International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600

http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

FRAMEWORK FOR LOW- HIGH INTRA CLUSTERING MEASURING COMMON WEIGHTED SIMILARITIES

B. Sundaramurthy1, P.K. Kumaresan2

1 Associate Professor, Department of Computer Science and Engineering, Vinayaka Missions Kirupananda Variyar Engineering College, a constituent College of Vinayaka Missions Research Foundation Deemed to be University, Salem.

2 Professor & Vice Principal, Department of Computer Science and Engineering, Vinayaka Missions Kirupananda Variyar Engineering College, a constituent College of Vinayaka Missions Research Foundation Deemed to be University, Salem.

Abstract- Distance functions like Euclidian, Manhattan etc. are the common traditions to measure the similarities between numeric values. Various text similarity techniques like Cosine similarity, Dice similarity etc. are used to measure similarities between text values. But generally an object is consisting of set of attributes of different data types. Clustering is a technique of creating group of similar objects. There are number of techniques available to measure the similarities between the objects. So measuring the similarity between two objects requires the similarity measurement of different data types which requires the combination of similarity measurement techniques. Also some attributes may be more relevant and some attributes may be less relevant for object similarities between the objects for clustering purpose. So similarity weights can be assigned for each pair of attributes between the objects to effectively measure the object similarities. In this paper a framework is proposed to measure the weighted similarities between the objects consist of attributes of different data types. The proposed framework is implemented using the open source technologies and results are also explained with the help of illustrative examples.

1. INTRODUCTION

Clustering is a technique of creating group of similar objects. An Object is a set of attributes. Attributes of the objects can be composed of different data types. The data types could be numeric (integer, double etc.), text and date etc. Good clusters have the following two properties:

1. Low inter-cluster similarity
2. High intra-cluster similarity

So creating good clusters from a set of objects required the object similarity calculations. For calculating the similarity between the objects, the similarities between the attributes of the objects are calculated. For different data types there are different similarity techniques. Following are the standard similarity measurement techniques for the different data types:

1.1 Similarities for Numeric Values

Similarity between the numeric values can be measured using distance functions. To compute whether a set of points are close enough to be considered a cluster, we need a distance measure - D(x, y). The usual axioms for a distance measure D are:

- D(x, x) = 0
- D(x, y) = D(y, x)
- D(x, y) <= D(x, z) + D(z, y)

Assume an N-dimensional Euclidean space, the distance between two points, $a=(a_1,a_2,...,a_n)$ and $b=(b_1,b_2,...,b_n)$ may be defined using one of the measures:

An important step in any clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. These distances (similarities) can be based on a single dimension or multiple dimensions, with each dimension representing a rule or condition for grouping objects. The various kinds of distances measurement techniques are:

1.1.1 Minkowski distance

The Minkowski distance between point $a=(a_1,a_2,...,a_n)$ and point $b=(b_1,b_2,...,b_n)$ is

$$d(a,b) = \sqrt[q]{\textstyle\sum_{i=1}^{n}(b_i -}$$

1.1.2 The Manhattan distance

Also known as city-block distance, this distance measurement is especially relevant for discrete data sets. While the Euclidean distance corresponds to the length of the shortest path between two samples, the Manhattan distance refers to the sum of distances along each dimension.

Manhattan distance is a special form of Minkowski distance, by putting q=1 in the formula of Minkowski distance, it can be calculated. The Manhattan distance between point $a=(a_1,a_2,...,a_n)$ and point $b=(b_1,b_2,...,b_n)$ is $d(a,b) = \sum_{i=1}^{n}|b_i$

1.1.3 The Euclidean distance

The Euclidean distance between point $a=(a_1,a_2,...,a_n)$ and point $b=(b_1,b_2,...,b_n)$ is

$$d(a,b) = \sqrt{\textstyle\sum_{i=1}^{n}(b_i -}$$

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600

http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

1.1.4 Pearson Correlation distance

This distance is based on the Pearson correlation coefficient that is calculated from the sample values and their standard deviations. The correlation coefficient 'r' takes values from –1 (large, negative correlation) to +1 (large, positive correlation). Effectively, the Pearson distance dp is computed as dp = 1 - r and lies between 0 (when correlation coefficient is +1, i.e. the two samples are most similar) and 2 (when correlation coefficient is -1).

