Enhancing Document Clustering Using Hybrid Models for Semantic Similarity

Ahmed K. Farahat* Mohamed S. Kamel†

Abstract

Different document representation models have been proposed to measure semantic similarity between documents using corpus statistics. Some of these models explicitly estimate semantic similarity based on measures of correlations between terms, while others apply dimension reduction techniques to obtain latent representation of concepts. This paper proposes new hybrid models that combine explicit and latent analysis to estimate semantic similarity between documents. The proposed models have been used to enhance the performance of document clustering algorithms. Experiments on thirteen benchmark data sets show that hybrid models achieve significant improvement in clustering performance when used with clustering algorithms that are sensitive to errors in estimating document similarity.

1 Introduction.

Document clustering is concerned with organizing documents into groups according to their topics. Algorithms for document clustering have been used since the early years of text retrieval systems to organize documents for end-user browsing and to improve the effectiveness and efficiency of the retrieval process. In recent years, document clustering has received considerable attention because of the huge number of documents which have become available on the web. This drives more research work to improve the efficiency of document clustering algorithms while maintaining their scalability.

Many clustering algorithms have been applied to document data sets. These algorithms include, but are not limited to: hierarchical methods [1], k-means [1], spectral clustering [2], and recently non-negative matrix factorization [3]. Most of these algorithms use the vector space model (VSM) [4] as their underlying model for document representation. VSM represents documents as vectors in the space of terms and measures proximity between documents based on the inner-product of their vectors. Some document clustering algorithms, like k-means and non-negative matrix factorization, are applied directly to document vectors, while others, like hierarchical methods and spectral clustering, are applied to the matrix of cosine similarities. The use of VSM as the underlying model for document representation totally ignores any semantic relations between terms. This means that documents with no common terms are considered dissimilar even if they have many terms that are semantically related. On the other hand, documents with many terms in common are considered similar even if these common terms are noisy and other terms in the two documents have no semantic relatedness.

Different models for document representation have been proposed to overcome the limitations of the VSM by capturing semantic similarity between documents based on the statistical analysis of term occurrence patterns. Some of these models, like the generalized vector space model (GVSM) [5], explicitly calculate measures of correlation between terms, and then use these measures to estimate document similarities. Other models, such as latent semantic indexing (LSI) [6], are based on using dimension reduction techniques to obtain latent representation of concepts.

In this paper, we propose new hybrid models for document representation that first map documents to a semantic space in which similarity between documents reflects how their terms are statistically correlated, and then apply dimension reduction techniques to obtain a concise representation that preserves semantic similarity between documents. The paper then performs empirical comparisons between the proposed models and four well-known models for semantic analysis, and studies how these models enhance the effectiveness of different document clustering algorithms. Experiments have been conducted on thirteen benchmark data sets using three well-known clustering algorithms.

The rest of the paper is organized as follows. Section 2 presents the necessary background. Sections 3 and 4 review two well-known methods for estimating statistical semantic similarity. Section 5 explains the hybrid approach in details. Experiments and results are presented in section 6. Finally, section 7 concludes the paper and discusses future work in using semantic similarity for enhancing document clustering.

*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada. Email: afarahat@pami.uwaterloo.ca.

†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada. Email: mkamel@pami.uwaterloo.ca.

2 Background.

2.1 Document Clustering. Document clustering is an unsupervised learning task which aims at organizing documents into groups according to their similarity. Different aspects of similarity between documents can be defined. The most commonly-used aspect is the topic similarity, which is usually estimated based on the proximity of document vectors in the space of terms.

Data clustering algorithms can be generally categorized into hierarchical and partitional [1]. Hierarchical clustering constructs a hierarchy of nested clusters, while partitional clustering divides data points into non-overlapped clusters such that a specific criterion function is optimized. Hierarchical algorithms are either agglomerative or divisive. In document clustering, hierarchical agglomerative clustering (HAC) is more common. HAC algorithms differ in the way they calculate similarity between clusters. The most commonly-used algorithms are single-link, complete-link and average-link clustering [1]. Partitional algorithms used for document clustering include, but are not limited to: the k-means algorithm [1], spectral clustering [2], and non-negative matrix factorization [3].

