Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SEMI-STRUCTURED TEXTS CLUSTERING ALGORITHM

[1,2]ZHANG PEI YUN, [2]CHEN EN HONG, [3]HUANG BO

[1]School of Mathematics and Computer Science, Anhui Normal University, 241003, China

[2]School of Computer Science, University of Science and Technology of China, Hefei, 230026, China

[3]School of Computer Science & Technology, NUST, Nanjing 210094,

ChinaE-mail: [1]zpyanu@mail.ahnu.edu.cn

ABSTRACT

In order to improve the clustering result of semi-structured texts, it needs to reduce the dimension and sparsity. To reduce the dimensions of semi-structured texts clustering, aimed at meta-data of semi-structured texts, we build the metadata feature vectors. Based on the domain concepts model, we build domain vector based on the domain concepts tree (set). With the help of the WordNet, we compute semantic similarity between the metadata feature vector and the domain vector. Finally, the clustering algorithm is designed to cluster semi-structured texts based on the semantic similarity between metadata feature vectors and domain vectors. The analysis shows that the clustering algorithm is feasible and has higher clustering accurate rate. It can ease the problem of lacking domain ontology and has the ability to improve the clustering quality.

Keywords: *Domain Concepts Model; Metadata; Semi-Structured Texts; Clustering; Semantic Similarity*

1 INTRODUCTION

Clustering is an important research content of data mining and pattern recognition. It plays an important role in recognition of the inner data structure [1]. Clustering is widely used in search engine; Web mining, information retrieval etc [2]. At present, there are several different clustering methods, such as hierarchical clustering, fuzzy clustering, probabilistic clustering, association rules based on clustering etc [1,3,4]. The typical clustering process mainly includes data preparation, feature selection and feature extraction, close degree calculation, clustering, the validity evaluation of the clustering results [1].Semi-structured texts as a kind of important network information resources; we can improve the efficiency of discovering semi-structured texts based on the clustering process. Because the current text clustering methods do not consider the feature of semi-structured texts (structured, unstructured, semi-structured), there are the high dimensions and high sparseness problems during the semi-structured text clustering process. For example, a text may have thousands of words, of which only a small portion of words are feature words, which may have high dimensions and high sparsity, so it results in inefficient clustering effect. There are existing huge semi-structured text resources in the network, for example, the digital library, open archives initiative, Electronic periodicals, et al. Therefore, it is necessary for us to study the problems of the semi-structured texts clustering and to improve the clustering effect. This paper studies the semi-structured texts clustering algorithm in order to reduce the dimension of text clustering and improve quality.

2 RELATED WORK

The researchers have done much work and obtain a lot of achievements of clustering. Halgamuge &Xu[5-6] point out that hierarchical clustering algorithm is a typical algorithm with higher clustering effect. Fung [2] proposes unsupervised hierarchical clustering algorithm(FIHC), the paper points out that the FIHC algorithm is more accurate, efficient and scalable than UPFMA、*k*-means. The improved FIHC algorithm [7], during the construction of cluster tree, it removes the pruning process, but it needs two mandatory input parameters. Medina [8] constructs the clustering tree based on the description of main concepts. The above methods of hierarchical clustering lack of lexical semantic analysis and prone to leak clustering. Hotho [9] provides a common OAI-PMH collector system, using the vector space model to calculate the text cosine similarity. Hamel [10] applies the word similarity measurement method

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and combines WordNet to improve the text vector representation model, but it does not reduce the dimension of text representation feature. The word clustering methods based on words semantics which follow the word frequency vector model and do not reduce the high dimension of the model, they limit to some specific application domain. Based on multiple feature comparison of texts data and grouping and aggregation, Martinez [11] explores the general and special ontology and designs the corresponding clustering algorithm. These methods rely on the prior label input and artificial and formal semantics of one or more knowledge structure (such as ontology, classification).Based on term semantic similarity of text similarity computing, Huang [12] proposes the weighting term similarity tree which is used to guide the text similarity computing and cluster. Günnemann [13] uses metadata and ontology resource to classify resources and identifies topics and subtopics by extracting key words and the domain-specific ontology. Navigli [14] proposes OntoLearn level algorithm and uses the WordNet to construct domain ontology; it is assumed that there have existed the classified documents. Sun [15] proposes the clustering method based on element path and the element path expresses the semantic information. Vicient [16] proposes domain independent and unsupervised methods to find the relevant feature of heterogeneous texts resources, and it is associated to the background of ontology concepts.