1.2 Weighted Distances

Weighted distances are different from the normal distances. In weighted distances functions weights are assigned for each pair of values (attributes) between two objects. The weighted Minkowski distance between point $a=(a_1,a_2,...,a_n)$ and point $b=(b_1,b_2,...,b_n)$ is

$$d(a,b) = \sqrt[q]{\sum_{i=1}^{n}(b_i - a_i)} \quad .$$

1.3 String Similarity and Methods

There are number of string similarity algorithms present today, some of the most commonly used are discussed here, they can be used for calculating the similarity of text data [4].

1.3.1 Dice Similarity

The Dice Coefficient is a word-based similarity measure. The similarity value is related to a ratio of the number of common words for both sentences and the number of total words of the two sentences.

When comparing two sentences Q and S, if N_{common} is the count of common words, N_Q is the total count of words of sentence Q, and N_S is the total count of words of sentence S, the Dice coefficient can be expressed as follows [9].

$$Dice(Q,S) = \frac{(2*N_{common})}{(N_Q + N_S)}$$

In the Vector space model, documents (S) and queries (Q) are decomposed into smaller word units. All words are used as elements in the vectors that will represent Q and S. Both vectors contain weights assigned to each word corresponding to the number of occurrence of that word within them.

1.3.2 Cosine Similarity

Cosine similarity measurement is very common way of calculating corpus based sentences or strings similarity. The formula is given in following equation (Salton et al., 1983 [11]) for t words.

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600

http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

$$COSINE(Q,S) = \frac{\sum_{k=1}^{t}(w_{qk} \cdot w_{sk})}{\sqrt{\sum_{k=1}^{t}(w_{qk})^2 \cdot \sum_{k=1}^{t}(w_{sk})^2}}$$

Where w_{qk} and w_{sk} are the words presents in sentences Q and S.

1.3.3 BLEU Similarity

BLEU is a method for automatic evaluation of machine translation. We use this algorithm to rank similar sentences by comparing the input sentence Q and only one reference sentence S. The implementation is based on the following formula:

$$\text{Log BLEU} = \min(1- r/c, 0) + \sum_{n=1}^{N} w_n \log p_n$$

1.3.4 TF-IDF Similarity

Term Frequency (TF) and Inverse Domain Frequency (IDF) are very common factors today to calculate the similarities among the strings. Both are used mostly in text mining to discover the similarities. Product of both gives the similarity of a text in a document.

Where D is the document and frequency(t) is the frequency of a word in a document.

$$\text{TF-IDF Sim} = \log\left(\frac{|D|}{\text{document - frequency(t)}}\right)$$

1.3.5 Jaccard Similarity

This is another token based vector space similarity measure like the cosine distance and the matching coefficient. Jaccard Similarity uses word sets from the comparison instances to evaluate similarity. The Jaccard similarity penalizes a small number of shared entries (as a portion of all non-zero entries) more than the Dice coefficient. The Jaccard similarity is frequently used as a similarity measure for chemical compounds. Each instance is represented as a Jaccard vector similarity function. The Jaccard between two vectors X and Y is

Jaccard Sim = (X*Y) / (|X||Y|-(X*Y))

Where (X*Y) is the inner product of X and Y, and |X| = (X*X)^1/2, i.e. the Euclidean norm of X. This can more easily be described as (|X & Y|) / (| X or Y |)

1.4 Similarity for Date Attributes

Date attributes can be transformed to the numeric values and then similarity techniques for the numeric attributes can be applied. There are number of ways by which the date value can be transformed to the numeric values. One technique is counting the number of milliseconds after 1st January, 1900 for a given date, that number will represent the given date.

1.5 Normalization Techniques

Normalization is used to transform the values to scale to fall within a small, specified range. There are number of normalization techniques available [12], some of the common normalization techniques which are included in the framework implementation are mentioned here:

1.5.1 Min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Where v' is the new value, v is the older value. min_A , max_A are the older minimum and maximum values and new_min_A, new_max_A are the new minimum and new maximum values for the normalization.

1.5.2 Z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

Where v' is the new value, v is the older value. $mean_A$ and $stand\_dev_A$ are the mean and standard deviation of the value sets.

