A Dynamic and Semantically-Aware Technique for Document Clustering in Biomedical Literature

Min Song, New Jersey Institute of Technology, USA
Xiaohua Hu, Drexel University, USA
Ilhui Yoo, University of Missouri, USA
Eric Koppel, New Jersey Institute of Technology, USA

ABSTRACT

As an unsupervised learning process, document clustering has been used to improve information retrieval performance by grouping similar documents and to help text mining approaches by providing a high-quality input for them. In this article, the authors propose a novel hybrid clustering technique that incorporates semantic smoothing of document models into a neural network framework. Recently, it has been reported that the semantic smoothing model enhances the retrieval quality in Information Retrieval (IR). Inspired by that, the authors developed and applied a context-sensitive semantic smoothing model to boost accuracy of clustering that is generated by a dynamic growing cell structure algorithm, a variation of the neural network technique. They evaluated the proposed technique on biomedical article sets from MEDLINE, the largest biomedical digital library in the world. Their experimental evaluations show that the proposed algorithm significantly improves the clustering quality over the traditional clustering techniques including k-means and self-organizing map (SOM).

Keywords: document clustering, feature selection, neural network

INTRODUCTION

Document clustering, unlike document classification, is an unsupervised learning process meaning that there is no known information about documents including the number of document groups (usually called $k$). Document clustering organizes textual documents into meaningful groups that represent topics in document collections without any known information about a document set. As a result, the documents in a document cluster are similar

DOI: 10.4018/jdwm.2009080703

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
to one another while documents from different clusters are dissimilar.

Document clustering was originally studied to enhance the performance of information retrieval (IR) because similar documents tend to be relevant to the same user queries (Wang et al., 2002; Zamir & Etzioni, 1998). Document clustering has been used to facilitate nearest-neighbor search (Buckley & Lewit, 1985), to support an interactive document browsing paradigm (Cutting et al., 1992; Gruber, 1993; Koller & Sahami, 1997; Gruber, 1993), and to construct hierarchical topic structures (van Rijsbergen, 1979). Thus, document clustering plays a more important role for IR and text mining communities since the most natural form for storing information is text and text information has increased exponentially.

In the biomedical domain, document clustering technologies have been used to facilitate the practice of evidence-based medicine. This is because document clustering enhances biomedical literature searching (e.g., MEDLINE searching) in several ways and literature searches are one of the core skills required for the practice of evidence-based medicine (Evidence-based Medicine Working Group, 1992). For example, Pratt and her colleagues (Pratt et al., 1999; Pratt & Fagan, 2000), and Lin and Demner-Fushman (2007) introduced interesting semantic document clustering approaches that automatically cluster biomedical literature (MEDLINE) search results into document groups for the better understanding of literature search results.

Current information technologies allow us to acquire, store, archive, and retrieve documents electronically. To this end, document clustering has been given focal attention because document clustering assists users in discovering hidden similarities and key concepts in documents. One of the most serious problems making document clustering difficult to deal with text information is that the size of text collections in digital libraries is increasing rapidly. To handle the increasing size of document collections, a clustering algorithm has to not only solve the incremental problem but it must also have high efficiency in a large dataset.

Most document clustering algorithms require a form of data pre-processing including stop-word removal and feature selection. Through the data pre-processing, unimportant features are eliminated and the original dimension is reduced to a more manageable size. However, the data pre-processing has two problems. First, although the data pre-processing can reduce the original dimension size, the reduced dimension is still sparse, which is called "the curse of dimensionality". As the result, clustering results are often low quality. Second, the reduction of dimensionality by the data-preprocessing may disturb the preservation of the original topological structure of the input data.

