PERFORMANCE EVALUATION OF SEMANTIC BASED AND ONTOLOGY BASED TEXT DOCUMENT CLUSTERING TECHNIQUES

***Ashish Punya, *Meenakshi Rana**

**Department of Computer Science & Engineering,*
KEC Ghaziabad

ABSTRACT

Text clustering typically involves clustering in a high dimensional space, which appears difficult with regard to virtually all practical settings. In addition, given a particular clustering result it is typically very hard to come up with a good explanation of why the text clusters have been constructed the way they are. In this paper, we propose a new approach for applying background knowledge during pre processing in order to improve clustering results and allow for selection between results. We built various views basing our selection of text features on a hierarchy of concepts. Based on these aggregations, we compute multiple clustering results using K-Means. The results may be distinguished and explained by the corresponding selection of concepts in the ontology. Our results compare favourably with a sophisticated baseline pre processing strategy.

The amount of digital information is created and used is steadily growing along with the development of sophisticated hardware and software. This has increased the need for powerful algorithms that can interpret and extract interesting knowledge from these data. Data mining is a technique that has been successfully exploited for this purpose. Text mining, a category of data mining, considers only digital documents or text. Text Clustering is the process of grouping text or documents such that the document in the same cluster are similar and are dissimilar from the one in other clusters. This paper studies the working of two sophisticated algorithms. The first work is a hybrid method that combines pattern recognition process with semantic driven methods for clustering documents, while the second uses an ontology-based approach to cluster documents. Through experiments, the performance of both the selected algorithms is analyzed in terms of clustering efficiency and speed of clustering.

INTRODUCTION

The information and communication industry has envisaged a dramatic increase in the amount of information or Data being stored in electronic format. With the enormous amount of data stored in files, databases, and other Repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps Interpretation of such data and for the extraction of interesting knowledge that could help in decision-making . Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology With great potential to help companies focus on the most important information in their data warehouses. Data Mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven

Decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events Provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions That traditionally were time consuming to resolve. They scour databases for hidden patterns, finding predictive Information that experts may miss because it lies outside their expectations .

With the abundance of text documents through World Wide Web and corporate document management systems, the dynamic partitioning of texts into previously unseen categories ranks top on the priority list for all business intelligence systems. However, current text clustering approaches still suffer from major problems that greatly limit their practical applicability.

Data mining is a multidisciplinary field, drawing work from areas including database technology, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, knowledge-based systems, knowledge acquisition, information retrieval, high-performance computing and data visualization . Even though, many researchers have probed into the field of data mining, it still has to go a long way for perfection. As the demand of customers grows the need for understanding the data and predict the future becomes crucial. In general, data mining basically performs three operations. They are

(i) explore the data
(ii) find patterns and
(iii) Perform prediction.

To perform these steps, a number of data mining methods including data characterization, data discrimination, association analysis, classification, prediction and clustering are available.

ONTOLOGY CONSTRUCTION FOR INFORMATION SELECTION

Technology in the field of digital media generates huge amounts of textual information. The potential for exchange and retrieval of information is vast and daunting. The key Problem in achieving efficient and user-friendly retrieval is the development of a search Mechanism to guarantee delivery of minimal irrelevant information (high precision) While insuring relevant information is not overlooked (high recall). The traditional Solution employs keyword-based search. The only documents retrieved are those Containing user specified keywords. But many documents convey desired semantic Information without containing these keywords. One can overcome this problem by Indexing documents according to meanings rather than words, although this will entail a Way of converting words to meanings and the creation of ontologies. We have solved the Problem of an index structure through the design and implementation of a concept-based Model using domain-dependent ontologies. Ontology is a collection of concepts and their Interrelationships, which provide an abstract view of an application domain. We propose A new mechanism that can generate ontologies automatically in order to make our Approach scalable. For this we modify the existing self-organizing tree algorithm (sota) That constructs a hierarchy.

Furthermore, in order to find an appropriate concept for each Node in the hierarchy we propose an automatic concept selection algorithm from Wordnet, a linguistic ontology.

1. RELATED WORK

Historically ontology has been employed to achieve better precision and recall in the text retrieval system .Here, attempts have taken two directions, query expansion through the use of semantically related-terms, and the use of conceptual distance measures .

For the construction of ontology, the above papers assume manual construction; however, only a few automatic methods are proposed. Elliman et al. propose a method for constructing ontology to represent a set of web pages on a specified site. Self organizing map is used to construct hierarchy. Bodner et al. propose a method to construct hierarchy based on statistical method (frequency of words). Hotho et al. propose various clustering techniques to view text documents with the help of ontology.