The k-means algorithm [1] is the most widely used algorithm for data clustering. The goal of the algorithm is to group data points into k clusters such that the Euclidean distances between data points in each cluster and its centroid are minimized. Spherical k-means [7] is a variant of the basic k-means algorithm that uses cosine similarity between data points instead of the Euclidean distance. Spherical k-means is usually used with document data sets where the cosine similarity is a measure more indicative of proximity between documents.

Although empirical comparisons have shown that partitional algorithms are more efficient and effective than hierarchical algorithms in the document clustering task [8], a hierarchy of document clusters is more informative than a flat partitioning as it naturally represents the hierarchy of topics. In addition, hierarchical algorithms are suitable for applications in which the number of clusters is not predefined.

2.2 Document Representation. Most of the models proposed for document representation originated in information retrieval systems to calculate the relevance of documents to a user query. The most commonly-used model for document representation is the vector space model (VSM) [4]. VSM represents documents as vectors in the space of index terms, and measures proximity between documents using the inner-product of their vectors. Given a set of n documents $D = \{d_j : j = 1, .., n\}$ and a set of m terms $T = \{t_i : i = 1, .., m\}$. Let X be an $m \times n$ term-document matrix whose element x_{ij} represents the weight of term t_i inside document d_j. VSM uses the columns of X to directly represent documents. The matrix that encodes the inner-products of document vectors can be calculated as:

$$K_{VSM} = X^T X \tag{2.1}$$

where K_{VSM} is an $n \times n$ matrix which is called the kernel matrix [9]. The matrix of cosine similarities can be calculated from the kernel matrix as:

$$Sim = L^{-1/2} K L^{-1/2} \tag{2.2}$$

where L is a $n \times n$ diagonal matrix whose diagonal elements are the diagonal elements of K.

VSM has been used in many text mining tasks, where it has achieved good results as well as an acceptable computational complexity. However, VSM in its original form assumes that terms are independent and accordingly ignores any semantic relations between them. This assumption implies that proximity between documents does not reflect their true topic similarity. In addition, redundancy in representation increases the dimensionality of document vectors and negatively affects the performance of the underlying algorithms.

To overcome the limitations of VSM, different representation models have been proposed to estimate statistical measures of semantic similarity between documents. Generalized vector space model (GVSM) [5], and latent semantic indexing (LSI) [6] are two models for estimating semantic similarity that are discussed in details in sections 3 and 4.

Other models for document representation are based on using lexical databases, such as WordNet [10], to represent documents in the space of concepts and calculate their similarity. WordNet-based models have been used in different tasks including clustering [11]. Similar representation models are based on exploiting knowledge from an encyclopedia (like Wikipedia). Explicit semantic analysis (ESA) [12] is such a model, which represents terms as vectors in a space of concepts represented by articles from Wikipedia. These models, however, use an external source of knowledge and are outside the scope of this paper.

3 Generalized Vector Space Model (GVSM).

The generalized vector space model (GVSM) is a document representation model that was proposed by Wang et al [5] to estimate similarity between documents based on how their terms are related. Wang et al highlighted that VSM in its original form assumes that term vectors form an orthonormal-basis, and proposed a new model which removes this orthogonality assumption. In

GVSM, term vectors are used as a non-orthogonal basis in which documents are represented. The kernel matrix in the new basis is calculated as:

$$K_{GVSM} = X^T G X \tag{3.3}$$

where G is an $m \times m$ Gram matrix (also called the association matrix) which represents the inner-products of term vectors in some space. The GVSM model in its original form estimates the Gram matrix G by representing terms in an orthonormal-basis of 2^m vectors which represent the min-terms[1] that can be formed by taking different combinations of terms. The association measures between terms are calculated as the cosine of the angle between their vectors in the new space. However, as the maximum number of min-terms that appear in n documents is n, the GVSM is usually simplified by assuming that each document has a unique min-term. This assumption means that terms are represented as vectors in the dual space of documents. Accordingly, G can be calculated as: $G = L^{-1/2} X X^T L^{-1/2}$, where L is a diagonal matrix whose diagonal elements are the lengths of term vectors in the dual space. Other versions of GVSM [13] calculate G as the inner-products of term vectors in the dual space of documents: $G = XX^T$.

