# Document Clustering Using the LSI Subspace Signature Model

**W.Z. Zhu and R. B. Allen**

College of Information Science and Technology, Drexel University, Philadelphia, USA

*weizz2004@yahoo.com and rba@boballen.info*

We describe the Latent Semantic Indexing Subspace Signature Model (LSISSM) for semantic content representation of unstructured text. Grounded on Singular Value Decomposition (SVD), the model represents terms and documents by the distribution signatures of their statistical contribution across the top-ranking latent concept dimensions. LSISSM matches term signatures with document signatures according to their mapping coherence between LSI term subspace and LSI document subspace. LSISSM does feature reduction and finds a low-rank approximation of scalable and sparse term-document matrices. Experiments demonstrate that this approach significantly improves the performance of major clustering algorithms such as standard K-means and Self-Organizing Maps compared to Vector Space Model (VSM) and the traditional LSI model. The unique contribution ranking mechanism in LSISSM also improves the initialization of standard K-means compared to random seeding procedure which sometimes causes low efficiency and effectiveness of clustering. A two-stage initialization strategy based on LSISSM significantly reduces the running time of standard K-means procedures.

## 1. Introduction

Document clustering faces several challenges such as feature selection, feature and document representation, determining the optimal number of the clusters, clustering accuracy and efficiency, labeling of clusters, scalability, and cluster similarity/distance. Targeting these issues, we propose a semantic representation model motivated by the LSI Probabilistic Model (Ding, 2005) along with the visual patterns of the concept mapping and the distribution of term contribution and document contribution on the latent concept dimensions explored by the Storylines interface (Zhu & Chen, 2007). The model is divided into three components: LSI subspace signature content representation, signature ranking for dimension reduction and novel signature similarity measures. Each term and each document is defined by an LSI subspace signature which represents the distribution of its local statistical contribution on the top-ranking LSI latent concept dimensions. Then, novel similarity measures bridge the term signatures and the document signatures. Finally, the initial term-document matrix is transformed into a low-rank approximation using the latent concept dimensions as intermediate layers and bootstrapping of term rankings to control information loss.

We evaluate the effectiveness and efficiency of the model for document clustering. Specifically, this paper examines how LSISSM may improve the performance of major clustering algorithms, such as standard K-means and Self-Organizing Maps (SOM) and how LSISSM solves the initialization problem of standard K-means (Dubes & Jain, 1988).

The remainder of this paper is organized as follows: Section 2 reviews various data models for semantic content representation and document clustering. Section 3 describes the three components of LSISSM. Section 4 presents the methodology of the LSISSM-based clustering applications and evaluation methods. Section 5 lists the data sets, procedures of text pre-processing and the experiment results. Finally, Section 6 discusses of the results and summarizes the work.

## 2. Literature Review

Here, we review previous approaches to semantic content representation and the application of semantic representation to document clustering.

### *2.1 Data Models for Semantic Content Representation*

The Vector Space Model (VSM, Salton et al., 1975) often uses term frequency or tf*idf scores (TFIDF) to weigh the associations between terms and documents. It does not capture the semantic relationships of concepts effectively and does not reduce dimensionality systematically. Richer representations can overcome some of the drawbacks of VSM. These semantic data models represent each topic as a unified signature which is a weighted vector. These signatures show how terms and documents are related and can be used to disambiguate those terms and documents. Similar signatures generally reflect the semantic relationships. There are three major types of statistical semantic content representation: LSI, probabilistic topic models and connectionist models.

### 2.1.1. Latent Semantic Indexing and Related Models

Traditional LSI (Deerwester et al., 1990; Koll, 1979) is the truncated Singular-Value Decomposition (SVD) which decomposes the term-document matrix $A$ into a term-concept matrix $U$, a diagonal singular value matrix $C$ and document-concept matrix $V$:

$$A = UCV^T \tag{1}$$

The entry values in matrix $A$ are term frequencies or TFIDF scores. The top $K$ dimensions of $A$ and $V$ are selected as the best proximity of the original matrixes. Each term or each document is represented by a vector which lists its least square distances to the top $K$ dimensions.

Traditional LSI has problems related to information loss, noise reduction and the interpretation of the latent concept dimensions. Information loss occurs when the number of the LSI latent concept dimensions selected for the low-rank approximate of the LSI subspaces, the $K$ values, is small. Empirically, the number of the latent concept dimensions is usually less than 400 even when the corpus contains thousands of documents. The values of SVD represent many of noisy and unimportant term relationships. Kontostathis et al. (2005) demonstrate that removing 70% of the SVD entries does not affect information retrieval performance. These values have a strong correlation with term second-order co-occurrence (Kontostathis & Pottenger, 2006) which can be either semantic relations or noise. In addition, the partial SVD processes such as "folding in" are sources of information loss. The "folding up" approaches (Mason & Spiteri, 2008; Tougas & Spiteri, 2006, 2007) adaptively combine matrix updating with folding in and use an error measure to control the accuracy loss of partial SVD. Another consideration is that the traditional LSI latent concept dimensions often do not correlate to topics. Because the projection scores in the LSI subspaces can be either positive or negative, they do not have direct statistical meanings. A group of terms that has higher projection scores in the same dimensions with the same sign might be associated with one topic and its context. But, two different topics often map to the same latent concept dimension with different signs and each latent concept dimension can be a mixture of topics.

Non-negative Matrix Factorization (Xu et al., 2003; Xu et al., 2004) captures the topics with the positive projection on the latent concept dimensions and linearly combines these topics to represent the documents. Cai et al. (2005) propose an unsupervised linear discriminant analysis method, so called Local Preserving Indexing (LPI). This model projects the documents to a lower dimension space and captures both geometric and discriminating structures. However, it still has to determine how many dimensions to use in the subspace.

Ding (2005) developed a probabilistic model which provides insight for LSI representations based on the dual relationship between words and documents. The model has established the contribution of dimensions to the latent semantic space. Specifically, the singular value squared is the contribution of the corresponding dimension. This quadratic dependence means that traditional LSI dimensions with small singular values are overrepresented by the linear relationship. Furthermore, Ding demonstrated that the importance of LSI dimensions follows the Zipf distribution, which explains why a small number of the most important dimensions can adequately approximate the overall semantic space. Based on the conclusion that the singular value squared is proportional to the statistical contribution of the latent concept dimensions, the normalized contribution of the approximated subspace is the ratio from the sum of the singular value squared of the selected subspace to the sum of the singular value squared of the overall subspace.

### 2.1.2. Probabilistic Topic Models

Hoffman (1999) proposed a probabilistic topic model in his Probabilistic Latent Semantic Indexing (PLSI) which is based on the idea that a latent topic can be represented as a distribution over terms and a document is a mixture of the latent topics. The model specifies the probabilistic relationships between a term $t$ and the latent topic set $Z$ with $T$ topics for a document:

$$P(t_i) = \sum_{j=1}^{T} P(t_i \mid z_i = j) P(z_i = j) \tag{2}$$

However, PLSI itself is not generative which makes it difficult to predict a new document. Blei et al. (2003) extended the model and smoothed the topic distribution by placing a Dirichlet prior to it, the so called Latent Dirichlet Allocation (LDA) which is a generative model. The probability density of a $T$ dimensional Dirichlet distribution over a multinomial distribution $P = (p_1, \ldots, p_T)$ is defined by:

$$Dir(\alpha_i,..., \alpha_T) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1} \quad (3)$$

In Eq. 2, $\alpha_j$ is the prior observation count that topic $j$ is sampled in a document before observing any term from the document. Using Expectation-Maximization (EM) or Gibbs Sampling, the probabilistic topic models directly extract those topics which are represented by the most associated terms (Griffiths et al., 2007). Using matrix factorization, PLSI and LDA split the term-document matrix into two matrixes, a term-topic matrix and a document-topic matrix. The feature values in PLSI and LDA are non-negative and sum to 1. The effectiveness of probabilistic topic models relies on two factors, the estimate of the term distribution – how many topics in the corpus and the smoothing which depends on the quality of the training sets.

#### 2.1.3. Connectionist Models

A third numerical approach to semantic content representation is based on connectionist models (e.g., Rosenblatt, 1958; Rumelhart, 1990; Rumelhart et al., 1986). These models represent their attributes using multiple-layer architectures. In Rumelhart's feed-forward network model a hidden layer between the input layer and the output layer is composed of the neurons which adapt the Back-Propagation supervised learning algorithm (Werbos, 1994) and learn the patterns of the semantic associations of concepts and attributes within a series of contexts using non-linear sigmoid activation functions to estimate the weights of the inputs and presents the outputs as the weighted-sum of the inputs with a minimized error. In a sense, the hidden layer is similar to the LSI latent concept dimensions. For unsupervised learning, Self-Organizing Maps (SOM, Kohonen, 1990) does not include the hidden layer and adjusts the weights of the inputs so that the similar inputs produce similar outputs.

