International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 6 (2015) pp. 15631-15641

http://www.ripublication.com

An Optimized Attribute Based Similarity Search In Metric Database

C. Swaraj Paul[1*], G.Gunasekaran[2]
[1]Research Scholar, Department of Computer Science and Engineering, St.Peter's University, Avadi, Chennai, India, email: swarajpaulresearch@gmail.com,*
[2]Principal, Meenakshi College of Engineering, Chennai, India, email:gunaguru@yahoo.com.

Abstract

Most of the research papers studied the structured data extracted from the web page templates. These templates are having most of the structured data on a web page. Web pages and data based on the internet are increasing very speedily. It is easy to judge about the pattern of a website according to the nature of the website developed and it is very difficult to find the exact similar data from the web. This paper provides a best solution for searching the similar data in two ways. First, the similarity data are clustered and classified using the **CSA-** [Clonal Selection Algorithm] according to the attributes priority. Second, a **FANN** – [Fast Artificial Neural Network] is applied for fast learning on data using the index of the metric data, which enables the Metasearch engine to handle efficiently the query terms within a short time. Also, it returns the relevant query results speedily with high accuracy due to the data learning method. The simulation results are experimenting with time series data in MATLAB 2012 software and evaluated by comparing the results with the existing metaheuristic algorithms

Keyword: Web Searching, Web Mining, Metaheuristic Searching, Data Mining, Data pattern.

Introduction

Currently, huge data are increasing in terms of number, size and the importance. So, it is necessary to provide computer processing in terms of both hardware and software, which can provide effective mean for analyzing the scalable data. Consequently, today we have reasons and means for efficiently analyzing huge data of considerable importance with a low cost. KDD-[Knowledge Discovery in Database] in data mining is useful in finding knowledge corresponding with the resource. It is well known that the internet and the World Wide Web are successfully running in the past 10 years.

Presently there are more than 900 million web pages in the web [1]. To improve the data mining efficiently various search engines have been developed. All the web pages are arranged using the index and keywords. Using the keywords, the search engine can retrieve the web pages accurately according to the query words. Searching the web is the largest new industry on the internet and some of them are Yahoo, Alta Vista, Excite and Infoseek [2]. Data mining refers to the process of finding interesting patterns in data, that are not explicitly part of the data (Witten & Frank, 2005, p. xxiii). The interesting patterns can be used to tell us something new and to make predictions. The process of data mining is composed of several steps including selecting data to analyze, preparing the data, applying the data mining algorithms and then interpreting and evaluating the results. Sometimes the term data mining refers to the step in which the data mining algorithms are applied. The first step in data mining is data cleaning, or pre-processing. All input data must meet certain conditions to ensure optimal performance including:

1. The data must be in a usable form.
2. There must be sufficient data to derive a solution.

KDD tasks were proposed to identify a novel, potentially useful and the knowledge [3]. Various similar search engines to KDD are proposing has novelty and interestingness, but they cannot satisfy fully [4]. Some of the visual mining approaches are also used [5, 6] are having visual techniques can improve the efficiency of the mining. Active mining approaches have been developed [7, 8] as automatic methods can be appended by visual methods effectively. Some of the machine learning approaches, such as hidden Markov model is also used for clustering based data extraction [9]. A metadata based target file searching is introduced for document searching in web [10]. A new binarization algorithm is used for document retrieval using historical information about the document [11]. The binarization algorithm is also used for image retrieval by comparing the pixels in the images [12]. In World Wide Web, all the individual elements such as html forms [13], tables [14, 16], are extracted using structured and highly structured Meta search engine tools [15].

Existing Approaches

Some of the specific approaches used in earlier researches are given below and are [17]. ***Artificial neural networks***-Non-linear predictive models that learn through training and resemble biological neural networks in structure. ***Genetic algorithms***-Optimization techniques that use process, such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution. ***Decision trees***-Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. ***Nearest neighbor method***-A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k 1). Sometimes it called the k-nearest neighbor technique.