GVSM has been used in information retrieval, where it has not achieved much improvement. However, it has been successfully used in multilingual information retrieval [13] where documents available in different languages are used to construct G. The similarity between a user query in one language and documents in another language can be calculated using the GVSM similarity kernel. Recent work [14] studied the use of different GVSM-based models for improving the effectiveness of document clustering algorithms.

4 Latent Semantic Indexing (LSI).

Latent semantic indexing (LSI), originally proposed by Deerwester et al [6], is another document representation model which is based on decomposing the term-document matrix using singular value decomposition (SVD): $X = U \Sigma V^T$. The leading left and right singular vectors are used to represent terms and documents in some semantic space. Let U_d and V_d be $m \times d$ and $n \times d$ matrices whose columns are the leading d left and right singular vectors of X respectively, Σ_d is a $d \times d$ diagonal matrix whose diagonal elements are the largest d singular values of X. The matrix U_d is directly used to represent terms in the d-dimensional semantic space. Each document is then represented in the semantic space as a linear combination of their term vectors: $X_d^{(LSI)} = U_d^T X$. The corresponding kernel matrix is:

$$K_{LSI} = X^T U_d U_d^T X \tag{4.4}$$

The use of SVD obtains the best rank d approximation of K in terms of the Frobenius norm[2]: $\|K - K_{LSI}\|_F$. This means that LSI preserves similarity between documents in the VSM as much as possible.

LSI is highly related to principal component analysis (PCA) [15] which is a well-known method for dimension reduction. PCA is equivalent to LSI if each column of the data matrix has a zero mean.

Latent semantic kernel (LSK) [16] is a kernel-based version of the LSI method. Given a kernel matrix K which represents the inner-product of documents in some space, the eigenvalue decomposition of K can be computed as:

$$K = U \Lambda U^T$$

where U is an $n \times n$ matrix whose columns are the eigenvectors of K, and Λ is an $n \times n$ diagonal matrix whose diagonal elements are the eigenvalues of X. The latent semantic indexing could be performed by taking the d eigenvectors of K which correspond to the largest eigenvalues and using them to represent document vectors: $X_d^{(LSK)} = \Lambda_d^{1/2} U_d^T$. Similarly, kernel PCA [17] is a kernel-based version of the PCA method.

5 Hybrid Models for Semantic Analysis.

This section proposes new hybrid models for document representation which combine GVSM-based models with dimension reduction techniques. The models first construct a semantic space in which similarity between documents encodes how their terms are statistically correlated. Dimension reduction algorithms are then applied to document vectors in the semantic space to obtain latent concepts.

Hybrid models differ from traditional latent models in the properties of documents they preserve in the latent space. LSI and PCA are applied directly to the term-document matrix, and they essentially preserve VSM-based similarity between documents. These similarities, as discussed in section 2.2, do not reflect any semantic relations between terms. Hybrid models, on the other hand, preserve semantic similarity that is explicitly estimated from measures of term-term correlations. We empirically show that preserving explicit measures of semantic similarity is more effective than preserving VSM-based similarity, and it achieves better performance with the document clustering task.

[1] In Boolean algebra, a min-term is a product term in which each variable appears once.

[2] The Frobenius norm $\|K\|_F$ is calculated as $\sqrt{\Sigma_{i,j} |k_{ij}|^2}$

5.1 Mapping Documents to Semantic Space. In the first step of the proposed approach, documents are mapped to a semantic space in which the proximity between their vectors represents how their terms are statistically correlated. The document representation model used is based on the generalized vector space model (GVSM) [5]. GVSM assumes that term vectors are linearly-independent and represents documents as a linear combination of term vectors.