Note that a set of hierarchies will be constructed for multiple views only; not for ontology construction purpose.

2. ONTOLOGY FOR INFORMATION SELECTION

Ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose . Therefore, ontology defines a set of representational terms that we call concepts. Inter-relationships among these concepts describe a target world. Ontology can be constructed in two ways, domain dependent and generic. CYC , WordNet , and Sensus are examples of generic ontology. WordNet is a linguistic database formed by synsets—terms grouped into semantic equivalence sets, each one assigned to a lexical category (noun, verb, adverb, adjective). Each synset represents a particular lexical concept of an English word and is usually expressed as a unique combination of synonym sets. In general, each word is associated to more than one synset and more than one lexical category.

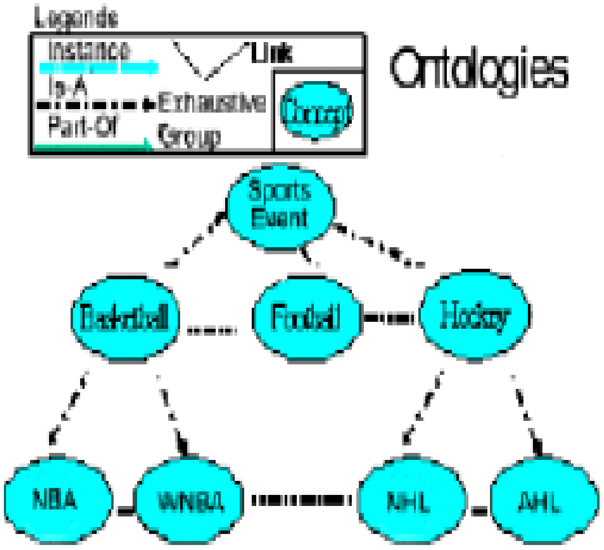

Figure 1. A Portion of Ontology for Sports Domain

A domain-dependent ontology provides concepts in a fine grain, while generic ontology provides concepts in coarser grain. Figure 1 illustrates example ontology for the sports domain. This ontology may be obtained from generic sports terminology and domain experts . The ontology is described by a directed acyclic graph (DAG). Here, each node in the DAG represents a concept. In general, each concept in the ontology contains a label name and a vector. A vector is simply a set of keywords and their weights. Furthermore, the weight of each keyword of a concept may not be equal. In other words, for a particular concept some keyword may serve as more discriminating as compared to some other; it will be assigned higher weight.

3. AUTOMATED ONTOLOGY CONSTRUCTION

We would like to build ontology automatically from a set of text documents. If documents are similar to each other in content they will be associated with the same concept in ontology. For this first we would like to use a hierarchical clustering algorithm to build a hierarchy. Then we need to assign concept for each node in the hierarchy. For this, we deploy two types of strategy and follow bottom up concept assign mechanism. First, for each cluster consisting of a set of documents we assign a topic based on a supervised self-organizing map algorithm for topic tracking. However, if multiple concepts are candidates for a topic we propose an intelligent method to arbitrate them.

4.1 HIERARCHY CONSTRUCTION

We would like to partition a set of documents S= {D1, D2… Dn} into a number of clusters C1, C2… Cm, where a cluster may contain more than one documents. Furthermore, we would like to extend our hierarchy into several levels. For this, several existing techniques are available to such as hierarchical agglomerative clustering (HAC) , selforganizing map (SOM) , self-organizing tree (SOTA), and so on.

4.1.1. SELF-ORGANIZING MAP (SOM)

The SOM, introduced by Kohonen , is one of the unsupervised neural networks. A SOM consists of two parts, the input data (i.e., document, image) and the output map. It maps the high dimensional input data into the low dimensional output topology space, which is two-dimensional. Furthermore, we can think of the SOM as a “nonlinear projection” of probability density function p(x) of the high-dimensional input data vector x onto the two dimensional display. This makes SOM optimally suitable for application to the problem of the visualization and cluster of complex data.

Each SOM input data is represented by a vector of features x. Each node in the output map has a reference vector w. The reference vector has the same dimension as the feature vector of input data. Figure 2 shows the basic architecture of SOM. Initially the reference vector is assigned to random values. During the learning process an input data vector is randomly chosen from the input data set and compare with all w. Various distance measure functions can be used to compare such as Euclidean distances ||x-wi|| or cosine distances (x,wi)/(||x||*||wi||). The best matching node c is the node, which has the minimum distance with the input data.

