The Personal Name Problem And a Recommended Data Mining Solution

Clifton Phua, Vincent Lee and Kate Smith

School of Business Systems, Faculty of Information Technology, Monash University,
Clayton campus, Wellington Road, Clayton, Victoria 3800, Australia

{clifton.phua, vincent.lee, kate.smith}@infotech.monash.edu

ABSTRACT

The personal name problem is the situation where the authenticity, ordering, gender, and other information cannot be determined correctly and automatically for every incoming personal name. A novel solution, tested on scoring data, is to mine a comprehensive external name dictionary with a set of chosen techniques made up of exact matching, phonetics (extended soundex), simmetrics (levenshtein), and classifiers (naïve Bayes algorithm). The main contribution of this paper is in the evaluation of and selection from five very different approaches and the empirical comparisons of multiple phonetical and string similarity techniques for the personal name problem. Other contributions include relating personal names mining to credit application fraud detection and other security systems, and making the labelled data and techniques available for future studies. In reality, there is no silver bullet solution to this problem but it can be alleviated with appropriate techniques on sufficient name data.

Keywords

Credit application fraud detection, deception and alias detection, phonetical matching, approximate string matching, naïve Bayes

1. INTRODUCTION

Identity crime, consisting of identity theft and fraud, is about taking advantage of real unsuspecting individuals' personal information and/or fictitious identities for the perpetrators' own financial gain. It is an enabler for a myriad of white-collar crime such as insurance, credit, telecommunications application fraud, as well as other more serious crimes. In recent years, it has been accepted as a significant problem in developed countries and in coming years, its cost and extent are projected to grow steadily. Identity crime is almost certainly associated with an entity name, such as a personal, place, or organisational name. This paper focuses solely on **verifying and extracting information from personal names.** This is done with the intention of incorporating that functionality into a **data mining-based credit application fraud detection system**.

Almost every person has a life-long personal name which is officially recognised and has only one correct version in their language. Each personal name typically has two components/parts: a first name (also known as given, fore, or Christian name) and a last name (also known as family name or surname). Both these name components are strongly influenced by cultural, economic, historical, political, and social backgrounds. In most cases, each of these two components can have more than a single word and the first name is usually gender-specific. (see Figure 1).

Credit application fraud, a manifestation of identity and demographical fraud, if present, poses three important practical considerations for personal name verification:

- Balance between manual checking and analytical computing. Intuitively, about twenty percent of applications should be manually reviewed, the result has to be reasonably accurate, and each personal name should not take too long to be verified.
- Reliability of the verification data has to be examined. By keeping the name verification database's updating process separate from incoming applicant names, it can prevent possible data manipulation/corruption by fraudsters. However, the incompatibility of names in databases can also be caused by genuine reasons as such as cultural and historical traditions, translation and transliteration, reporting and recording variations, and typographical and phonetic errors [4].
- Domain knowledge has to be incorporated into the entire process. Within the Australian context, the majority of names will be Anglo-Saxon but the minority will consist of very diverse groups of cultures and nationalities. Therefore the content of the name verification database has to include a significant number of popular Asian, African, Middle Eastern, and other names.

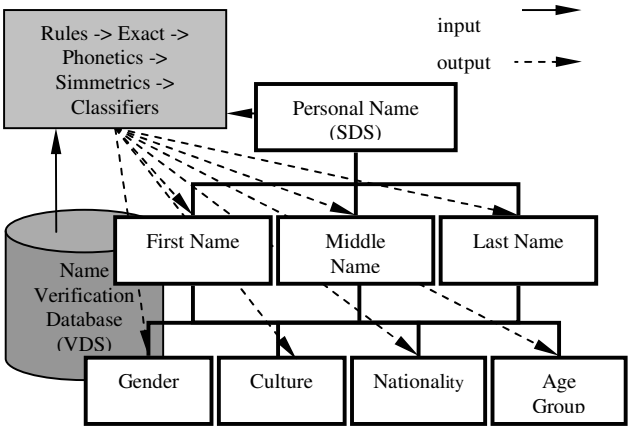

Figure 1: Hierarchy chart on the inputs, process, and outputs of the name verification task.

Figure 1 in the previous page illustrates the input, process, and output sections. The input refers to the incoming names and those in the verification database (which acts like a external dictionary of legal names); process/program refers to the possible five approaches for personal name analysis; and output refers to the insights correctly provided by the process. For simplicity, this paper uses first name to denote both first and middle names; and culture to represent culture and nationality. While the scope here explicitly seeks to verify the authenticity and extract first/last name and gender information from a personal name, culture can be inferred to a large extent and age group can be inferred to a limited extent.

Section 2 briefly describes related applications and academic disciplines. Section 3 explains the importance of personal names in the context of credit application fraud and provides four reasons on why sometimes names cannot be matched in the name verification database. Section 4 describes the verification and scoring data sets, and evaluation measures. Section 5 discusses the main experimental conditions and results. Section 6 highlights the main discussion points and limitations. Section 7 concludes the paper and considers the future work.