Neural Network

A neural network based classification algorithm was proposed to decide the percentage of the relevant content in the web pages and reorder them appropriately [2]. Only the top most, less percentage of the results of this approach are more relevant. Also the most relevant web pages taken from various standard search engines are more similar to each other than to irrelevant web pages and vice versa. This approach is the core of the Anvish search engine. The entire content of the two web page comparison may take too much of the time. So in this paper, the titles and the summaries of two web pages are compared and analyzed to find out the similarities and return less results. To overcome these issues in the existing system an Artificial Neural Network based similarity search engine is proposed in this paper.

Neural network works in considered in two steps, the generalization ability of the NN and the fastness of the training process. In this paper, due to increase the speed, CC4 NN was chosen to process the searching results. The CC4 algorithm is a new type of corner classification training algorithm for three-layered feed forward NNs.

Materials and Methods

There are two methods is used in this paper, one is Clonal selection and other one is AFNN. For experimenting and evaluating the proposed approach FANN tool (18) is used and it have several features than other searching tools. It has inbuilt neural network libraries for evaluating the search engine process. The data used in this paper for the experiment is a metric data, file named as SunSpotsRaw-1980-2006.txt in ASCII format. The data represent the monthly mean value of the sunspots from January 1980 to October 2006. The total number of data in the file is 322. In the experiment, it is aimed to predict the past values of the sunspots recorded. In the initial step, the data file is fed into the FANN Tool. The data is processed and predict the results using FANN approach and is described fully below.

Fast Artificial Neural Network Tool

FANN is a free open source tool having NN library, which implements multilayer based ANN using C language. This supports the fully and partially connected networks. FANN is generally called as NN and it is a mathematical model, which tries to simulate the structural and functional characteristics of biological NNs. ANN performs a non-parametric, non-linear, multivariate with multiple regression. The FANN works in cross platform execution in both fixed and floating point modes. FANN has its own framework to handle the training data sets easily. It is an easy usable software and fast. The step by step process followed by FANN is:

- Prepare the data in such a way that the FANN library can understand
- Design an ANN with relevant inputs
- Train the ANN which designed already
- Test the trained ANN
- Run the trained ANN

Prepare the Data in FANN

Choose the appropriate data file, such as "data1.txt" from the data folder. This file consists of monthly mean value of the sunspots. In this experiment, it is aimed to predict the past values of the sunspots recorded and the file in the form of ASCII. Now the uploaded data file is processed and various kinds of reports are generated. One is Time series Based Data Processing, where this operation provides a time series based plot for the mean value of the sunspots in terms of time. The data processing reads the ASCII raw data and determine the scaling value of the data. In this metric data, the range of the data is available within the range of (0, 1) or (-1, 1), which can be applied to any real time data scaling process. The entire data is categorized as incremental based, batch based, property based, quick property based and so on. The necessary settings given in FANN, cascade tuning is given in Table-1.

Table 1: FANN Parameter Settings

Maximum Number of Neurons	10
Fraction to be changed in the output	0.01
Epochs	12
Weight Multiplier	0.4
Limit of Candidate	1000
Candidate per groups	2 to 4
Hidden Layer	2
Training Method	Batch Based
Activation Function	Linear Regression
Number of Input dim	6
Minimum Output Value	1
Maximum Output Value	1
Fraction of data for training	70%
Fraction of data for testing	30%
Shuffle	Scrambling the sequence

Scale:

For This Example:

min = 0.0, max = 200.3 (estimated from the data)
(Desired) Minimum Output Value = 0
(Desired) Maximum Output Value = 1
Scaled Value = (Input Value - Minimum) / (Maximum - Minimum)

For This Example:

Scaled Value = (Input Value – 0,0) / (200.3 - 0.0)
And the reverse way (de scaling):
Input Value = (Scaled Value *(200.3 - 0.0)) + 0.0

In the *Time series Data Processing* dialog you can switch between two types of graphs Fig.2 and Fig.3 and the format of the input and output files are given in Table-2:

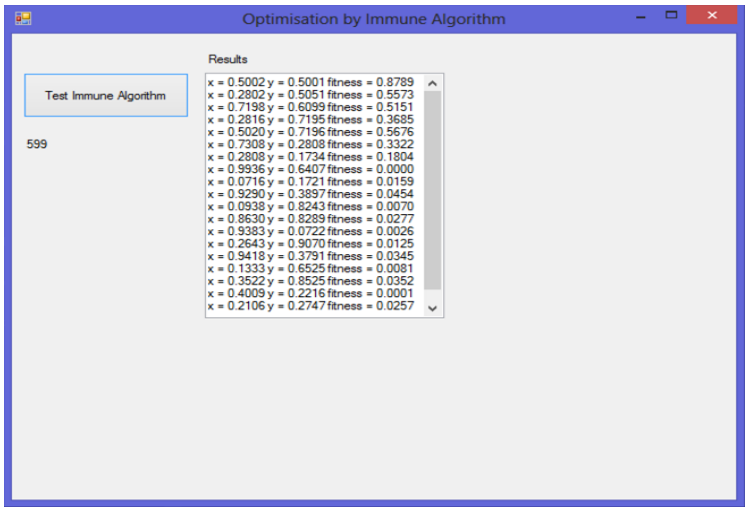
The format of the data is in the form of

Table 2: The Data processed format in FANN

Num_ Train_ data	Num_ Input_ Nodes	Num_ Output_ Nodes
Num_ Input_ Nodes	Input Data	
Num_ Output_ Nodes	Output Data	
:	:	:
Num_ Input_ Nodes	Input Data	
Num_ Output_ Nodes	Output Data	

Experiments and Results

The experiment is carried out in two stages. In the first stage, the data is preprocessed like check the irrelevant data and normalized. After normalization, the data is fed as input into the Clonal selection algorithm for clustering and classifying the data for fast analysis. The Clonal selection algorithm chooses the best data within a threshold assigned as the objective function value using a fitness function. If the satisfies the fitness condition then it will be added into the cluster. Else the clones are expanded by randomly selecting new dynamic populations. The Clonal selection algorithm is implemented in C#.net language, and the clustered immunes fits within the fitness function is shown in Fig.2.



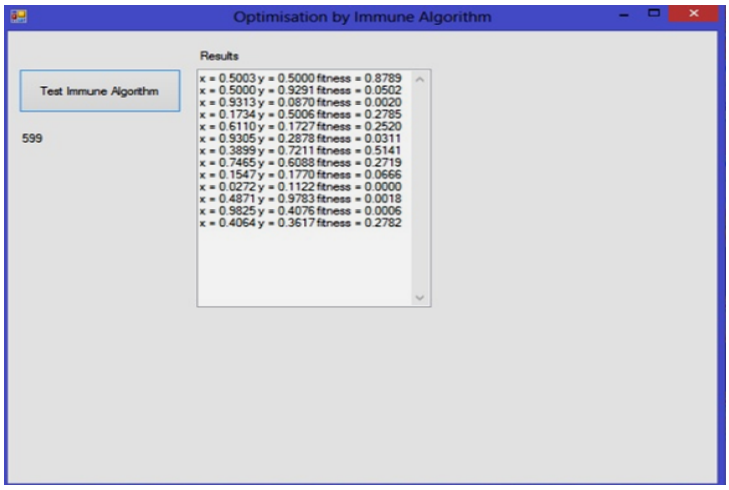


Figure 1: Clonal Selection Based Clustering and Classification

The scalar value of the entire data is fetched and analyzed to improve, due to the efficiency of the search engine in term of accuracy and relevant data retrieval. The scalar data threshold value is assigned in the training data set for batch wise comparison with linear regression. During FANN process the layers compare all the data with the threshold value for each batch and group the data for classification. According to this process, the entire scalar data is processed from 0.6 to 21.0 and it is plotted as a histogram, is shown in Fig.3.