:|| || min{|| ||} c i i c x − w = x − w (1)

Then the reference vector of the best matching node and its neighboring nodes which are topologically close in the map are updated by Equation 2. In this way, eventually neighboring nodes will become more similar to the best match nodes. Therefore, the topologically close regions of the output map gain an affinity for clusters of similar data vectors .

$\Delta w = \eta\ t \times \Lambda\ i\ c \times x - w$ (2)

Where, i, t, and η(t) denote the neighboring node, discrete time coordinate, and learning rate function respectively. The convergence of the algorithm depends on a proper choice of η. During beginning of the learning η should be chosen close to 1, Thereafter it decreases monotonically. One choice can be η(t) = 1/t. Note that in Equation 2 Λ(i,c) is the neighborhood function. A Gaussian function can be used to define Λ(i,c):

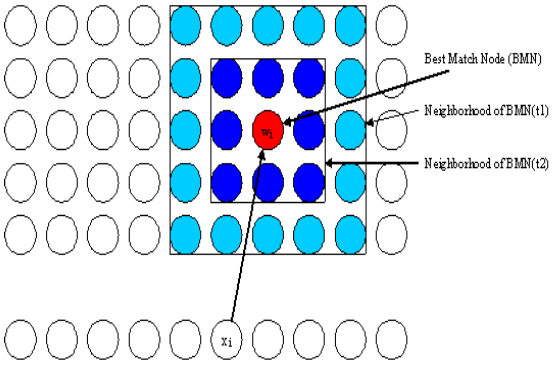

Figure 2. Basic Architecture of Self-Organizing Map (SOM)

4.1.2. Self-Organizing Tree Algorithm (SOTA)

The predetermined structure of classical SOM implies a limitation on the result mapping. A number of models have been proposed to build a topology of output nodes. Fritzke proposes a growing cell structure (GCS) model which facilitates finding the suitable output mapping structure and size automatically. Based on the SOM and GCS, Dopazo et al. introduce a new unsupervised growing and tree-structured SOM called selforganizing tree algorithm (SOTA) . The topology of SOTA is a binary tree.

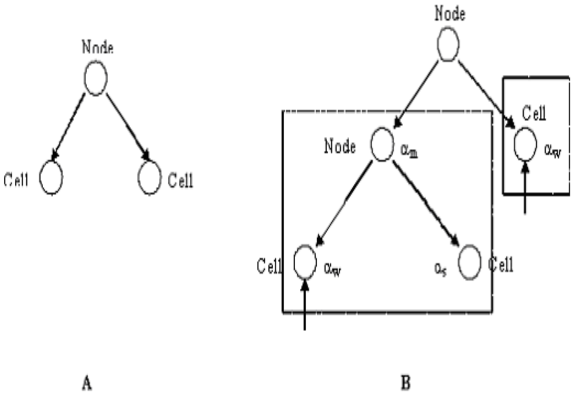

Figure3. (A) Initial Architecture of SOTA

(B) Two Different Reference Vector Updating Schemes

Initially the system is a binary tree with three nodes (Figure 3 (A)). The leaf of the tree is called cell and internal node of the tree is called node. Each cell and the node have a reference vector w. The values of the reference vector are randomly initialized. In SOTA only cells are used for comparing with the input data. After distributing all the input data into two cells, the cell which is most heterogeneous will be changed itself to a node and create two descendent cells. This procedure is called cycle. To determine heterogeneity a Resource is introduced. Resource of a cell i is calculated based on the average of the distances of the input data assigned to the cell from cell vector.

A cell which has the maximum Resource will expand. Therefore, the algorithm proceeds the cycle until each input data is associated with a single cell or it reach at the desired level of heterogeneity. Each adaptation cycle is contained a series of epochs. Each epoch consists of presentation of all the input data and each presentation has two steps. First, we find the best match cell which is known as winning cell. This is similar to the SOM. The cell that has the minimum distance with the input data is the best match cell/winning cell. The second is updating the reference vector wi of winning cell and its neighborhood using the following function:
$\Delta w = \phi\ t \times x - w$ (5)

Where $\phi(t)$ is the learning function:
$\phi(t) = \alpha \times \eta(t)$ (6)
$\eta(t)$ is function similar in SOM and α is a learning constant. For different neighbors α have different values. Two different neighborhoods are here. If the sibling of the winning cell is a cell, then the neighborhood includes the winning cell, the parent node and the sibling cell. On the other hand, it includes only the winning cell itself [4] (see Figure 3 (B)).
Furthermore, parameters αw, αm and α s are used for the winning cell, the ancestor node and the sibling cell, respectively. For example, values of αw, αm and αs can be set as 0.1, 0.05, and 0.01 respectively. Note that parameter values are not equal. This is because these non-equal values are critical to partition the input data set into various cells. A cycle is converged when the relative increase of total error falls below a certain threshold.

