ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2015.0148

SUPERVISED ALIAS NAME VALIDATION USING STATISTICAL SIMILARITY COEFFICIENTS

A. Suruliandi[1], P. Selvaperumal[2] and T. Dhiliphan Rajkumar[3]
Department of Computer Science and Engineering, Manonmaniam Sundaranar University, India
E-mail: [1]suruliandi@yahoo.com, [2]selvaperumal.p@gmail.com, [3]dhiipanrajkumar@gmail.com

Abstract:

Alias name is the surnames for a known name. Extracting and validating alias names is an interesting research topic in language processing and has a number of Natural language processing applications like Information extraction, Information retrieval, Sentimental analysis, Question and answering. Alias name validation involves the process of validating whether a name is alias name or not. In this work, seven statistical similarity coefficients were used as features in classifier to validate alias names. For each name-alias pair, seven statistical similarity coefficient values were calculated and used as features to train a classifier. The trained classifier is then employed to classify whether a name-alias pair is valid or not. Experiments were conducted using Indian name-alias data that has data for 15 persons containing 35 name-alias pairs. Results show that SVM classifier with Radial Basis Function Kernel outperforms all the other classifiers in terms of overall accuracy.

Keywords:

Alias name extraction, Information Extraction, Web Mining

1. INTRODUCTION

Alias name extraction and validation involves extracting all the surnames of a person from the web and validating them. In the web, these surnames are scattered in the form of standard lexical patterns like "Name aka Alias name" or as prefix or suffix pattern like "alias name Primary name", and mostly in non-standard format. Even spelling variant of a primary name is considered as alias name of the person [1]. Alias extraction is closely related to co-reference resolution, where the primary work would be to extract the co-reference chain, which refers to different expressions that refers to an entity [2]. Co-reference resolution can be extended to multiple documents which are referred as cross document co-reference resolution [3]. Since web contains billions of web pages, it makes impossible to implement co-reference resolution. Another closely related problem is known as anaphora resolution. In anaphora resolution, references of pronouns that occurs earlier or later in the discourse is resolved [4]. Web people search has been an important part of web search queries, as around 30% of queries are pertaining to personal names. Due to unstructured nature of the web documents, it has always been difficult in extracting documents of personal names. In the web, two types of name ambiguity are prevalent. A single name may be shared by multiple persons. On the other hand, a person can have more than one name. While the former is called lexical ambiguity, the latter is called referential ambiguity [5]. Solving the former is known as personal name disambiguation and solving the latter is called Alias name extraction.

Named ambiguity arises when information from multiple sources are integrated for a task. Alias names do occur in standard lexical patterns. There are many alias names that are not in standard lexical patterns and difficult to extract. The use of statistical similarity measures for alias validation is based on the notion that primary name and alias name do co-occur in web pages. While this is true in many cases, it is also equally true that alias names do not co-occur. A special case of alias name is previous name, where an entity name that was previously referred by a name is now changed to another name. For example the entity name Chennai was previously referred to as Madras. It is true that both the names of the entity co-occur in many web pages, there are web pages where they do not co-occur. In this case, before the name change Chennai was referred as Madras only and after the change of name, it was referred as Chennai. It is difficult for an Information retrieval system to identify all the alias names for an input primary name and retrieve web pages containing primary name and alias names.

Alias extraction involves two tasks namely alias extraction and alias validation. Alias extraction involves extracting alias names from the web and alias validation involves validating name-alias pairs. Alias validation is a binary classification problem where alias and non-alias are the class labels. The use of statistical similarity coefficients in validating name-alias pair is based on the notion that name and alias frequently co-occurs in a page. So their strength can be measured in terms of similarity coefficients. Based on this idea, in this work different associative measures have been used to validate alias names.

1.1 MOTIVATION AND JUSTIFICATION OF THE PROPOSED APPROACH

Statistical similarity coefficients have long been used for finding the strength between two entities. Statistical similarity metrics like Dice, Jaccard and cosine coefficients can be used as co-occurrence similarity metrics [6]. Although methods like latent semantic indexing, co-reference resolution may help in extracting alias, the sheer size of web containing billions of pages render it an infeasible job? In this scenario, researchers started focusing on robust cum feasible method for alias extraction. Statistical similarity measures provide a way to find the statistical similarity between any two entities.