1.5.3 Normalization by decimal scaling

Where v' is the new value, v is the older value. Where j is the smallest integer such that Max $(|v'|) < 1$

$$v' = \frac{v}{10^j}$$

This paper is organized in the following sections. Section 2 represents the background and related work done in the similar field. Section 3 depicts the proposed prediction model. Section 4 and 5 presents the implementation of proposed model and results of the implementation. Section 6 is the conclusion and future scope of the proposed work.

2. FRAMEWORK

The proposed framework is studied and mentioned. Research area where the similar logical model is used is also mentioned in framework for computing semantic similarity between objects represented as arbitrary RDF graphs, based on similarities of specified object properties. In the proposed framework three kinds of concept matching techniques were implemented: the string matcher, the numeric matcher, and the taxonomic matcher [8]. The problem of finding objects in an image that are similar to a given query object.

Machine learning techniques can retrieve more objects that are similar to the query than distance-based similarity methods [1]. The compression-based, parameter-free, similarity distance measures. They have considered two types of objects literal objects and objects which have literal embodiments. For the first type families of computable distance measures corresponding to parameters expressing similarity according to particular features between pairs of literal objects have been studied. For the second type similarity distances generated by web users corresponding to particular semantic relations between the designated objects have been studied. [6]

A framework named similarity spreading, to include interrelationship between objects and improve the similarity calculation.[2] Similarity assessment that allow comparing two differently structured objects. They also analyzes several situations in which class hierarchies are used in different ways for case modeling and proposed a technique for specifying similarity measures for comparing arbitrary objects.

Semantic proximity to address the dual schematic and semantic perspectives by enumerating possible semantic similarities between objects having schema and data conflicts. [9]

A Common weighted similarity model to capture the duplicate and similar software bug is a software bug repository. Theoretical framework to study and extract the properties of a similarity function which are sufficient for clustering.[3]

3. PROPOSED TECHNIQUE

Various similarity techniques can be used based on the data types to measure the similarities. The three major categories of the data types are:

Number Data Type – It includes the integer, double, float numbers etc. the normal distances techniques like Euclidian, Manhattan etc. can be used directly to measure the distance between the numeric data type.

String Data Type - Text similarity techniques can be used to measure the distance between the texts attributes.

Date Data Type – To measure the distance between the date data type, first the granular level for the date is set in order to make the uniformity for distance similarity calculations.

The various granular levels for the date data type are: Hours, Days, Weeks, Months, and

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

Years etc. Once granular level of the date data type are set, the date values are converted to these granular level values and then normal distance techniques like Euclidian, Manhattan etc can be applied. Similarities between two objects can be expressed using eq. (1).

$$Sim_{Object} = W_1 * Sim_{A1} + W_2 * Sim_{A2} + ...+ W_N * Sim_{AN} \quad (1)$$

Where Sim_{Object} is the overall object similarity between a pair of objects. W_i is the similarity weight for the attribute pair Ai for two objects and Sim_{Ai} is the similarity value between the attribute pair.

4. FRAMEWORK FOR THE PROPOSED TECHNIQUE

The proposed framework concept can be explained with the help of Figure-1. An object is consisting of number of attributes. In the given diagram it is assumed that object is consist of N number of attributes. Pair wise attributes are matched against the similarities based on their data types and each similarity value is assigned some similarity weight denoted by W_i. The overall similarity is the sum of all the pair wise similarities values among the attribute sets.

The proposed framework is implemented using the open source technologies like Java, Java Reflection API, and Java Swing etc.

A stemming algorithm is a process of linguistic normalization, in which the variant forms of a word are reduced to a common form.

