***The Republic of Iraq***

***Ministry of higher education and scientific research***

***University of Al-Qadisiyah***

***College of Computer Science and Information Technology***

***Computer Science Department***

***Web User Profiling in java Using Semantic Matching***

***A report submitted to the department of computer science as part of the requirements for obtaining a bachelor's degree in computer science and information technology / computer department***

**2017 2018**

**Set by:**

**Nijod Faisal Abd Al-Sattar**

**Haneen Thamir Kadhim**

**Supervisor:**

**Dr. Muntasir Jaber Jawad**

***بســــــم الله الرحمن الرحيــــــم***

 ***"اقرأ بأسم ربك الذي خلق خلق الأنسان من علق اقرأ وربك الأكرم الذي علم بالقلم علم الأنسان مالم يعلم***

***صدق الله العلي العظيـــــــم***

***"اهداء"***

***جل وعلا شأنه امرنا بالقراءة واكدها مرتين. الاولى انارت الطريق والثانية لأناره الطريق للأخرين. انا اقرأ وهم يحموني ويعلموني الحشد الشعبي المقدس واستاذي الدكتور منتصر جابر جواد***

# **Chapter one**

# **Introduction**

# **1.1-XML**

XML is a shortcut to (extensible markup language). It designed for transmitting data and stored. Some believe that XML used to display the data as in HTML but not, and is the markup language general to create language coding with your purpose have the ability to describe a number of different types of data that means it's way to describe the data as in the data base. XML is used as a format to store and process documents that are connected and not connected to the internet.

Currently there are two version of XML:

The first version is XML (1.0), which appeared in 1998 and is now in it's fourth version, which appeared in 2006 and it widely used.

The second version is XML (1.1), which appeared in 2004 and is now in it's second version, which appeared in 2006 and it is not widely used:

Characteristics of the XML which makes the language of appropriate for data transfer.

1-The formula humanity and read automatically.

2-Her support for characters international standard system that allows for any information in any language written connect.

3-Its ability to represent the most common computer science data structure such as list and tree.

4-It has a self-documentation format that describes the structure, domain names, and assigned values. [1]

# **1.2-Matching**

Matching: is the process of discovering mapping between two graphs through the application of matching algorithm, there are two types of matching: [7]

# **1.2.1-Exact Matching**

the exact matching method is for improving the estimation of causal effects by reducing imbalance in covariates between treated and control groups. The exact method is faster and easier to used and understand, requires fewer assumption, more easily automated, and processes more attractive statistical properties for many applications than existing matching methods. In exact matching, user temporarily coarsen their data, exact match on these coarsened data then run their analysis on the coarsened, it bounds the degree of model dependence and causa effect estimation error by extant user choice, it is reducing the maximum imbalance on one variables has not effect on others, doesn't require a separate procedure to restrict data to common support meets the congruence principle, is approximately invariant to measurement error and it is enable to balances all nonlinearities and it is interaction in simple and can works with multiply imputed data set. In the other type of matching methods inhered many of CEM's properties when applied to further match data preprocessed by exact matching. [2]

# **1.2.2-Semantic Matching**

Semantic matching is a new approach and discussed some of it's key properties. For performing generic matching. We search for semantic correspond by mapping meaning (concepts), and not labels, as in syntactic matching. When we match two nodes, it is not sufficient to consider the meanings of labels of this node, but also need the positions of the nodes in the graph. The semantic is the similarity relations between elements (concepts) rather than the syntactic similarity. Compute element-level semantic matching for each node, compute semantic relation holding among all concepts denoted by labels at nodes under consideration. [7][8]

# **1.3-User Profiling**

The user profile is represented by using XML language. We represent it using XML because it has many features in representation profile information. Such as data and documents represented in XML can be processed with different types of applications, that makes it more applicable switch and sharing data or documents between components (application, database).

The user profile contains both general user information that is applicable over all applications and more unstable application particular data the user profile data is approved against an XML pattern to secure the integrity of the data. User profile is showing personal data associated with specific user or customized desktop environment. A profile indicates therefore to the explicit digital representation of a person's identity. A user profile also can be considered as the computer representation of user form. A profile can be used to store description the characteristics of a person. This information can be used to by systems taking into account the persons characteristics and preferences. [6]