### *2.2 Document Clustering*

Clustering algorithms fall into two main categories, hierarchical and non-hierarchical. Non-hierarchical clustering generally partitions a set of objects into non-overlapping groups so as to maximize the within-cluster inter-object similarities and minimized the between-cluster similarities due to some heuristic criterion of 'goodness of clustering'. Currently, the best K-means partition algorithm is bisecting-K-means. It divides the corpus into two clusters with K-means first. Then, it iteratively partitions the currently largest cluster into two clusters, again using K-means, until K clusters have been identified (Steinbach et al. 2000). However, these K-means document clustering algorithms need a predefined number of the clusters.

The number of the document clusters in a text corpus is difficult to predict. SOM (Kohonen, 1990) automatically decides the number of the regions or the clusters. Willett (1990) introduced hierarchical agglomerative clustering, as mentioned above, which emphasizes the discovery of the boundaries of the clusters but does not interpret them. Wang et al. (1999) describes an association-rule based non-hierarchical clustering algorithm, which compares the similarity among the frequent feature sets occur in the documents instead of the pair-wise similarity between documents. Beil et al. (2002) extend the idea to create a hierarchical clustering method called HFTC (Hierarchal Frequent Term-based Clustering) which greedily picks frequent feature sets to minimize the overlap among the documents. Fung et al. (2003) propose the Frequent Itemset-based Hierarchical Clustering (FIHC) algorithm that applies terms in the global frequent feature sets to reduce dimensionality and significantly improve the clustering accuracy. These methods help interpretation to the document clusters with frequently used terms but have weakness in explaining the relationships between concepts and documents. Schütze and Silverstein (1997) used the LSI projections to the LSI document space to improve the efficiency of the document clustering. Each document is represented as a vector of the least-square distances from the latent concept dimensions to the original documents with a fixed number of the top-ranking dimensions. Ampazis and Perantonis (2004) introduce LSISOM which applies the LSI term subspace to improve SOM document clustering.

## 3. LSI Subspace Signature Model

We propose the LSI Subspace Signature Model (LSISSM) for semantic content representation. Figure 1 shows the key components of LSISSM: signature representation, term/document ranking and similarity between signatures.

**"Insert Fig #1 here ( assist_fig1.tif )"**

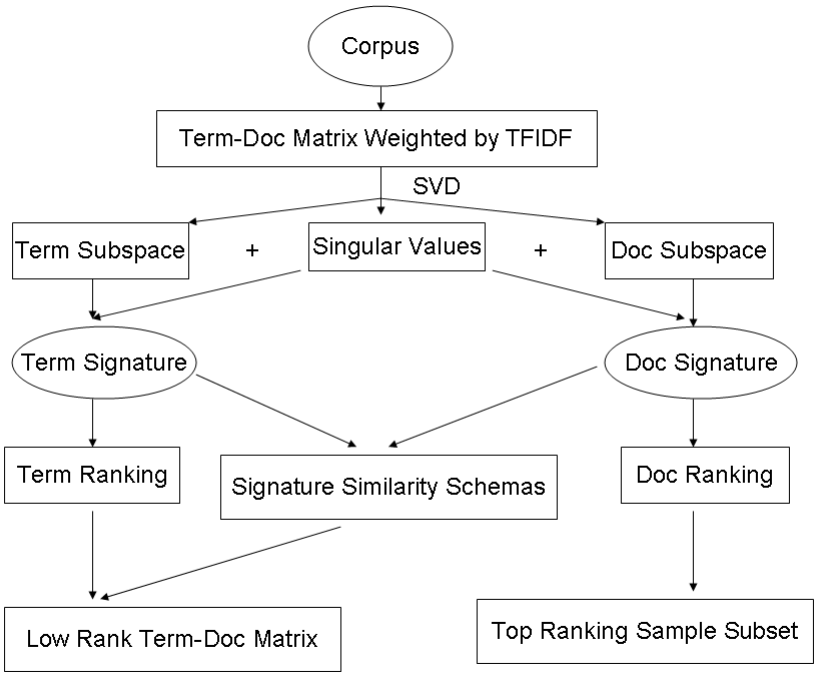



**Figure 1**: Overview of the LSI Subspace Signature Model.

The model derives a spectral signature representation for terms and documents, ranks them according to their contribution and determines relatedness between the signatures by their matching patterns on the identical LSI latent concept dimensions.

LSISSM has several advantages compared to traditional LSI: The top-ranking dimensions are picked based on their global statistical contribution to the LSI subspaces instead of a predefined fixed number; we use an optimized feature set to present the documents instead of a number of the LSI latent concept dimensions because the meaning of the latent concept dimensions is implicit; the similarity schemas depend on the association between term signatures and document signatures and reduce the noise in the associations by bridging and matching the LSI term subspace and the LSI document subspace. The output of the model is a low-dimension term-document matrix weighted by the signature similarity measures.

#### *3.1 LSI Subspace Term Signatures and Document Signatures*

Ding's (2005) model shows that the singular value squared is proportional to the statistical contribution of the corresponding latent dimensions. Based on this result, we built feature and document representations according to their local statistical contribution to the latent concept dimensions. The contribution is calculated by the squared product of the singular value of the latent concept dimension and the project score to the dimension. The values of the signatures are non-negative and have an explicit statistical meaning -- the contribution to the latent concept dimensions. The terms and documents are represented separately by the contributions to the top $K$ dimensions in the identical LSI latent concept dimensions.

The signatures are generated as follows:

**Step 1:** Use SVD to decompose the term-by-document matrix $A$ into a term-concept matrix $U$, a diagonal matrix $C$ and a document-concept matrix $V$.

**Step 2:** For the $U$ or $V$ matrices, the top $K$ dimensions selected as the proximity of the overall subspace is determined by the ratio $T_d$:

$$T_d = \sum_{j=1}^{K} D_j / \sum_{n=1}^{M} D_n \quad (4)$$

In Eq. 4, $D$ is the square of the singular value S and $M$ is the number of the latent concept dimensions with a non-zero contribution. $T_d$ defines the contribution portion of the top $K$ dimensions that contribute to the overall latent subspaces. Users can set it according to their statistical confidence interval on errors. For example, setting $T_d$ at 0.95 the top $K$ dimensions that contribute 95% are used and the remainder is ignored. The experiments show that if selecting a subspace which makes a high percent contribution, the number of the latent dimensions is large. For instance, we test the corpus with 2527 Reuters news articles described in Section 5 and find that 917 top latent concept dimensions make about 80 percent global contribution ($T_d$=0.8). This indicates that LSISSM uses more latent dimensions than that are typically used in traditional LSI.

**Step 3:** $w_{in}$ is the local contribution of a term/document $X_i$ to a dimension $n$ formulated by Eq. 5, where $X_{in}$ is the $n^{th}$ dimension projection score of $X_i$.

$$w_{in} = D_n X_{in}^2 \quad (5)$$

The global contribution of a term or document to the selected LSI subspaces, $G_{in}$, is:

$$G_{in} = \sum_{n=1}^{K} w_{in} \quad (6)$$

In Eq. 6, $K$ is determined by $T_d$. Normalized by the contribution of all the terms or documents, $G_{in}$ is converted to $NG_i$:

$$NG_i = G_{in} / \sum_{i=1}^{P} G_{in} \quad (7)$$

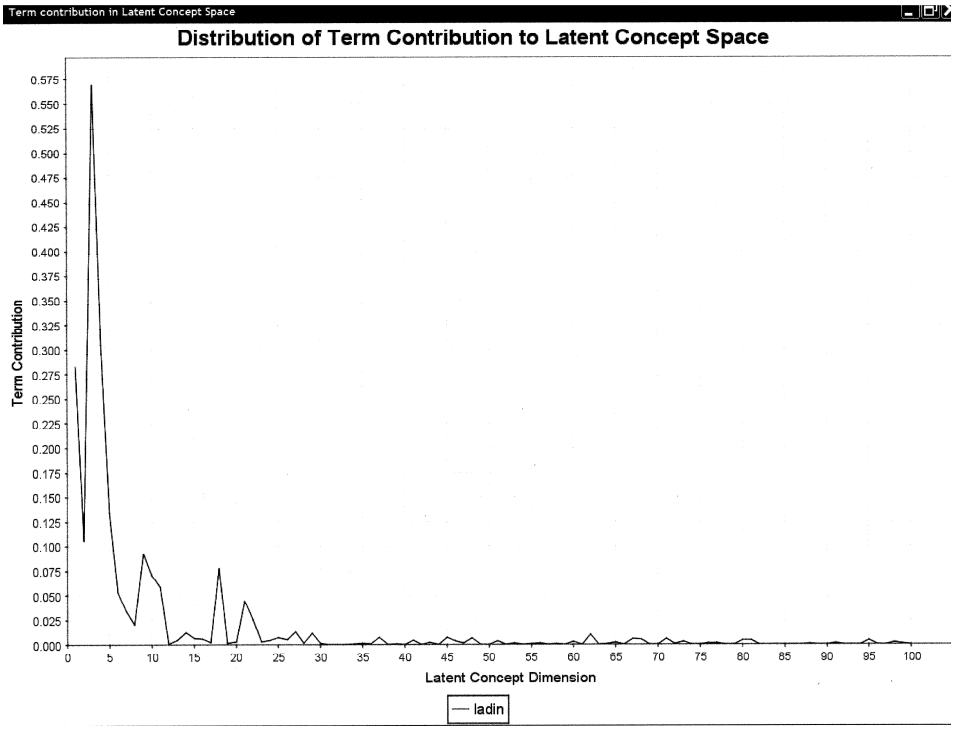
In Eq. 7, $P$ is the number of the terms or the documents in the corpus. The global contribution of any subset of the corpus, $T_g$,, is:

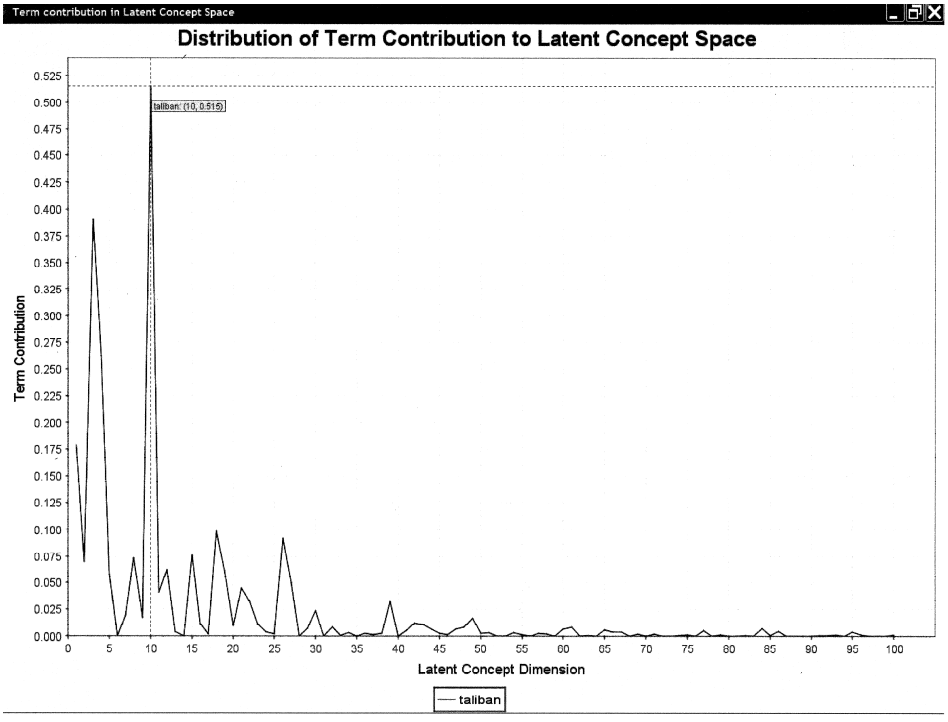
$$T_g = \sum_{i=1}^{S} NG_i \quad (8)$$

Where S is the number of the terms or documents in the subset.

**Step 4:** $X_i$ is represented by a vector of $w_{in}$ as a signature, where $n$ belongs to {1,.., K}.

An example of term and document signatures is shown in Fig. 2. This was taken from our earlier analysis of a collection of 948 news articles extracted from a terrorism news collection. In each of the three panels, the X axes indicate the LSI latent concept dimensions and the coordinates in Y axes are weighted by $w_{in}$. The positions and height of the peaks in the document signature suggests that the term signatures capture the main themes of the news article. This motivates us to create a similarity metric which links the term signatures and the document signatures through the projections to the same LSI latent concept dimensions across the latent term and document subspaces.


Term contribution in Latent Concept Space
Distribution of Term Contribution to Latent Concept Space
Term Contribution
Latent Concept Dimension
ladin


Term contribution in Latent Concept Space
Distribution of Term Contribution to Latent Concept Space
taliban: (10, 0.515)
Term Contribution
Latent Concept Dimension
taliban

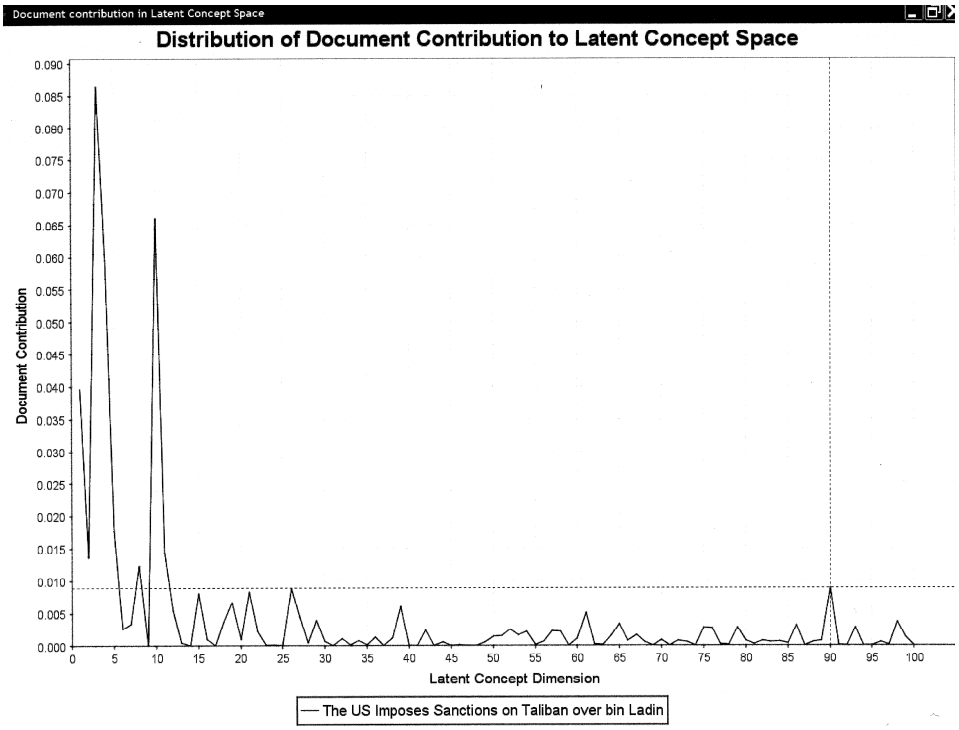



**Figure 2:** Note the similarity between the term signatures for "taliban" and "Ladin" (upper two panels) and the document signature for a news article with a title "The US Imposes Sanctions on Taliban over bin Ladin" (lower panel) generated from Storylines (Zhu and Chen, 2007).

### *3.2 LSI Subspace Signature Ranking*

The signature ranking algorithm, Global and the Local Contribution Ranking (GLCR), iteratively picks terms or documents based on their contribution rankings until these terms and documents collectively reach a predefined threshold, the ratio $T_g$ as calculated by Eq. 8. The term frequency distribution in a corpus follows Zipf's Law (Gelbukh & Sidorov, 2001) because a small number of terms make the major statistical contribution. Accordingly, when using GLCR to select terms, the value of $T_g$ is set high enough to ensure the term subset makes a major contribution to the overall semantic subspace.

GLCR has two steps. The global contribution of one term or one document is calculated by Eq. 6 or 7 and the local contribution of one term or one document is estimated by the absolute value of its projection score to one latent concept dimension. GLCR first selects a threshold $T_l$ for the local contribution $w_{\text{in}}$. $T_l$ is derived empirically and is initialized by a ratio to the mean of the absolute values of all the projection scores across the top $K$ dimensions. Some terms have higher scores for the global contribution but the projection value for each of the top $K$ dimensions is lower than $T_l$. Such terms lack discriminating power and are ignored. GLCR scans the signature of each term or document. The selection starts from the dimension with the highest singular value and follows step-wise. This strategy means that although a term or document has high projection scores in many dimensions, while in a given dimension the absolute value of its projection score is larger than $T_l$, it will be selected and the searching process goes to another candidate. Then, each term or each document in the subset is ranked according to Eq. 6 or 7.

### *3.3 Measuring the Similarity of the Term and Document Signatures*

We build novel similarity measures to correlate term signatures and document signatures. The Concept and Document Signature Similarity (CDSS) between a term signature $r_a$ and a document signature $r_u$ is calculated by Eq. 9:

$$CDSS \quad (r_a, r_u) = \frac{\alpha_{a,u} \sum_{i=1}^{K} (r_{a,i} - \beta \bar{r}_a)(r_{u,i} - \beta \bar{r}_u)}{\sqrt{\sum_{i=1}^{K} (r_{a,i} - \beta \bar{r}_a)^2 \sum_{i=1}^{K} (r_{u,i} - \beta \bar{r}_u)^2}}$$

(9)

where

$$\bar{r}_x = \frac{\sum_{i=1}^{K} r_{x,i}}{K} \qquad \alpha_{a,u} = \left\{ \begin{array}{l} 1 \text{ if } d_{a,u} \geq d_{avg} \\ \frac{d_{a,u}}{d_{avg}} \text{ if } d_{a,u} \leq d_{avg} \end{array} \right\}$$

$K$ is determined by $T_d$. $\alpha_{a,u}$ is the significance weight which is determined by the number of co-projected latent concept dimensions of the two signatures with a range from 0.0 to 1.0 and β is a pruning parameter which determines how many of the top $K$ latent concept dimensions are included in the similarity score. If β equals 0.0, all the top K dimensions are used in the similarity measure. If β equals 1.0, only the dimensions with values above $\bar{r}$ are used. The value of β can be larger than 1.0. $\bar{r}$ is the average score of the signature across the top $K$ dimensions. $d_{avg}$ in $\alpha_{a,u}$ is the average number of the co-projected dimensions shared by each pair of the signatures. The sign of projection scores affect the matching of the term and document signatures. For instance, by observation of the matrix U and V, if the projection score of a term is negative in one dimension, the most related documents might have negative projection scores in the same dimension. The association is counted only when the two signatures have projection scores with the same sign on a given dimension.