Bollegala et al, [5] used 23 features including co-occurrence measures, page count based association measure and frequency of lexical pattern to extract lexically structured alias names from the web. Tomoko Hokama et al, [7] exploited prefix and suffix patterns to extract alias name of a person from the web. It is based on the notion that most of the prefix string and suffix string occurring before and after primary

name and alias name are same. Vinay Bhat et al, [8] used latent semantic analysis (LSA) for finding alias name in a text corpus. They proposed an extended two stage LSA based algorithm that extracts nearest names for an input name than LSA. Paul Hsiung et al, [1] method uses both orthographic similarity measure and semantic similarity measure to find the alias of a known name. They used string edit distance, Normalized string edit distance, discretized and exponential string edit distances as orthographic similarity measures along with semantic similarity measures as features for classification. Hsin-Hsi Chen et al, [9] compared five associative measures namely Dice, Overlap, Jaccard, Cosine and Co-occurrence double check.

Their experiments show that co-occurrence double check achieves better co-relation coefficient compared to other coefficients. William Cohen et al, [10] compared different string similarity metrics or string similarity. They found that among token based distance measure, hybrid distance that combines TF-IDF and Jaro-Winkler performs slightly better than other measures. Tarique Anwar et al, [11] used Association score, Dice score and Similarity score to find the strength of association between name and alias name. These scores were aggregated and the name-alias pair whose aggregated score was higher was considered as valid name-alias pair. Coefficient similarity measures like Jaccard coefficient [12] and Overlap coefficient [13] have been used to find the relationships between personal names on the Web.

Among the various existing statistical similarity coefficients, it is observed from the literature survey that Dice, Jaccard, PMI, Overlap, Cosine distance measures are used to find similarity between entities. It is also expected that combining these similarity metrics with Google distance and Associative score can be used to effectively validate alias names in the web. Justified by this fact, seven aforementioned statistical similarity metrics were used to validate alias names in the web.

1.2 OUTLINE OF THE PAPER

The proposed method for alias extraction using similarity coefficients is shown in the Fig.1. Query "Name" AND "Alias name" is given to search engine and snippets are retrieved. The number of web pages containing the "Primary name" and "Alias name" can be known by the hit count for the query. The count value is used to find the similarity coefficient values. The coefficient values are then used as feature vectors for training the classifier. The trained classifier is then used to classify between alias and non-alias names.

1.3 ORGANIZATION OF THE PAPER

The second section gives a detailed description of different statistical similarity coefficients and classifiers used for alia name validation. The third section describes experimental setup, experimental data, performance metric, experiments and results. Fourth section concludes the paper.

2. METHOD

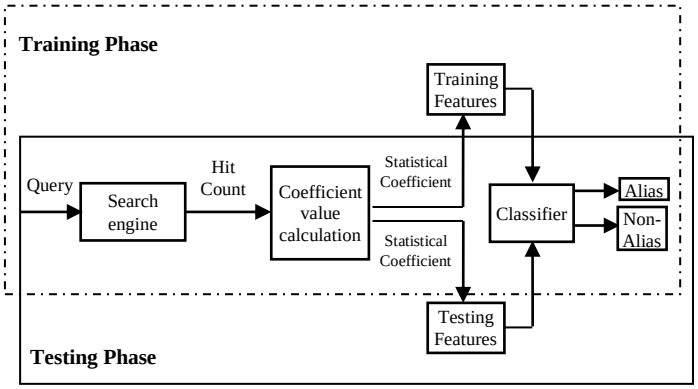

Fig.1. Outline of the proposed method

2.1 CO-OCCURRENCE SIMILARITY METRICS

Bollegala et al, [14] used Web Jaccard, web overlap, Web dice and Web Pmi coefficient similarity measures for finding semantic similarity between words. They used these coefficients along with lexical patterns extracted from web pages as features for SVM, which classifies words pairs into synonymous and non-synonymous word pairs.

2.1.1 Web Jaccard:

The Jaccard similarity, [14] is used to find binary differences between two or more objects. It is also called jaccard index. The Web Jaccard coefficient of Name and alias can be defined as,

$$\text{Web Jaccard}(name, alias) = \frac{H(name \cap alias)}{H(name) + H(alias) - H(name \cap ALIAS)} \quad (1)$$

where, $H(name \cap alias)$ refers to Hit count returned for the conjunctive query "Name" AND "Alias" given to the search engine, $H(name)$ is Hit count returned for the query "Name" and $H(alias)$ is Hit count returned for the query "alias".