To solve these problems, we propose a context-sensitive semantic smoothing of a document model and incorporate it into Dynamic Growing Cell Structure (DynGCS). The effect of model smoothing has not been extensively studied in the context of document clustering (Zhang et al., 2006). Most model-based clustering approaches simply use Laplacian smoothing to prevent zero probability (Nigam & McCallum, 1998; Zhong & Ghosh, 2005), while most similarity-based clustering approaches employ the heuristic TF*IDF scheme to discount the effect of general words (Steinbach et al., 2000). As showed in (Zhong & Ghosh, 2005), model-based clustering has several advantages over discriminative based approaches. One of the advantages of model-based approaches is that it learns generative models from the documents, with each model representing one particular document set. Due to the promising results reported in model-based clustering approaches, we propose a novel semantic smoothing technique to improve clustering quality.

DynGCS is an adaptive variant of an artificial neural network model, Self-Organizing Map (SOM), which is well suited for mapping high-dimensional data into a 2-dimensional representation space. The training process is based on weight vector adaptation with respect to the input vectors. SOM has shown to be a highly
effective tool for document clustering (Kohonen et al., 1999). One of the disadvantages of SOM in document clustering is its fixed size in terms of the number of units and their particular arrangement, which must be defined prior to the start of the training process. Without knowing the type and the organization of documents, it is difficult to get satisfying results without multiple training runs using different parameter settings. This is extremely time consuming, given the high-dimensional data representation. DynGCS solves the problem of fixed-sized structure by dynamically generating multiple layers of SOM.

The organization of this article is as follows: Section 2 describes semantic smoothing of document model. Section 3 denotes the DynGCS algorithm. Section 4 presents results. We conclude our article in Section 5.

RELATED WORK

In this section, we highlight work done on document clustering. Jain et al. (1999) provided an extensive survey of various clustering techniques. Zamir and Etzioni (1998) provided a survey on applying hierarchical clustering algorithms into clustering documents. Hierarchical clustering algorithm has been widely used among the numerous document clustering algorithms. Several variants from this algorithm include single-link, group-average and complete-link. Recently, partitional clustering algorithms were proposed to cluster large document datasets due to their relatively low computational requirements (Eissen & Potthast, 2005; Larsen & Aone, 1999; Aggarwal et al., 1999; Allen & Littman, 1993).

Cutting et al. (1992) adapted two partition-based clustering algorithms, Buckshot and Fractionation, to clustering documents. Buckshot selects a small sample of documents to pre-cluster them using a standard clustering algorithm and assigns the rest of the documents to the clusters formed. Fractionation splits the \( N \) documents into \( m \) buckets where each bucket contains \( N/m \) documents.

Zamir et al. introduce the notion of phrase-based document clustering. They use a generalized suffix-tree to obtain information about the phrases and use them to cluster the documents (Zamir et al., 1997).

According to Allen and Littman, the two cluster selection methods of BiSecting K-means that are used to select the cluster to be bisected, do not significantly affect clustering quality; the two methods are selecting the largest cluster and the cluster with the least overall similarity (Allen & Littman, 1993). We believe the choice of the cluster selection methods does affect the clustering quality because the choice may lead to different document clustering results. Allen and Littman (1993) and Beil et al. (2002) show BiSecting K-means is better than K-means while Pantel and Lin (2002) shows K-means is superior to Bisecting K-means.

Suffix tree document model was firstly proposed in 1997 (Nigam & McCallum, 1998; Zamir et al., 1997). Different from document models, which treat a document as a set of words and ignores the sequence order of the words, suffix tree document model considers a document to be a set of suffix substrings such that common prefixes of the suffix substrings are selected as phrases to label the edges of a suffix tree. The STC algorithm is based on this model and works well in clustering Web document snippets returned from several search engines. However, the properties of the suffix tree model and STC have not been analyzed in their papers (Zamir et al., 1997). Eissen et al. continued their work and pointed out that the STC algorithm is a fusion heuristic that efficiently evaluates the graph-based similarity measure for large document collections (Eissen et al., 2005). Furthermore, they also proposed several new graph-based similarity measures to compute document similarities. Their experimental evaluation results showed that the similarity measures, especially the hybrid similarity measure, had achieved significant performance improvements in the MajorClust algorithm and the GAHC algorithm.