CDSS calculates the normalized association between a term and a document by reducing noise. In addition, we propose a variation of the CDSS measure. Specifically, Global Contribution enhanced CDSS (GCDSS) (Eq. 10) multiples the CDSS similarity score with the global contribution score of each term signature:

$$GCDSS(r_a, r_u) = \sum_{n=1}^{K} D_n X_{an}^2 CDSS(r_a, r_u) \qquad (10)$$

GCDSS is used only between term and document signatures. And, CDSS can be applied to measure the relationships of term-term, document-document or term-document.

After the three steps of LSISSM, a sparse TFIDF document-term matrix is re-constructed and the association between a term and a document is weighted by CDSS or GCDSS. In this stage, the similarity measures only count the relationships between the terms and the documents where the terms appear. Using GCDSS the top-ranking terms have higher scores compared to those generated by CDSS, which suggests the representatives for the documents. By observation, topical terms (i.e., those relevant to the class labels of the documents) are often ranked highest within the documents (see the results in Section 5.1.3).

## 4. Text Clustering Using LSISSM

LSISSM has controls on the level of the dimension reduction. The term-signature ranking narrows the scope of the features to a limited number of conceptual terms and the parameters of the model ensure the term subset make the main global contribution to the LSI term subspace. This mechanism is determined by the characteristics of the term distribution for a text corpus, Zipf's distribution. Using the term subset to present the overall semantic space, a highly scalable term-document matrix is transformed into a low-dimension term-document matrix weighted by CDSS or GCDSS. These similarity schemas match any conceptual term signature with any document signature and prune the association between the signatures. So, the model reduces noise with parameters.

### *4.2. Applying LSISSM to Basic K-means and SOM*

There are two ways to apply LSISSM to standard K-means and SOM (see Figure 3). One way is to use the output of LSISSM directly, the low-dimension term-document matrix, as input for the two algorithms. Another way is to group top-ranked conceptual terms first by the clustering algorithms and then rank term clusters according to the sum-up global contribution of the terms in them. The term clustering algorithm applied here is SOM (Kohonen, 1990; Lin et al., 1991) because the number of term clusters cannot be pre-defined. The term-term similarity matrix is constructed by CDSS. The value of β is set from 1.0 to 1.5 in the experiments. After term clustering, each term cluster is ranked by the

accumulated value of the global statistical contribution of all the terms in the cluster. Only those terms which appear in the top-ranking term clusters are included in the input matrix.

### *4.3. Applying LSISSM-based Two-stage K-means*

Standard K-means minimizes the sum of the squared distances between each point in the dataset and the closest center. The K centers start with randomization seeding. The random initialization procedure dramatically affects the performance of the K-means algorithm and sometimes causes very poor performance. We propose a two-stage initialization procedure based on LSISSM. In the first stage, standard K-means is run with a fixed number of the clusters and a small feature subset which only includes the top-ranking terms generated by LSISSM. In the second stage, the centroids of the K clusters produced by the first step are used as the initial centroids of the second stage using an expanded feature subset. Then, the standard K-means algorithm is run again. The results in section 5.2 show that this strategy improves both the efficiency and accuracy of standard K-means.

K-means++ (Arthur & Vassilvitskii, 2007) mathematically shows that the initial clusters positioned in the dense data regions which make the major statistical contributions improve both the efficiently and effectively of standard K-means. And, that method is applied to non-textual data with low dimensionality. Silic et al. (2008) used the seeding method, K-means++, but did not report the effect. The performance of K-means++ is affected by the dimensions used for each data point, while our model targets both the selection of data points and dimension reduction.

Our two-stage K-means approach tries to find and use the key conceptual term subsets to represent the dense regions of the document clusters. Our assumption is that the key term subset (i.e., the one which makes the main contribution to the semantic concept space) is most likely associated with the dense regions of document clusters. The stochastic processing of the standard K-means converges very fast if the key-term subset is very small and the resulting centroids are likely to be better positions than those generated randomly. The conceptual feature subset is selected by GLCR. The procedures for the two-stage K-means are as follows:

**Step 1:** Arbitrarily choose k initial centers $C = \{c_1, \ldots, c_k\}$.

**Step 2:** For each document $\chi_i$, set the cluster $C_i$ to be the set of documents in $\chi$ that are closer to $c_i$ than any other centers.

**Step 3:** For each document $\chi_i$, set $c_i$ to be the center of mass of all points in $C_i$:

$$c_i = 1 / |C_i| \sum_{x_i \in C_i} x \tag{11}$$

**Step 4:** Repeat Steps 2 and 3 until $C$ no longer changes.

In the first stage, k initial centers are generated randomly and each document is represented by a small feature subset of the top-ranking terms. After the algorithm converges, the centers of the clusters produced by the first stage are calculated using Eq. 11. Then, these centers are used as the seeds for the second stage to repeat Steps 2 to 4. In the second stage, the corpus is represented by a much larger feature subset with additional top terms. We explore two strategies to implement the second stage of the two-stage K-means algorithm. One strategy uses the feature subset directly. The second strategy uses the term clusters. The feature subset is clustered by SOM and then the term clusters are ranked by the accumulated statistical contribution of the terms inside the clusters. The terms in the top-ranking clusters are then used in the second stage.

## 5. Experiments

The experiments evaluate how LSISSM affects the performance of the major clustering algorithms because the major challenges for text clustering are selecting the discriminated terms and reducing the noise from the relationships between term and documents. Two clustering algorithms are tested: standard K-means (Dubes & Jain, 1988) and SOM (Kohonen, 1990; Lin et al., 1991). Moreover, to solve the initialization problem of standard K-means, we propose a novel K-means algorithm, two-stage K-means.

### *5.1. Research Strategy and Considerations*

#### 5.1.1. Data Sets, Text Pre-Processing and Document Clustering Evaluation Methods

Three standard news corpora, Reuters 21587[1], TDT1 and TDT2[2], are used. From the Reuters21578–Apte-90Cat corpus, 2527 training news articles were selected. These belong to 10 categories, acq, coffee, interest, iron-steel, oat, palmkernel, sugar, sun-meal, veg-oil and wheat. The sample size for each category varied from 1 to 1646. For TDT1, we generate a subset which consists of all the 25 categories and contains 1131 documents that are evaluated as "YES" rather than "BRIEF" in the evaluation sheet of TDT1. The sample size for each category varied from 2 to 273. The full TDT2 corpus has 100 categories (news story topics). We randomly selected a subset consisting of 30 categories (topics) containing 3349 documents which are evaluated as "YES" rather than "BRIEF" in the evaluation sheet of TDT2. The sample size for the categories varied from 1 to 1132. The three datasets used as the training sets for text clustering.

Pre-processing directly influences the quality of the clustering. First, Stanford part-of-speech (POS) tagging (Toutanova & Manning, 2000; Toutanova et al., 2003) is applied to the corpus. Then, stop-word filtering using the Google stop word list and Porter stemming are applied. Each article for a given category has to include at least one noun as identified by the POS tagging. All the nouns are included in the analysis because this approach emphasizes the concept representations of the documents. The association between a noun term and a document in the initial term-document matrix is weighted by traditional TFIDF. Each column of the matrix is then normalized to 1.0. The VSM baselines of the clustering algorithms are generated with the full-feature sets and their term-document matrix weighted by TFIDF.

For non-hierarchical clustering, the pureness of the clusters is evaluated using Purity (Manning et al., 2008) and Entropy (Beil et al., 2002). Xu et al. (2003) and Cai (2005) use Normalized Mutual Information (NMI) which can be applied to all clustering methods. The advantage of NMI is that its value is not affected by the number of the clusters. For Purity and NMI, higher values show better performance, while lower entropy values indicate better performance. In our experiments the entropy value is normalized by the size of the categories to ensure that it has a range between 0.0 and 1.0, and so it is called relative entropy.