2.1.2 Web Overlap:

Web Overlap is a modified form of Overlap coefficient. Web Overlap, [14] of name and alias is defined as follows,

$$\text{Web Overlap}(name, alias) = \frac{H(name \cap alias)}{\min(H(name), H(alias))} \quad (2)$$

2.1.3 Web Dice:

D. Bollegala et al, [15] used this score for calculating similarity between "name" and "alias". It can be calculated as below,

$$\text{DiceScore}(name, alias) = \frac{2 \times \text{hits}(name\ AND\ alias)}{\text{hits}(name) + \text{hits}(alias)} \quad (3)$$

2.1.4 Web PMI:

Point wise mutual information (PMI) is a coefficient inspired by information theory [16]. Web PMI, [14] of name and alias can be calculated as below,

$$\text{Web Pmi}(name, alias) = \log_2 \frac{L * Hits(\text{"}name\ AND\ alias\text{"})}{\text{Hits}(name) * \text{Hits}(alias)} \quad (4)$$

where, L is number of pages indexed in Google search engine. (It is approximately 1010).

2.1.5 Normalized Google Distance:

Cilibrasi et al, [17] proposed Google similarity distance that gives the semantic distance between any two words based on the appearance of these words in the World Wide Web.

$$NGD(name, alias) = \frac{\max\{\log f(name), \log f(alias)\} - \log f(name, alias)}{\log M - \min\{\log f(name), \log f(alias)\}} \quad (5)$$

M is the total number of web pages searched by Google. $f(name)$ and $f(alias)$ are the counts for search terms "name" and "alias" respectively. $f(name, alias)$ is the number of web pages found on which both "name" and "alias" occur.

2.1.6 Cosine:

Cosine angle between any two objects gives the degree of similarity between those objects in a vector space.

$$\text{Cosine}(name, alias) = \frac{H(name \cap alias)}{\sqrt{(H(name), H(alias))}} \quad (6)$$

2.1.7 Associative score:

Tariq anwar et al, [11] used Associative score as one of three score used to validate alias names. AssoScore is calculated by summing up of multiplied values of f-score pattern value in Table.1 with the number of times that frequency occurs when issuing appropriate query.

$$\text{AssoScore} = \sum_i (F - Score(P_i) \times freq(P_i)) \quad (7)$$

where, P_i is the pattern with i values ranging from 1 to 5 as in Table.1. The value obtained is then normalized to the range of 0 to 1.

Table.1. F-Scores for various patterns [15]

Pattern Id	Pattern	F-Score
P1	SearchQuery(realName, aka, *)	0.335
P2	SearchQuery(*, aka, realName)	0.322
P3	SearchQuery(realName, better known as, *)	0.310
P4	SearchQuery(realName, alias, *)	0.286
P5	SearchQuery(realName, also known as, *)	0.281

2.2 CLASSIFIERS

2.2.1 Naïve Bayes:

Naive Bayes, [18] is based on Bayes rule and it assumes that attributes are independent of each other.The working principle of naïve Bayes classifier is as follows:

1. **Training step:** Using the training data, the method estimates the parameters of a probability distribution, assuming predictors are conditionally independent given the class.
2. **Prediction step:** For any unseen test data, the method computes the posterior probability of that sample belonging to each class. The method then classifies the test data according the largest posterior probability.

2.2.2 SVM:

SVM is non-probabilistic binary linear classifier considered to be the most robust and accurate.It has a sound theoretical foundation, requires only a dozen examples for training, and is insensitive to the number of dimensions [19]. SVM constructs hyperplanes in Multidimensional space which is used to classify dataset.

The working principle of SVM classifier is as follows:

1. Project the training data set in feature space.
2. Project the testing data set in the same features space.
3. Find a hyperplane such that it should maximize the distance between the closest data point.

2.2.3 KNN Classifier:

KNN classifier, [19] is a lazy learner algorithm, which memorizes the entire training data and performs classification of the test data based on the training data. The Working principle of KNN classifier is given below:

1. Extract Training features from the training data set and train the classifier.
2. Extract Testing features from the testing data set.
3. Find the distance of each test sample with k-nearest training samples.
4. Assign the class label for test sample, the class labels of majority of k-nearest neighbors.

2.2.4 Decision tree – C 4.5:

Weka provides J48 a java implementation of C 4.5 decision tree [20]. C 4.5 is an extension of earlier ID3 algorithm. The Working principle of C 4.5 Decision tree classifier is given below:

1. If all the data set belongs to the same class, then tree is labeled with that class.
2. If there are more than one class, calculate the normalized information gain for each attribute for best splitting
3. Create a decision node that best splits attribute
4. Repeat the above steps to construct the tree until no more attribute is left to divide.