# **Chapter two**

# **Literature review**

# **Related works**

# **1-According to the work of "Anna Formica":**

The method that aimed at finding the best matches between a user request and service offered by several enterprises, that suggest a given business ecosystem (for example, the tourism sector) a group of SMEs (SMEs it is represent the backbone of the European industrial system) agree on the adoption of reference ontology. It is used to build the company profiles depending on the services provided. Thus; a user gives set of desired features the represent user required, is expressed in terms of the reference ontology terminology (concepts). Explanation of SemSim, a method used to collectively search the SME profiles to identify the services that match at better the user required. SemSim is the approach that used to evaluated the semantic similarity among concepts depending on the well-known information content. The similarity assessment is analyzed by studying the correlation among the selected similarity methods and human judgment. The correlation reflects the noisiness in the linear relationship among a human judgment and for instance semsim values that essentially means that higher scores on HJ tend to be paired with higher scores on semsim, analogously for lower scores. [3]

# **2- According to the work of "Changbo ke":**

We depend a web service modeling ontology discovering frame work. Through the proposing conception similarity and structure similarity based on taxonomic and a hierarchical methodology, we avoid a complex logic reasoning effectively and by defining a serial of restructuring constraints according to the relationship among two similarities and restructuring. With development of web service standard and a maturity of platform that support the web service development that leads to make the web service a major software paradigm and computing the resource. To describe web service by web service description language that based on XML. It includes grammatical compatibility but lacks the consideration of semantic information. In addition, the service registration and discovery mechanism is based on the global description of its discovery and generation and supports grammatical operation This level has two double points of this service to describe the service cannot describe the service more accurately, on the other hand, the services are obtained only by the presence of a corresponding word in the process of discovery of the service and therefore difficult to satisfy the user requirements in the functional and non-functional Internet services And be unable to detect the services of the agent or user accurately but the features of ontology web service, semantic web service it's smart to accurately describe the service and increase the efficiency of discovery

The results of this work are to increase the accuracy of the description of the services of users to the Internet sites and to increase the efficiency of discovery of these services.

 [4]

Finally, in work of Anna Formica, we have been able to get the results of intensive and good for medium enterprises and get the results confirmed, but we will not be able to use this work in large enterprises or large businesses and control of small and medium companies are a small part of our work in comparison to our other needs in the field of control.

And in work of Changbo Ke, we were able to cut down the services provided by users and increase the efficiency of their discovery of application submission by users but we could not contain all the requirements of users in this work, but we controlled a small part of the most important of them.

# **Chapter three**

# **Suggested Method and Algorithms**

# **1-Similarity Distance**

We have used java programming language to implanted matching algorithm between user profile between the two persons by using the java language depending on the XML for any two people, for example, first person "A" and second person "B" using the same as the search engine. We compared them and find out the behavior of each of. At follow-up to the behavior of "A" we see that the most word used in the search engine (for example, computer) and he used (1000 times) and "B" also use the word (computer) and they are the most commonly used to him in the same search engine and used (100 times) in this case when the compare between them can we say they have close behavior and distance similarity among them (less than the distance increased similarity). Similarity measurements are obtained by using an inverse measure of distance this means that the farther we move away from two things the more different they are and the distance between them is greater the things are more similar as we get closer to them and here the distance between them is smaller and there are several ways to measure the distance between two points of data may contain specific properties that may look a bit of mathematic but we focus on clear concepts only and the dimension law between two points on the assumption that the points are A and B :

(AB)2= (X1 – X2)2+(Y1-Y2)2

For example : If the point A is (3,1) and the point B is (6,5) the distance between the two points is calculated as follows

(AB)2=(X1-X2)2+(Y1-Y2)2

(AB)2=(3-6)2+(1-5)2

(AB)2=(-3)2+(-4)2

(AB)2=9+16

(AB)2=25

(AB)=5

befor gathering the data must specify a measure of similarity/distance this measurement reflects degree of proximity or separation between the goals of the target and must comply with the characteristics that thinks it distinguish between data collection in many cases dependent these characteristics of the data or the context of the problem there is no better global measure for all types of data collection problems .however, the choice of an appropriate similarity measure is also crucial for analysis especially for a particular kind of clustering algorithm for example the exiting clustering algorithms rely on density and largely at the expense of similarity and find density-based clustering on dense areas groups in the data set and the density of a particular point is estimated as the proximity of the object of the corresponding data to the neighboring objects by indicating that the proximity is quantitatively the value of distance / similarity. we can see that a large number of sites where distance/ similarity is required to find dense areas and determine the block of new data objects under influence of the various assembly efficiencies that are of great importance in helping to choose the best measure in general determines the similarity measurement between the symbolic description of two objects one single value that depends on two factors the characteristics of objects and the measurement itself.[10]

# **2- Contextual matching**

To enhance the semantic expression of the texts along with the traditional keyword matching strategy so as to effectively improve the contextual matching in our approach we take two aspects of the similarities among pages and ads.

