I.J. Intelligent Systems and Applications, **2016, 2, 59-65**
Published Online February 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2016.02.08

String Variant Alias Extraction Method using Ensemble Learner

P.Selvaperumal
Research student, Department of Computer science and Engineering, Manonmaniam Sundaranar University Tirunelveli, India
E-mail: Selvaperumal.p@gmail.com

Dr.A.Suruliandi
Professor, Department of Compute science and Engineering, Manonmaniam Sundaranar University Tirunelveli, India

Abstract—String variant alias names are surnames which are string variant form of the primary name. Extracting string variant aliases are important in tasks such as information retrieval, information extraction, and name resolution etc. String variant alias extraction involves candidate alias name extraction and string variant alias validation. In this paper, string variant aliases are first extracted from the web and then using seven different string similarity metrics as features, candidate aliases are validated using ensemble classifier random forest. Experiments were conducted using string variant name-alias dataset containing name-alias data for 15 persons containing 30 name-alias pairs. Experimental results show that the proposed method outperforms other similar methods in terms of accuracy.

Index Terms—String variant alias, name disambiguation, Entity disambiguation, Information extraction.

I. Introduction

Aliases are surnames or nicknames for a known name. Alias extraction is an information extraction problem that involves extracting and validating alias names. There are many kinds of aliases are there in the web like string variant aliases, lexically structured and semantic aliases etc [1] [2] [3]. The problem of string variant alias name extraction is referred by names like approximate name matching, string similarity calculation, duplicate record detection etc.Approximate name matching is the problem of searching for approximate matches of name in the web and hence it is called as approximate name matching.

Name variation occurs because of transcription errors, translation errors, and lack of standard format etc. Presence of spelling variation of same name makes integration of data, ontology integration etc difficult. String variation may be because of unintentional misspelling of the name or may be because of translation of web page contents from one language to another. In either of the aforementioned cases, the variant names are aliases of the primary name. For example, Arnold Schwarzenegger name has several name variants like Arnoldschwarzeneger, Arnold schwazenegger and also non string similar alias names like '*the Governator*',' terminator', 'Arnie'. Some alias names can be found by simple approximate string matching algorithms while others are difficult to find since many are syntactically dissimilar [4]. This makes extracting all the aliases of a known name a challenging task. Identifying spelling variants of an entity is difficult in web because they do not share any common pattern of variation. A widely used notion of string similarity is the edit distance which is the minimum number of insertions, deletions, and substitutions required to transform one string into the other [5].It is the commonly used similarity measure for measuring string similarity [6].Various studies has been conducted in the past to investigate the performance of string similarity metrics [7] [8] [9]. No string similarity measure suits for all different types of domains. Some string similarity metrics operate at letter level (like Levenshtein, Smith Waterman Similarityetc) and some other at token level (like Jaccard, tf-idf).String similarity can be calculated using string similarity metrics, token based distance function and Hybrid distance functions [7]. Thus different string similarity metrics helps in identifying different string variant names for a known name.

A Closely related field to string variant alias name extraction is spelling correction [10], where the aim is to find the nearest similar lexicon word. Another closely related field is record matching [11] in statistics field, where the aim is to use statistical models to whether a pair of records relate to same. In Duplicate record detection, the aim is to find similarity between two records in the database systems.

The problem of string variant alias name extraction is difficult because of different kinds of string variant aliases, size of web, difficulty in validating the extracted aliases etc. Spelling variant and orthographic variants of primary names are considered as string variant aliases. Orthographic variation includes hyphenation, punctuation, capitalization, word beaks etc and string variant alias name includes addition, substitution or elimination of one or more letters in the name to form alias name. String variant name is prevalent in web pages, blogs and posts because of the typographical errors, misspellings, abbreviations, pronunciation variation in the names. There also phonetic variants of names where phonemes

of the name are modified to form alias names like Sinclair and St. Clair [12]. It is difficult to say whether one string is a variant form of another string or not. This is because there is no uniform pattern of string variation and there are different similarity metrics to measure the degree of similarity between strings.Thus extracting string variant alias names from the web is a challenging task and validating those string variant forms are equally challenging. The absence of any standard benchmark dataset for string similarity detection makes it difficult to compare metrics and algorithms to compare with each other to find best metrics [13].