According to Zamir et al., Suffix Tree Clustering (STC) provides better clustering...
quality for web documents than K-means in terms of precision (Zamir et al., 1997). On the other hand, STC showed poor clustering results for Eissen et al. (2005). Larsen and Aone (1999) claim hierarchical clustering is better than K-means based on the experiments where one document set is used, while Zhao and Karypis (2002) and Steinbach et al. (2000) indicate BiSecting K-means and K-means are better than hierarchical clustering.

Most document clustering studies (Beil et al., 2002; Cutting et al., 1992; Hotho et al., 2002; Larsen & Aone, 1999; Pantel & Lin, 2002; Steinbach et al., 2000; Eissen et al., 2005) uses only 1000 to 20,000 documents in their experiments. To test the scalability of document clustering approaches, much larger document sets are required. Hotho et al. (2002) claim the use of ontology may improve document clustering. However, the authors used their own manually modeled ontology for tourism domain for document clustering.

SEMANTIC SMOOTHING OF DOCUMENT MODELS

In document clustering, a TF*IDF score is often used as the dimension values of document vectors. In the context of language model, a TF*IDF scheme is roughly equivalent to the background model smoothing. Since TF*IDF is a pure probabilistic scheme, it does not convey semantics of content represented by terms and phrases. As an alternative, (Lafferty & Zhai, 2001) proposes semantic smoothing where context and sense information are incorporated into the model. Latent Semantic Indexing (LSI) is an early attempt at semantic smoothing which projects documents in a corpus into a reduced space where document semantics becomes clear. LSI explores the structure of term co-occurrence with Singular Value Decomposition (SVD). However, the problem of LSI is that it increases noise while reducing the dimensionality because it is unable to recognize polysemy. In practice, it is also criticized for the lack of scalability and ability to interpret.

Our semantic smoothing technique is similar to the one proposed in (Zhang et al., 2006). Their approach utilizes multi-word phrases (e.g. "star war", "movie star") as topic signatures. Using multi-word phrases has several advantages: 1) a multi-word phrase is often unambiguous; 2) multi-word phrases can be extracted from a corpus by existing statistical approaches without human knowledge; and 3) documents are often full of multi-word phrases; thus, it is robust to smooth a document model through statistical translation of multi-word phrases in a document to individual terms. Unlike (Zhang et al., 2006), we employ an information gain-based keyphrase extraction technique (Song et al., 2003) to generate a set of phrases in documents that achieves competitive performance in biomedical data collections.

Our keyphrase extraction procedure consists of two stages: building an extraction model and extracting keyphrases. The extraction model is trained for keyphrases before the keyphrase extraction technique is applied. Keyphrases are extracted by referencing the keyphrase model, which was built based on a decision tree technique (Song et al., 2003). Both training and test data are processed by the following three components: 1) Data Cleaning, 2) Data Tokenizing, and 3) Data Discretizing. After that, there are three features to calculate information gain: 1) TF-IDF, 2) Part-OF-Speech (POS), and 3) First Occurrence of Phrases. Based on these features, we build a discretization table and rank the candidate keyphrases based on its information gain measure. We compared information gain with other techniques such as Naïve Bayesian used in KEA (Willett et al., 1999) and found that information gain gave us the best performance in our previous experiments. In addition, we adopted the Mahalanobis Distance to measure distance among models. Mahalanobis distance takes into account the covariance among the variables in calculating distances (Pearson, 2005). Suppose we have indexed all documents in a given collection C with terms and phrases as illustrated in Figure 1. Note that Vp denotes phrase vector, Vw denotes word vector, and Vd denotes document vector. The translation
probabilities from a keyphrase \( t_k \) to any individual term \( w \), denoted as \( p(w \mid t_k) \), are also given. Then we can easily obtain a document model below:

\[
p(w \mid d) = \sum_{k} p(w \mid t_k) p_m(t_k \mid d)
\]  

(1)

The likelihood of a given document generating the keyphrase \( t_k \) can be estimated with

\[
p_m(t_k \mid d) = \frac{c(t_k, d)}{\sum_{t} c(t, d)}
\]  

(2)

where \( c(t, d) \) is the frequency of the keyphrase \( t_k \) in a given document \( d \).