The kernel matrix K that represents the inner-products of document vectors in the semantic space is calculated using equation 3.3. The Gram matrix of term vectors G encodes measures of statistical correlations between terms, which can be estimated from the documents to be clustered. Different measures of correlations can be used to estimate G. The proposed similarity model requires G to be a positive semi-definite as it represents the inner-products of term vectors in the semantic space.

Our recent work [14] provided a theoretical and empirical analysis of the effectiveness of different estimates of G in improving the performance of document clustering algorithms. These estimates include the association and normalized association matrices, which measure term-term correlations using the inner-products and the cosine similarities of term vectors in the space of documents respectively. We also studied estimates based on the term-term covariance matrix and the matrix of Pearson's correlation coefficients between terms. The use of these two matrices implies the assumption that terms are random variables with Gaussian distributions. Our analysis showed that using the term-term covariance matrix achieves the best performance with different clustering algorithms.

In the proposed hybrid models, the term-term covariance matrix G_{COV} is calculated from the term document matrix as:

$$G_{COV} = \frac{1}{n-1}\widetilde{X}\widetilde{X}^T, \tag{5.5}$$

where $\widetilde{X}$ is a matrix that is obtained from X by centering its columns (i.e., subtracting the mean of each column from its elements).

The kernel matrix K that represents the inner-products of document vectors in the semantic space can be expressed as $K = W^T W$, where W is an $n \times n$ matrix whose columns represent the document vectors in the semantic space:

$$W = \frac{1}{\sqrt{n-1}}\widetilde{X}^T X, \tag{5.6}$$

To calculate W, the matrix $\widetilde{X}$ does not have to be calculated and stored. Instead, $X^T X$ can be first calculated and then multiplied by an $n \times n$ centering matrix H to obtain $\widetilde{X}^T X$. The centering matrix H implicitly subtracts the mean of each column from its elements. H can be expressed as: $H = I - \frac{1}{n}ee^T$, where e is the all-ones vector of length n. Equation 5.6 can be simplified to: $W = \frac{1}{\sqrt{n-1}}(X^T X - X^T \tilde{x} e^T)$, where $\tilde{x}$ is a vector of length n whose element j is the mean of the j^{th} column of X. Using this formula and given that X is very space, W can be calculated in an efficient way.

Representing documents in the semantic space formed by the columns of W has many desirable properties compared to the traditional VSM representation. One property is that the length of a term vector in the semantic space is proportional to its term variance quality (TVQ) [18] which has been used as a feature selection measure in document clustering. This means that terms are automatically assigned weights that reflect how important they are in the corpus of documents. Another property is that the cosine similarity between term vectors reflects their statistical correlation. G_{COV} maps positively correlated terms to vectors in almost the same direction in the semantic space ($cos \approx 1$), uncorrelated terms to near-orthogonal directions ($cos \approx 0$), and negatively correlated terms to opposite directions ($cos \approx -1$). As the document vectors are represented as a linear combination of term vectors, their proximity in the semantic space reflects the significance of their terms and how these terms are statistically correlated.

In the next section, we discuss how dimension reduction techniques are applied to the document vectors in the semantic space.

5.2 Dimension Reduction. Given the representation of documents in the semantic space, W (equation 5.6), dimension reduction techniques can be applied to the matrix W or to the kernel matrix $K = W^T W$ to obtain a concise representation of document vectors that preserves semantic similarity between documents. Dimension reduction removes the noise or irrelevant information in the original term-document matrix, and in the calculation of term-term correlations. It also allows documents to be clustered in a more efficient way.