Personal names is different from other words in the text. While there is only one spelling for many words, names cannot be considered so. There is a subtle difference between string similarity matching and string variant alias name validation and in this paper both are used in the same sense. There are different N-gram algorithms available for approximate string matching process. According to salton, [14] bigram and tri gram does best suit for approximate string matching. Most string similarity methods are based on a pattern matching, phonetic encoding or hybrid of these methods.

In database systems, this task of string similarity detection is called duplicate entity or record detection [13]. The difference between alias extraction from the web and record duplication detection is, in database records are more structured and formal in design. In web, these strings variants are scattered across web pages and extracting them is difficult. The presence of spell variant or phoneme variant names decreases the accuracy of information retrieval system. In bibliographic databases, knowing different forms of same name of the author helps in increasing the accuracy of the system. In this paper, a new method of extracting string variant alias name is proposed and evaluated using synthetic name-alias dataset.

The contribution of the work includes

- Extracting string variant alias names from the web
- Validating alias names using robust ensemble classifier Random forest.

A. Related Work

Paul Hsiung et al [2], used link data sets to extract string variant and semantic aliases. He used orthographic features such as string edit distance and semantic features like friends information to to traing a classifier which classifies between an alias or not. Cohen et al, [7] compare different string matching algorithms for name matching tasks and found that a hybrid metrics that combines Jaro-wrinkler with TF-IDF works better in name matching tasks than other metrics like string edit distance or Jaccard coefficient etc.

Lait et al, [15] proposed Phonex, a name matching algorithm of improved version of Soundex algorithm converts each name to four character code to compare two strings achieves significant results that its predecessor Phonex algorithm. Mengmeng du, [16] made an exhaustive study of various approximate name matching methods and found that algorithms based on edit distance with a trie data structure outclasses other methods in terms of language independency and accuracy. Wei Lu et al, [17] proposed edit distance based string similarity search using B^+-tree data structure. First they split string collection into partitions and then the strings are indexed using B^+ tree based on distance of strings in the partition to the string to be compared. The constructed B+ tree then can be used to answer string similarity queries. Peter Christen, [18]made a detailed study on different approximate name matching techniques and came out that no single technique can detection all the string variant names. He found that if the name has large nicknames and name variations, then dictionary based name standardization should be applied before name matching. Elmagarmid et al, [13] done a detailed survey on duplicate record detection in database including character based, token based and phonetic based similarity metrics. While many of the string similarity metrics, approximate name matching algorithm works well in finding similarity between two names, they are not suitable for web given that sheer size of web makes it difficult to use these techniques alone in finding string variant alias names. Yancey[19] compared Jaro-Winkler with edit distance metric and found that Jaro-Winkler works well for name matching tasks for US census data. Bilenko et al., [20] compared the performance of token-based and character-based similarity metrics and found soft tf-idf performs much better than other alternatives. The detailed literature survey conducted shows that there are no single good metric and algorithm for web based name string variant name extraction and validation and hence in this work, a method is proposed to extract string variant alias name from the web. Meijuan Yin, et al,[21] proposed an alias extraction method in emails corpus that extracts aliases of sender and receiver. Their method first extracts email ids of sender and receiver, and then extracts their aliases from salutation and signature blocks using NER tools and name boundary word template.

There has been a great deal of attention towards ensemble learning from the machine learning community. The use of ensemble learner, for classification purpose is attractive area of research in recent times. Govindarajan et al, [22] used Support vector machine as base classifier with radial basis function for classification in the applications like intrusion detection, direct marketing and signature verification. This method proposed of three phase namely preprocessing, classification and combining, where an ensemble classifier is constructed by resampling and the final decision is taken by voting of classifiers. Ensemble learning algorithms perform consistently well for biological data. Ensemble of bagging, boosting performs well for gene expression data for cancer classification [23].