We refer to the above model as translation model indicated in (Lafferty & Zhai, 2001). As we discussed in the introduction, the translation from multi-word phrases to individual terms would be very specific. Thus, the translation model not only weakens the effect of “general” words, but also relieves the sparseness of class-specific “core” words. However, not all topics in a document can be expressed by key phrases. If only the translation model is used, there will be serious information loss. A natural extension is to interpolate the translation model with a unigram language model below:

\[
p_{\alpha}(w \mid d) = (1 - \alpha)p_m(w \mid d) + \alpha p(w \mid C)
\]  

(3)

\( P_{m}(w \mid d) \) is a maximum likelihood estimator. We refer to this unigram model as simple language model or baseline language model. We use Jelinek-Mercer smoothing on the purpose of further discounting “general” words that do not convey the high discriminated power.

The final document model for clustering use is described in equation (4). It is a mixture model with two components: a simple language model and a translation model.

\[
p_{\alpha}(w \mid d) = (1 - \lambda)p_{\xi}(w \mid d) + \lambda p_{\mu}(w \mid d)
\]  

(4)

The translation coefficient \( \lambda \) is used to control the influence of two components in the mixture model. In our experiment, we set the background translation coefficient to 0.5. With training data, the translation coefficient can be trained by optimizing the clustering quality. After estimating a language model for each document in the corpus with context-sensitive semantic smoothing, we use the Mahalanobis Distance of two language models as the distance.

Figure 1. Phrase/term and document relation
measure of the corresponding two documents. Given two probabilistic document models
$p(w \mid d_x)$ and $p(w \mid d_y)$, the Mahalanobis Distance of $p(w \mid d_x)$ to $p(w \mid d_y)$ is defined as:

$$d(p_x, p_y) = \sqrt{\sum_{i=1}^{D} \frac{(p_x - p_y)^2}{\sigma_i^2}}$$

(5)

where $\sigma$ is the standard deviation of the $p_x$ over the sample set.

**DYNGCS**

DynGCS is a self-organizing neural network model that incrementally builds a Dirichlet Voronoi tessellation of input space while automatically finding its structure and size. DynGCCs are used in artificial neural network (ANN) architectures. ANN can allow clusters to search and reduce search time. Unsupervised ANN learning for document clustering offers a number of advantages. Most commonly unsupervised ANN learning methods are Hard Competitive Learning (HCL), Self-Organizing Map (SOM) algorithm, and Adaptive Resonance Theory (ART). All these methods have one thing in common: the winner-takes-all concept. HCL and SOM are dependent on input data density, therefore it would not be wise to use these for purpose of improved information clustering and retrieval.

The main drawback of the existing algorithms is that they require either pre-specification of the number of clusters (K-means clustering and SOM), or have the user decide the number of clusters to the user (hierarchical clustering). Particularly for a large dataset, SOM suffers from its fixed network architecture, which has motivated the development of a number of adaptive variants (Zhong & Ghosh, 2005).

In this article, we introduce an adaptive variant of SOM to resolve the aforementioned issue. DynGCC is an unsupervised clustering method that adaptively decides on the best architecture for the self-organizing map. This stems from Growing Cell Structure (GCS), introduced by Fritzke (1994, 1995). Fritzke introduced an incremental self-organizing network with variable topology, known as GCS based on SOM and Hebbian learning. The GCS has three main advantages over the SOM: first, the network structure is determined automatically by the input pattern. Second, the network size needs not to be predefined. Third, all parameters of the model are constant. Therefore, a “cooling schedule” is not required which is a contrast to the conventional SOM. The problem with GCS, however, is that it tends to overspill as the map grows larger.