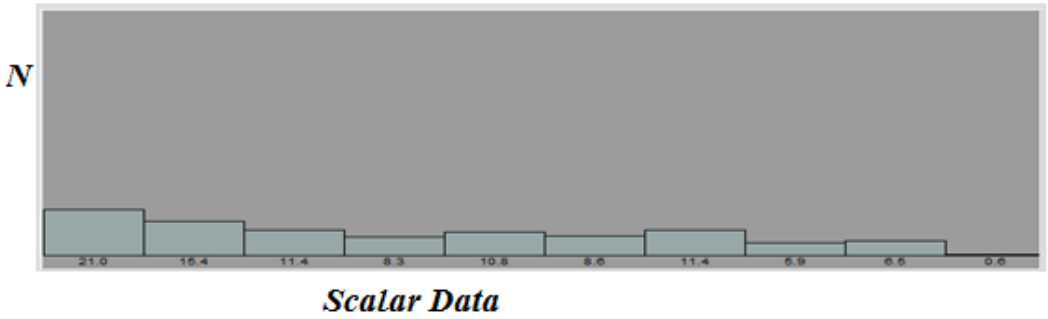


Figure 2: Histogram [N-Number of Data]

The scalar value is considered and fetched in terms their vector values,(ie) the data are time series data, the scalar data can be represented in terms of time and date. The time series data are uncertain data and it cannot be defined. The scalar data are represented in terms of time is shown in Fig.4 as X-Y plot.

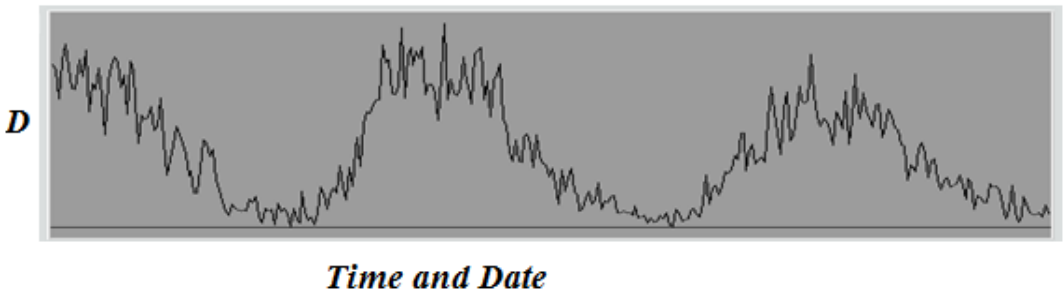


Figure 3: X-Y Plot[D – Data]

Neural Network Setup for the sunspot time series data is given below as a sequence of steps:

Step1: Load the training data and testing data

Step2: Verify the Number of nodes [Dimension] for Input and output shown in the GUI

Step3: Number of Hidden Layers to be specified and it may be from 3 to 5. Where the number of layers = input(1) + Hidden (1 to 3) + output(1), so that the minimum 3 layers with 1 hidden layer and a maximum of 5 layers with 3 hidden layers.

Hidden layer 1: # of neurons in the 1st hidden layer

vHidden layer 2: # of neurons in the 2nd hidden layer ($\#\, \boldsymbol{of\ Layer} \geq \mathbf{4}$)

Hidden layer 3: # of neurons in the 3rd hidden layer ($\#\, \boldsymbol{of\ Layer} = \mathbf{5}$)

Training Method chosen as the FANN_TRAIN_BATCH method. With all setting given in AFNN tool to configure the artificial neural network with appropriate libraries involved in predicting the past data from the given data set.