#### 5.1.2. VSM as the Baseline Model of the Clustering Technique

Using Purity, Entropy and NMI, we see that traditional LSI improved the efficiency of document clustering (Schütze & Silverstein, 1997) but it does not improve its effectiveness: in Tables 1 and 2 in which no consistent pattern is found. The results indicate that VSM is not worse than traditional LSI in the two of the three corpora. With the increase of the category size from the Reuters corpus to TDT2, traditional LSI becomes less effective. Thus, we picked VSM as the baseline for the following studies and compared the results of LSISSM in section 5.3.1.

| Corpus | Category Size | Similarity | Top K | Purity | Entropy | NMI |
|---|---|---|---|---|---|---|
| Reuters | 10 | TFIDF | ------ | 0.795 | 0.250 | 0.389 |
| Reuters | 10 | LSI | 10 | **0.888** | **0.223** | **0.492** |
| TDT1 | 25 | TFIDF | ------ | 0.815 | **0.133** | **0.789** |
| TDT1 | 25 | LSI | 10 | **0.828** | 0.170 | 0.745 |
| TDT2 | 30 | TFIDF | ------- | **0.913** | **0.072** | **0.765** |
| TDT2 | 30 | LSI | 10 | 0.905 | 0.091 | 0.719 |

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[2] http://www.ldc.upenn.edu/

**Table 1:** Comparison of the clustering performance between VSM and traditional LSI using the standard K-means for Reuters, TDT1 and TDT2.

| Corpus | Category Size | Similarity | Top K | NMI |
|---|---|---|---|---|
| Reuters | 10 | TFIDF | ------ | 0.257 |
| Reuters | 10 | LSI | 20 | **0.315** |
| TDT1 | 25 | TFIDF | ------ | **0.690** |
| TDT1 | 25 | LSI | 30 | 0.631 |
| TDT2 | 30 | TFIDF | ------- | **0.517** |
| TDT2 | 30 | LSI | 30 | 0.475 |

**Table 2:** Comparison of the clustering performance between VSM and traditional LSI using SOM with Reuters, TDT1 and TDT2. Note: In these tables, Top K denotes the top K latent concept dimensions which represent the selected LSI document subspace. These K values are the dimension numbers which achieve the best scores of Purity, Entropy and NMI for each data set. The bold values indicate the better performance if compared VSM with traditional LSI.

5.1.3. Determining Top Ranked Terms

We use the top-rank terms to seed the clustering tests described in the Section 5.2. GLCR selects the terms which make the major contribution to the corpus and reflect the most popular topics in the corpus. For instance, the top 20 terms listed in Table 3 ranked by GLCR and extracted from the subset of the Reuters 21578 corpus cover most topics related to the ten class labels, "acq, coffee, interest, iron-steel, sugar, wheat, palmkernel, oat, sun-meal, and veg-oil".

| Rank | Term | Rank | Term |
|---|---|---|---|
| 1 | stg | 11 | reserve |
| 2 | tonn | 12 | taft |
| 3 | rate | 13 | oil |
| 4 | bank | 14 | bill |
| 5 | sugar | 15 | twa |
| 6 | wheat | 16 | federal |
| 7 | coffee | 17 | export |
| 8 | gencor | 18 | purol |
| 9 | cyclop | 19 | chemla |
| 10 | usair | 20 | money |

**Table 3:** Top 20 terms which make the highest contribution to the LSI term subspace for the Reuters corpus ranked by GLCR.

The size of the term subsets is determined by the parameters, $T_g$, $T_l$ and $T_d$, which are between 0.0 and 1.0. $T_d$ determines how many top dimensions are selected. Limiting the number of dimensions reduces noise. In general a larger $T_l$ value means less noise. $T_g$ determines the upper boundary of the overall contribution of the term subset if $T_l$ and $T_d$ are predefined. We varied $T_d$, $T_l$ and $T_g$ to maximize the contribution included while minimizing the noise. For example, from the Reuters dataset, if $T_d$ is set to 0.95 and $T_l$ is set to 0.5, GLCR selects a subset of 2673 conceptual terms out of the whole set (9070 terms). This set makes an 83.3% (maximum value of $T_g$) statistical contribution to the selected LSI term subspace. From TDT1, GLCR selects a subset of 1614 terms out of the whole set (8338 terms), which makes an 82.30% (maximum value of $T_g$) statistical contribution to the overall term subspace. From the TDT2,

GLCR selects a subset of 3007 concept terms out of the whole set (17083 terms), which makes an 85.14% (maximum value of $T_g$) statistical contribution to the overall LSI term subspace.

### *5.2. Results*

We report a number of tests of clustering using LSISSM. The formal comparison to the VSM Baseline is described in Section 5.3.1. The standard K-means algorithm uses the same parameters across the three datasets. The predefined number of clusters for the K-means is 10, 25 and 30 respectively for the three data sets with 10 iterations. SOM uses the same parameters across the three datasets. Epsilon is 0.25, and each run has 2500 iterations with 12 nearest neighbors.

#### 5.2.1. K-means Using Top-Ranking Terms

By focusing on just the top terms as obtained by LSISSM, the overall clustering performance is improved. As shown in Figure 3, the standard K-means reaches the best points with the top 2600 terms on both Purity and Entropy measures, and receive the maximum value of NMI with a feature subset consisting of the top 600 terms. The numbers of the top-ranking conceptual terms included in the feature subset are also listed in the figure. For example, in Figure 3 the top 100 terms contribute 19.1% ($T_g$) to the LSI term subspace.

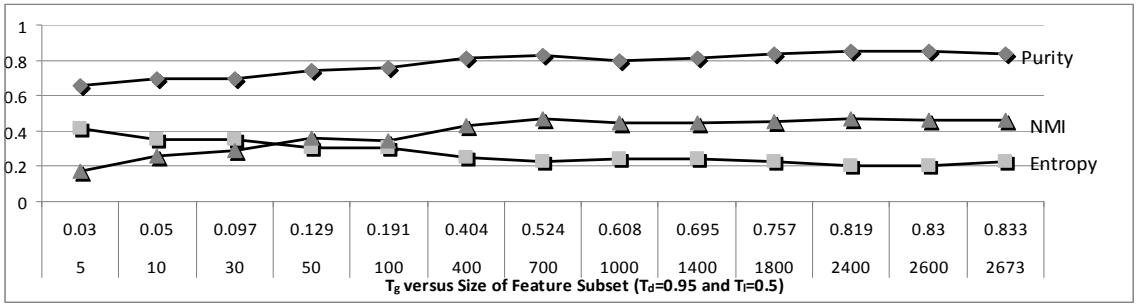



**Figure 3:** Evaluation of the standard K-means with $T_g$ using Reuters and CDSS. Note that lower entropy scores indicate better performance.

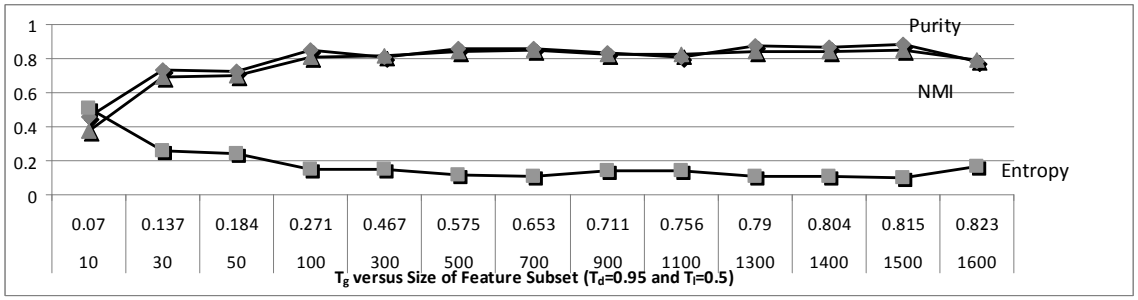



**Figure 4:** Evaluation of standard K-means with $T_g$ using TDT1 and CDSS.

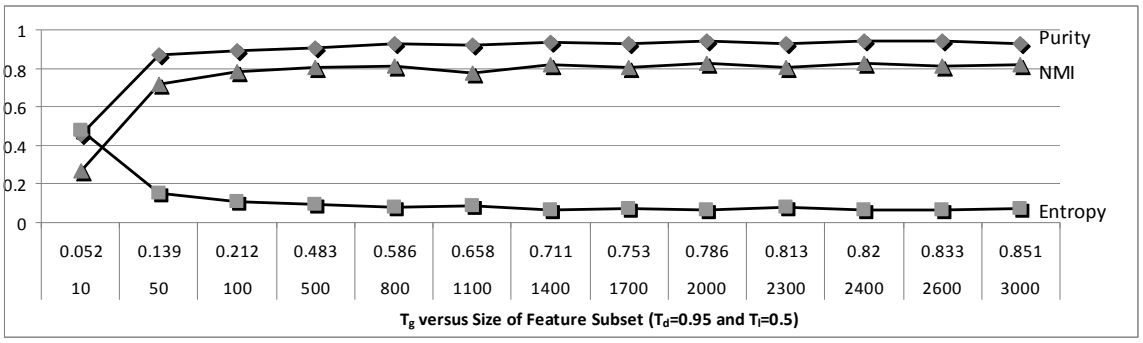



**Figure 5:** Evaluation of standard K-means with $T_g$ using TDT2 and CDSS.

In Figure 4, standard K-means reaches the best scores with the top 1500 terms evaluated by the three evaluation methods. In Figure 5, standard K-means reaches the best scores with the top 2400 terms for Purity and Entropy. For the NMI measure, standard K-means reaches the maximum score with the subset that includes the top 2000 terms.