2. RELATED WORK

There are three broad application categories of related work in name matching [4]:

- Information retrieval - Finding exact or variant form(s) of incoming name in verification database with no changes to the database. This present work is the most similar to this where an incoming personal name is used as a search key to retrieve first/last name and gender information.
- Name authority control - Mapping the incoming name upon initial entry to the most commonly used form in database. Current publications on author citation matching within the ACM portal, CiteSeer and DBLP databases are examples of this [10, 11]. Unlike author names where the first names are usually abbreviated, credit applicant names have complete first and last names.
- Record linkage/duplication detection - Detecting duplicates for incoming multiple data streams at input or during database cleanup. Recent publications focused on supervised learning on limited labelled data [28] and on approximate string matching [3]. Unlike their matching work which uses comparatively smaller data sets and has other informative address and phone data. Intelligently matching incoming names-only data with a comprehensive verification database is a harder problem.

Other specific applications of personal name matching include art history [4], name entity extraction from free text [9, 21, 2], genealogy, law enforcement [29], law [19, 5], and registry identity resolution [27]. Name matching has been explicitly or implicitly researched under databases, digital libraries, machine learning, natural language processing, statistics, and other research communities; and also known as identity uncertainty, identity matching, and name disambiguation.

3. PROBLEM

3.1 Some Aspects of Any Personal Name and Justifications to Find Them

For personal names, there are requirements in systems to determine *authenticity* - to detect fake/fictitious names (pseudonyms); *ordering* - to find transposed first and last names; *gender* – determined from first names, to check for inconsistencies with the other attributes, such as title and gender attributes; and *culture* – to know the name's background (note that culture determination is not within the scope of this paper).

These four aspects can also be useful for the detection of synthetic identity fraud which is more prevalent than identity theft in application data [20]. They can also determine if fraudsters consciously or unconsciously favour a certain style of fraudulent personal names. They can be fake (i.e. "Fantastic Four" or "ABcd EfgH"), use of first names as last names (i.e. "George Timothy") and vice versa (i.e. Smith Jones). Fraudsters may prefer either exclusively male first names (i.e. Michael) or exclusively female first names (i.e. Sarah).

Determining ordering, gender, and culture from names can be crucial in characterising legitimate social networks of applications when these information are not explicitly given. Name ordering and derived culture are fairly consistent within normal social networks. Name ordering, derived gender and culture can help define the husband-wife, parent-child, siblings and other relationships across different cultures.

Fraudsters may be inclined to abuse personal names of a particular culture. There is a remote possibility that this is associated with organised crime and/or cultural-based crime. For example, in Australia, major organised crime syndicates are moving towards identity crime and others [26] and they are already or increasingly cultural-based [15]. In the US, [16] outlines the sophisticated white collar crimes by various cultural syndicates.

3.2 Reasons for Finding No Exact Match – Personal Name Problem

There are four main possibilities when the incoming first and last name does not match any name in the verification database exactly. First, this could be part of the objective - that the personal name is not *authentic* and should be manually checked. Second, it is most likely due to an incomplete white list. It is impossible to have a name verification database which has every legitimate name, especially rare ones. Third, the incoming name does not have any variant spelling of name(s) in the database (i.e. Western European last names). Fourth, there are virtually millions of potential name combinations or forms (i.e. East Asian first names).

The last three reasons are problems which prevent legitimate incoming personal names from being verified correctly by the database. Without finding an exact match in the name verification database, the personal name problem in this paper refers to scenario where the *authenticity, ordering*, *gender* (possibly *culture* and *age group*) cannot be determined **correctly** and **automatically** for **every** incoming personal name. Therefore, additional processing is required.

4. DATA SETS AND EVALUATION MEASURES

4.1 The Personal Name Verification Data Set (VDS)

The experimental VDS has 99570 unique names (examples) which are evenly distributed amongst gender, culture, and nationality. The original data set has 1 attribute and 3 class labels.

- The *Name* attribute consists of the name component in uppercase.
- The *First/Last* class label is binary with "F" and "L" representing first and last name respectively. A name which can either be a first or last name will be duplicated, one labelled with "F", and the other "L".
- This *Gender* class label is actually a subset of the *First/Last* class label, breaking "F" into "m" and "f" representing male and female in the former respectively. Some names which are unisex will have the name attribute value duplicated, one labelled with "m", and the other "f".
- The *Culture* class label has thirty-seven three-letter acronyms representing culture and country origin of a name. Any cross-cultural name can be duplicated and be labelled with a different acronym. This class is used here to segment the verification data for experiments.

