of greater important in most communities.

From the technical point of view, the system wants to link and match as many names as possible with the correct individuals. Names are also important pieces of information when databases are deduplicated and when data sets are linked or integrated and no unique entity

identifiers are available. Personal names have characteristics that make them different from general text. While there is only one correct spelling for many words, there are often several valid variations for personal names. People also frequently use nicknames in daily life.

2. BACKGROUND THEORY

Personal names are also heavily influenced by a person's cultural background, and in certain situations people do change their names. All these issues make matching of personal names more challenging compared to matching of general text. Name equivalence can be discussed in many different ways.

First of all, one needs to separate the various types of names that can be observed and second of all, one need to define name equivalence. Gustafson describes some of the partitions between name types; a name can refer to a person (first name, surname, nickname etc.), a geographical site (place names and street names) and various non-human object names (animal names, company names etc.).

The second name related problem concerns the definition of equivalence between names. Many different definitions can be said to be true. The definitions distinguish between three different types of equivalence: same spelling, same pronunciation and same origin. Names have been regarded as equivalent when they have or could have the same pronunciation [3].

The spelling and the pronunciation of a name may have altered when people moved and due to different and uncertain spelling conventions. To be able to match these types of names, a name matching algorithm needs to be somewhat "loose" since the spellings can have diverted to a considerable extent. Most techniques are based on pattern matching, phonetic encoding, or a combination of these two approaches.

2.1. Meaning of Names

Names are more than just strings of characters. Names provide different elements which may differ between cultures. Some cultures show the marital status very clearly, others do not refer to it.

Boys can be named by: (1) taking names of leading or important male characters accepted as having lasting literary value and (2) combining monosyllables evidently indicating the male.

Girl can be named by: (1) taking name of leading female characters from well-known literature and (2) combining monosyllables evidently indicating the female [1].

2.2. Personal Name Characteristics

Even when only considering the English-speaking world, a name can have several different spelling forms for a variety of reasons. Compound names are often used by married women, while in certain countries husbands can take on the surname of their wives. In today's multi-cultural societies and worldwide data collections, the challenge is to be able to match names coming from different cultural backgrounds [8].

2.3. Categories of Name Variations

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. Most of these variations can be categorized as follows:

- Spelling variations
- Phonetic variations
- Double names
- Double first names
- Alternate first names

In these situations an algorithm that recognizes simple variations in spelling or phonetics would not be able to identify two such names as referring to the same person. Any name matching scheme can be but a contribution to the frequently encountered task of trying, in the absence of any scheme of personal identity numbers, to determine whether two sets of personal information in fact relate to the same individual.

3. NAME MATCHING METHOD

A lot of data collected and processed contains information about people. Personal names are often used as identifiers to access data or when searching for people. Three main application areas for name matching: Text data mining, Information retrieval and Data linkage and deduplication.

Name matching can be defined as the process of determining whether two name strings are instances of the same name. As name variations and errors are quite common, name string comparison without result in good matching quality. The two main approaches for matching names are phonetic encoding and pattern matching [2].

3.1. Phonetic Encoding

Common to all phonetic encoding techniques is that they convert a name string into a code according to how a name is pronounced (i.e. the way a name is spoken). Naturally, this process is language dependent. When matching names, phonetic encoding can be used as a filtering step, i.e. only names having the same phonetic code will be compared using a computationally more expensive pattern matching algorithm [6]. Alternatively, exact string comparison of the phonetic encodings can be used. Most techniques have been developed mainly with English in mind: Soundex, Phonex, NYSIIS (New York State Identification Intelligence System), Double-Metaphone, Fuzzy Soundex [4].

3.2. Pattern Matching

Pattern matching techniques are commonly used in approximate string matching, which has widespread applications, from data linkage and duplicate detection, information retrieval, correction of spelling errors, to bio and health informatics [10].

- Levenshtein or Edit distance
- Damerau-Levenshtein distance
- Bag distance
- Smith-Waterman distance
- Longest common sub-string (LCS)
- Q-grams
- Positional q-grams
- Skip-grams
- Compression
- Jaro algorithm
- Winkler (or Jaro-Winkler) algorithm

4. LEVENSHTEIN DISTANCE

Levenshtein Distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. For example,

- If s is "test" and t is "test", then LD(s,t) = 0, because no transformations are needed. The strings are already identical.
- If s is "test" and t is "tent", then LD(s,t) = 1, because one substitution (change "s" to "n") is sufficient to transform s into t.