5.2.2. K-means Using Top-ranking Term Clusters

In this section, the value of β is set to 1.0 in CDSS to produce term-term similarity matrix for term clustering with SOM. SOM generates 317, 227 and 348 term clusters for the Reuters, TDT1 and TDT2 respectively.

As shown in Figure 6, using the Reuters collection standard K-means reaches the best points with the top 200 term clusters for both Purity and Entropy measures, and it receives the maximum value of NMI with the top 175 term clusters.

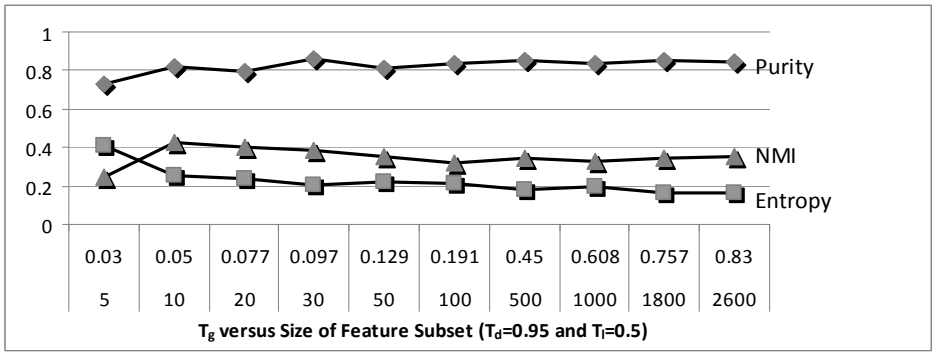



**Figure 6:** Evaluation of standard K-means with $T_g$ using the Reuters, CDSS and term clusters.

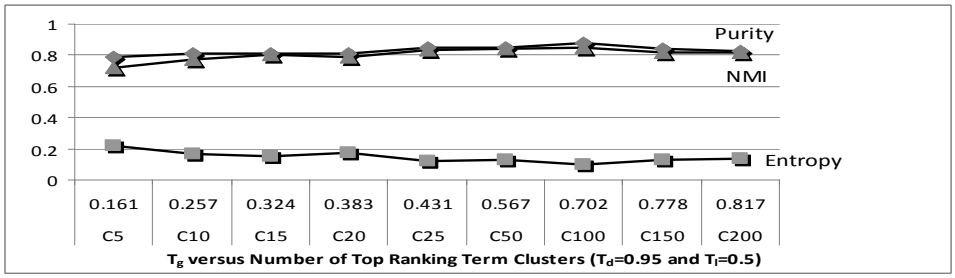



**Figure 7:** Evaluation of standard K-means with $T_g$ using TDT1, CDSS and term clusters.

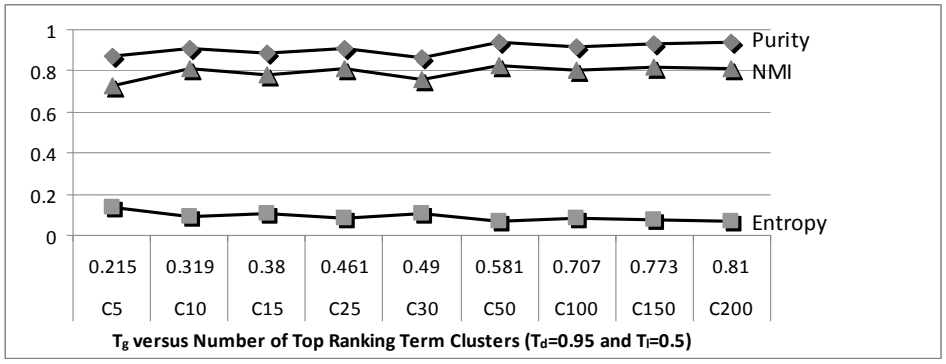



**Figure 8:** Evaluation of standard K-means with $T_g$ using TDT2, CDSS, and term clusters.

In Figure 7, standard K-means reaches the best scores with the feature subset including the top 100 term clusters evaluated by any of the three evaluation methods. In Figure 8, standard K-means has the maximum scores of the three among methods with the top 50 term clusters.

5.2.3. Two-stage K-means

We pause in the comparison of the clustering algorithms to consider an extension of K-Means using LSISSM, so called two-stage K-means. Two-stage K-means picks a term subset of no more than 100 in the first stage so that the document clustering algorithm can converge quickly.

| Corpus | Dataset | Running Time: 1st stage | Running Time: 2nd stage | Overall Running Time |
|---|---|---|---|---|
| Reuters | 2600 | ------ | ------ | 9368.8 |
| Reuters | 20+2600 | 43.2 | 2842.5 | 2885.7 |
| Reuters | C200 | ------ | ------ | 6991.8 |
| Reuters | 20+C200 | 42.6 | 2437.7 | 2480.3 |
| TDT1 | 1500 | ------ | ------ | 1360.4 |
| TDT1 | 100+1500 | 114.7 | 111.6 | 226.3 |
| TDT1 | C100 | ------ | ------ | 1124.4 |
| TDT1 | 100+C100 | 114.7 | 75.5 | 190.2 |
| TDT2 | 2400 | ------ | ------ | 15804.6 |
| TDT2 | 40+2400 | 497.3 | 3998.5 | 4495.8 |
| TDT2 | C200 | ------ | ------ | 13298.9 |
| TDT2 | 40+C200 | 492.0 | 2766.8 | 3258.9 |

**Table 4:** Comparison of the running time (seconds) between standard K-means and Two–stage K-means using the top-ranking terms and the top-ranking term clusters for Reuters, TDT1 and TDT2.

For instance, in Table 4, 20+2600 indicates that in the first stage a feature subset with the top 20 terms is used and in the second stage the feature subset is expanded to 2600 features. 20+C200 denotes that the first stage is the same as the previous example and in the second stage 200 term clusters are used. The term clusters applied in this section are the same as ones in Section 5.2.2. Table 7 reports the running time for both stages. Compared to standard K-means, two-stage K-means is consistently faster. For instance, compared to Reuters (C200) and TDT1 (1500), the runs of the two-stage K-means Reuters (20+C200) and TDT1 (20+1500) are much faster. TDT1 (20+1500) saves 83% of the time compared to TDT1 (1500).

As shown in Tables 5 and 6, two-stage K-means not only saves time but also improves the clustering accuracy. For instance, in Table 5, compared to TDT1 (1500) and TDT2 (2400), TDT1 (100+1500) and TDT2 (40+2400) are top ranked for each of the three evaluation measures. And, Reuters 20+2600 has a better score than Reuters 2600 only for NMI. And as shown in Table 6, compared to TDT1 (C100) and TDT2 (C200), TDT1 (100+C100) and TDT2 (40+C200) have better scores with each of the three evaluation measures. For Reuters, with the top 200 term clusters the two-stage K-means has better scores for Purity and Entropy.

| Corpus | Dataset | Purity | Entropy | NMI |
|---|---|---|---|---|
| Reuters | 2600 | **0.853** | **0.200** | 0.462 |
| Reuters | 20+2600 | 0.845 | 0.211 | **0.470** |
| TDT1 | 1500 | 0.884 | 0.097 | 0.849 |
| TDT1 | 100+1500 | **0.913** | **0.090** | **0.864** |
| TDT2 | 2400 | 0.943 | 0.063 | 0.823 |
| TDT2 | 40+2400 | **0.947** | **0.059** | **0.827** |

**Table 5:** Comparison of the clustering performance between standard K-means and two–stage K-means using CDSS and the top-ranking terms for Reuters, TDT1 and TDT2.

| Corpus | Dataset | Purity | Entropy | NMI |
|---|---|---|---|---|
| Reuters | C200 | 0.833 | 0.224 | **0.471** |
| Reuters | 20+C200 | **0.841** | **0.218** | 0.455 |
| TDT1 | C100 | 0.882 | 0.098 | 0.850 |
| TDT1 | 100+C100 | **0.906** | **0.095** | **0.861** |
| TDT2 | C200 | 0.936 | 0.069 | 0.811 |
| TDT2 | 40+C200 | **0.947** | **0.060** | **0.836** |

**Table 6:** Comparison of the clustering performance between standard K-means and two–stage K-means using CDSS and the top-ranking term clusters for Reuters, TDT1 and TDT2.

5.2.4. SOM Using Top-Ranking Terms

The input matrix for SOM is weighted by GCDSS. In Figure 9, SOM finds the turning point which has the maximum score from the feature subset with the top 2600 terms if evaluated by Entropy. The best score for Purity is received by the subset with the top 30 terms and the best score for NMI is generated by the subset with only 10 top terms.