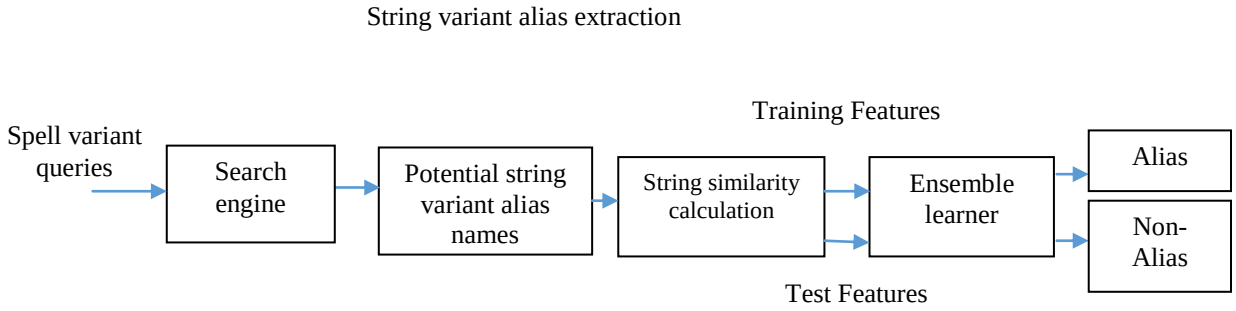

Fig.1. Outline of the proposed String variant alias extraction method

B. Motivation and Justification of the Proposed Approach

There are many algorithms and string similarity metrics available for validating similarity of two names but there are very few methods proposed by researchers for web based string variant alias name extraction and validation. Motivated by this, in this paper a new method for web based string variant alias name extraction is proposed.

There are plethora of algorithms available for approximate name matching and string similarity detection tasks. Many of these algorithms and string similarity metrics provides good results in name matching process. Most of the name errors in the web are simple errors where either a letter was deleted or added or replaced with another letter. String similarity may be the best way to find similar strings in web pages. Boosted trees and random forest performs well compared to other supervised classification algorithms on a variety of datasets [24]. Justified by this, in this paper, a novel method for string variant alias name extraction from the web using best of string similarity metrics as features for a random forest ensemble classifier is proposed.

C. Organization of the paper

The section II gives overview of the proposed method, string similarity metrics, procedure involved in extracting candidate aliases. Section III deals with experiments conducted, result and discussion. Section IV concludes the paper.

II. METHOD

A. Proposed Method

Fig 1 shows the proposed method of string variant alias name extraction process from the web. This process consists of two phases namely string variant name extraction and string variant name validation. During string variant name extraction, spell variant queries are issued to the search engine to obtain different string variants of the same name. Each name-string variant pair is then used to calculate seven different string similarity metrics which are then used as features for a trained ensemble classifier random forest. The ensemble classifier random forest then classifies whether the input string variant name is alias or non-alias.

B. Procedure for Extracting String Variant Alias from the Web

Proposing an algorithm for extracting string variant alias name is still a research problem yet to be solved. The following procedure extracts a small number of string variant alias names out of many scattered across web pages.

Step 1: Query the search engine with queries of the form "first name" –"name" Object (for example **"**Arnold sw***"* **–"** Arnold Schwarzenegger" terminator**)** to obtain 'n' number of snippets and extract nouns that occurs in the place of the pattern "sw*".

Step 2: similarly Search with the query of the form "first name" –"name" Object (**"**Ar* schwarzeneger**"** – **"**Arnold Schwarzenegger" terminator**)** and extract nouns that occurs in the place of the pattern "Ar*".

Step 3: Add the extracted strings to string variant alias name pool.

Similarly if the name contains two consecutive alphabets like in the case of Arnold Schwarzenegger, one of the alphabet is dropped and then a query like "Arnold schwarzeneger" is issued to find the number of web pages. If it surpasses the threshold, it is added to the alias pool. If the name is in the form of acronym in many number of web pages, it is also added to the alias pool.

C. String Similarity Metrics

Seven string similarity metrics were used as features to find string variant alias names. As Hamming distance can be applied only if the strings are of same length, it cannot be applied to check string variant names, as it cannot be always ensured that the primary and string variant aliases will be of same size.

1. Levenshtein Distance(or) String Edit Distance

String edit distance, [25] is the minimum number of single-character edits (insertions, deletions and substitutions) required to change name into the alias.

Levenshtein distance=

$$Min(\sum edit\ operations)\{\frac{is\ 0\ \ for\ similar\ strings}{is+ve\ \ for\ dissimilar\ strings} \quad (1)$$

2. Smith Waterman Similarity

SmithWatermanSimilarity, [26] between name and alias finds an optimal local alignment between name and alias, and returns the number of one-element matches.

$$\text{Smith Waterman similarity} = (\sum element\ matches) \quad (2)$$

Where element refers to letter in string.