1. Similarity based on keywords and common text capture
2. Similarity based on Wikipedia measure the relevance of the semantic perspective of oneness-like techniques. This approach consists of the following steps:
* First: -of all we choose enough articles from Wikipedia to share many of the semantic concepts.
* Second: -we build the keyword phrase for each page. Finally, we suggest combining the two types above and similar in a uniform way. To make the top-nods selection.
* In order to evaluate the effectiveness of our approach we conducted a set of experiments containing real ads and pages, the results show that the approach combines Wikipedia based semantic matching with word matching can greatly improve the accuracy of the measurement of similarity among pages and ads and thus improve the effectiveness of contextual advertising. In addition, the results of full-text matches take a long time between al articles and pages.[5]

# **Term Frequency Inverse Document Frequency (TF-IDF) Algorithm:**

TF-IDF to identify words in a set of documents that are more suitable for use in the query. TF-IDF refers to the calculation of the value of each word found in the documents through the inverse proportion of the frequency of words in the specified parts of the documents and calculating thepercentageof them. -IDF means the strength of the relationship with the specific documents indicating that if these words appear in the query, this document is useful to the user as a result of his request [9] . there are several ways to model documents text for example, can be represented as a group of words where is assumed that appear words independently and sorting is a word . this model widely used in information retrieval and exploration is the expense of the words in the group, which different from the definition of sports the group corresponding to each word dimension in an area of the data generated then become each document vector is made up of non-negative in each dimension here we use the frequency of each term which means the terms and the most important descriptive document D={d1,….,dn} consists of a group of documents and T={t1,……,tm} consists of a group of terms distance occurring in D a moment only assume it is a word is then representation of the document as vector three-dimensional td . tf(d,t) refers to repeat the term (t belong T) in the document (d belong D) then the representation of vector document (d) is td=(tf(d,t1 ),…..,tf(d,tm)) although the words the most importance for example words such as (a,the) is probably the words the most frequently that appear in the text of the English but are not descriptive and is not an important issue of the document strategies for the most complex such as scheme (tfidf) used with documents submitted as vectors same degree of similarity between the two documents as a correlation between the vector corresponding which can legalize it more as a cosin of the angle between two vector contains an area of the document is usually the tens and thousands of dimension words are basically words but we apply many standard vector-reaching the basic, The first we removed words stop in the document such as (a ,are ,do) it is the most commonly used .Secondly is set words with endings away in a word single such as production ,produce .produces and product . and the main objective is to differences logical that have the same root should be treated as a word of one .Third , we take into account the embed terms is repeated in the representation of the documents in a clustering are all through the exclusion of terminology repeated.in assembly process also need to compare the similarity between the two groups or between block and each object . in the clustering hierarchical calculates usually as a link the full or link one or a distance of connectivity medium, however in algorithms assembly partial usually are representation of the block with object to the point of central .and the point of central is the value of the medium of all vectors term in the group . and you should know that is not necessarily be terms most common is the most useful on the reverse of the terms that appear frequently in a small number of documents. But they are rarely in other documents the most appropriate and specific for a particular group of documents and therefore it is the most beneficial to find documents similar in order to capture these terms reflect their importance. We convert frequencies (tf(d,t)) in the long the basic (tfidf) (term frequency inverse document frequency). Tfidf is calculated the frequency of the term (t) in the document (d) in a factor reduces the importance with appear in a document entire which is defined as **tfidf(d,t)=tf(d,t)\* log (D/df(t))** here df(t) is the number of document which appears in which the term of (t).

The example of TF-IDF Algorithm: -

package com. guendouz. textclustering. preprocessing;

 import java. util.Arrays;

 import java. util. List;

 public class TFIDFCalculator {

public double tf(List<String> doc, String term) {

double result = 0; for (String word: doc) {

if (term. equalsIgnoreCase(word)) result++;} return result / doc. size ();

}

public double idf(List<List<String>> docs, String term) {

double n = 0;

for (List<String> doc: docs) {

for (String word: doc) {

 if (term. equalsIgnoreCase(word)) {

 n++;

 break;

 } } }

return Math.log (docs. size () / n);

 }

 public double tfIdf(List<String> doc, List<List<String>> docs, String term) {

 return tf (doc, term) \* idf (docs, term);

}

public static void main(String[] args) {

List<String> doc1 = Arrays.asList("Lorem", "ipsum", "dolor", "ipsum", "sit", "ipsum");