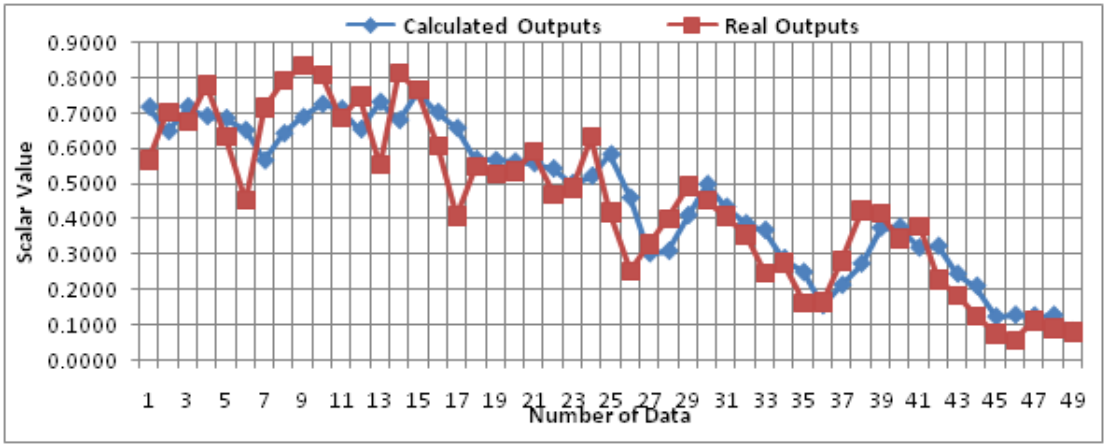


Figure 4: Predicted Data Results

Figure-5 shows that, the predicted data from the real time data [Cricket - data] is obtained using our FANN tool and the true data already available. The comparison result shows that the experimented result merely equal to the true value and it prove it

is better. The existing true values are compared with the output of the implementation. The predicted data size is much better than the true data. The effective retrieval of similarity data from the core data storage is continuously researched by past 30 years. So many measures are used to find out the effectiveness of the information retrieval in terms of similarity. The scalar data retrieval and the vector relation with the scalar is shown in Fig.3 and Fig.4. The FANN tool is implemented for the uncertain data to fetch the similarity. Based on the implementation experiments, our approach works well on the numerical data it is due to the applied ontological relationship.

Performance Evaluation

The experiment is repeated for various data sets taken from UCI repository and the results are compared to evaluate the performance of the proposed approach. The comparison result is shown in Table-2 and in Fig.6, which proves that the proposed approach is more efficient than the existing approaches.

Table 3: Performance Analysis

Data	Available Data	Retrieved Data
Tennis Major Tournament Match Statistics	127	100
Wholesale customers	440	387
Time series data	500	498

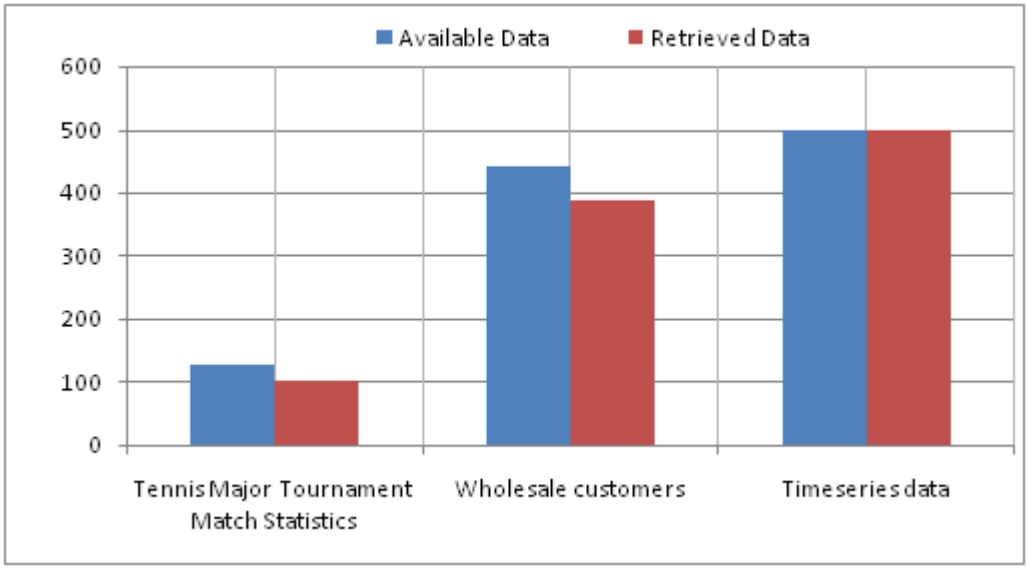


Figure 5: Performance Analysis

From Table-2 and Fig.6, it is clear that, there are three data sets are experimenting using our proposed approach. The number of data in all the three data sets are 127, 440 and 500. After preprocessing, normalization, clustering and classification, then

FANN based retrieved data are 100, 387 and 498 respectively from all the three data sets. Similarly the proposed approach is executed on the web data and the results are obtained to evaluate the performance. Instead of the metric data the web data are taken and a keywords is applied as the query word. In our experiment, any query related to metric data services, such as Hockey Data, 20-20 News Group Data, Cricket Data, and Breast cancer data. The search results for the query which we sent are classified as the number of relevance's and number of irrelevance by the FANN tool. The classification accuracy is given in Table-2, Table-3, Table-4 and Fig.6, Fig.7.