3. Jaro-Winkler Distance

Jaro wrinkle distance, [27] is used to find similarity between two strings.

$$d_j = \frac{1}{3}(\frac{m}{|s1|}+\frac{m}{|s2|}+\frac{m-t}{m}) \quad (3)$$

Where, dj=Jaro Distance, m = number of matching characters, t = number of transposed characters, |s1| = length of the name and |s2| = length of the alias.

4. Jaccard Coefficient

$$J\ (A,B) = \frac{|n\ grams\ (name) \cap n\ grams\ (alias)|}{|n\ grams\ (name) + n\ grams\ (alias)|} \quad (4)$$

5. Dice Coefficient

Dice coefficient between any two words can be used in information retrieval system [28].

$$\text{Dice}(A,B) = \frac{2\times|n\ grams\ (name) \cap n\ grams\ (alias)|}{|n\ grams\ (name) + n\ grams\ (alias)|} \quad (5)$$

6. Q Gram Distance

Q gram distance, is the number of q grams that are similar between the name and alias. If q=2, then the method will return number of bi-gram words that are similar between the two strings. Elmagarmid et al,[13] used q gram distance for duplicate record detection.

$$Q\ gram\ dis\tan ce = (\sum Q\ gram\ matches) \quad (6)$$

7. Cosine Similarity

Cohen William, [7] combined cosine similarity with tf.idf similarity to compute similarity of two strings. Cosine similarity is a common vector based similarity measure, in which input strings are transformed into vector space to find the cosine of angle between them.

$$CS(A,B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (7)$$

Where A,B are input string vectors.

D. Random Forest Classifier

The idea of ensemble classifier [29], is to build different models and combining them for the better prediction .In ensemble learning, the main aim is to build an ensemble of classifiers and for classification of a new instance, their individual decisions are combined. Unlike ordinary machine learning algorithms, which construct a single hypothesis using training data, ensemble learning algorithms construct a series of hypothesis and use them for prediction by combining the hypothesis. The generalization ability of the ensemble learner will be more compared to the individual learners.

There are two types of ensemble learning methods [30].In averaging method, several models are built from the training set and their results are averaged out. Example of such learners includes bagging, random forest etc. In contrast, boosting method builds models sequentially in order to reduce models bias. Example of such learners include Boosting, Well known ensemble learning methods are boosting [31], bagging [32], voting [33], and stacking [34].

Random forest, [35] is an ensemble learning method for regression and classification. Random forest constructs a number of decision trees from the bootstrap samples drawn from the training set. Apart from this, it also introduces another randomness. During splitting of attributes, best attributes among a subset of randomly selected attributes are used for constructing decision trees. Thus it adds an additional layer of randomness compared to bagging. Finding the optimal number of trees required for the random forest is a challenging task. There is not always guarantee that, as the number of tree goes, prediction accuracy will go up. It is also equally true that beyond an optimal number of trees, a further addition to the trees would yield no improvement to the prediction accuracy [36].

The working strategy of random forest classifier is as follows.

For 1 to number of trees
    Draw a bootstrap sample of size N from dataset and construct a tree
    For each sample from the bootstrap sample
        Select a set of variables random from all available variables
        Among the selected variables pick the best split point
        Split the node into two daughter nodes
    End of for
End of for

For test instance, predict is as follows

$$\text{Regression} = \frac{1}{b}\sum_{b=1}^{B} T_b(x) \quad (8)$$

Where $T_b(x)$ is the regression value of a tree in the random forest and B is the number of trees in the forest.

$$\text{Classification} = \text{Majority vote}\left\{Prediction_x\right\}_1^B \quad (9)$$

Where B is the number of trees in the random forest.

The error rate of prediction is obtained as follows

For each iteration, predict the instances that are not used for learning (out-of-bag) using the decision tree and the averaging out of these decisions will yield Out of Bag error rate. Unlike simple decision tree, random forest trains using bootstrapping of samples from the training set. Thus by randomly sampling the instances with replacement, each decision tree is constructed and thus reduces the error rate.