4.1.3. Modified SOTA

SOTA is specifically designed for molecular bio-sequence classification and phylogenetic analysis. Inspired by the self-organized tree structure and low time complexity we develop a modified self-organizing tree for image classification which is called MSOT.

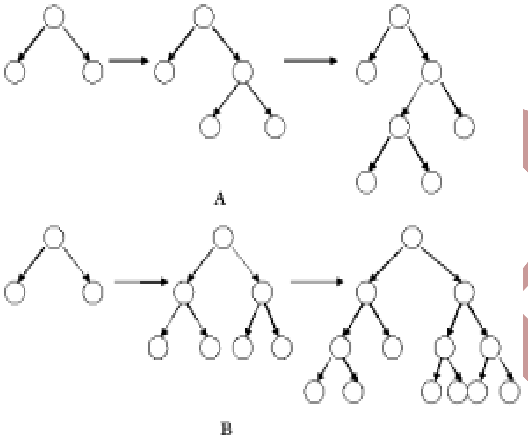

Figure 4. Ontology Construction A) SOTA B) MSOT

Although MSOT is similar to SOTA, it differs in two ways. First, in SOTA, during expansion, only one cell, which has the maximum resources, will be selected for expansion. On the other hand, in MSOT more than one cell may be selected for expansion, depending on their resources (see Figure 4). Here cells whose resources exceed a threshold will participate in the expansion. Thus, the aspect of threshold plays an important role here. Expansion of more than one node allows the algorithm to grow a tree quickly. Second, in SOTA during the expansion phase of a selected node two new cells will be created. Furthermore, initially each new cell will replicated with the same reference vector of the selected cell. Now, input data associated with selected node will be distributed between two new cells; the reference vector of each new cell will be updated. Therefore, input data will only be considered for the distribution of within two new cells, (i.e., locally). On the other hand, in MSOT more than one node may be selected, and two new cells will be created for each selected node. Now the question is how we can distribute input data of selected node among these new created cells. One approach is that input data of each selected node will be distributed to two new created children cells which are similar to the SOTA approach. This kind of distribution cannot provide a good cluster result because data may be poorly distributed between two new cells. Furthermore, once data is wrongly assigned in a group at earlier stage we cannot adjust them later. This is also one of the shortcomings of the classical HAC algorithm. The other approach is aggressive; input data of selected node will be distributed not only its new created cells but also its neighbor cells. For this, first we determine K level apart ancestor node of selected node. Next, we determine a sub-tree rooted by the ancestor node and input data of selected cell will be distributed among all cells (leaf) of this subtree. The latter approach is known as K level distribution (KLD).

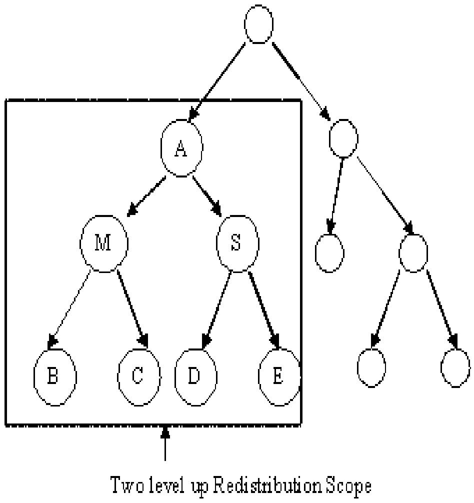

Figure 5. Two Level Distribution Scope of MSOT

For example, Figure 5 shows the scope of K=1. Now, we need to distribute data associated with node M to new created cells. For K=1, immediate ancestor of M will be determined which is A. Data of node M will be distributed to cells (i.e., B, C, D, and E) of a sub-tree rooted by A. For each data we will determine winning cell among B, C, D, and E using Equation 1. Note that if K=0, data of M will be distributed between cells B and C; the latter approach is simply turned into the former approach. After distributing each input data, winning cell and neighbor reference vector will be updated.