2.2.5 Logistical regression:

Logistical regression, [21] is a probabilistic classification model that can be used to classify data. Logistical regression can be binomial or multinomial where, binominal can be for binary classification problems and multinomial for multi class problems.

The Working principle of Logistical regression classifier is given below:

1. From the training data, a logistic regression probability model is first estimated that is composed of a list of vectors and their corresponding categories.

2. Test data values are coded as dense vector instances, and the outputs are a category number with a probability. The conditional probability about how likely the class fits is then calculated.

2.2.6 Simple perceptron

Perceptron, [20] is a supervised linear classification algorithm that works online. Thus instead of considering the entire dataset, it operates one data after another.

1. Initialize the weights and threshold to small random numbers
2. For each training sample, compute output and update weights.
3. Repeat the above steps for all the training data set.

3. EXPERIMENTAL RESULTS AND DISCUSSION

3.1 DATA SET

Since there is no standard name-alias data set for Indian names to the best of authors' knowledge, a list of Indian name-alias pair were collected from the web with the help of informative web pages like Wikipedia. The information is collected from the web, keeping in view that if the alias name is not prevalent in the web, then even a robust alias extraction method may fail to extract it. For this work, fifteen Indians containing 35 name-alias pairs were collected from the web and were used as data set for the experiments. The Table.2 shows list of Indian name-alias data set used for the experiments.

Table.2. Indian name-alias data set

Sl. No	Name	Alias Names
1.	Rajinikanth	Shivaji Rao Gaekwad, Thalaivar, Superstar
2.	Sachin Tendulkar	Sachin Ramesh Tendulkar(SRT), The Little Master, Master Blaster.
3.	Sonia Gandhi	Antonia Maino, Sonia Maino, Madam
4.	Mother Teresa	Agnes *Gonxha* Bojaxhiu, The Saint of the Gutters, Teresa of Calcutta
5.	Mahendra Singh Dhoni	Mahi, MS Dhoni, Captain Cool
6.	Mahesh Babu	Mahesh, Prince, Superstar
7.	Viswanathan Anand	Vishy, lightning kid
8.	Abhishek Bachchan	Junior B
9.	Narendra Modi	NaMo
10.	P. T. Usha	Golden Women
11.	Mamata Banerjee	Didi
12.	Salman Khan	Sallubhai, Chulbul Pandey, Abdul Rashid Salim Salman
13.	Saurav Ganguly	Dada, Bengal tiger, Prince of Kolkata
14.	Virender Sehwag	Nawab of Najafgarh, viru
15.	Mahatma Gandhi	The Father of the Nation, Bapu, Mohandas Karamchand Gandhi

3.2 PERFORMANCE METRICS

The efficiency of a search engine can well be evaluated using precision and recall [22]. Each statistical similarity coefficient is measured in terms of precision, recall and f-score.

$$\text{Overall Accuracy} = \sum_{i=0}^{n} \frac{\text{Number of Alias Validated}}{\text{Total number of Aliases in the dataset}} \quad (8)$$

3.3 EXPERIMENTAL RESULTS AND ANALYSIS

For each name-alias pair in the data set, appropriate query to calculate statistical similarity coefficient is given to the Google search engine. For example queries "Primary name", "Alias name", "Primary name" AND "Alias name" is issued to search engine and the hit counts that refers to the number of web pages containing the appropriate word is obtained. A word inside the quotes refers to the fact that we need an exact match of the primary or alias name and not the spelling variant names.

The number of results returned by the search engine for queries (Hit count) is then used to calculate seven similarity coefficients values. The calculated seven similarity coefficient values are then used as feature vector for the subsequent classification process. The performance of different classifiers in alias name validation is then compared. The Table.3 shows the Hit count for queries used to calculate coefficient similarity values.

Table.3. Hit counts returned by Google search engine for "sachin Tendulkar" and "Master blaster"queries

Sl. No	Query	Hit Count
1.	"Sachin tendulkar"	1,47,00,000
2.	"Sachin tendulkar" AND "master blaster"	3,61,000
3.	"Master blaster"	6,78,000

It is evident from the hit count that primary name (Sachin Tendulkar) occurs in majority of web pages compared to alias names. Alias name (Master blaster) occurs in relatively less pages than primary name. Primary name and alias name co-occurs in relatively lesser web pages than primary and alias names alone. The hit count then used to calculate seven different similarity coefficients for each name-alias pair. Similarity coefficients for the name-alias pair "Rajinikanth-Superstar" calculated is tabulated in Table.4. These similarity coefficients serves as feature vectors for alias names and are fed to classifier for alias name validation.