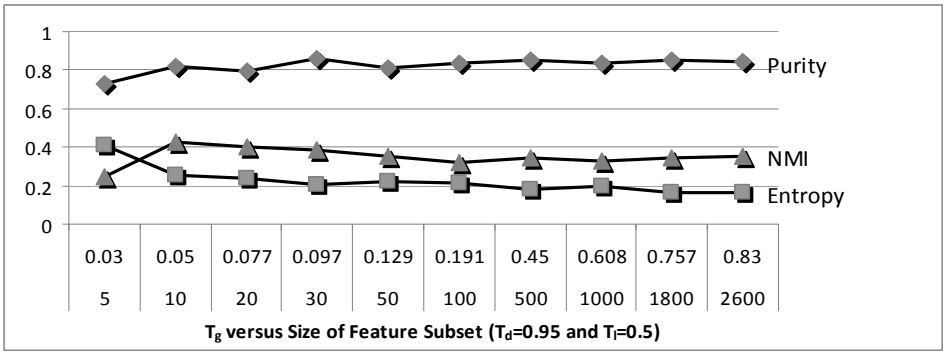



**Figure 9:** Evaluation of SOM with $T_g$ using Reuters and GCDSS.

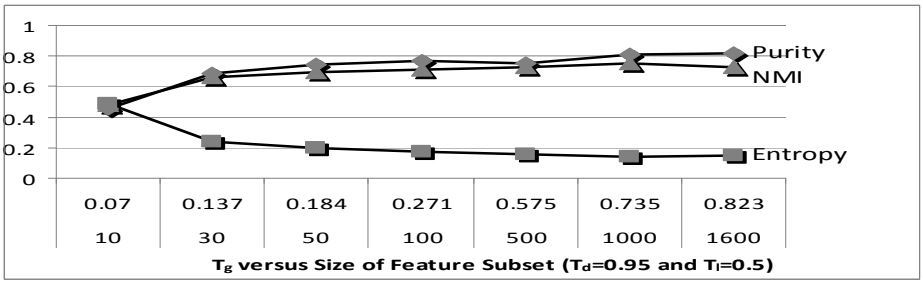



**Figure 10:** Evaluation of SOM with $T_g$ using TDT1 and GCDSS.

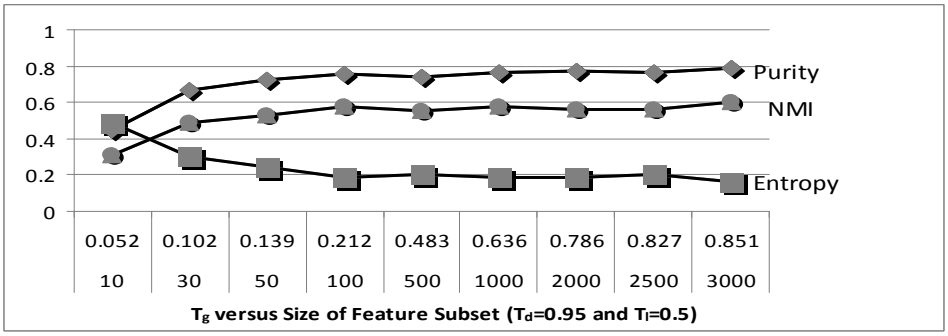



**Figure 11:** Evaluation of SOM with $T_g$ using TDT2 and GCDSS.

In Figure 10, SOM gets the maximum scores for the NMI and Entropy measures with the top 1000 terms respectively and reaches the peak score of Purity with the top 1600 terms. In Figure 11, SOM achieves the maximum scores of the three evaluation methods with the top 3000 terms.

### *5.3. Summary of Results*

5.3.1. Comparison to VSM

In Tables 7, 8, and 9, the best points selected from Figure 3 to 11 are compared with the VSM baselines. Generally these points have at least two best evaluation scores generated by two different evaluation measures. Compared to VSM, when using the top-ranking terms, the Purity test to LSISSM showed in Table 7 is significantly better across the three corpora ($t(2)=4.51$, $p<0.05$) and

NMI is also statistically significant ($t(2)=13.54$, $p<0.05$). When using top-ranking term clusters, the NMI t-test to LSISSM showed in Table 8 is statistically significant across the three corpora ($t(2)=9.43$, $p<0.05$) if β is set at 1.0 in CDSS for SOM term clustering. If β is set as 1.5 in CDSS for SOM term clustering, t(2) value will increase to 43.6 ($p < 0.001$). In Table 9, NMI t-test is statistically significant ($t(2)=8.61$, $p<0.05$) across the three corpora. Because SOM produces different numbers of the clusters for the same corpus with the different feature subsets, only NMI is suitable to be used to compare the performance.

| Corpus | Similarity | Top Terms | Purity | Entropy | NMI |
|---|---|---|---|---|---|
| Reuters | TFIDF | ALL | 0.795 | 0.250 | 0.389 |
| Reuters | CDSS | 2600 | **0.853** | **0.200** | **0.462** |
| TDT1 | TFIDF | ALL | 0.815 | 0.133 | 0.789 |
| TDT1 | CDSS | 1500 | **0.884** | **0.097** | **0.849** |
| TDT2 | TFIDF | ALL | 0.913 | 0.072 | 0.765 |
| TDT2 | CDSS | 2400 | **0.943** | **0.063** | **0.823** |

**Table 7:** Comparison of the clustering performance between the VSM baseline and LSISSM using standard K-means, CDSS and the top-ranking terms for Reuters, TDT1 and TDT2.

| Corpus | Similarity | Top Term Clusters | Purity | Entropy | NMI |
|---|---|---|---|---|---|
| Reuters | TFIDF | ALL | 0.795 | 0.250 | 0.389 |
| Reuters | CDSS | C200 (β=1.0) | **0.833** | **0.224** | **0.471** |
| Reuters | CDSS | C200(β=1.5) | **0.846** | **0.210** | **0.460** |
| TDT1 | TFIDF | ALL | 0.815 | 0.133 | 0.789 |
| TDT1 | CDSS | C100(β=1.0) | **0.882** | **0.098** | **0.85** |
| TDT1 | CDSS | C100(β=1.5) | **0.923** | **0.069** | **0.865** |
| TDT2 | TFIDF | ALL | 0.913 | 0.072 | 0.765 |
| TDT2 | CDSS | C50(β=1.0) | **0.936** | **0.068** | **0.825** |
| TDT2 | CDSS | C200(β=1.5) | **0.938** | **0.064** | **0.836** |

**Table 8:** Comparison of the clustering performance between the VSM baseline and LSISSM using standard K-means, CDSS and the top-ranking term clusters for Reuters, TDT1 and TDT2.

| Corpus | Similarity | Top Terms | NMI |
|---|---|---|---|
| Reuters | TFIDF | ALL | 0.257 |
| Reuters | GCDSS | 2600 | **0.355** |
| TDT1 | TFIDF | ALL | 0.690 |
| TDT1 | GCDSS | 1000 | **0.755** |

| | | | |
|---|---|---|---|
| TDT2 | TFIDF | ALL | 0.517 |
| TDT2 | GCDSS | 3000 | **0.602** |

**Table 9:** Comparison of the clustering performance between the VSM baseline and LSISSM using the SOM, GCDSS and the top-ranking terms for Reuters, TDT1 and TDT2.

Thus, the GCDSS measure significantly improves the performance of SOM compared to the VSM baseline with the top ranking terms. And, the CDSS measure significantly enhances the performance of standard K-means compared to the VSM baseline using either top-ranking terms or top-ranking term clusters. There is no obvious difference between the VSM baseline and GCDSS measure for standard K-means algorithm, and there is no obvious difference between the VSM baseline and CDSS measure for the SOM algorithm. That might be due to the different mechanisms of K-means and SOM. GCDSS measure highlights the key terms and it is helpful for SOM to pick the documents as the winning nodes through competitive learning if the documents have stronger relationships with these key terms. The K-means averages the contributions of these key terms on the centroid while GCDSS does not.

LSISSM decreases the number of the required features at least 71.3% (2600 out of 9070). The running time of the algorithms drops from overnight to just a few hours. The two-stage K-means improves the efficiency compared to the model-based standard K-means and speeds the clustering up to five times without losing clustering accuracy. Two-stage K-means reduces the initialization problem of standard K-means. Compared to the VSM baseline, the two-stage K-means algorithm runs faster by one or two orders of magnitude.

### 5.3.2. Model Parameters

Overall, LSISSM does not require strict parameter tuning and allows each parameter a large range of values which ensure the significance of the clustering results.

Besides, the parameters of the clustering algorithms themselves, the CDSS and GCDSS parameters $T_g$, $T_l$, $T_d$ and $\beta$ change the performance substantially. The range of $T_d$ in these experiments varies from 0.50 to 0.99. $T_d$ is often set as 0.95, which means the top K latent concept dimensions selected make 95% contribution to the LSI subspaces. $T_l$ ranges from 0.3 to 1.0. The initial value of $T_l$ is often set as 0.5. The value of $T_g$ represents the accumulated statistical global contribution of the feature subsets to the LSI term subspace calculated by GLCR. If $T_l$ and $T_d$ are predefined, the upper boundary of $T_g$ determines how many terms are included in the feature subset. The results shown in Figures 3 to 11 demonstrate the range of $T_g$ in which the clustering performance of the standard K-means and SOM are augmented is very large. The graphs in Figure 3 show that if $T_g$ is larger than 0.5, the performance of the standard K-means is consistently higher than the VSM baseline. The plateau in Figure 4 indicates that if $T_g$ is larger than 0.50 and less than 0.82, the performance of standard K-means is consistently higher than the VSM baseline. The plateau in Figure 5 indicates that if $T_g$ is larger than 0.55 and less than 0.85, the performance of standard K-means is consistently higher than the VSM baseline. The curves in Figure 6, 7 and 8 show that if $T_g$ is larger than 0.6, 0.5 and 0.58, the performance of standard K-means is consistently higher than the VSM baseline respectively. The trends in Figure 9, 10 and 11 show that if $T_g$ is larger than 0.05, 0.2 and 0.15, the performance of SOM is consistently higher than the VSM baseline respectively.