An open source stemmer named Xapian is used for normalization and cleaning of the text. Xapian stemmer is based on the Snowball Stemming Algorithms. [11]

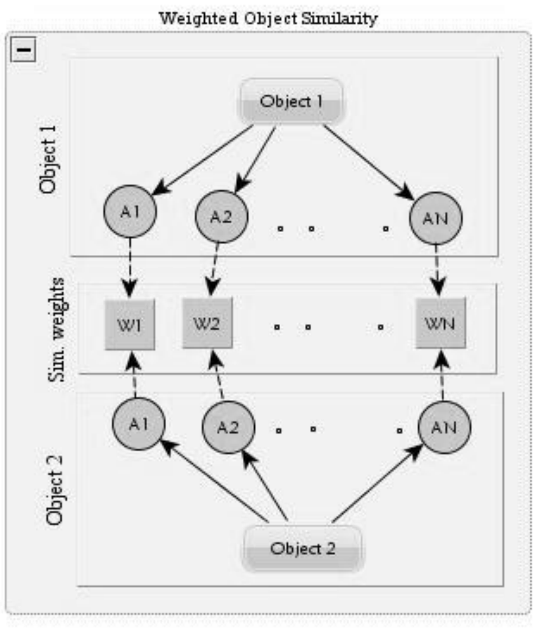


Figure 1. Weighted object similarity between two objects.

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

The process of the proposed technique and various modules for the proposed framework are depicted in figure-2.

The object retriever, the first module of the framework fetches the objects from the object source.

The object source could be either a database, a flat file or it can be a web data source consist of online object repositories. Once all the objects are retrieved at the local machine, it enters into the object preprocessor phase, which is the second module for the proposed framework.

In this module all the attribute values based on their data types are cleaned and prepared for similarity measurement.

For example number data type attributes are normalized, string data type is cleaned using stemmer program etc.

After the attribute preparation and cleaning weights are assigned for each attribute pair for the similarity, also similarity techniques are selected for attribute similarity in the next module which is called specify weight and similarity technique module.

At last the similarities among the attributes are calculated using the similarity calculation techniques and the overall similarity between the objects are calculated in the last module.

Normalization techniques are also applied to measure the similarity in the interval [0…1].

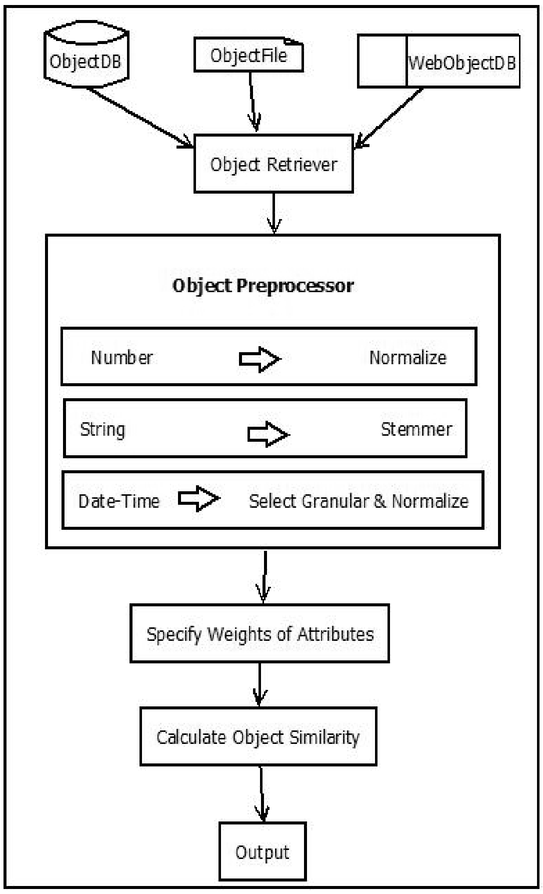


Figure 2. Various stages in OSim framework.

The main GUI (Graphical User Interface) for the proposed OSim framework is depicted in figure-3. The framework implementation is divided into the four modules, using the main GUI of the OSim framework any of the modules can be

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600

http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

selected for performing the operation. Names and responsibilities of each module are:

1. Fetch Objects – This module is responsible for retrieving the objects from object source or repository to the local machine.

2. Preprocess Objects – In this module the attributes of object based on their data types are pre-processed and cleaned for the similarity measurement.

3. Specify weights and distance techniques – Using this module a user can specify the weights for the different attribute pair similarity and user can also specify the distance techniques for measuring the similarities.

4. Calculate Similarity – This section calculated the overall similarities for a given object pair and applies the normalization technique to measure the similarity in a specific interval.