To tackle the problem of overspill, we combine GCS with the Growing Hierarchical Self-Organizing Map (GH-SOM). GH-SOM adopts a hierarchical structure with multiple layers, where each layer consists of a set of independent self-organizing maps (Dittenbach et al, 2000). With the probability of adding every cell in a SOM from one layer to the next layer of the hierarchy, it shares the sample adaptation steps with GCS. There is, however, one exception that uses a decreasing learning rate and a decreasing neighborhood radius. The mean quantization error of the map is used to decide whether a new level of the hierarchy needs to be created. For instance, at level 0, the single SOM unit is assigned a weight vector $m_0$, such that $m_0 = [\mu_0, \mu_0, ..., \mu_0]$ is computed as the average of all input data. The mean quantization error of this single unit is computed as the following with $d$ representing the number of input data $x$:

$$m_{qe_0} = \frac{1}{d} || m_0 - x ||$$

(6)

DynGCS produces distribution-preserving mappings as other SOM related algorithms. DynGCS operates on the following principle of GCS (Bruske & Sommer, 1995):

1. During the training stage, the number of clusters and the connections among them are dynamically assigned.
2. Adaptation strength is constant over time.
3. Adaptation occurs only in the best-matching cell (BMU) and its neighborhood.
4. Adaptation increments the signal counter for BMU and decrements the remaining cells.
5. For the adaptation of the output map to the distribution of the input vectors, insertion of new cells and deletion of existing cells occur.

The DynGCS hierarchy is constructed from and superimposed onto the standard GCS algorithm detailed above (Figure 2). DynGCS starts with the small architecture. The DynGCS is a hierarchical self-organizing neural network designed to preserve topological structure of input data based on GCS and Growing Hierarchical Self-Organizing Map. DynGCS grows dynamically. In each growth, the DynGCS adds two children to the leaf whose heterogeneity exceeds a threshold and turns it to a node. This process goes on until the heterogeneity of all cells is less than the threshold. A learning process similar to GCS is adopted.

The DynGCS algorithm is shown in Table 1. Initially there is only one root node. All the input data is linked to the root. The reference vector of the root node is initialized with the centroid of the data. In growth mode, two child nodes are appended to the root node. All input data linked to the root node is distributed between these child nodes by employing a learning process. Once the learning process is finished, the heterogeneities of the leaf nodes are scrutinized to decide whether expansion to another level is necessary. If another expansion is needed, then a new growth step is invoked. Two child nodes are appended to the leaf nodes if the level of heterogeneity is greater than the threshold. All input data is distributed again with the learning process and a new growth begins. This process continues until the heterogeneity of all the leaves is less than the threshold. In DynGCS, each leaf represents a cluster that includes all data linked to it. The reference vector of a leaf is the

Figure 2. Hierarchical structure of DynGSC
centroid of all data linked to it. Therefore, all reference vectors of the leaves form a Voronoi set of the original dataset. Each internal node represents a cluster that includes all data linked to its leaf descendants. The reference vector of an internal node is the centroid of all data linked to its leaf descendants.

The height of DynGCC is \( \log_d M \), where \( d \) is the branch factor and \( M \) is the number of nodes in the hierarchy. \( M \) is \( O(N) \) where \( N \) is the number of data. Let \( J \) be the average number of learning iterations for each learning process. Thus, the time complexity factor for DynGCC will be \( O[\log_d N \times (J \times N + d \times J \times N)] \). Since \( J \) and \( d \) are constants, the complexity will be \( O(cN \times \log_d N) \). The metric, misclassification index (MI), purity, confusion matrix, F-measure, and Entropy. In our experiment we used misclassification index, purity, and Entropy as clustering evaluation metrics. MI is the ratio of the number of misclassified objects to the size of the whole dataset; thus, MI with 0% means the perfect clustering.

\[
MI = \frac{\text{no. misclassified objects}}{\text{total no. of objects}}
\]

The cluster purity indicates the percentage of the dominant class members in a given cluster; the percentage is nothing more than the maximum precision over the classes.

\[
Purity = \frac{1}{n} \max_i \{ \text{precision}(i, j) \},
\]

where \( n \) is the number of documents.

The entropy of a cluster implies how the members of the \( k \) classes are distributed within each cluster. We use weight average entropy as an overall clustering metric as shown below:

\[
Entropy = -\sum_{j} \frac{1}{n} \sum_{i} P(i, j) \times \log_2 P(i, j)
\]

### EXPERIMENTS

In this section, we report our evaluation method, data collection, and experiment results.