From the above analysis, the main problems existing in the texts clustering are as below. It lacks of different process for structured, unstructured and semi-structured text respectively; they are mainly based on the frequency of the words [9-10]. Because of lacking of semantic information to understand and automatic process, it increases the workload and the difficulty for text clustering. This paper does research on semi-structured text clustering algorithm based on the metadata and the domain concepts model, it can reduce the dimension of text clustering, and therefore it can improve the semantic interoperability and text clustering effect.

3 RESEARCH BACKGROUND

3.1 Semi-Structured Text

The semi-structured texts refers to a special kind of natural language texts, which have a standardized text structure , but the expression of the content is free, such as SGML, HTML, email, all kinds of databases, knowledge database and digital library, et al. Semi-structured texts may have information such as titles, authors, abstract, keywords, main body, appendix etc.

3.2 Metadata

The metadata is "data about data". Semi-structured texts contain metadata information. By extracting important metadata which are important for clustering (such as title, abstract, keywords, et al.), the generated metadata feature vector will be low dimension and low sparsity, therefore it is better to solve the high dimension and height data sparsity problem during the clustering process. Usually there are two main kinds of methods for metadata extraction which are based on rules or statistical model. The existing metadata extraction methods lay a foundation for our work.

3.3 Domain Concepts Model

In general, the important concepts in domain concepts model can characterize the field characteristics. Domain concepts model can be divided into domain concepts tree and domain concepts set.

(1) Domain concepts tree (concepts tree in abbreviation)

The concepts of the domain can be constructed to a concepts tree. For example, figure 1 is the concepts tree with some domain concepts. If the domain concepts are formed into the concepts tree, the semantic similarity calculation will be executed based on the sub-tree and its parent concept in the concepts tree.

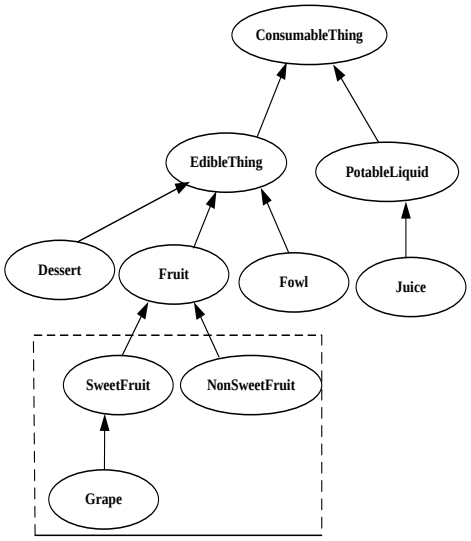

Figure 1. Part Of The Domain Concepts Tree

In figure 1, the ovals represent concepts, arrow lines point from sub-concepts to parent-concepts. The sub-concepts in the dashed box have “Fruit” as their parent-concept and these sub-concepts are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

built as the domain vector [SweetFruit, NonSweetFruit,Grape]. In the process of clustering, if the semantic similarity between the metadata feature vector and the domain vector reaches the required threshold, then the text will be clustered to the "Fruit" class. If there is complete ontology in the domain, then the concepts tree can be extracted from the domain ontology based on the parent-concepts and sub-concepts relation. The concepts tree can be built manually as well. The Yahoo directory concepts tree and subject classification tree and domain ontology can be used for building concepts tree. Based on the concepts tree, we can build many domain vectors.

(2)Domain concepts set (concepts set in abbreviation)

If the domain important concepts do not be built into the concepts tree, they can be organized into concepts set. For example, in economy domain, there are the important concepts such as "bank, cash, spending, lending, deposits, reputation, passbook, inflation", et al. These concepts are organized into concepts set. Based on the concepts set, we can build a domain vector. The texts will be clustered into a category. From above analysis, the clustering effect based on the concepts tree is finer than based on concepts set.