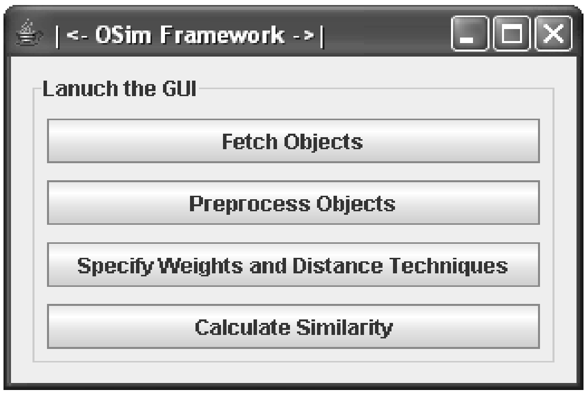


Figure 3. Main GUI for OSim framework.

The GUI of the object retriever module implementation is depicted in figure-4. This is the important module of the proposed framework.

This module fetches the objects from a specified object repository. The object repository could be a database, a flat file system consists of serialized objects or it could be web based online object repository source. The implementation supports all three types of behavior to extract objects from these repositories by three separate interfaces as shown in figure.

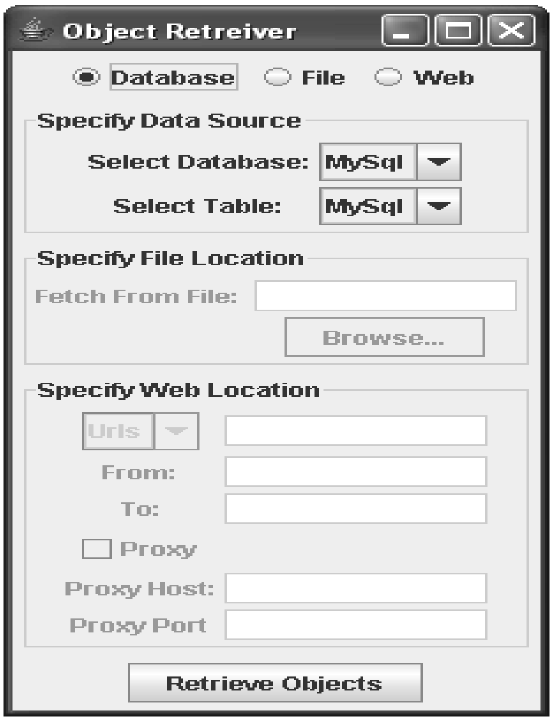
If the object repository source is web based online repository then user can also specify the proxy IP (internet protocol) address and port number if required for the internet connection.

The object preprocessing implementation GUI for the proposed framework is shown in Figure-5. In this module all the attributes and their data-types are extracted from an object and then they can be taken for preprocessing.

Based on the data-types of the attributes the type of attributes the preprocessing techniques can be selected. For date data type, one of granular level of the date representation can be selected using this GUI to make the date type attribute similarity uniformity.

A date value can be represented at various granular levels e.g. day, week, month, year etc. So a common representation technique is selected for similarity measurement. String data can be pre-processed using stemmer program.

In the proposed framework Xapian's open source stemmer [11] is selected for implementation.



Object Retreiver

[✓ Database] File Web

Specify Data Source

Select Database:	MySql
Select Table:	MySql

Specify File Location

Fetch From File:

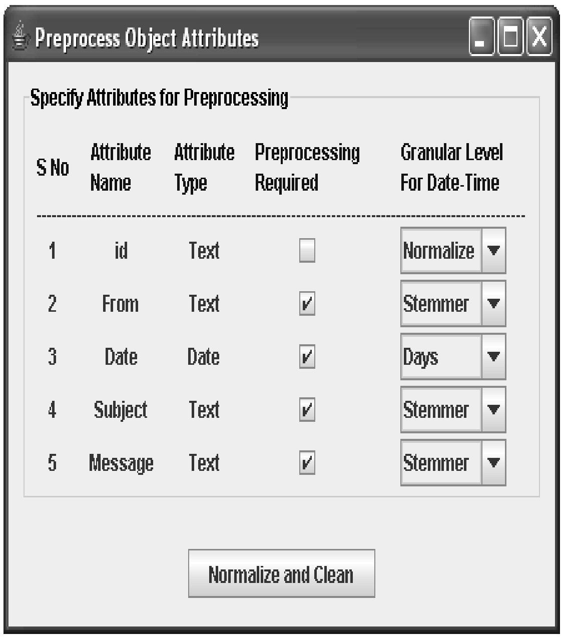
Browse...