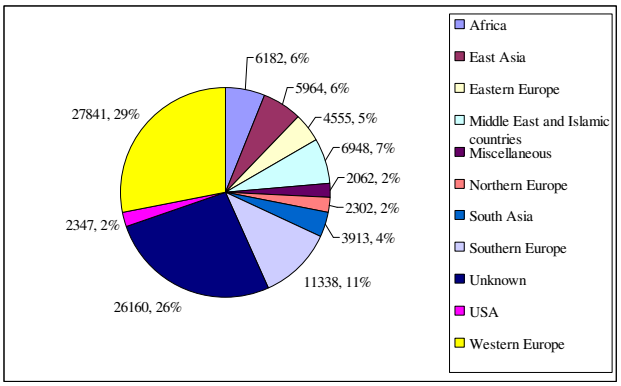

Figure 2: Pie chart of VDS distribution by international origins.

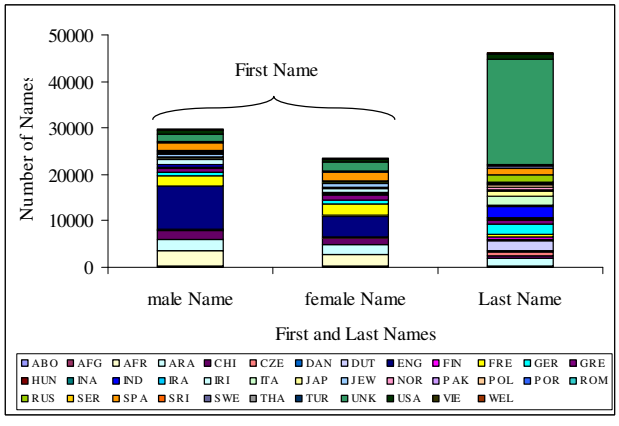

Figure 3: Bar chart of VDS distribution by first/last names, gender, and the culture.

Figures 2 and 3 give a broad and detailed view of the VDS respectively. Most of the main VDS data sources are publicly available, and is compiled from dozens of websites and a few individual contributors over a few months. Although it is a relatively large sample of the possible names, it does not have frequency information to indicate the popularity of any name part.

4.2 The Personal Name Scoring Data Set (SDS)

The SDS has 8623 personal names/instances (each with first and last names) and is an assembly of four publicly available data sources which are multi-cultural and within the Australian context (all labelled with first/last and manually labelled with gender):

- 82 staff and research students from the School of Business Systems.
- 6829 graduating students from Monash University who graduated in the second half of 2004 and the first half of 2005. 18 "personal names" within are degree titles - fake ones.
- 1689 graduating students from Melbourne University who graduated in April 2005.
- 23 fake personal names which describe facilities and items found in a hotel.

4.3 Evaluation Measures

To evaluate the effectiveness across all five approaches, this paper proposes a simple-to-use normalised net value (NNV) for the test set (SDS). The net value for an approach is the summation of all scores: where a score of 1 is given for a correct match/prediction, 0 for no match/prediction, and -1 for an incorrect match/prediction.

Every name component is assessed by:

$$\text{NNV} = \frac{\text{sum of scores} + \text{number of incoming names from SDS}}{\text{number of incoming names from SDS x 2}}$$

Classifiers are also evaluated by:

$$\text{F} - \text{measure} = \frac{2 \text{ x recall x precision}}{\text{recall} + \text{precision}} \quad [30]$$

5. EXPERIMENTS

All experiments are performed on a single computer with Pentium IV 3.0 GHz, 2 Gb RAM, running on Windows XP platform. The Nickname-Rules, Exact-Matching, Phonetics software is written in Visual Basic .NET, and Simmetrics [6] in C# .NET. Attribute/feature selection and classifiers are carried out from WEKA [30].

5.1 Rules

First, *Display_Nicknames* sub-procedure is used to output the most possible forms of a male/female nickname for the first name. For example, "VICKY" and "PETE" are output with "VICTORIA" and "PETER" respectively. Next, general name characteristics of first/last, gender, and culture are hard-coded as

rules and are applied directly to each personal name. Table 1 below shows *Implement_Rules* accepts either a first or last name and calls two second-level sub-procedures: *First_Name_Rules* which calls thirteen and *Last_Name_Rules* which calls twenty-nine third-level sub-procedures. Each third-level sub-procedure represents a specific rule set (of up to twenty name rules) which checks for particular substrings within the name component, and then outputs possible first/last name and/or gender. The ordering and gender matches/predictions for the name component will depend on which one of the two possible outcomes has the highest number of counts/votes.

Table 1: Pseudo-code of rules (top-level sub-procedure only).

```
Sub Implement_Rules(ByVal name_component, ByRef ordering,
ByRef gender)
   individual_word = name_component.Split(" ")
   FOR each individual_word
     First_Name_Rules(individual_word)
     Last_Name_Rules(individual_word)
   NEXT
   ordering = Maximum(first_name_counts|last_name_counts)
   gender = Maximum(male_counts|female_counts)
End Sub
```

These nickname rules are derived from [27, 10] and name component rules are compiled from several websites. Although these rules are not able to determine name *authenticity* (except to flag strings which are too short, too long, have non-alphabetic or rare punctation characters), it is the only approach here which guesses gender from last names without utilising the VDS. Referring to Table 2 below, despite outcomes on only 31.8% to 42.7% of the data, the NNV from rules is still good on last name and gender.