4 THE KEY PROBLEMS AND THE ALGORITHMS

Different from the traditional distance-based clustering method (such as hierarchical clustering method and plane dividing method etc), the implementation of the paper is based on metadata and domain concepts models and their semantic similarities. We realize the semi-structured texts clustering based on the domain important concepts and metadata. Aimed at semi-structured texts, the key problems of the paper are that how to build the metadata feature vectors and build domain vectors and compute the semantic similarity between the two vectors. In order to solve the key problems, we design the corresponding algorithms (algorithm 1 and algorithm 2 and algorithm 3). Semi-structured text clustering framework diagram is shown in figure 2.

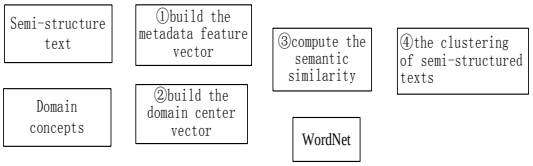

Figure 2. Semi-structured text clustering framework diagram

The figure 2 mainly has four parts. ①Build the metadata feature vector. It is based on the metadata extracted from semi-structured texts; the purified metadata is built into feature vector. The process is shown in algorithm 1.②Build the domain vector. It is based on the domain concepts tree or concepts set to build the domain vector. The process is shown in algorithm 2. ③Compute the semantic similarity between the two vectors based on WordNet. The process is shown in algorithm 3. ④The clustering of semi-structured texts. It is based on the semantic similarity to realize the clustering. The process is shown in algorithm 4.

4.1 Build The Metadata Feature Vector

In order to realize the semi-structured texts clustering, we utilize metadata to reduce the dimension. Firstly, we need to extract users' interested metadata based on document metadata standard. After automatically harvesting metadata from semi-structured texts, we store them in files. After reading the metadata and delete stop words from the metadata, we obtain the purified metadata. Based on them, we build the metadata feature vector. The process is shown in algorithm 1.

Algorithm 1. *getMetadataVector* (*metadataFile*)
Inputs：*metadataFile*
Outputs：*metadataVector*
1. *metadata* ← read metadata from *metadataFile*;
2.*metadataFined* ← delete the stop words from *metadata*, get the purified metadata;
3. **For each** *metadataFined*[*i*] in *metadataFined*
4.{
5. **If**(*metadataFined*[*i*] does not exist in *metadataVector*)
6. {*metadataVector*[*i*].*metadata* ← *metadataFined*[*i*];
7. int *n*= count the number of *metadataFined*[*i*];
8. *metadataVector*[*i*]. *metadata.weight* ← *n* ; /* *n* is as the weight of *metadataVector*[*i*]. *metadata* */;
9. }**Endif**
10.}**Endfor**
11.**Return** *metadataVector*;

Algorithm 1 is the base for the semi-structured text clustering. In algorithm 1, we need to read metadata from the metadata file (see sentence 1). The obtained metadata will be useless words for clustering. Therefore, we need to delete the stop words, such as "at, with, to, by" etc. the stop words are stored in the stop words list files. We can add or remove the stop words to the stop list file. After deleting stop words, we obtain the purified metadata

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and store them in variable *metadataFined*(see sentence 2). We build the metadata feature vector; the process is that put the purified metadata into the variable *metadataVector* and the purified metadata do not exist in *metadataVector* (see from sentence 3 to and sentence 6). Because it is very necessary for count the metadata's frequency, the frequency mean the occurrence of feature metadata. The bigger the frequency is, the more contribution it is to the clustering result. We need to count the frequency of each feature word as its weight and the value of the frequency is as the weight of metadata feature item (see sentence 7 and sentence 8). In the end, the algorithm 1 returns the
metadata feature vector is as below.
metadataVector={(*metadataFined*$_1$,n_1), (*metadataFined*$_2$,n_2)…(*metadataFined*$_i$,n_i)};
If the text has *n* metadata, the time complexity is *O*(*n*).

4.2 BUILD THE DOMAIN VECTOR

To build the domain vector, we need to obtain the important concepts from the corresponding domain. In the case of lacking the domain expert supporting, we use top-down method to identify the important concepts and relationships. The process to build the domain vector is shown in figure 2.