Specify Web Location

Urls	
From:	
To:	
Proxy	
Proxy Host:	
Proxy Port	

Retrieve Objects

Figure 4. Retrieve objects module GUI.



Preprocess Object Attributes

Specify Attributes for Preprocessing

S No	Attribute Name	Attribute Type	Preprocessing Required	Granular Level For Date-Time
1	id	Text	☐	Normalize
2	From	Text	✓	Stemmer
3	Date	Date	✓	Days
4	Subject	Text	✓	Stemmer
5	Message	Text	✓	Stemmer

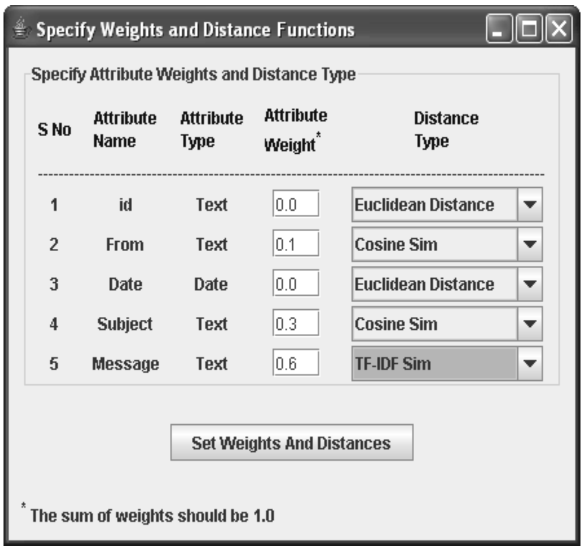
Normalize and Clean

Figure 5. Objects preprocessing GUI.

Figure-6 depicts the GUI for the specify weights and similarity functions for the attributes. Using this GUI the weights can be assigned for attribute pair similarities and also the distance functions based on the data types of attributes can be selected to measuring the similarities. In the figure the attributes of an email object is shown to measure the similarities among the email objects. The data types of attributes and their pair wise weights along with the distance techniques are also shown in the figure. The sum of the weights should be one to normalize the similarity values.

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 6, Issue 2, No 1, 2018. ISSN: 2348-6600
Reference ID: IJCS-346 PAGE NO: 2311-2323

Figure-7 depicts the result GUI for the OSim framework. Here two object-id's for which similarities is required are mentioned and their corresponding similarity is calculated and shown to the user.

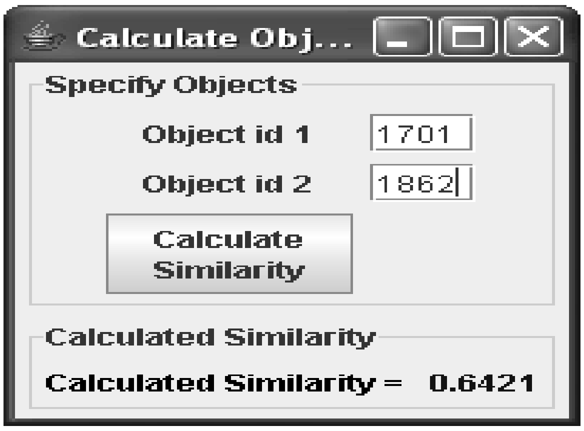

Specify Weights and Distance Functions

Specify Attribute Weights and Distance Type

S No	Attribute Name	Attribute Type	Attribute Weight*	Distance Type
1	id	Text	0.0	Euclidean Distance
2	From	Text	0.1	Cosine Sim
3	Date	Date	0.0	Euclidean Distance
4	Subject	Text	0.3	Cosine Sim
5	Message	Text	0.6	TF-IDF Sim

* The sum of weights should be 1.0

Figure 6. Specify attribute weights for similarity using GUI.


Calculate Obj...

Specify Objects

Object id 1: 1701

Object id 2: 1862

Calculated Similarity

Calculated Similarity = 0.6421

Figure 7. Osim result GUI.