The pruning parameter $\beta$ in CDSS or GCDSS affects the clustering performance by determining how many latent concept dimensions are used in the comparison of the two signatures. For instance, changing the value of $\beta$ from 0.0 to 0.5 and then to 1.0, the NMI scores for the standard K-means decreases from 0.407 to 0.389 and then increase to 0.455 with a feature subset of 1800 top-ranking terms for the Reuters corpus. Empirically, the value of $\beta$ is set to 0.0 comparing a term signature and a document signature. If matching two term signatures, the scope of $\beta$ is 1.0 to 1.5. The effect of the term clusters on the performance of the standard K-means varies with the value of $\beta$ while CDSS is used to calculate the similarity between the term signatures. For instance, if the

value of β is changed from 1.0 to 1.5, for TDT1 with the top 100 term clusters, the Purity increases from 0.882 to 0.923, see Table 8.

## 6. Conclusion

We designed and developed a novel model, LSISSM, for semantic content analysis of unstructured text. The central components of the model follow the Zipf's Law -- the term frequency distribution rule in the documents. The model gives unified and comparable spectral signature representations for terms and documents in an unsupervised manner. A unique ranking mechanism for signatures sorts the terms and documents and controls information loss during dimension reduction. The similarity measures between the signatures reflect the coherence of term maps and document clusters on the LSI latent concept dimensions. The value of our model is demonstrated across the document clustering applications systematically controlled by a large range of model parameters.

The model should be able to automatically identify the topic labels for the document clusters. The feature ranking algorithm can also be applied to document ranking to select small representative sample subsets for active learning (Zhu, 2009; Zhu & Allen, in preparation). The LSISSM document signature ranking algorithm (GLCR) picks good training samples and maintaining the sampling distribution of the full set of text categories, even outlier categories. This method does not need to be given the labels of the categories. That is because the ranking algorithm selects the samples following the order of their importance concern both the global and local statistical contribution of the samples to the LSI document subspace. The samples with the highest statistical contribution to the corpus will be included in the subset first. Each sample picked is a good statistical representative of the corpus. So, the average contribution for each sample in the subset generated by our approach is higher than the average. Even compared to the independent tests on the full training set, our method does not decrease the performance and exceeds it in some cases. In addition, the results from these studies indicate that the sample subsets represented by a small feature subset improve and stabilize the effectiveness and efficiency of the classifiers without affecting the sample distribution of the text categories. The effects of the LSI subspace signature ranking on the unsupervised classification and supervised classification are consistent and identical.

## Acknowledgement

This paper is based on Zhu (2009). Weizhong Zhu was supported by IMLS Grant RE-05-05-0085-05. Robert B. Allen is now in the School of Information Management at Victoria University of Wellington.

## References

Ampazis, N. & Perantonis, S.J. (2004). LSISOM — A Latent Semantic Indexing Approach to Self-Organizing Maps of Document Collections, *Neural Processing Letters,* 19(2),157-173.

Arthur, D. & Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027-1035.

Beil, F., Ester, M. & Xu, X. (2002). Frequent term-based text clustering. *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, 436-442.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning* Research, 3, 993–1022.

Cai, D., He, X. & Han, J., (2005). Document Clustering Using Locality Preserving Indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12), 1624-1637.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6), 391-407.

Ding, C. H. (2005). A Probabilistic Model for Latent Semantic Indexing. *Journal of the American Society for Information Science*, 56(6), 597-608.

Dubes, R.C. & Jain, A.K. (1988). *Algorithms for Clustering Data,* Prentice Hall, New York.

Fung, B. C. M., Wang, K. and Ester, M. (2003). Hierarchical Document Clustering Using Frequent Itemsets, *SIAM International Conference on Data Mining* (SDM'03), San Francisco, CA, 59-70.

Gelbukh, A. & Sidorov, G. (2001). Zipf and Heaps Laws' Coefficients Depend on Language. Proc. CICLing-2001, *Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City. Lecture Notes in Computer Science N 2004, ISSN 0302-9743, ISBN 3-540-41687-0, Springer-Verlag, 332–335.

Griffiths, T. L., Steyvers, M. & Tenenbaum, J. B. (2007). Topics in Semantic Representation. *Psychological Review,* 114, 211-244.

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. *ACM SIGIR*, 50–57, New York: ACM Press.

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE* 78(9), 1464–1480.

Koll, M. (1979). WEIRD: An approach to concept-based information retrieval. ACM SIGIR Forum, 13(4), 32-50.

Kontostathis, A. & Pottenger, W. M., (2006). A Framework for Understanding Latent Semantic Indexing (LSI) Performance. *Information Processing and Management*, 42(1), 56-73.

Kontostathis, A., Pottenger W. M., and Davison B. D. (2005). Identification of Critical Values in Latent Semantic Indexing (LSI). In T.Y. Lin et. editors, *Foundations of Data Mining and Knowledge Discovery*, 333-346. Springer-Verlag.

Lin, X., Soergel, D. & Marchionini, G. (1991). A Self-Organizing Semantic Map for Information Retrieval, *ACM SIGIR*, 262-269.

Manning, D. C., Raghavan, P. & Schutze, H., (2008). *Introduction to Information Retrieval,* Cambridge University Press, ISBN 0521865719, 357-359.

Mason J. E. and Spiteri R. J. (2008) A new adaptive folding-up algorithm for information retrieval, *Proceedings of the Text Mining Workshop 2007*, 2008.

Ponte, J. & Croft, W.B. (1998). A Language Modeling Approach to Information Retrieval, *ACM SIGIR*, 275-281.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, *Psychological Review*, 65, 6, 386-408.

Rumelhart, D. E. (1990). Brain Style Computation: Learning and Generalization. In Zornetzer, S. F., Davis, J. L., and Lau, C. (eds.), *An Introduction to Neural and Electronic Networks*, 405-420. San Diego, CA: Academic Press.

Rumelhart, D. E., McClelland, J.L. & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.* Cambridge, MA: MIT Press.

Salton, G., Wong, A. & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18, 11, 613-620.

Schutze, H. & Silverstein, C. (1997). Projections for Efficient Document Clustering. *SIGIR Forum* 31, SI, 74-81.

Steinbach, M., Karypis, G. & Kumar, V. (2000). *A Comparison of Document Clustering Techniques*, Proc. TextMining Workshop at KDD 2000.

Tougas, J. E. & Spiteri, R. J. (2006) Updating the Partial Singular Value Decomposition in Latent Semantic Indexing, *Computational Statistics & Data Analysis*, 52, 174-183, 2006.

Tougas, J. E. & Spiteri, R. J. (2007) Two Uses for Updating the Partial Singular Value Decomposition in Latent Semantic Indexing. *Applied Numerical Mathematics*, 58, 499-510, 2007.

Toutanova, K. & Manning, D. C. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (EMNLP/VLC-2000), 63-70.

Toutanova, K., Klein, D., Manning, D. C. & Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL* 2003, 252-259.

Wang, K., Xu, C. & Liu, B. (1999). Clustering transactions using large items. *ACM CIKM*, 483–490.

Willett, P. (1998). Recent Trends in Hierarchic Document Clustering: A Critical Review. *Information Processing and Management*, 24(5), 577-597.

Werbos, P. J. (1994). The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting. New York: Wiley-Interscience.

Xu, W. & Gong, Y. (2004). Document Clustering by Concept Factorization, *ACM SIGIR*, 202–209.

Xu, W., Liu, X. & Gong, Y. (2003). Document Clustering Based on Non-Negative Matrix Factorization. *ACM SIGIR*, 267–273.

Zhu, W. (2009). *Text Clustering and Active Learning Using a Latent Semantic Indexing (LSI) Subspace Signature Model and Query Expansion*, Doctorate Dissertation, Drexel University. (http://idea.library.drexel.edu/bitstream/1860/3077/1/Zhu_Weizhong.pdf)

Zhu, W. & Allen, R. B., (in preparation). Active Learning for Text Classification Using the LSI Subspace Signature Model.

Zhu, W., & Chen, C. (2007). Storylines: Visual Exploration and Analysis in Latent Semantic Spaces, *International Journal of Computers and Graphics, Special Issue on Visual Analytics*. 31(3), 338-349.

Zhu, W., Chen, C. & Allen, R. B. (2008). Analyzing the Propagation of Influence and Concept Evolution in Enterprise Social Networks through Centrality and Latent Semantic Analysis. Washio, T. et al. (Eds.): *Advances in Knowledge Discovery and Data Mining, PAKDD 2008*, Osaka, Japan, LNCS, 5012, 1090-1098.