Algorithm 2. *getDomainVector* (*domainConcepts*)
Inputs：*domainConcepts*
Outputs：*domainVector*
1.If(*domainConcepts* come from domain concepts tree)
2.For each *domainConcept*[*i*] in *domainConcepts*
/* *domainConcept*[*i*] represent one concept of the concepts tree */
3.{[c_1,c_2,c_3,…..c_n] ← get each subclass of *domainConcept*[*i*];
4.*domainVector*[*i*].*concepts* ← {c_1,c_2,…c_n,*domainConcept*[*i*]};
5. }**Endfor**
6. **Else** /* If the domain concepts come from the domain concepts set */
7. {[c_1, c_2,…c_n] ← get each *domainConcept*[*i*] from *domainConcepts*;
8. *domainVector.concepts* ← {c_1, c_2, c_3, …..c_n , *domainConcept*}; /* c_1, c_2, c_3, …..c_n as a set belong to *domainConcept* */
9. } **EndIf**
10. **Return** *domainVector*;

The algorithm 2 reads domain concepts from variable *domainConcepts*(see sentence 1). Because domain concepts may derive from the concepts tree and concepts set, the algorithm 2 deals with them respectively. If the input parameters of algorithm 1 come from concepts tree, then aimed at the each concept in the concepts tree, we build the *domainVector*[*i*].*concepts* for each concept variable *domainConcept*[*i*], and the items of the domain vector come from the sub-concepts of *domainConcept*[*i*](see from sentence 2 to sentence 5). When the concepts tree has *n* nodes, there will be *n* domain vectors built. If the input parameters come from domain concepts set, then all the concepts in the set are assigned to the domain vector variable *domainVector*.*concepts* (see from sentence 7 to sentence 9) and we build one domain vector. Let *n*= *domainConcepts.size*(), the time complexity of algorithm 2 is *O*(*n*). The algorithm 2 returns *domainVector* of one domain.

4.3 Computer The Semantic Similarity Betweenn The Metadata Feature Vector And The Domain Vector

In order to determine the field of the semi-structured texts, we need to compute the semantic similarity between the metadata feature vectors and the domain vectors. Because there are many feature items in the two vectors, we need to compare the semantic similarity between two feature items from the two vectors respectively. Based on the rich semantic information of WordNet, the paper computes the semantic similarity. The process is shown in algorithm 3.

Algorithm 3.
***computeSemanticSimilarity*(*metadataVector* ,*domainVector*)**
Inputs：*metadataVector* & *domainVector*
Outputs： *similarityVector* /* the semantic similarity between two vectors ***/**
1 *simVec*=*null*; *k*= *metadataVector.size*(); *l*= *domainVector.size*();
2. **For** *i*=1 to *k*-1
3. {**For** *j*=1 to *l*
4.{similarity ← getSimilarity(*metadataVector*[*i*],*domainVector*[*j*]);
5. **If** (*similarity* >*simVec*[*i*].*sim*)
6. { *simVec*[*i*].*sim* ← *similarity*;
7. *simVec*[*i*].*metadata* ← *metadataVector*[*i*];
8. *simVec*[*i*].*concept* ← *domainVector*[*j*];
9. }**Endif**
10. }**Endfor**
11. $imVec.sim = \sum_{i=1}^{n} simVec[i].sim * simVec[i].metedata.weight$;
12. }**Endfor**
13. **Return** *simVec*;

In algorithm 3,when computing semantic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

similarity between two items from the metadata feature vector and the domain vector respectively(see sentence 2 and sentence 3), we adopt JWNL to compute the semantic similarity value and assign the similarity value to the variable *similarity*(see sentence 4). We compare the values and get the maximum value as the last matched result and save it in variable *simVec*[*i*].*sim*(see sentence 5 and sentence 6), and the corresponding items are store as well(see sentence 7 and sentence 8). In the end, *simVec*[*i*].*sim* multiplies the weight of the corresponding metadata item and the weight is as the contribution value to semantic similarity. We add all the maximum similarity values between two items as the last semantic similarity of the two vectors (see sentence11). The value of variable *simVec*[*i*].*metadata*.*weight* comes from the sentence 13 in the algorithm 1. The algorithm 3 returns the similarity value of the metadata feature vector and the domain vector. The time complexity of the algorithm 3 is $O(k*l)$.