5. APPLICATION OF THE FRAMEWORK

In this section some of the research area where the proposed framework can be used is mentioned. Two such recent research topics from data mining are selected as examples where the proposed framework can be applied for discovering the useful knowledge patterns.

6.1 Email Mining

Email mining includes email documents classification, clustering, emails thread summarization etc. An email can be represented as an object with various attributes like from-id, to-id, cc-id, bcc-id, subject, message etc. This framework can be used for measuring the accurate similarities between two email documents. If user wants that only subject and message should be considered for the email distance measurement then weights should be assigned to these two attributes only for the remaining attributes the weight assigned will be zero. In this way the proposed framework provides a flexible way to measure the similarity between email documents.

6.2 Software Repository Mining

Software repository mining includes applying data mining techniques over source code repository, project related documents, design documents etc. In case of source code mining of object oriented projects, a class can be represented as an object the various fields of the class can be represented as attributes of the class object. The user can specify the weights to different attribute pair to measure the similarities among the different classes represented as an object.

6. CONCLUSION AND FUTURE SCOPE

A novel framework is proposed in this paper to measure the similarities between the objects. There are number of applications where the proposed framework can be effectively used. Some of the recent research areas are also mentioned in this paper. The framework proposed is generic in nature and is there is the scope of future expansion of future requirements. Semantic similarity measures can also be included as the future scope of the proposed work. Using semantic similarity techniques, the contextual and meaningful similarities between the objects can also be calculated.

7. REFERENCES

[1] Cantu-Paz, E., Cheung, S-C., and Kamath, C., "Retrieval of Similar Objects in Simulation Data Using Machine Learning Techniques," Image Processing: Algorithms and Systems III, SPIE Volume 5298, pp 251-258. SPIE Electronic Imaging, San Jose, January 2004. UCRL-JC-153866

[2] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen,Wei-Ying Ma,Yong Yu , Similarity spreading: a unified framework for similarity calculation of interrelated objects, Pages: 460 - 461 , 2004, ISBN:1-58113-912-8 , International World Wide Web Conference archive, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters table of contents

[3] Maria-Florina Balcan, Avrim Blum, Santosh Vempala, "A Discriminative Framework for Clustering via Similarity Functions", Proceedings of the 40th annual ACM symposium on Theory of computing , Victoria, British Columbia, Canada , Pages 671-680 , Year of Publication: 2008

[4] Naresh Kumar Nagwani, Pradeep Singh, "Weight Similarity Measurement Model Based, Object Oriented Approach for Bug Databases Mining to Detect Similar and Duplicate bugs", Proceedings of the International Conference on Advances in Computing, Communication and Control , Mumbai, India , Pages 202-207 ,ICAC 2009.

[5] Ralph Bergmann and Armin Stahl, Similarity Measures for Object-Oriented Case Representations, Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning, Pages: 25 - 36 , Year of Publication: 1998.

[6] R. Cilibrasi, P.M.B. Vitanyi, Similarity of objects and the meaning of words, Proc. 3rd Conf. Theory and Applications of Models of Computation (TAMC), J.-Y. Cai, S. B. Cooper, and A. Li (Eds.), Lecture Notes in Computer Science, Vol. 3959, Springer-Verlag, Berlin, 2006, 21--45.

[7] R.L. Cilibrasi and P.M.B. Vitányi. The Google similarity distance.IEEE Trans. Knowledge and Data Engineering, 19(3):370–383, 2007.Preliminary version: Automatic meaning discovery using Google, http://xxx.lanl.gov/abs/cs.CL/0412098 (2004)
[8] Radoslaw Oldakowski, Christian Bizer, SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs, In Poster at the 4th International Semantic Web Conference (ISWC 2005), 2005.
[9] Vipul Kashyap Amit Sheth, Semantic and Schematic Similarities between Database Objects: A Context based approach, September , The VLDB Journal — The International Journal on Very Large Data Bases archive, Volume 5 , Issue 4 (December 1996) , Pages: 276 - 304 , Year of Publication: 1996
[10] Java: http://www.java.sun.com
[11] Xapian stemmer: http://xapian.org/docs/stemming.html
[12] Jiawei Han and Micheline Kamber: “Data Mining: Concepts and Techniques”,2006, ISBN 1-55860-901-6.