Table 2: Results of rules on first/last name and gender prediction on SDS.

	First Name	Last Name	Gender
% of incoming names predicted	40.0%	31.8%	**42.7%**
% of correct predictions	38.5%	**84.1%**	79.8%
Normalised Net Value	45.4%	60.8%	**62.7%**

5.2 Exact Matching

Each component in SDS is compared to every example in the VDS to find exact matches. For exact matching, phonetics, and simmetrics, when a match is found, its corresponding ordering and gender labels are counted, and the sum of candidate matches are returned. The label(s) with the highest count will be the final output. For example, if the name component "CLIFTON" is recognised in the VDS as both a first name and last name, the final prediction will be "F|L". But, it is important to note that in all subsequent experiments, predictions with conflicting outcomes such as "F|L" will be considered as a "no match/prediction" even though the correct class label is "F".

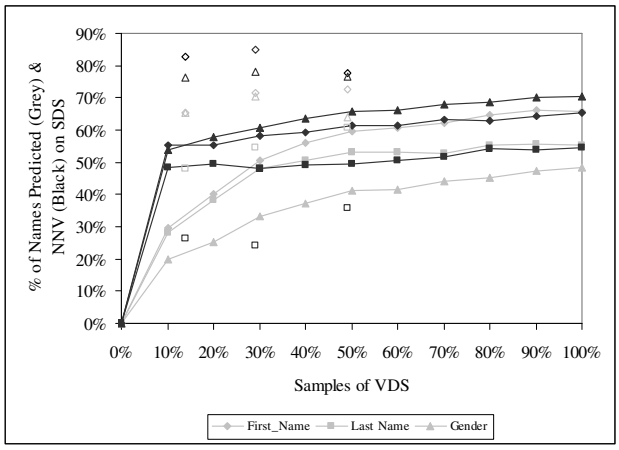

Figure 4: Results of exact matching with different sample sizes and data selections of VDS on SDS.

Figure 4 above shows the effect of having a larger random sample of VDS on the exact matching of first names, last names, and gender. Exact matching with the entire VDS can barely determine name *authenticity*, as it will involve manually checking 34.2% to 51.6% of the SDS which has no matches from the VDS (better than rules though). Its highest NNV is with the entire VDS: first names is 65.5% (better than rules), of last names is 54.7%, and of gender is 70.6% (better than rules). Unsurprisingly, with larger data samples, the number of predictions (plotted lines in light grey) and normalised net value (plotted lines in dark grey) increase marginally. Interestingly, although the number of matches on gender is the lowest among the three, gender predictions are the most accurate.

In Figure 4, the hollow scatter points at 14%, 29%, and 49% are selected samples of the VDS which represent the choice of English names (ENG) only, Western European names only (ENG, FRE, GER, DUT, IRI, WEL), and Western European plus five other major name groups (CHI, IND, SPA, ARA, RUS) accordingly. The dark grey hollow scatter points illustrate high accuracy for first name and gender probably because first names are repetitive and gender is determined from first names. However, there should not be any assumptions at this stage about the background of the incoming names, as they will most likely come from many groups.

5.3 Phonetics

Phonetics (sounds-like algorithms) operate by encoding the SDS instance and then matching that against the encoded names from the VDS. The ones used in the experiments here are briefly described below:

- Soundex: *Soundex* is the most widely used phonetic algorithm, has seven numerical codes and transforms the name into a four-letter code with the first letter of name preserved. *Extended Soundex* is a "relaxed" version of the original proposed by this paper: no preservation of the first letter of name, not restricted to a four-letter string, no trailing zeros to "pad" the four-letter code, and place "R" in the same numerical code group as "L".

- Soundex variants: *Phonix* applies more than a hundred string standardisation rules before a nine numerical code transformation [31]. *Phonetex* [12] is a combination of Soundex and Phonix. *Phonex* [17] is a combination of Soundex and Metaphone.
- Metaphone and extension: *Metaphone* [22] reduces English words into sixteen consonant sounds and *Double Metaphone* [22] to twelve consonant sounds.
- Others: *Caverphone* has more than fifty sequential rules and *NYSIIS* uses five sets of rule sets to map name into code [13].

There are two obvious flaws in phonetics: Most of them were mainly designed for the Anglo-Saxon last names, yet many last names in most databases are of many different cultures. Also, many of them were designed for different purposes or were not general enough for widespread use. For example, phonetex was designed for spell checkers; phonex was adapted to British surnames, caverphone was designed for New Zealand accents, and NYSIIS was mainly used for the New York city population.