The pseudo code for the MSOT algorithm is as follows:

Step 1: [Initialization] initialize as like as SOTA—one node with two cells.

Step 2: [Distribution] distribute each input data between newly created cells; find the winning cell using KLD and assign the input data to this winning cell, update the reference vector of the winning cell and its neighbor using Equation 5 & 6.

Step 3: [Error] while error of the entire tree is larger than a threshold go back to Step 2.

Step 4: [Expand] for each cell, calculate resource, and check whether it exceeds a threshold. If yes, change this selected cell as node, and create two new children cells from it. If there are no more cells for expansion, the system is converged; else go back to Step 2.

Step 5: prune the tree and delete a cell that does not have any input on it.

CONCLUSION AND FUTURE WORK

In this paper we have proposed a potentially powerful and novel approach for the automatic construction of ontologies. The crux of our innovation is the development of a hierarchy, and the concept selection from WordNet for each node in the hierarchy. For developing a hierarchy we have modified the existing self-organizing tree (SOTA) algorithm that constructs a hierarchy from top to bottom; we have developed K-level Distribution (KLD) strategy. This algorithm improves the clustering result as compared to traditional hierarchical agglomerative clustering (HAC) algorithm. We would like to extend this work in the following directions. First, we would like to do more experiments for clustering and topic tracking techniques. Next, we would like to address this ontology construction in the domain of software component libraries.

ACKNOWLEDGMENT

We thank to Dr. Mayank Singh for his support and our University for motivating and encouraging doing our Research work in a Successful.

REFERENCES

[1] Shawkat Ali, A.B.W. (2008) K-means Clustering Adopting RBF-Kernel, Data Mining and Knowledge Discovery Technologies, David Taniar (Ed.), Pp. 118-142.

[2] Cao, T.H., Do, H.T., Hong, D.T. and Quan, T.T. (2008) Fuzzy named entity-based document clustering, Proceedings of IEEE International Conference on Fuzzy Systems, Hong Kong, Pp. 2028-2034.

[3] P. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proc. of KDD-1998, New York, NY, USA, August 1998*, pages 9–15, Menlo Park, CA, USA, 1998. AAAI Press.

[4] A. Hinneburg and D.A. Keim. Optimal gridclustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proc. of VLDB-1999, Edinburgh, Scotland, September 2000*. Morgan Kaufmann, 1999.

[5] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[6] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2), 2001.

[7] M. Devaneyand A. Ram. Efficient feature selection in conceptual clustering. In *Proc. of ICML-1997, Nashville, TN, 1998*. Morgan Kaufmann, 1998. G. Bisson, C. N_edellec, and D. Ca namero. Designing clustering methods for ontology building: The Mo'K workbench. pages 13{19, 2000. 3

[8] Macskassy, S. A., Banerjee, A., Davison, B., & Hirsh, H. (1998). Human performance on clustering web pages: a preliminary study. In *Proceedings of KDD-1998,* pages 264-268. AAAI Press.

[9]Schuetze, H. & Silverstein, C. (1997). Projections for efficient document clustering. In *Proceedings of SIGIR-1997,* pages 74-81. Morgan Kaufmann.

[10] A. Maedche and S. Staab, "The Text-To-Onto ontology learning environment,"in Proc. 8th Int.Conf. Conceptual Struct., Darmstadt, Germany,2000, pp. 14–18.

[11] D. Roussinov and H. Chen, "Document clustering for electronic meetings:An experimental comparison of two techniques," Decis. Support Syst.,vol. 27, no. 1/2, pp. 67–79, Nov. 1999.

[12] H. Li, K. Zhang, and T. Jiang, "Minimum entropy clustering and applications to gene expression analysis," in Proc. 3rd IEEE Comput. Syst. Bioinform. Conf., Stanford, CA, 2004, pp. 142–151.

[13] T. H. Cheng and C. P. Wei, "A clustering-based approach for integrating document-category hierarchies," IEEE Trans. Syst., Man, Cybern.A,Syst., Humans, vol. 38, no. 2, pp. 410–424, Mar. 2008.

[14] H. J. Kim and S. G. Lee, "An effective document clustering method using user- adaptable distance metrics," in Proc. ACM Symp. Appl.Comput.,Madrid, Spain, 2002, pp. 16–20.

[15] Fabiano D. Beppler,"An Architecture for an Ontology-Enabled Information Retrieval"

[16] "Automatic Ontology Generation:State of the Art "Ivan Bedini, Benjamin Nguye, Orange.