III. Experiments, Results and Performance Analysis

A. Data Set

Since to the best of authors' knowledge there is no significant benchmark dataset for string variant alias name, a synthetic dataset of string variant alias name was created. The dataset set contains 30 name-alias pairs for 15 persons containing were collected from the web by manually searching with spell variant queries to the web. If the results surpasses a predefined threshold, then that does mean that such spell variant forms are prevalent in the web. Such prevalent spell variant names are then added in the dataset and are used here for the experiments. Even though other forms of string variants like acronyms of names, names separated by periods etc are also string variant forms, for this experiment these kinds of string variant forms are rarely considered as valid variant forms. It is because string similarity metrics considered for the experiments does poorly for such forms of variants. All the string variant alias names were extracted from the web in order to test the proposed method with real world dataset. Table 1 shows partial list of string variant name-alias dataset used for the experiment

Table 1. Partial list of string variant name-alias dataset.

S.No	Name	String variant alias names
1	Arnold schwarzenegger	Arnold schwarzeneger, Arnold schwazenegger,
2	Barack obama	Barrack Obama, obama
3	Sachin tendulkar	Sachin Ramesh Tendulkar, Tendulkar,
4	Mahendra Singh Dhoni	M.S Dhoni, Dhoni, Mahandra singh Dhoni
5	J. K. Rowling	Joanne "Jo" Rowling, Joanne Rowling, JKR
6.	Shah Rukh Khan	Shahrukh Khan, sharuh khan
7.	warren buffett	Waren Buffett, Warren Buffet, Warren Bufett
8.	Michael Schumacher	Michel Schumacher
9.	Sylvester stallone	Sylvester stalone, sylvesterstellone
10.	Rajinikanth	Raajinikanth

B. Performance Metric

The performance of string variant alias name process can be measured by its accuracy.

$$Accuracy = \frac{\sum_{i=0}^{n} Number\ of\ correctly\ classified\ alias}{Total\ number\ of\ aliases}$$

Where n is the number of name-alias pairs to be classified.

C. Experiments, Results and Discussion

First, candidate string variant aliases are extracted from the web for every name-alias pair in the dataset. The extracted spell variant forms of the names are added to the potential string variant alias name list. In our experiments, potential names were in unigram, mostly bigram and trigrams were potential string variant names. In the experiments, apart from spell variant forms of names, various other kinds of string variant forms like names in acronym form, shortened form of names separated by periods etc are encountered. All the variations of name extracted except duplicates were added to the potential string variant alias list. Table 2 shows the list of candidate aliases for "Mahendra Singh Dhoni".

Table 2. list of String variant alias extracted for the name Mahendra Singh Dhoni

M.S Dhoni
Mahi
Dhoni
MSD
Mahandra Singh Dhoni
MS
Mahendra Sing Dhoni
mahendra singh doni

Three letter level string similarity coefficients and four string level similarity coefficients discussed in section 2.2 are calculated for each pair of name - string variant alias in the dataset. The seven feature vectors for each pair of name-string variant alias are then input to random forest classifier to classify between alias and non-alias names. The Performance of different methods and metrics for string variant alias name detection is compared to find the accuracy of the proposed method. For this experiment, few important string similarity metrics are taken out of

large number of metrics available for the purpose of comparison. For the proposed method using random forest, 100 trees were constructed and the results were averaged out. The overall accuracy of the proposed method using random forest classifier is determined by performing 10 fold cross validation, which is the default cross validation value in weka [37] and results are tabulated in table 3.

Table 3. Performance of different methods and metrics.

Method/ Coefficient	Accuracy
Proposed method	0.80
Paul Husing using SVM classifier	0.76
Q-Gram	0.70
K-Approximate String-Matching	0.66
Cosine	0.66
Edit distance	0.60
Jaccard	0.56
Dice	0.56

It is evident from the table 3 that proposed method outperforms the existing method and metrics that are normally used for string similarity calculation. String variant alias name extraction method proposed by Paul Husing using edit and normalized string edit distance and using these as features in SVM classifier also performs well compared to other simple string similarity metrics.

It should also be noted that usage of simple metrics like string edit distances, Levenshtein distance gives good results because many string variant aliases can be detected simply by letter replacements between name and aliases.The proposed method uses random forest classifier for alias name classification. Performance of different classifiers in alias name classification is studied and is noted down in table 4.

Table 4. Accuracy of various classifiers in string variant alias detection

Classifier	Accuracy
Random forest	0.80
SVM	0.76
KNN	0.70
Logistical regression	0.70
J48	0.63
Decision table	0.56
Na ïve bayes	0.56
ZeroR	0.50

It can be inferred from the table 4 that ensemble classifier random forest better classifies string variant alias name than others i.e., Random forest classifiers identify string variant alias names more accurately than other classifiers.