4.4 Semi-Structured Texts Clustering

Based on the result of the algorithm1, the algorithm 2 and the algorithm 3, we design a semi-structured text clustering algorithm (Semi-Structured Text Clustering, SSTC for short). The process is shown in algorithm 4.

Algorithm 4. *SSTC* (*metadataFileSet, domainConceptsSet, threshold*)

Inputs：*metadataFileSet* , *domainConceptsSet*, *threshold* /*the metadata files collection, domain concepts tree (set), the threshold */

Outputs：the clustering result **/*** the clustering result ***/**

1.**For each** *metadataFile* in *metadataFileSet*

2.$metadataVector_i$ **=*getMetadataVector*** (*metadataFile*$_i$); /* build metadata feature vector, the function ***getMetadataVector*** comes from the algorithm 1 */

3.**For each** $domainConcepts_j$ in *domainConceptSet*

4. $domainVector_j$ = ***getDomainVector*** ($domainConcepts_j$); /* build domain vector ,*getDomainVector* function comes from the algorithm 2 */

5. *sim*=0； /**sim* is used to store the current maximum similarity value of a metadata feature vector and a domain vector*/

6. *n*=*metadataFileSet.size*();

7. *m*= *domainConceptSet.size*();

8. **For** *i*=1 to *n*

9. { **For** *j*=1 to *m*

10. { *similarity*= ***computeSemanticSimilarity***(*metadataVector*$_i$, $domainVector_j$); /****computeSemanticSimilarity*** function comes from the algorithm 3 */

11. **if**(*sim*< *similarity*)

12. *sim*= *similarity*; /*save the maximum semantic similarity value of two vectors */

13. } **Endfor**

14. **If** (*sim* > *threshold*) /* determine the maximum semantic similarity is greater than a threshold requirements */

15. *clustering*[*i*] ← The text resource belongs to the corresponding $domainVector_j$;

16. } **Endfor**

17. **Return** *clustering* ;

In algorithm 4, for each file $metadataFile_i$ in metadata files collection *metadataFileSet*(see sentence 1), we invoke algorithm 1 to build the metadata feature vector based on $metadataFile_i$(see sentence 2). For each domain concept (see sentence 3), the domain vectors are built by invoking the algorithm 2 (see sentence 4).By invoking the algorithm 3, the semantic similarity values between a metadata feature vector $metadataVector_i$ and other different $domainVector_j$ are computed (see sentence 9). The similarity value is stored in variable *similarity* (see sentence 10), and by comparing, we get the maximum similarity value and store it in variable *sim* (see sentence 11 and sentence 12).When the value of *sim* is bigger; it indicates that the semantic similarity of $metadataVector_i$ and $domainVector_j$ is higher. If *sim* is greater than required *threshold*, then the $metadataVector_i$ of the corresponding text belongs to the $domainVector_j$ of the corresponding classification (see sentence 14 and sentence 15). The algorithm 4 returns the clustering result of semi-structured texts. The time complexity of the algorithm 4 is $O(n*m*k*l)$, and the $O(k*l)$ results from executing sentence 10.

5 ANALYSIS OF OUR EXPERIMENTS

The test samples are papers from Web of Knowledge. We test eight domains; they are geography, economy, communications, weather, travel, chemistry, history, art. They are abbreviated to geo, com, eco, wea, tra, che, his, art respectively. A total of 11868 semi-structured texts are tested of the eight domains. We adopt precision, recall and F-measure to evaluate performance of the clustering algorithm (SSTC). We set variable C as the ideally clustering number, M represents complete clustering number, $I = C \bigcap M$,I represents correctly clustering number. Based on these premises, some corresponding formulas are illustrated as below.

Journal of Theoretical and Applied Information Technology
30th April 2013. Vol. 50 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\text{precision} = \frac{|I|}{|M|} * 100\% \qquad \text{recall} = \frac{|I|}{|C|} * 100\%$$

$$\text{F} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} * 100\%$$

We test the performance of our SSTC algorithm in 8 different domains. The analysis results are shown in figure 3.