### Evaluation Method

We evaluated the proposed algorithm by comparing clustering output with known classes as answer keys. There have been a number of comparison metrics, such as mutual information

<table>
<thead>
<tr>
<th>SM/DynGCS Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. /** Parse input data ***/</td>
</tr>
<tr>
<td>2. /** Apply information gain based keyphrase extraction techniques to input data ***/</td>
</tr>
<tr>
<td>3. /** Apply semantic smoothing technique to documents ***/</td>
</tr>
<tr>
<td>4. /** Apply Mahalanobis Distance to feature vectors ***/</td>
</tr>
<tr>
<td>5. /** Initialization ***/</td>
</tr>
<tr>
<td>6. <strong>Do</strong></td>
</tr>
<tr>
<td>7. For any leaf whose heterogeneity is greater than the threshold</td>
</tr>
<tr>
<td>8. Changes the leaf to a node and create two descendent leaves.</td>
</tr>
<tr>
<td>9. Initialize the reference vector of the new leaves with the node’s reference vector</td>
</tr>
<tr>
<td>10. Set the cell growing flag of the new leaves to true</td>
</tr>
<tr>
<td>11. <strong>Do</strong></td>
</tr>
<tr>
<td>12. For each input data</td>
</tr>
<tr>
<td>13. Find BMU winner</td>
</tr>
<tr>
<td>14. Update reference vectors of winner and its neighborhood</td>
</tr>
<tr>
<td>15. Increase time parameter, ( t = t + 1 )</td>
</tr>
<tr>
<td>16. <strong>While</strong> the cell growing flag of all lowest level node are false</td>
</tr>
<tr>
<td>17. <strong>While</strong> the heterogeneity of all leaf nodes are less than the threshold</td>
</tr>
</tbody>
</table>

---

Table 1. SM/DynGCS algorithm
where \( p(i, j) \) is precision\((i, j) \) and \( n \) is the number of documents.

Note that the smaller \( MI \) and Entropy imply the better clustering quality while the larger purity indicates the better clustering quality.

**Data Collections**

We used public MEDLINE data for the experiments by collecting document sets related to various diseases. We use the “MajorTopic” tag along with the MeSH disease terms as queries to MEDLINE. Table 2 shows the document sets used in our experiments.

Each corpus name in Table 3 indicates the number of document sets (i.e. \( k \)) used for the corpus generation, as well as what document sets are used (document set IDs (see Table 2) are delimited by “-”). The format of the corpus ID is [Ck.n], where \( k \) is the number of document sets (classes) and \( n \) is a sequence number.

Once we retrieve the datasets, we generate various document combinations by randomly mixing the document sets whose numbers of classes are 2 to 12 (Table 3).

The document sets used for generating the combinations are later used as answer keys on the performance measure. Refer to (Yoo et al, 2007) for details on data collections and test data sets.

**Experiment Results**

In this section, we will report experiment results with average (\( \mu \)) and standard deviation (\( \sigma \)). In our experiments, we compared our method, SM/DynGCS with three other algorithms such as TF*IDF/DynGCS, K-means, and SOM.

**TF*IDF/DynGCS**: TF*IDF, which is widely used in information retrieval, was implemented for feature vectors. It is a measure of importance for a term in a document or class. As indicated by formula 7, TF*IDF is a term frequency in a document or class, relative to overall frequency. The TF*IDF feature is a well-known weighting scheme in information retrieval.

\[
W_{ij} = tf_{ij} \cdot \log_{\theta} \frac{N}{n_{i}}
\]

(7)

\( W_{ij} \) weight of term \( t_{i} \) in document \( D_{j} \), and \( tf_{ij} \) is frequency of term \( t_{i} \) in document \( D_{j} \). \( N \) is the number of documents in a collection and \( n \) is the number of documents where term \( t_{i} \) occurs at least once.