IV. Conclusion and Future Work

Alias extraction involves extracting candidate aliases and validating those candidate alias names. Alias extraction can be used in natural language processing applications like Question and answering, Information retrieval, Information extraction etc. In this paper, a novel method of string variant alias extraction from the web is proposed. First, for every known personal name, candidate alias names are extracted from the web. Then, seven different string similarity metrics were calculated for every name-alias pairs, and used as features for Random forest classifier. Random forest classifier classifies every string variant alias as valid or invalid alias names. Experiments were conducted using string variant name-alias data set for 15 persons containing 30 name-alias pairs. Results shows that proposed method outperforms other existing works.

Future work includes extracting other kinds of alias names like semantic aliases. Although lot of methods are there for string variant and orthographically alias name extraction and identification, scaling them to suit the need to web corpus remains a challenging task. Usage of string variant alias names in information retrieval, information extraction are still a research problem. There many of kinds of string variant alias names like phonetic variant aliases, variations because of translations, variations because of mixing up name and middle or last name etc. Extracting all these variants from the web requires robust algorithm for extraction as well as validation. Working with non-English language for extracting string variant alias has its own challenges.

References

[1] Bollegala, Danushka, Yutaka Matsuo, and Mitsuru Ishizuka. "Automatic discovery of personal name aliases from the web." IEEE Transactions on Knowledge and Data Engineering 23, no. 6 (2011): 831-844.

[2] Hsiung, Paul, Andrew Moore, Daniel Neill, and Jeff Schneider. "Alias detection in link data sets." In *Proceedings of the International Conference on Intelligence Analysis*, vol. 4, no. 4.6. 2005.

[3] Bhat, Vinay, Tim Oates, Vishal Shanbhag, and Charles Nicholas. "Finding aliases on the web using latent semantic analysis." Data & Knowledge Engineering 49, no. 2 (2004): 129-143.

[4] Ning an, Lilli Jiang and Jianyonngwang ,"Towards detecting of alias without string similarity", Information science Journal, March 2014, Pages 89–100

[5] Ristad, Eric Sven, and Peter N. Yianilos. "Learning string-edit distance." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20, no. 5 (1998): 522-532.

[6] Navarro, Gonzalo. "A guided tour to approximate string matching." *ACM computing surveys (CSUR)* 33, no. 1 (2001): 31-88.

[7] Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records." In *Kdd workshop on data cleaning and object consolidation*, vol. 3, pp. 73-78. 2003.

[8] Jokinen, Petteri, JormaTarhio, and EskoUkkonen. "A comparison of approximate string matching algorithms." *Software: Practice and Experience* 26, no. 12 (1996): 1439-1458.