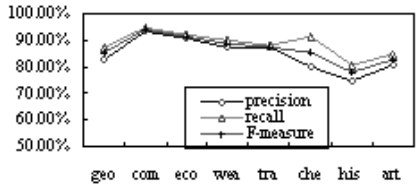

Figure 3. The Performance Of The SSTC Algorithm In Different Domains

In figure 3, aimed at our SSTC algorithm, the precision, the recall and F-measure of our experiments are shown. Take "communication" domain for example, the ideal clustering number is 819, the correct clustering number is 773, and the recall is 94.38%. Based on the tested 11868 semi-structured texts, the recall is pretty good. Because the establishment of the domain concepts tree (set) lacks of experts' participation, it is hard to determine the classification concepts and the boundary is fuzzy, it produces some loss of clustering precision and recall. In this paper, we compare SSTC algorithm and the traditional TF-IDF based clustering algorithm (abbreviation TF-IDF) and hierarchical clustering method (abbreviation HC), the analysis results are illustrated from figure 4 to figure 6.

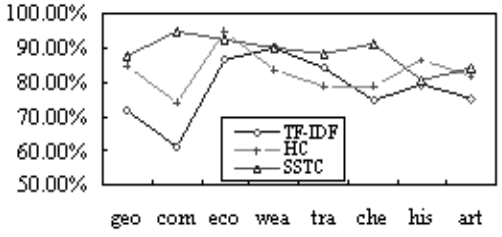

Figure 4. The Recall Of Different Methods

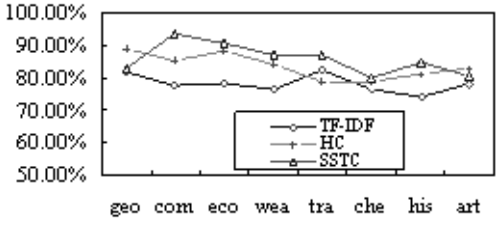

Figure 5. The Precision Of Different Methods

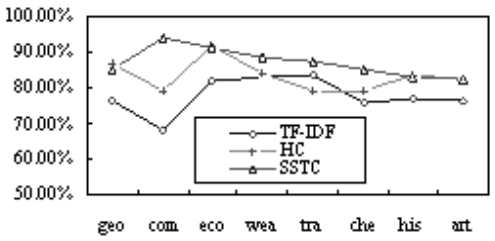

Figure 6. The F-Measure Of Different Methods

Different methods have different precision and recall and F-measure. Compared with TF-IDF and HC methods, the average clustering precision and recall and F-measure of our SSTC method are higher than other two methods. We adopt the purified metadata as feature vector; it can effectively reduce the dimension of semi-structured texts model. Compared with the traditional TF-IDF method, the efficiency has a larger improvement. TF-IDF method adopt the word frequency vector model, for most texts storage database, the number of words and texts are usually larger, the vector matrix not only has a very high dimension and extremely sparse, but also ignores the meaning of words, it eventually leads to low efficiency of cluster computing. Although the HC method is improved and better than TF-IDF method, but because it does not consider the semantic distance between vectors, its performance is between the SSTC method and TF-IDF method. The SSTC method both considers the influence of the word frequency and it also introduces the semantic dictionary WordNet to compute semantic similarity between vectors. SSTC method has good clustering performance; it can effectively complete semi-structured texts clustering.

6. CONCLUSIONS

As important network information resources, semi-structured texts have standardized structure. To improve clustering quality of semi-structured texts, the paper does research on clustering algorithm based on metadata and domain concepts model. Metadata as the important terms of semi-structured texts, they provide suitable characterization vectors for texts semantic similarity (proximity) calculation. On the condition that we do not influence the texts features extraction, the algorithm uses the metadata to minimize the text feature vector dimension and to improve clustering effect. To compute the semantic similarities, we build the domain vectors based on domain concepts tree (set). And with the help of WordNet, the semantic similarities are computed based on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

metadata feature vectors and domain vectors. Based on the analysis of our experiments, we can see that the clustering algorithm proposed in the paper can effectively implement the clustering for the semi-structured texts. It improves the semantic comprehension and clustering quality.