<table>
<thead>
<tr>
<th>Document Sets</th>
<th>ID</th>
<th>No. of Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otitis</td>
<td>Ot</td>
<td>5,233</td>
</tr>
<tr>
<td>Osteoarthrosis</td>
<td>OA</td>
<td>8,987</td>
</tr>
<tr>
<td>Osteoporosis</td>
<td>Ost</td>
<td>8,754</td>
</tr>
<tr>
<td>Migraine</td>
<td>Mg</td>
<td>4,174</td>
</tr>
<tr>
<td>Coronary Heart Disease</td>
<td>CHD</td>
<td>53,664</td>
</tr>
<tr>
<td>Breast Neoplasm</td>
<td>Bre</td>
<td>56,975</td>
</tr>
<tr>
<td>Depressive Disorder</td>
<td>Dep</td>
<td>19,926</td>
</tr>
<tr>
<td>AIDS</td>
<td>AIDS</td>
<td>19,671</td>
</tr>
<tr>
<td>Alzheimer Disease</td>
<td>Alz</td>
<td>18,033</td>
</tr>
<tr>
<td>Diabetes Type 2</td>
<td>Diab</td>
<td>18,726</td>
</tr>
<tr>
<td>Age-related Macular Degeneration</td>
<td>AMD</td>
<td>3,277</td>
</tr>
<tr>
<td>Parkinson Disease</td>
<td>Pk</td>
<td>9,933</td>
</tr>
</tbody>
</table>

Table 2. Document collections and their size

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Table 3. Overview of test data

<table>
<thead>
<tr>
<th>Corpus Name</th>
<th>ID</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>2_Bre-CHD</em></td>
<td>C2.1</td>
<td>110k</td>
</tr>
<tr>
<td><em>4_Dep-AIDS-Alz-Diab</em></td>
<td>C4.2</td>
<td>76k</td>
</tr>
<tr>
<td><em>2_Mg-Alz</em></td>
<td>C2.3</td>
<td>22k</td>
</tr>
<tr>
<td><em>3_OA-Ost-Pk</em></td>
<td>C3.3</td>
<td>28k</td>
</tr>
<tr>
<td><em>4_Alz-AMD-Ost-Ost</em></td>
<td>C4.1</td>
<td>35k</td>
</tr>
</tbody>
</table>

SOM: Self-Organizing Map (SOM) is a well accepted neural network technique in document clustering. In this experiment, we used 10,000 iterations and set the initial learning rate to 0.1. The 10 x 10 SOM is trained to cluster and the final number of cells is 250.

K-Means: K-Means is a simple but powerful unsupervised learning algorithm that solves the well known clustering problem. Because K-Means may produce different clustering results every time due to its random initializations, we ran it five times and averaged the values of clustering evaluation metrics.

Table 4 shows the comparison of the overall clustering quality of SM/DynGCS, TF*IDF/DynGCS, K-means, and SOM. The clustering results are from 5 datasets from which the averages and standard deviations are calculated. We notice that SM/DynGCS is superior to the hierarchical algorithms.

Comparing miscalculation indexes, SM/DynGCS outperforms the other three. Compared to K-means, it is 0.16% better in terms of μ.

In terms of entropy, the best performance was made by SM/DynGCS. SM/DynGCS improves accuracy by about 0.14% when compared to K-means. TF*IDF/DynGCS and SOM perform almost at the same level. With purity, the results show that SM/DynGCS performs 0.14% better than K-means. As indicated by the results, integrating semantic smoothing into clustering algorithm improves accuracy more than DynGCS with TF*IDF does.

Figure 3 shows how four different clustering techniques perform on five different datasets. Unlike the other three techniques, TF*IDF/DynGCS, K-means, and SOM, the performance of semantic smoothing-based DynGCS is stable across five datasets.

Figure 4 also indicates that the semantic smoothing-based DynGCS outperforms the other three techniques, TF*IDF/DynGCS, K-means, and SOM in terms of entropy measure. The superior performance of our semantic smoothing-based DynGCS is observed by MI measure (Figure 5).