In our experiments, we use both the latent semantic indexing (LSI), and the principal component analysis (PCA) methods for reducing the dimension of document vectors. In the case of LSI, the singular value decomposition of W is calculated as:

$$W = U\Sigma V^T, \tag{5.7}$$

where U and V are $n \times n$ matrices whose columns are the left and right singular vectors of W respectively. The singular vectors that correspond to the leading singular values can be used to represent the document vectors in

Algorithm 1 Spherical k-Means with Hybrid Models

Inputs: X, Q, k, d, t_{max}
Outputs: $\Pi = \{\pi_1, ..., \pi_k\}$
Steps:

1. $W = \frac{1}{\sqrt{n-1}} \widetilde{X}^T X$,
2. $[U, \Sigma, V] = svd\,(W)$
3. $W_d = U_d^T W = \Sigma_d V_d^T$
4. Initialize: $\Pi_0 = \{\pi_1, ..., \pi_k\}$, $t = 1$
5. $\overrightarrow{\mu_j} = \frac{\sum_{x_i \in \pi_j} \overrightarrow{w_{di}}}{\left\| \sum_{x_i \in \pi_j} \overrightarrow{w_{di}} \right\|},\ j = 1..k$
6. $y_i = arg\ max_j\ cos\,(\overrightarrow{w_{di}}, \overrightarrow{\mu_j})\,,\ i = 1..n$
7. $\pi_j = \{x_i :\ y_i = j\},\ j = 1..k$
8. $\Pi_t = \{\pi_1, ..., \pi_k\}$
9. If $(\Pi_t \neq \Pi_{t-1}\ \&\ t < t_{max})$ $t = t + 1$, Go to 5.
   Else Return Π_t

the latent semantic space as follows:

$$W_d = U_d^T W = \Sigma_d V_d^T, \tag{5.8}$$

where W_d is an $d \times n$ matrix whose columns represent the document vectors in the latent semantic space. U_d and V_d are $n \times d$ matrices whose columns are the leading d left and right singular vectors of W respectively. Σ_d is a diagonal matrix of the singular values of W. In the case of PCA, the representation of document vectors can be obtained by applying singular value decomposition to the matrix $\widetilde{W}$ obtained by centering the columns of W.

The representation of documents in the semantic space can also be obtained by applying the latent semantic kernel (LSK) to the kernel matrix $K = W^T W$ (or kernel PCA [17] in the case of PCA). The kernel matrix K is decomposed using eigenvalue decomposition: $K = U \Lambda U^T$. The leading eigenvectors U_d are then used to represent the document vectors in the low-dimension space. This method is equivalent to applying LSI on W (or PCA on $\widetilde{W}$). The rank-d kernel matrix K_d of document vectors in the low-dimension space can be obtained as: $K_d = U_d \Lambda_d U_d^T$.

5.3 Clustering in the Latent Semantic Space. Document clustering algorithms that are vector-based (such as k-means) can be applied directly in the semantic space on the columns of matrix W_d. Other algorithms that are based on similarities between documents (like hierarchical clustering) can be applied to the matrix of cosine similarities Sim. Kernel-based algorithms can also be applied to the kernel matrix K. However, clustering algorithms that require non-negative values in data vectors (like NMF) or the kernel matrix (like spectral clustering) cannot be directly applied to W_d and K. One way to apply these algorithms is to remove negative entries from the matrices. This can be done by simply setting all negative values to zero or adding some constant to all the elements of the matrix. The study of more advanced techniques for removing negative values is a subject for future work.

Algorithm 1 shows the steps of applying spherical k-means in the latent semantic space. k is the number of clusters, d is the number of dimensions of the semantic space, t_{max} is the maximum number of iterations, Π is the output partitioning of documents, and svd is the singular value decomposition function.

6 Experiments and Results.

6.1 Data Sets. Experiments have been conducted on thirteen benchmark data sets. The properties of the different data sets are summarized in table 1. These data sets have been previously used to evaluate different algorithms for document clustering. The *20ng* is a benchmark data set which consists of newsgroup documents. We used the mini-newsgroups version which is available at the UCI KDD Archive[3]. The pre-processing steps include the removal of message headers, stop-word removal, and stemming. The other data sets were used by Zhao and Karypis [8][21] to evaluate the performance of different document clustering algorithms. The *classic* data set consists of CACM, CISI, CRANFIELD, and MEDLINE abstracts[4]. The *fbis*, *hitech*, *reviews*, *la12*, *tr31* and *tr41* data sets are from TREC collections[5]. The *re0* and *re1* data sets are two subsets of Reuters-21578 [19]. The *k1a*, *k1b*, and *wap* data sets are from the WebACE project [20]. We used the pre-processed versions of these data sets distributed with the CLUTO Toolkit [22]. The pre-processing steps which have been applied to these data sets are stop-word removal and stemming. In all data sets, words that appear in only one document are removed and the normalized term frequency - inverse document frequency (tf-idf) measures are used to weight terms inside documents.