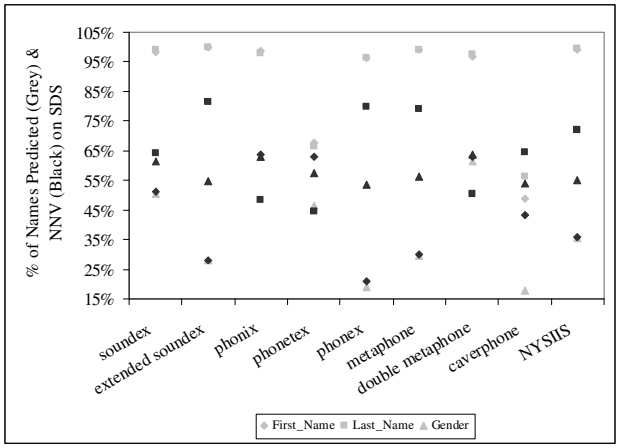

Figure 5: Results of phonetics on first/last name and gender prediction on SDS.

In Figure 5 above, there are phonetical matches to almost all SDS instances except for phonetex and caverphone. The highest NNV of first name is phonix at 63.9%, of last name is extended soundex at 81.4%, and of gender is double metaphone at 63.7%. The superior performance of extended soundex against other more complex phonetical techniques for this problem is unexpected. The results confirm that phonetics were originally designed and is still effective for matching last names. In contrast, first names are neglected by phonetics.

5.4 Simmetrics / Distance metrics

While phonetics examine dissimilar groups of letters to identify similarities in pronunciation, simmetrics use similar or identical groups of letters. Simmetrics (looks-like algorithms) are character-based similarity metrics which output the scaled similarity score (between 0 and 1) between two names components, where 0 represents totally different and 1 represents identical. Similarity measure = 1 - distance measure. Those used in the experiments here are briefly described below:

- Edit distances [6, 8]: *Levenshtein* calculates minimal number of single-character insertions, deletions, and substitutions to transform one string into another. *Needleman-Wunsch* is an extension of Levenshtein with a variable gap cost for insertions and deletions. *Smith-Waterman* is similar to Levenshtein but allows for context dependent gap costs.
- Non-edit distances [6, 8, 25]: *N-gram* calculates the percentage of matches of all possible substrings of size *n* between two strings. *Trigram* is used in the experiments here. *Block* is the absolute difference of coordinates between two names. *Jaro* is dependent on number and order of common characters between two names. *Jaro-Winkler*, an extension of Jaro, takes in account the length of common prefix between two names. *Ratcliff-Obershelp* matches characters in the longest common subsequence, and recursively matching characters on the both sides of the longest common subsequence.

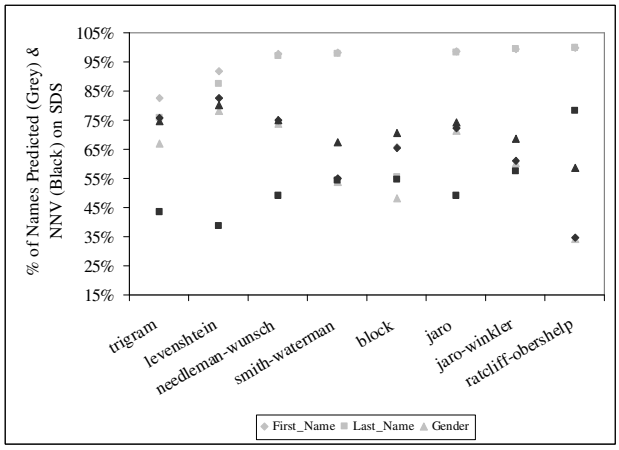

Figure 6: Results of simmetrics on first/last name and gender prediction on SDS.

With similarity threshold set at 0.8, Figure 6 above shows that simmetrics enables a very high match rate between VDS and SDS except for trigram, levenshtein, and block. The highest NNV of first name is levenshtein at 82.4%, of last name is ratcliff-obershelp at 78.1%, and of gender is levenshtein at 80.3%. Like extended soundex, the basic levenshtein produces the best NNV for first name and gender However, the percentage of names matched/predicted is relatively low at 91.7% and 78.2% respectively. Perhaps these names without matches/predictions should be manually investigated. There are other theoretically sound similarity metrics which have been experimented on VDS and SDS, but they are inefficient (*Editex*, *Gotoh,* and *Monge-Elkan*), or ineffective (*Hirschberg* and *Ukkonen*) for this problem.

5.5 Classifiers

5.5.1 Data Sets Construction for Classification

Twenty-eight derived attributes, consisting of five numerical and twenty-three nominal, were created from each name part. The numerical derived attributes include name length, 2 vowels counts and 2 normalised vowels counts (of first six letters and of last four

letters). The nominal derived attributes include *n*-grams of each first (5), second (3), third (3), third last (3), second last (3), last letters of each name component (6).