Table 4: Efficiency In terms of Time

Approach	**Search Time [in Seconds]**			
	Anvish	**Yahoo**	**Web Crawler**	**Meta Crawler**
Existing System	6	2	5	4
Proposed system	5	1.23	2.34	3

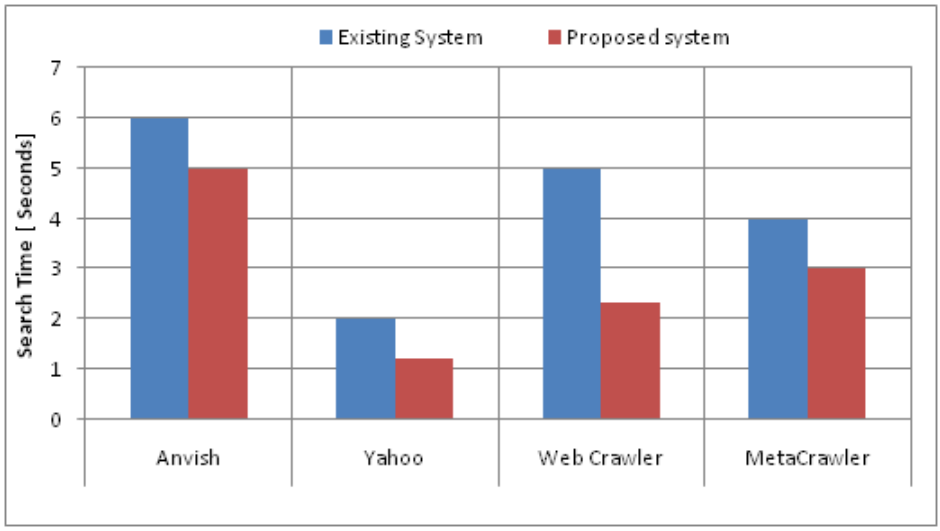


Figure 6: Efficiency In Terms of Time

According to the input query, the time taken to search the entire data is calculated for the existing approach and the proposed approach is given in Table-3 and in Figure-7. Our approach takes less time than the existing approaches.

Table 5: Efficiency In terms of Relevant Retrieval

Approaches	Number of Web pages	Correctly classified	In-Correctly Classified	Accuracy
Existing System	50	48	2	96%
Proposed System	50	41	9	82%