ACKNOWLEDGEMENTS

This paper is supported by National Natural Science Foundation of China (61201252; 61203173) and Natural Science Foundation of Anhui Province(The title: research on personalized demand-driven and dynamic collaborative service composition model and algorithm based on social network, 2013) and Humanities and Social Sciences Foundation of the Ministry of Education (10YJC870046) and Natural Science Research Key Project of Anhui Provincial Higher Education (KJ2011A128) and Soft Science Project of Anhui Province (11020503009).

REFERENCES

[1] J.g. Sun, j. Liu and L.Y. Zhao, "Clustering Algorithms Research", *Journal of Software*, Vol. 19, No. 1,2008,pp:48-61.

[2] B.C.M Fung, K. Wang and M. Ester, "Hierarchical document clustering, the Encyclopedia of Data Warehousing and Mining", *Idea Gro*up Reference, Hershey, New York, 2006,pp: 553-559.

[3] Y. Zong, G.D. Xu, P. Jin,Y.C. Zhang and E.H. Chen, "HC_AB:A new heuristic clustering algorithm based on Approximate Backbone", *Information Processing Letter*, No.17,2011,pp: 857-863.

[4] B. Liu, S.X. Xia, Y. Zhou, "A Sample- Weighted Possibilistic Fuzzy Clustering Algorithm", *Acta Electronica Sinica*, Vol. 40, No. 2, 2012, pp:371-375.

[5] S.K. Halgamuge and L.P. Wang, "Classification and Clustering for Knowledge Discovery", *Springer-Verlag*, Berlin and Heidelberg ,2005,pp:158-300.

[6] R. Xu and D.C. Wunsch. "Clustering", *Wiley/IEEE Press*, New York, 2008,pp: 200-358.

[7] J.A. Sánchez, M.A. Medina, O. Starostenko, A. Benitez and E.L. Domínguez, "Organizing open archives via lightweight ontologies to facilitate the use of heterogeneous collections", *Aslib Proceed-ings*, Vol. 64, No. 1, 2012, pp: 46-66.

[8] M.A. Medina and J.A. Sánchez, "OntOAIr: a method to construct lightweight ontologies from document collections", *Proceedings of the Ninth Mexican International Conference on Computer Science 2008 (ENC 08)*, Los Alamitos, America:IEEE Computer Society, 2008,pp:115-125.

[9] A. Hotho, S. Staab and G. Stumme, "Wordnet improves text document clustering", *Proceedings of the Semantic Web Workshop at SIGIR 2003,26th Annual International ACM SIGIR Conference*, *ACM SIGKDD*, Toronto, Canada, 2003,pp:541-550.

[10] L. Hamel, "Knowledge Discovery with Support Vector Machines", *Wiley Series on Methods and Applications in Data Mining*, Chichester: John Wiley & Sons, 2009, pp:100-150.

[11] S. Martinez, D. Sanchez and A. Valls, "Semantic adaptive microaggregation of categorical microdata", *Computer Security*, Vol. 31, No. 5, 2012,pp:653-672.

[12] C.H. Huang, J. Yin and F. Hou, "A Text Similarity Measurement Combining Word Semantic Information with TF-IDF Method",*Chinese Journal of Computers*, Vol. 34, No. 5, 2011, pp:856-864.

[13] S. Günnemann, I. Färber and T. Seidl, "Multi-view clustering using mixture models in subspace projections", *2012 Knowledge Discovery and Data Mining. ACM SIGKDD*, Beijing, China: 2012, pp: 132-140.

[14] R. Navigli and P. Velardi, "Learning domain ontologies from document warehouses and dedicated web sites", *Computational Linguistics*, Vol. 30, No. 2, 2004,pp:151-179.

[15] Y.Z. Sun, B. Norick, J.W. Han, X.F. Yan, P.S. Yu and X. Yu, "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks", *2012 Knowledge Discovery and Data Mining, ACM SIGKDD*, Beijing, China, 2012, pp: 1348-1356.

[16] C. Vicient, D. Sánchez and A. Moreno, "An automatic approach for ontology-based feature extraction from heterogeneous textual resources",*Engineering Applications of Artificial Intelligence*,Vol. 25, No. 78,2012,pp:1-15.