Table.4. various similarity coefficients values between Rajinikanth and Superstar

Sl. No	Method	Similarity coefficient
1.	Web Jaccard	0.01
2.	Web Overlap	0.11
3.	Web Dice	0.01
4.	Web PMI	0.93
5.	Normalized Google Distance	0.70
6.	Cosine	0.02
7.	AssoScore	0.63

The Seven similarity coefficients were used as feature vectors for each name-alias pair and are given as test data to a trained classifier. Then the performance of each classifier in classifying name-alias pair is noted. The performance of different classifiers in classifying between alias and non-alias names is tabulated in Table.5.

Table.5. Average co-occurrence similarity values for name-alias data set

Sl. No	Classifier	Number of correctly classified aliases	Number of Misclassified aliases	Overall Accuracy
1.	SVM	21	14	0.60
2.	Logistic regression	20	15	0.57
3.	Decision tree	20	15	0.57
4.	K-Nearest neighbour	18	17	0.51
5.	Naïve bayes	16	19	0.45
6.	Simpler perceptron	15	20	0.42

The output of the classifier is discrete values i.e “alias” or “non-alias”. The results show that, Support vector machine classifier achieves higher overall accuracy rate compare to logistical regression and decision tree. The accuracy of logistical regression and decision tree is more or less similar.

In order to further evaluate the efficiency of SVM classifier, the experiment was repeated by changing the kernels of SVM classifier. The number of alias names correctly validated using different kernels of SVM classifier is shown in the Fig.2. The performance accuracy of RBF Kernel was 0.60, linear and polynomial kernel is 0.57, and sigmoid kernel was 0.51.

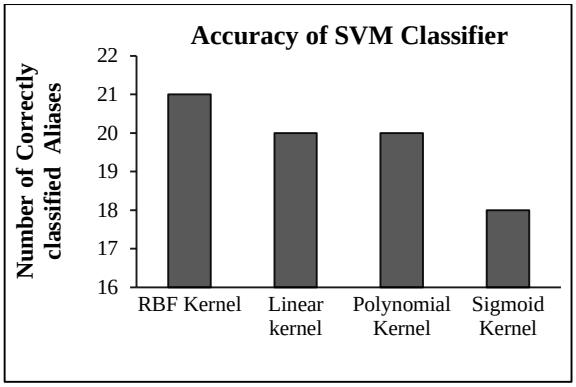

Fig.2. Performance comparison of SVM Kernels

The Table.6 shows the validated name-alias pairs using SVM classifier. Since majority of name-alias pairs in the name-alias data set are in lexically structured format in web pages, the accuracy of these classifiers is good. It should be noted that many alias names in the web pages are not lexically structured and are difficult to extract.

All the alias names need not co-occur with primary names. This means that co-occurrence similarity measures alone are not sufficient enough to validate name-alias pairs.

4. CONCLUSION

Extracting alias name for a known name is an interesting problem in Information Extraction. In this work, seven different statistical similarity coefficients were used as features in classifier to validate alias names. Experiments were conducted using Indian name-alias data set for fifteen Indian names containing 35 name-alias pairs. Results show that Support vector Machine classifier achieves high precision rate compared to other classifiers. Since a name-alias pair may not always co-occur in a web page, usage of other metrics along with this is advisable. An interesting future extension of this work includes changing features for classification and proposing techniques for alias extraction. The use of alias names in Question and answering, Sentimental analysis and in many other Natural language processing tasks is also interesting.

Table.6. Classified alias names by the SVM classifier with RBF kernel

Sl. No	Name	Correctly classified Aliases	Incorrectly classified Aliases
1.	Rajinikanth	Shivaji Rao Gaekwad, Superstar	Thalaivar
2.	Sachin Tendulkar	Sachin Ramesh Tendulkar, The Little Master,	Master Blaster
3.	Sonia Gandhi	Antonia Maino, Madam	Sonia Maino
4.	Mother Teresa	Agnes Gonxha*Bojaxhiu,The Saint of the Gutters*	Teresa of Calcutta

5.	Mahendra Singh Dhoni	Mahi, MS Dhoni	Captain Cool
6.	Mahesh Babu	Prince , Superstar	Mahesh
7.	Viswanathan Anand	-	Vishy, lightning kid
8.	Abhishek Bachchan	-	Junior B
9.	Narendra Modi	-	NaMo
10.	P. T. Usha	Golden Women	
11.	Mamata Banerjee	Didi	
12.	Salman Khan	Sallubhai,Chulbul Pandey	Abdul RashidSalim Salman
13.	Saurav Ganguly	Dada, Bengal tiger	Prince of Kolkata
14.	Virender Sehwag	viru	Nawab of Najafgarh
15.	Mahatma Gandhi	The Father of the Nation, Mohandas Karamchand Gandhi	Bapu

REFERENCES

[1] Paul Hsiung, Andrew Moore, Daniel Neill and Jeff Schneider, "Alias detection in link data sets", *Proceedings of the International Conference on Intelligence Analysis*, 2005.