6.2 Experimental Setup. Different experiments have been conducted to evaluate the effectiveness of document clustering using hybrid models compared to well-known document representation models.

[3] http://kdd.ics.uci.edu
[4] ftp://ftp.cs.cornell.edu/pub/smart
[5] http://trec.nist.gov

Table 1: The properties of data sets used to evaluate different representation models. n, m, and k are the number of documents, terms, and classes respectively. m_{doc} is the average number of terms per document, and n_{class} is the average number of documents per class.

ID	Data set	Source	n	m	m_{doc}	k	n_{class}
D01	20ng	20 Newsgroups	2000	28839	23.3 ± 49.1	20	100.0 ± 0.0
D02	classic	Different Abstracts	7094	41681	6.2 ± 7.7	4	1773.5 ± 971.4
D03	fbis	TREC	2463	2000	68.5 ± 88.7	17	144.9 ± 139.3
D04	hitech	TREC	2301	126321	37.9 ± 27.9	6	383.5 ± 189.9
D05	reviews	TREC	4069	126354	43.3 ± 34.8	5	813.8 ± 520.9
D06	la12	TREC	6279	31472	43.5 ± 38.0	6	1046.5 ± 526.5
D07	tr31	TREC	927	10128	111.9 ± 248.3	7	132.4 ± 124.0
D08	tr41	TREC	878	7454	66.5 ± 100.5	10	87.8 ± 80.1
D09	re0	Reuters-21578 [19]	1504	2886	15.0 ± 14.5	13	115.7 ± 173.8
D10	re1	Reuters-21578[19]	1657	3758	15.4 ± 12.3	25	66.3 ± 91.8
D11	k1a	WebACE [20]	2340	21839	44.5 ± 20.8	20	117.0 ± 117.5
D12	k1b	WebACE [20]	2340	21839	44.5 ± 20.8	6	390.0 ± 513.3
D13	wap	WebACE [20]	1560	8460	43.2 ± 20.5	20	78.0 ± 81.1

Three document clustering algorithms are used for evaluation: spherical k-means, and hierarchical agglomerative clustering (HAC) with both complete and average linkage. For spherical k-means, a MATLAB implementation of Algorithm 1 is used. The output clusters are refined by using an incremental optimization technique which moves individual documents between clusters. As Algorithm 1 is non-deterministic, it is repeated ten times using different initial solutions, and the solution with the best value of the objective function is selected. This experiment is repeated fifty times, and the average and standard deviation of quality measures are calculated. In each run of spherical k-means, the maximum number of iterations used is 100. For HAC algorithms, the MATLAB function *linkage* is used.

We compare six document representation models, including four well-known models (VSM, GVSM-COV - GVSM based on a covariance matrix [14] - LSI, and PCA), and two proposed models (LSI-COV - LSI with GVSM based on a covariance matrix - and PCA-COV - PCA with GVSM based on a covariance matrix). The spherical k-means algorithm is directly applied to document vectors, and the HAC algorithm is applied to the matrix Sim obtained by normalizing the kernel matrix K as shown in equation 2.2.

6.3 Performance Evaluation. The clusters obtained by different algorithms are compared to the ground-truth partitioning of documents. In order to evaluate the output of HAC algorithms, a flat partitioning of documents is obtained by traversing the hierarchy from the top cluster until the predefined number of clusters is reached.