The greater the Levenshtein Distance, the more different the strings are [7]. The Levenshtein Distance algorithm has been used in: Spell checking, Speech recognition, DNA analysis and Plagiarism detection.

4.1. Levenshtein Distance Algorithm

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit.
2	Construct a matrix containing 0..m rows and 0..n columns. Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.

6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d [i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d [i-1,j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

4.2. Operation of Levenshtein Distance Algorithm

The Levenshtein algorithm (also called Edit-Distance) calculates the least number of edit operations that are necessary to modify one string to obtain another string. A matrix is initialized measuring in the (m,n)-cell the Levenshtein Distance between the m-character prefix of one with the n-prefix of the other word. The matrix can be filled from the upper left to the lower right corner. Each jump horizontally or vertically corresponds to an insert or a delete, respectively. The cost is normally set to 1 for each of the operations. The diagonal jump can cost either one, if the two characters in the row and column do not match or 0, if they do. Each cell always minimizes the cost locally. This way the number in the lower right corner is the Levenshtein Distance between both words. There are two possible paths through the matrix that actually produce the least cost solution. Namely"=" Match; "o" Substitution; "+" Insertion; "-" Deletion [9].

5. NAME MATCHING SYSTEM FOR SHAN NAME

Currently we are constructing and implementing a naming system which offers to “correct” Shan names according to the methodology mentioned above. The system will give us the letters for each input name. We use these letters to compare names based on the basic algorithm. The user will be able to choose from a list of resulting possible names according to their respective meaning. By using this algorithm, the correct name will be known and the amount of variations and needed character also will be understood. Finally, this system will give to the user for the nationality, relation, race, gender, English and Myanmar meaning of Shan cultures [5]. We use a dictionary database of more than 2000 Shan names which contains not only the spelling, but also the meaning and correct pronunciation. Then the user can select the best name from the resulting list of names.

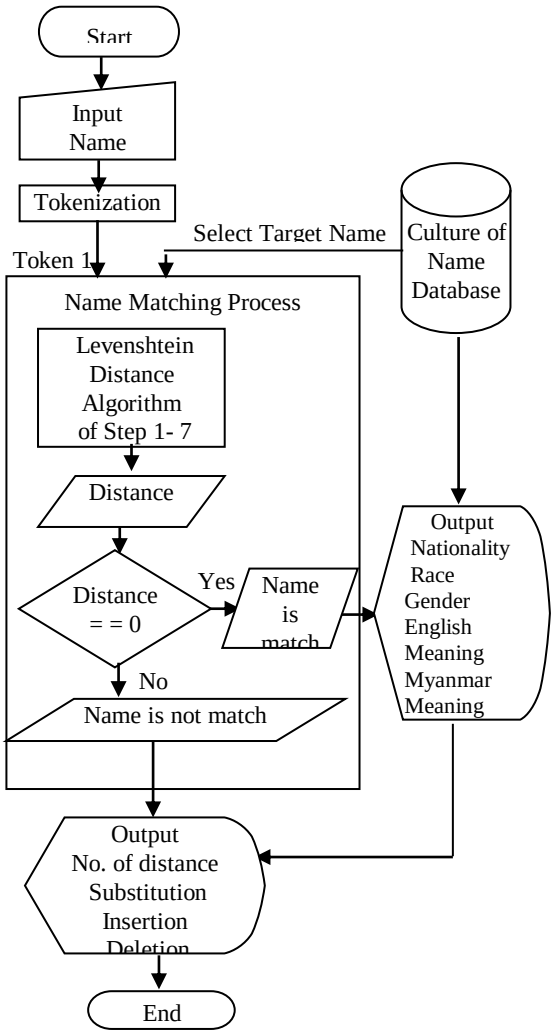

Figure 1. System Flow Diagram

6. IMPlEMENTATION RESULT

The paper proposed name variations for different cultures to guide the implementation of naming system, currently worked out for Shan names. This paper will be understood the characteristics of Levenshtein Distance algorithm which help to find reasonable variants of names.