[2] Wee Meng Soon, Hwee Tou Ng and Daniel Chung Yong Lim, "A machine learning approach to coreference resolution of noun phrases", *Association for Computational Linguistics*, Vol. 27, No. 4, pp. 521-544, 2001.

[3] James Mayfield, David Alexander, Bonnie J. Dorr, Jason Eisner, Tamer Elsayed, Tim Finin, Clayton Fink et al. "Cross-Document Coreference Resolution: A Key Technology for Learning by Reading", *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pp. 65-70, 2009.

[4] Shalom Lappin and Herbert J. Leass, "An algorithm for pronominal anaphora resolution", *Association for Computational linguistics*, Vol. 20, No. 4, pp. 535-561, 1994.

[5] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka, "Automatic discovery of personal name aliases from the web", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 6, pp. 831-844, 2011.

[6] Christopher D. Manning and Hinrich Schutze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 1999.

[7] Tomoko Hokama and Hiroyuki Kitagawa, "Extracting mnemonic names of people from the web", *Digital Libraries: Achievements, Challenges and Opportunities: Lecture Notes in Computer Science*, Vol. 4312, pp. 121-130, 2006.

[8] Vinay Bhat, Tim Oates, Vishal Shanbhag and Charles Nicholas, "Finding aliases on the web using latent semantic analysis", *Data & Knowledge Engineering*, Vol. 49, No. 2, pp. 129-143, 2004.

[9] Hsin-Hsi Chen, Ming-Shun Lin and Yu-Chuan Wei, "Novel association measures using web search with double checking", *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 1009-1016, 2006.

[10] William W. Cohen, Pradeep Ravikumar and Stephen E. Fienberg, "A comparison of string metrics for matching names and records", *KDD Workshop On Data Cleaning And Object Consolidation*, Vol. 3, pp. 73-78, 2003.

[11] Tarique Anwar, Muhammad Abulaish and Khaled Alghathbar, "Web content mining for alias identification: A first step towards suspect tracking", *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, pp. 195-197, 2011.

[12] Peter Mika, "Bootstrapping the foaf-web: An experiment in social networking network mining", *Proceedings of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[13] Yutaka Matsuo, Junichiro Mori, Masahiro Hamasaki, Takuichi Nishimura, Hideaki Takeda, Koiti Hasida and Mitsuru Ishizuka, "POLYPHONET: an advanced social network extraction system from the web", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 4, pp. 262-278, 2006.

[14] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka, "A web search engine-based approach to measure semantic similarity between words", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 7, pp. 977-990, 2011.

[15] Danushka Bollegala, Yutaka Matsuo, Taiki Honma and Mitsuru Ishizuka, "Identification of personal name aliases on the web", *Proceedings of 17th International Conference on World Wide Web*, 2008.

[16] Kenneth Ward Church and Patrick Hanks, "Word association norms, mutual information, and lexicography", *Computational linguistics*, Vol. 16, No. 1, pp. 22-29, 1990.

[17] Rudi L Cilibrasi and Paul MB Vitanyi, "The Google similarity distance", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 3, pp. 370-383, 2007.

[18] http://in.mathworks.com/help/stats/naive-bayes-classification.html

[19] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan et al. "Top 10 algorithms in data mining", *Knowledge and Information Systems*, Vol. 14, No. 1, pp. 1-37, 2008.

[20] Ian H Witten and Eibe Frank, "*Data Mining: Practical Machine Learning Tools and Techniques*", Morgan Kaufmann, 2005.

[21] Ning An, Lilli Jiang, Jianyonng Wang, Ping Luoe, Min Wange and Bing Nan Li, "Towards detecting of alias without string similarity", *Information Sciences*, Vol. 261, pp. 89-100, 2014.

[22] W. Bruce Croft, Donald Metzler and Trevor Strohman, "*Search engines: Information retrieval in practice*", Addison-Wesley, 2010.