We used three external quality measures to evaluate the performance of the clustering algorithms: F-measure, entropy, and purity. Higher values of F-measure and purity, and lower values of entropy indicate better clustering solutions. Let n be the total number of documents, n_{ij} be the number of documents that belong to class i and cluster j, n_i be the number of documents in class i, and n_j be the number of documents in cluster j. To calculate F-measure, the precision, recall, and F-measure of mapping class i to cluster j are first calculated as: $P_{ij} = n_{ij}/n_i$, $R_{ij} = n_{ij}/n_j$, $F_{ij} = 2P_{ij}R_{ij}/(P_{ij} + R_{ij})$. The F-measure of class i is then calculated as the maximum of F-measure of mapping this class to all clusters: $F_i = max_j \{F_{ij}\}$. The overall F-measure is calculated as:

$$F = \sum_{i=1}^{c} \frac{n_i}{n} F_i.$$

The entropy measures the homogeneity of clusters with respect to classes. Let $p_{ij} = n_{ij}/n_i$ be the probability that a member of cluster j belongs to class i, then the entropy of a cluster j is calculated as: $E_j = -\sum_{i=1}^{c} p_{ij} log(p_{ij})$. The overall entropy is then calculated as:

$$E = \sum_{j=1}^{k} \frac{n_j}{n} E_j.$$

The purity measures the average precision of clusters relative to their best matching classes. The purity of a cluster j is calculated by first assigning cluster

j to the most dominant cluster j, and then dividing the number of documents that belong to cluster j and class i by the total number of documents in cluster j: $P_j = \frac{1}{n_j} max_i \{n_{ij}\}$. The overall purity is then calculated as:

$$P = \sum_{j=1}^{k} \frac{n_j}{n} P_j$$

In the case of representation models that are based on dimension reduction, determining the intrinsic dimension of the semantic space, d, is a common problem in all dimension reduction techniques. Different heuristics exist for estimating d, however none of them is proven to achieve the best performance. In order to compare different representation models, we used an approach similar to the one suggested by Cai et al [23]. In this approach, the average of the best values of each quality measure for different values of d is used to estimate the performance of the representation model. In particular, we change the number of dimensions in the semantic space, d, from 5 to 100 with increments of 1, and evaluate all quality measures for each value of d. The best 10 values of each quality measure are obtained. The average and standard deviation of these best values are calculated and used to represent the quality of the representation model.

To compare two representation models, M_1 and M_2, for a specific clustering algorithm and data set, the average and standard deviation of quality measures are calculated for both representation models: ($\overline{q_1}$, s_1) and ($\overline{q_2}$, s_2) (for deterministic models, the standard deviation is 0). The t-test is then used to assess the significance of one method with respect to the other. We consider the null-hypothesis that the two methods are equivalent. The t-statistic is calculated as:

$$t = \frac{\overline{q_1} - \overline{q_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}},$$

where n_1 and n_2 are the number of observations used to estimate $\overline{q_1}$ and $\overline{q_2}$ respectively. The value of t-statistic is then compared to the critical value ($t_{critical}$) obtained from the t-distribution table for a 95% confidence interval. If $t > t_{critical}$, the null-hypothesis is rejected and the two representation models are considered inequivalent. In this case, if $\overline{q_1}$ is better than $\overline{q_2}$, the representation model M_1 is considered superior to M_2 for this clustering algorithm and data set.

To compare two representation models over all data sets, we use an approach similar to that suggested by Zhao and Karypis [21]. In this approach, the quality measures for a particular data set and clustering algorithm are normalized relative to the best value of the quality measure obtained by using different representation models. In the case of entropy, the relative entropy E_r is calculated by dividing the minimum entropy obtained by the original entropy value:

$$E_r = \frac{min(E)}{E}.$$

In the case of F-measure, and purity, the relative measure is calculated by dividing the value of the quality measure by the maximum value obtained:

$$F_r = \frac{F}{max(F)},\ P_r = \frac{P}{max(P)}.$$

The relative quality measures range from 0 to 1. The best representation models have relative quality measures that are close to 1. The higher the relative measures, the better is the representation model. The relative quality measures are then averaged for different data sets. The average values of relative quality measures for two representation models are then compared by applying a statistical significance test on these averages. Similar to [21], we use the paired t-test in which the original quality measures of one representation model M_1 are subtracted from their corresponding measures for the other representation model M_2 for all data sets. The distribution of these differences are tested for statistical significance. Pairs of representation models for which the null-hypothesis is rejected are considered statistically equivalent. In all t-tests, we use a confidence interval of 95%.