[9] Pfeifer, Ulrich, Thomas Poersch, and Norbert Fuhr. "Retrieval effectiveness of proper name search methods." *Information Processing & Management* 32, no. 6 (1996): 667-679.
[10] Angell, Richard C., George E. Freund, and Peter Willett. "Automatic spelling correction using a trigram similarity measure." *Information Processing & Management* 19, no. 4 (1983): 255-261.
[11] H.B. Newcombe, Handbook of Record Linkage. Oxford Univ. Press,1988.
[12] Lait, A. J., and Brian Randell. "An assessment of name matching algorithms." *Technical Report Series-University of Newcastle Upon Tyne Computing Science* (1996).
[13] Elmagarmid, Ahmed K., Panagiotis G. Ipeirotis, and Vassilios S. Verykios. "Duplicate record detection: A survey." *Knowledge and Data Engineering, IEEE Transactions on* 19, no. 1 (2007): 1-16.
[14] G. Salton. Automatic text transformations. In G. Salton, editor, AutomaticText Processing: The Transformation, Analysis, and Retrieval of Informationby Computer, pages 425–470. Addison-Wesley, MA, USA, 1988.
[15] Lait, A. J., and Brian Randell. "An assessment of name matching algorithms." *Technical Report Series-University of Newcastle Upon Tyne Computing Science* (1996).
[16] Du, Mengmeng. "Approximate Name Matching-Finding Similar Personal Names in Large International Name Lists."
[17] Lu, Wei, Xiaoyong Du, MariosHadjieleftheriou, and B. Ooi. "Efficiently Supporting Edit Distance based String Similarity Search Using B+-trees." (2014): 1-1.
[18] Christen, Peter. "A comparison of personal name matching: Techniques and practical issues." In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pp. 290-294. IEEE, 2006.
[19] W.E. Yancey, "Evaluating String Comparator Performance for Record Linkage," Technical Report Statistical Research Report Series RRS2005/05, US Bureau of the Census, Washington, D.C., June 2005.
[20] Bilenko, Mikhail, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. "Adaptive name matching in information integration." *IEEE Intelligent Systems* 18, no. 5 (2003): 16-23.
[21] Yin, Meijuan, Junyong Luo, Ding Cao, Xiaonan Liu, and Yongxing Tan. "User Name Alias Extraction in Emails." *International Journal of Image, Graphics and Signal Processing (IJIGSP)* 3, no. 3 (2011): 1.
[22] Govindarajan, M. "A Hybrid RBF-SVM Ensemble Approach for Data Mining Applications." *International Journal of Intelligent Systems and Applications (IJISA)* 6, no. 3 (2014): 84.
[23] Tan, Aik Choon, and David Gilbert. "Ensemble machine learning on gene expression data for cancer classification." (2003).
[24] Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." In *Proceedings of the 23rd international conference on Machine learning*, pp. 161-168. ACM, 2006.
[25] Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals." In *Soviet physics doklady*, vol. 10, no. 8, pp. 707-710. 1966.
[26] Smith, Temple F., and Michael S. Waterman. "Identification of common molecular subsequences." *Journal of molecular biology* 147, no. 1 (1981): 195-197.
[27] M.A. Jaro, "Unimatch: A Record Linkage System: User's Manual,"technical report, US Bureau of the Census, Washington, D.C., 1976.
[28] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Vol. 1. Cambridge: Cambridge university press, 2008.
[29] Dietterich, Thomas G. "Ensemble methods in machine learning." In *Multiple classifier systems*, pp. 1-15. Springer Berlin Heidelberg, 2000.
[30] http://scikit-learn.org/stable/modules/ensemble.html
[31] Schapire, Robert E., Yoav Freund, Peter Bartlett, and Wee Sun Lee. "Boosting the margin: A new explanation for the effectiveness of voting methods." *Annals of statistics* (1998): 1651-1686.
[32] Breiman, Leo. "Bagging predictors." *Machine learning* 24, no. 2 (1996): 123-140.
[33] Bauer, Eric, and Ron Kohavi. "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants." *Machine learning* 36, no. 1-2 (1999): 105-139.
[34] Džeroski, Saso, and Bernard Ženko. "Is combining classifiers with stacking better than selecting the best one?." *Machine learning* 54, no. 3 (2004): 255-273.
[35] Breiman, Leo. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.
[36] Oshiro, Thais Mayumi, Pedro Santoro Perez, and José Augusto Baranauskas. "How many trees in a random forest?." In *MLDM*, pp. 154-168. 2012.
[37] Witten, Ian H., and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

Authors' Profiles

P.Selvaperumal received his Bachelor degree(2006) in Computer science from Sacred heart college, Tirupattur and Master degree in Computer science(2009) from Bannari Amman Institute of Technology,Sathyamangalam and a second masters in technology in Veltech Technical university, Chennai. He is currently a Ph.D student pursuing Computer engineering in M.S University, Tirunelveli, Tamilnadu, India. His areas of interest include Text Mining, Machine Learning, NLP, Data science and big data exploration and Information Retrieval. He is an ACM Student member and a member of Indian society of technical education (ISTE).

A.Suruliandi received his B.E(1987) in Electronics and Communication Engineering from Coimbatore Institute of Technology- Coimbatore, Bharathiyar University, Tamilnadu, India, M.E (2000) in Computer Science and Engineering from Government College of Engineering-Tirunelveli, Manonmaniam Sundaranar University, Tamilnadu, India, Ph.D (2009) from Manonmaniam Sundaranar University as well. He started his academic career in 1987 and held his various positions in the Department of Computer Science, Kamaraj College, Tuticorin, Tamilnadu, India. Currently he is working as Professor in Department of Computer Science and Engineering Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India. He has published many papers in international journals and conferences. His research interest includes Datamining, pattern recognition, image processing, and remote sensing.