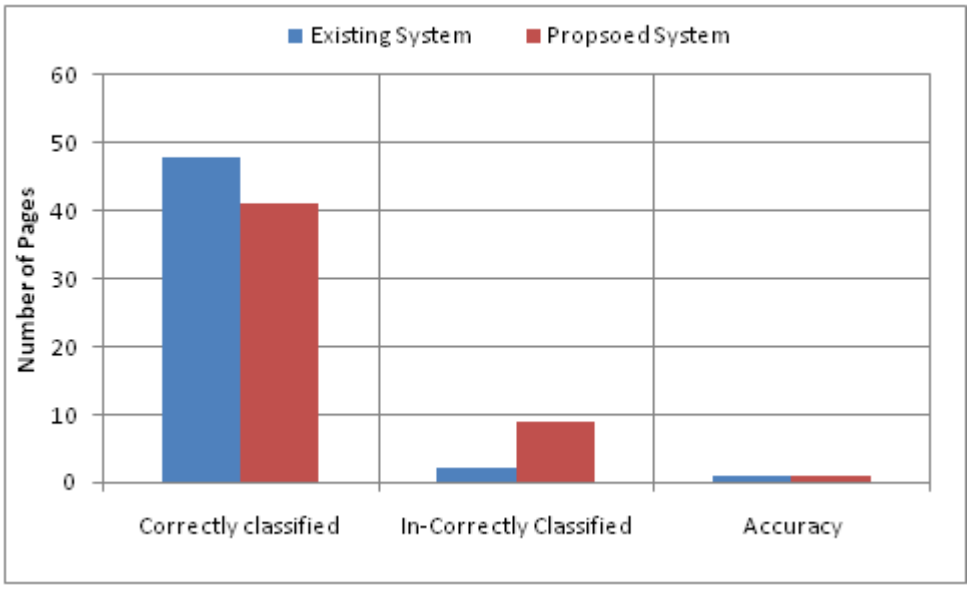


Figure 7: Efficiency In Terms Of Relevant Retrieval

Conclusion

In this paper, it is aimed to design and develop a framework for obtaining the similarity content from a huge set of data set. There are two main approaches are used for obtaining the objective of this paper. A Clonal selection algorithm is used for classifying the similar data as groups and FANN tool is used for similarity data retrieval and the results obtained. From the tables and Figures, it is clear that our proposed approach proves its efficiency in terms of similar data classification and retrieval within a stipulated time interval.

References

[1] S. Lawrence, C. Lee Giles, Accessibility of information on the Web, Nature 400 (1999) 107-09.

[2] Bo Shu, Subhash Kak, “A neural network-based intelligent metasearch engine”, Information Sciences, Vol 120, Nov (1999).

[3] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padraic Smyth, “FromData Mining to Knowledge Discovery: An Overview”, Advances in Knowledge Discoveryand Data Mining, pages 1–34. AAAI Press, 1996.

[4] Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher, “Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery”, IJCAI-05, pages 647–652, 2005.

[5] Willi Kl¨osgen and Stephan R. W. Lauer, “Handbook of Data Mining and Knowledge Discovery”, Visualization of Data Mining Results. Oxford University Press, New York, 2002.

[6] Graham J. Wills and Daniel Keim, “Handbook of Data Mining and Knowledge Discovery”, chapter 15.1: Interactive Statistical Graphics. Oxford University Press, New York, 2002.

[7] Hiroshi Motoda, “Active Mining-A Spiral Model of Knowledge Discovery”, IEEE Intl. Conference on Data Mining, Maebashi City, Japan, 2002.

[8] Dragan Gamberger, Nada Lavrac, and Dietrich Wettschereck, “Subgroupvisualization: A method and application in population screening”, IDAMAP-2002.

[9] Caillet, M. Pessiot, J. Amini, M. Gallinari, “Unsupervised learning with term clustering for thematic segmentation of texts”; Proceedings of Rercherche d’Information Assiste par Ordinateur (2004), Avignon, France, 1-11.

[10] Ramalho, Jose C, Ferreira, Miguel, Faria, Luıs, “Relational Database Preservation through XML modeling”, In Proceedings of Extreme Markup Languages 2007.

[11] Vonikakis, V., Andreadis, I., and Papamarkos, "Robust Document Binarization with OFF Center-surround Cells", -Pattern Analysis & Applications, to appear - 2008.

[12] Lins, R. D..da Silva, J. M. M, “A Quantitative Method for Assessing Algorithms to Remove Back-to-Front Interference in Documents”; SAC’07, 2007, Seoul, Korea, pp.610-616.

[13] Madhavan, J., Jeffery, S.R., Cohen, S., "Web-scale Data Integration : You can only afford to Pay As You Go", -CIDR -2007.

[14] Cafarella, M.J., Halevy, A., Wang, Z.D. And Wu.E, "WebTables : Exploring the Power of Tables on the Web", International Conference in VLDB-2008.

[15] Elmeleegy, H., Madhavan, J. And Halevy, “Harvesting relational tables from lists on the Web"; International Conference on Very Large Data Bases (VLDB), 20, 2, (2009), 209–226.

[16] Madhavan, J. And Halevy.A, "Harnessing the Deep Web: Present and Future", In Biennial Conf. on Innovative Data Sys. Res-CIDR -2009.

[17] S.Shahar Banu, V.Saravanan, R. Shriram, “Extracting Peculiar Data from Multidatabases Using Agent Mining”, IJRTE-ISSN-2277-3878: 2013.