For example, the first name "CLIFTON" results in the following attribute values: 7, 2, 1, 0.33, 0.25, "C", "CL", "CLI", "CLIF", "CLIFT", "L", "LI", "LIF", "I", "IF", "IFT", "IFT", "FT", "T", "FTO", "TO", "O", "LIFTON", "IFTON", "FTON", "TON", "ON", "N". And the last name "PHUA" gets: 4, 2, 2, 0.33, 0.5, "P", "PH", "PHU", "PHUA", "PHUA", "H", "HU", "HUA", "U", "UA", "UA", <NULL>, "PH", "H", "PHU", "HU", "U", "PHUA", "PHUA", "PHUA", "HUA", "UA", "A". Initially, sixty-seven frequently occurring bigrams, from "AB", "AE", "AH" to "YA", "YS", "ZU" as were included as attributes, but predictive accuracy remained the same while computational complexity increased significantly. Maybe it is because *n*-gram repeats are rare in names. This decision to remove them was confirmed through information gain feature selection.

From the feature selection, it was ascertained that the predictiveness of first/last names and gender come predominantly from the last few letters of the name component (last letter, last two letters, and last three letters). This discovery is in sync with [1] that the last letter in the first name is the most predictive for gender, even in a multi-cultural context.

5.5.2 Classification Results

Classifiers (discriminant functions) are trained with certain subsets of the VDS and score the SDS accordingly (it is the only approach which gives a ranked output). This is a hard problem because of the large numbers of nominal attribute values in the *n*-gram attributes. Due to this, typical decision trees (cannot handle identifier-like attributes) and support vector machines (extensive pre-processing required to convert data to numerical format) are not suited to this task. On the other hand, naïve Bayes is the only suitable classification algorithm which is extremely efficient and still comparable to other state-of-the-art algorithms.

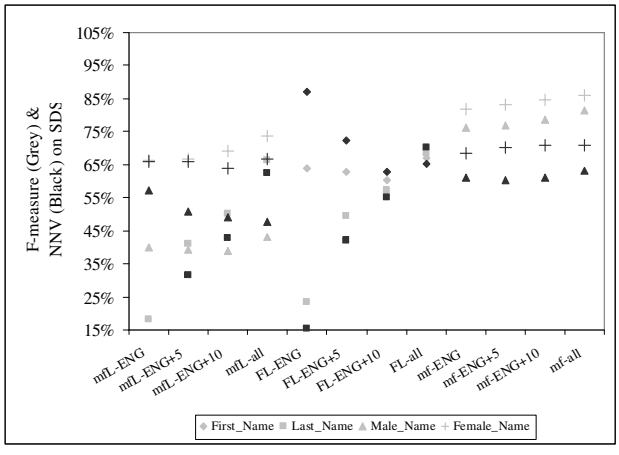

Figure 7: Results of classification with different class labelling and data selection on SDS.

With reference to the *x*-axis of Figure 7 above, the classifiers are built from *mfL* which has a trinary class label (male, female, or last), *FL* which has a binary class label (first or last), and *mf* (from first names only) which has a binary class label (male or female). Finer-grained classifiers are built using selected samples of the VDS described in section 5.2 (ENG, ENG + 5, ENG + 10). The highest NNV of first name is FL-ENG at 87.0%, of last name is FL-all at 70.1%, of male name is mf-all at 63.0%, and of female name is mf-all at 71.0%. Although there is a conflict where the highest F-measure of first name is FL-all instead of FL-ENG, the latter is still preferred for SDS as most first names will be of English origin (even for different cultures). Also, it seems that results with binary class labels are slightly better than the trinary class label. The main disadvantage of classifiers, compared to the rest, is in the pre-processing effort required to transform data into a suitable format and the expertise needed to interpret the results.

5.6 Combinations / Hybrids

Previous research claims that combination of results/evidence almost always improves performance [14, 12, 31], but this depends on the diversity of the individual results. Table 3 below shows the solution for the name components. Although the first name and gender predictions are the best by combining results from simmetrics and classifiers; for last name, phonetics remain the single most important algorithm (combining results decreases NNV significantly). The NNVs here are a few percentage points lower than the highest NNVs from other approaches but all the names from the SDS will now have a match/prediction to indicate whether a name component is a first or last (*ordering*), and/or male or female (*gender*) name.

Table 3: Results of the optimum combinations on SDS.