List<String> doc2 = Arrays.asList("Vituperata", "incorrupte", "at", "ipsum", "pro", "quo");

List<String> doc3 = Arrays.asList("Has", "persius", "disputationi", "id", "simul");

 List<List<String>> documents = Arrays.asList(doc1, doc2, doc3);

 TFIDFCalculator calculator = new TFIDFCalculator (); double tfidf = calculator. tfIdf (doc1, documents, "ipsum"); System.out.println("TF-IDF (ipsum) = " + tfidf);

 }

 }

# **Flow chart**

Web log file

Sort User interest

Figure(1) overall system flowchart

No

yes

No

yes

Sort user by URLS

More URLS

More users

TF-IDF

Clean URL data

Fetch URL data

Select URL from user

Select user

preprocessing

* Start :we run our program.
* Web log file: we are working on taking the URL of any web page in the internet and put it in the program . as picture below :



And the URL is:



* Preprocessing: the sub program that responsible for bringing the information in HTML from begin working after that we working to remove the tags that brought by sub program to be ready for use.

As the code below :



* Sort user by URLS :in this step we sort the URLS in log file depending on the use of users to the internet and interests .
* Select user: in this step we select any user from the log file who owns several URLS.
* Select URL from user: after we select the user we take any URLS for this user and use it in sub program to process it .
* Fetch URL data: bring the data of web page throng put the URLS in the sub program and process it . as the code below :



* Clean URL data: after fetching data from URLS in HTML form we need to clean it up of tags in order and remove words the most frequently in the language such as (a, and , the , and) to be ready to use by the tf-idf algorithm . as the code below:



* TF-IDF: in this step we used the algorithm (TF-IDF) this algorithm is working on the breakdown the words and calculate the iteration of each word in the text through the inverse proportion of the frequency of word in the particular parts of the log file and calculating the percentage of them.
* Sort user interest: through the previous work we are able to order the users by their interests in any site use it .
* More URLS and more users : if there more URL we choose new URL and if there is no URL we choose a new user that found and if there is no we finish the program .

# **References**

1. Author:/عماد عدلي XMLطريقك الى /الموسوعة العربية للكمبيوتر والانترنت /2005
2. Author :Jine Peng /User profiling in intrusion detection/Available online 1July 2016/ by Journal of network and imputer application 72(2016) 14-27/page 1,8.
3. Author: Anna Formica /Semantic search for matching user requests with profiled enterprises/Available on line 11 December 2012 /Journal home page [www.elsevier.com/locate/compind](http://www.elsevier.com/locate/compind) /page 1,11
4. Self-adaptive semantic web service matching method
5. Author :Guandong Xu .Zongda Wu .Guiling Li .Enhong Chen / Improving contextual advertising matching by using Wikipedia thesaurus knowledge /Springer-Verlag London /2014
6. Author :Daniel L Silver . Elhadi Shakshuki /User profile management reference model and web services implementation/International Journal of Web and Grid Services /2010
7. Semantic matching
8. Author :Juan Ramos Java platform, Enterprise edition (Java EE) specification, v7
9. Author : Juan Ramos/Using TF-IDF to determine word relevance in document queries/Available on line 15 April 2013 by oracle America, Inc. ("specification lead ")/page 1,4.
10. Author :Anna Huang /Similarity measures for text document clustering /New Zealand/April 2008

# **Contents**

**Contents**

[**Chapter one** 4](#_Toc512506856)

[**Introduction** 4](#_Toc512506857)

[**1.1-XML** 5](#_Toc512506858)

[**1.2-Matching** 5](#_Toc512506859)

[**1.2.1-Exact Matching** 6](#_Toc512506860)

[**1.2.2-Semantic Matching** 6](#_Toc512506861)

[**1.3-User Profiling** 7](#_Toc512506862)

[**Chapter two** 8](#_Toc512506863)

[**Literature review** 8](#_Toc512506864)

[**Related works** 9](#_Toc512506865)

[**1-According to the work of "Anna Formica":** 9](#_Toc512506866)

[**2- According to the work of "Changbo ke":** 9](#_Toc512506867)

[**Chapter three** 12](#_Toc512506868)

[**Suggested Method and Algorithms** 12](#_Toc512506869)

[**1-Similarity Distance** 13](#_Toc512506870)

[**2- Contextual matching** 14](#_Toc512506871)

[**Term Frequency Inverse Document Frequency (TF-IDF) Algorithm:** 15](#_Toc512506872)

[**Flow chart** 20](#_Toc512506873)

[**References** 24](#_Toc512506874)

[**Contents** 25](#_Toc512506875)