Experimental results on different real name data sets have shown that there is no single best technique available. The characteristics of the name data to be matched have to be considered when selecting a name matching technique. The contributions of this system are a detailed discussion of the characteristics of personal names for Shan people such as Pao, Palaun, Taungyoo and Yunn, etc and possible sources of variations and errors in them, an overview of a range of name matching techniques. The paper provides the output result for many variations using several real world data sets containing relational Shan names.

This system provides a comprehensive review of existing culture distance literature with particular emphasis on data representation. The second and main objective will be to design and implement a comprehensive similarity and text distance measure incorporating new algorithm.

Perceptual measures and Levenshtein Distances calculated on the basis of transcriptions will form a baseline for the comparison of results obtained in our thesis. The main evaluation of the system will be based for the most part on its correctness on Shan name.

7. CONCLUSION

The paper discusses the characteristics of personal names and the potential sources of variations and errors in them, and present an overview of pattern matching based on name matching techniques. 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 any type of data.

More detailed analysis into the types and distributions of errors are needed to better understand how certain types of errors influence the performance of matching techniques.

The name matching techniques covered by this investigation comprise only a small selection of those existing, but they are the representative of many of the current approaches to the problem of name matching. Due to the diversity of applications for name matching techniques, choosing a particular algorithm will depend largely on the nature of the data it is to be applied to. This paper develops the problems involved in approximate string matching in general and in name matching specifically. The paper work suggests that methods based on distance measures are the best in these situations for the obvious reason that a pronunciation bias is unlikely to be reflected in a spelling bias. It was possible to identify areas of improvement to the design of this system by using the Phonex name matching method, based on the Soundex coding technique.

This system uses Shan name usages as a naming methodology, the Levenshtein or Edit-distance algorithm for personal name matching. Our proposed system uses the hybrid name matching algorithms to return the variants of names from a database with the relative probability of their similarity.

7.1 Advantages of the system

The advantage of this process would be an improvement of searching algorithms for Shan names in databases as well as in the internet. Here the system will need name matching algorithms. A further benefit of this process would be an optimization for name searching. The next step of development is to take into account different cultures. The benefits in number of matches being able to find all, or almost all, of the possible matches contained in the database, were with the current database and set of search words.

The results show that of the two transcription methods, the Levenshtein Distances based on the phonetic transcriptions more closely match the perceptual distances and group the dialects into their regional groups more closely.

REFERENCES

[1] Bouchard, G. and Pouyez, C. (1980). Name Variations And Computerised Record Linkage. Historical Methods, 13(2), 119-125.

[2] Branting, L.K. (2002). Name Matching Algorithms for Legal Case-Management Systems. Refereed article in: The Journal of Information, Law and Technology (JILT).

[3] Domeij, Rickard, Hollman, Joachim, and Kann, Viggo. Detection of Spelling Errors in Swedish Not Using a Word List en Clair, Journal of Quantitative Linguistics, Vol 1(3), 1994.

[4] Du, M. W., and Chang, S. C. A model and a fast algorithm for multiple errors spelling correction, Acta Informatica, Springer-Verlag, Vol 29, 1992, pp 281-302.

[5] Jonathan A. Zdziarski, Ending Spam, No Starch Press, Copyright © 2005 by "tokenization: the building blocks of spam".

[6] Jurafsky, D. and Martin, J.H. (2000). Speech and Language Processing, Prentice Hall.

[7] Lait, A. J., and Randell, B. An Assessment of Name Matching Algorithms, Department of Computing Science, University of Newcastle upon Tyne, UK 1993.

[8] Palmer, D. D. and Hearst, M.A. (1994). Adaptive sentence boundary disambiguation. Technical Report UCB / CSD / 94/797, University of California Berkeley, Computer Science Division.

[9] Peter C., Department of Computer Science, the Australian National University Canberra ACT 0200, Australia, "A Comparison of Personal Name Matching: Techniques and Practical Issues".

[10] Aroonmanakun, W. (2004). Thai Romanization, Retrived November, 19, 2005, from http://www.arts.chula.ac.th/%7Eling/tts/.