Table 4. Comparison of evaluation metrics

<table>
<thead>
<tr>
<th></th>
<th>MI</th>
<th>Entropy</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF*IDF/DynGCS</td>
<td>μ:0.23</td>
<td>σ:0.13</td>
<td>μ:0.26</td>
</tr>
<tr>
<td>SM/DynGCS</td>
<td>μ:0.35</td>
<td>σ:0.17</td>
<td>μ:0.31</td>
</tr>
<tr>
<td>K-means</td>
<td>μ:0.19</td>
<td>σ:0.12</td>
<td>μ:0.17</td>
</tr>
<tr>
<td>SOM</td>
<td>μ:0.28</td>
<td>σ:0.13</td>
<td>μ:0.27</td>
</tr>
</tbody>
</table>

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
CONCLUSION

In this article, we proposed a hybrid clustering algorithm that combines semantic smoothing of document model and dynamic growing cell structure. We developed a context-sensitive smoothing method for document models and used Mahalanobis distance for smoothed probabilistic models as a document similarity measure for clustering. These feature vectors were used as input for document clustering. Our document clustering technique combined GCS with the Growing Hierarchical Self-Organizing Map (GH-SOM). GH-SOM adopts a hierarchical structure with multiple layers, where each layer consists of a set of independent self-organizing maps.

We performed a comparison study of SM/DynGCS with the other three techniques: TF*IDF/DynGCS, K-means, and SOM on 5 MEDLINE corpora. The experimental results indicated that our SM/DynGCS is superior to the other three approaches in terms of $\mu$ and $\sigma$. For future work, we will conduct more comprehensive experiments including other model-based partitional approaches. We also plan to apply our technique to dynamic document clustering, i.e. clustering search results.

REFERENCES

Figure 5. Performance comparison on different datasets by MI


Min Song is an assistant professor of Department of Information Systems at NJIT. He received his MS in School of Information Science from Indiana University in 1996 and received PhD degree in information systems from Drexel University in 2005. Min has a background in text mining, bioinformatics, information retrieval and information visualization. Song received the Drexel Dissertation Award in 2005. In 2006, Song's work received an honorable mention award in the 2006 Greater Philadelphia Bioinformatics Symposium. In addition, The paper entitled Extracting and Mining Protein-protein interaction Network from Biomedical Literature has received the best paper award from 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, which was held in San Diego, USA, Oct. 7-8, 2004. In addition, another paper entitled Ontology-based Scalable and Portable Information Extraction System to Extract Biological Knowledge from Huge Collection of Biomedical Web Documents was nominated as the best paper at 2004 IEEE/ACM Web Intelligence Conference, which was held in Beijing, China, Sept. 20-24, 2004.

Xiaohua (Tony) Hu is currently an associate professor (early tenured in 2007) and the founding director of the data mining and bioinformatics lab at the College of Information Science and Technology. He is the now also serving as the IEEE Computer Society Bioinformatics and Biomedicine Steering Committee Chair, and the IEEE Computational Intelligence Society Granular Computing Technical Committee Chair (2007-2008). Tony is a scientist, teacher and entrepreneur. He joined Drexel University in 2002, founded the International Journal of Data Mining and Bioinformatics (SCI indexed) in 2006, International Journal of Granular Computing, Rough Sets and Intelligent Systems in 2008. Earlier, he worked as a research scientist in the world-leading R&D centers such as Nortel Research Center, GTE labs and HP Labs. In 2001, he founded the DMW Software in Silicon Valley, California. His research ideas have been integrated into many commercial products and applications.

Illhoi Yoo is an assistant professor of the Department of Health Management and Informatics, University of Missouri School of Medicine. He gained his MS degree in information science from University of Pittsburgh and his PhD in information science and technology from Drexel University. His research interest is in semantic-oriented MEDLINE search and biomedical literature mining using biomedical ontologies. He has published 35 peer reviewed papers in various journals, conferences and book chapters. He won the best paper award as the co-author of a paper published in the Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology on Oct. 7, 2004.

Eric Koppel is an undergraduate at the New Jersey Institute of Technology, working toward a BS in Computer Science. His research interests include knowledge discovery, pattern mining and Web 2.0 technologies.