	First Name	**Last Name**	**Gender**
Hybrid	Simmetrics (levenshtein) and Classifiers (FL-ENG)	Phonetics (extended soundex)	Simmetrics (levenshtein) and Classifiers (MF-all)
% of incoming names predicted	100%	100%	100%
Normalised Net Value	78.6%	81.2%	78.8%
Computation time per personal name (see Figure 8)	6.5 secs	1.2 secs	6.5 secs

These are simple and effective techniques for mining interestingness patterns in personal names. For example, through the techniques, a previously unknown, interesting, statistically reliable, and actionable piece of knowledge about the authentication of names was discovered. There is a high probability that a personal name is not *authentic* if both the following rules are satisfied:

- Rule 1: With exact matching, if a SDS instance not found in VDS, and
- Rule 2: With simmetrics (levenshtein) and classifiers (mfL-all), if both predicts the same SDS instance to be a last name and/or male name.

With Rule 1, the number of SDS first names and SDS last names with no exact matches is 2945 (34.2%) and 3863 (44.8%) respectively. It is irrational to manually check through all these names to find 82 fake names. But if Rule 2 is incorporated too, by manually investigating 95 (1.1%) first names and 178 (2.1%) last names, 21 (51.2%) and 22 (53.7%) fake names can be found respectively. This is probably because first names and female names have more distinct patterns than last names and male names. In credit applications, once these fake names are detected, they will be rejected immediately.

6. DISCUSSION & LIMITATIONS

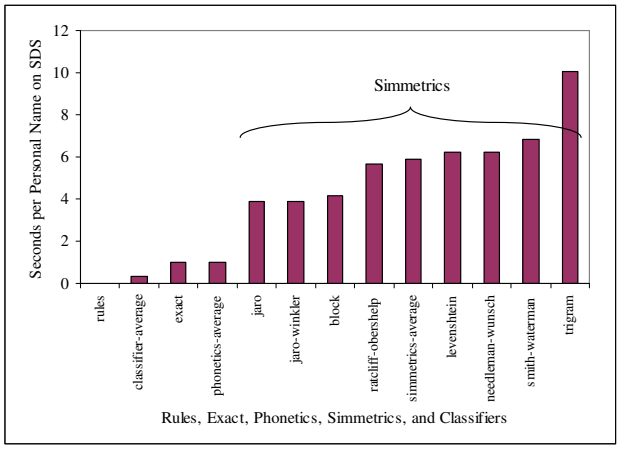

Figure 8: Computation time of the five approaches on SDS.

Are the chosen techniques efficient? Figure 8 above shows the computational times of all five approaches for first/last name and gender on a single SDS instance. On average for each SDS instance, simmetrics is 6 times computationally more expensive then phonetics and exact matching, 17 times more than classifiers, and 2400 times more than rules. Therefore, under these experimental conditions, the efficient usage of simmetrics for this problem will be restricted to several thousand personal names.

Which data mining research and applications can use the VDS and its recommended techniques reliably? Below is a list of likely candidates:

- Research: author recognition in digital libraries, database marketing, fraud detection, homeland security, recommendation systems, and social network analysis. It can even be used to increase the name variety and quality of artificial data generators such as [7].
- Private sector: account opening application, address change, payment activities in internet-based businesses, insurance, banks, and telecommunications.
- Public sectors: name verification activities in card issuing authorities, customs, electoral registers, health sectors, law enforcement, immigration, social security, and tax offices.

How can this work be incorporated into say, a credit application data mining-based fraud detection system? For our experimental system *Hesperus*, it has to process several thousands of synthetic applications a week, each applicant's personal name need to be verified automatically (authenticity), and its additional information (ordering and gender) to be extracted as useful derived attributes for defining legitimate and anomalous social networks [24].

What are the limitations in this paper? Individual privacy is an extremely important issue which has to be addressed to implement any form of real fraud monitoring system. The classifiers and combinations analysis of names are still a form of data mining "by hand" (not fully automated). The adjustment of different similarity thresholds with simmetrics for VDS and SDS can be investigated. Multi-cultural phonetical techniques such as the International Phonetic Alphabet [18] can be explored.

7. CONCLUSION

In this paper, personal name problem is defined for the case where the authenticity, ordering, and gender cannot be determined correctly and automatically for every incoming personal name. The recommended solution is to use the data from VDS with a set of techniques made up of exact matching, phonetics (extended soundex), simmetrics (levenshtein), and classifiers (naïve Bayes algorithm).

The original labelled training, and test data will be available at:

http://www.bsys.monash.edu.au/people/cphua/

The source code of phonetics and simmetrics are available at:

http://sourceforge.net/projects/simmetrics/

ACKNOWLEDGMENTS

This research is financially supported by the Australian Research Council under Linkage Grant Number LP0454077. Special thanks to Herbert Barry, Sam Chapman, Vanessa Oltman, Haidong Wang, Gerhard Fries, and other anonymous people who have contributed data, ideas, papers, and/or software for this paper.

REFERENCES

[1] Barry H and Harper A. "The Majority of Female First Names Ended in A or E Throughout the Twentieth Century", *in Gender Roles*, Nova Science Publishers Inc., **Chapter VI**, pp117-143, 2005.

[2] Bikel D, Schwartz R and Weischedel R. "An Algorithm that Learns What's in a Name", *Machine Learning*, **34**, pp211-231, 1999.

[3] Bilenko M, Mooney R, Cohen W, Ravikumar P, and Fienberg S. "Adaptive Name Matching in Information Integration", *IEEE Intelligent Systems*, **18**(5), pp16-23, September/October 2003.

[4] Borgman C and Siegfried S. "Getty's Synoname and Its Cousins: A Survey of Applications of Personal Name-Matching Algorithms", *Journal of the American Society for Information Science*, **43**(7), pp459-476, 1992.

[5] Branting K. "Name-matching Algorithms for Legal Case-management Systems", *Journal of Information, Law and Technology*, 2002.

[6] Chapman S. "SimMetrics – Open Source Similarity Measure Library", accessed from http://sourceforge.net/ projects/ simmetrics/, accessed in April 2005.

[7] Christen P. "Probabilistic Data Generation for Deduplication and Data Linkage", *Sixth International Conference on Intelligent Data Engineering and Automated Learning*, Brisbane, July 2005, accepted.

[8] Cohen W, Ravikumar P and Fienberg S. "A Comparison of String Distance Metrics for Name Matching Tasks", *in Proceedings of AAAI03*, 2003.

[9] Cohen W and Sarawagi S. "Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Market Extraction Processes and Data Integration Methods", *In Proceedings of SIGKDD04*, 2004.

[10] Feitelson D. "On Identifying Name Equivalences in Digital Libraries", *Information Research*, **9**(4), 2004.

[11] Han H, Giles L, Zha H, Li C and Tsioutsiouliklis K. "Two Supervised Learning Approaches for Name Disambiguation in Author Citations", *in Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, pp296-305, 2004.

[12] Hodge V and Austin J. "A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach", *IEEE Transactions on Knowledge and Data Engineering*, **15**(5), pp1073-1081, 2003.

[13] Hood D. "Caverphone: Phonetic Matching Algorithm", Technical Paper CTP060902, University of Otago, New Zealand, 2002.

[14] Hsiung P. "Alias Detection in Link Data Sets", Technical Report CMU-RI-TR-04-22, Robotics Institute, Carnegie Mellon University, USA, 2004.

[15] Jenkins C and Colley A. "Asian Gangs Join Cyber Fraudsters", *The Australian*, 24th May 2005.

[16] Kleinknecht W. "The New Ethnic Mobs: The Changing Face of Organised Crime in America", New York Free Press, 1996.

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

[18] Lutz R and Green S. "Measuring Phonological Similarity: The Case of Personal Names", LAS Whitepaper, 2003.

[19] Navarro G, Baeza-Yates R and Arcoverde J. "Matchsimile: A Flexible Approximate Matching Tool for Searching Proper Names", *Journal of the American Society for Information Science and Technology*, **54**(1), pp3-15, 2003.

[20] Oscherwitz T. "Synthetic Identity Fraud: Unseen Identity Challenge", *Bank Security News*, **3**(7), April 2005.

[21] Patman F and Thompson P. "Names: A New Frontier in Text Mining", *in Proceedings of Intelligence and Security Informatics*, pp27-38, 2003.

[22] Philips, L. "Hanging on the Metaphone", *Computer Language,* **7**(12), December 1990.

[23] Philips, L. "The Double Metaphone Search Algorithm", *C/C++* Users Journal, June 2000.

[24] Phua C, Lee V, Smith K and Gayler R. "On the Empirical Scoring of Anomalous Credit Applications with Pair-wise Matching", *Credit Scoring and Credit Control IX*, 2005, accepted.

[25] Ratcliff J and Metzener D. "Ratcliff-Obershelp Pattern Recognition", *Dictionary of Algorithms and Data Structures*, Black P (ed.), NIST, 1998.

[26] Riches S. "Criminals Branch Out", Herald Sun, 4th November 2004.

[27] Stanford Information Technology Systems and Services. "Person Registry Identity Resolution", accessed from http://www.stanford.edu/dept/itss/infrastructure/registry/project/person_registry/attributes/matching.html, accessed in April 2005.

[28] Tejada S, Knoblock C and Minton S. "Learning Domain-independent String Transformation Weights for High Accuracy Object Identification", *in Proceedings of SIGKDD02*, 2002.

[29] Wang G, Chen H and Atabakhsh H. "Automatically Detecting Deceptive Criminal Identities", *Communications of the ACM*, **47**(3), pp71-76, March 2004.

[30] Witten I and Frank E. *Data Mining: Practical Machine Learning Tools and Techniques with Java*, Morgan Kauffman Publishers, California, USA, 2000.

[31] Zobel J and Dart P. "Phonetic String Matching: Lessons from Information Retrieval", *in Proceedings of 19th International Conference on Research and Development in Information Retrieval*, pp166-172, 1996.