6.4 Results and Analysis. Table 2 shows the average of the relative quality measures over all data sets for different clustering algorithms. Table 3 shows a comparison between different pairs of representation models using significance tests. Each column represents a comparison between two methods (A, B). The symbols $\gg$, $\ll$, and $=$ indicate that A is significantly superior, inferior, and equivalent to B respectively.

We can observe from tables 2 and 3 that using different representation models that capture semantic similarity achieves significant improvement (based on t-test) compared to the basic VSM model for all clustering algorithms. However, the improvement is much larger in the case of agglomerative algorithms than spherical k-means. For instance, when comparing GVSM with VSM, the improvements in relative F-measure are around 25% for HAC-complete, 15% for HAC-average, and 5% for spherical k-means.

We can also observe from table 3 that models which capture semantic similarity are statistically equivalent when used with the spherical k-means. In addition, GVSM is equivalent to LSI (column 2,3) for most quality

Table 2: Average of relative quality measures for different representation models and clustering algorithms.

Models	**1.VSM**	**2.GVSM-COV**	**3.LSI**	**4.PCA**	**5.LSI-COV**	**6.PCA-COV**
Algorithms	**Relative F-Measure**					
Spherical k-means	0.9186	0.9668	0.9750	0.9640	0.9780	0.9676
HAC-Complete	0.6633	0.9113	0.8503	0.8943	0.9828	0.9742
HAC-Average	0.7109	0.8605	0.9160	0.9712	0.9662	0.9557
Algorithms	**Relative Entropy**					
Spherical k-means	0.9000	0.9720	0.9706	0.9619	0.9529	0.9383
HAC-Complete	0.5941	0.8897	0.8187	0.8879	0.9732	0.9616
HAC-Average	0.6066	0.7887	0.9002	0.9824	0.9462	0.9493
Algorithms	**Relative Purity**					
Spherical k-means	0.9408	0.9826	0.9850	0.9887	0.9857	0.9797
HAC-Complete	0.6583	0.9136	0.8997	0.9382	0.9869	0.9809
HAC-Average	0.6775	0.8296	0.9219	0.9873	0.9470	0.9517

measures. On the other hand, for HAC-complete, PCA is equivalent to GVSM (column 2,4), and superior to LSI (column 3,4) for all quality measures. For HAC average, PCA is superior to GVSM (column 2,4) for all quality measures, and equivalent to LSI (column 3,4) for most quality measures.

In the case of the hybrid models LSI-COV and PCA-COV, we can observe that they are superior to all other models when used with HAC-complete. For instance, when comparing LSI-COV to LSI, the improvements with HAC-complete are around 13% in F-measure, 15% in entropy, and 8% in purity. However, LSI-COV and PCA-COV are statistically equivalent to LSI and PCA respectively for HAC-average and spherical k-means. LSI-COV and PCA-COV (column 5,6) are statistically equivalent for all clustering algorithms.

Based on these observations, we can conclude that the effectiveness of different models for estimating semantic similarity depends on how the clustering algorithm works. Partitional algorithms, like spherical k-means, assign data points to clusters such that a global criterion function is optimized. This criterion function is calculated based on the similarity measures between cluster centroids and all data points. So, if there are errors in estimating some of these similarities (e.g., between some document and a centroid), the effect of these errors will be compensated by other measures of similarity (e.g., between the same document and other centroids). We think that is the reason why there is no difference in performance when using different models for estimating semantic similarity, and why the improvements achieved by using semantic similarity relative to VSM are small compared to the improvements in the case of HAC algorithms. On the other hand, HAC algorithms construct clusters in a hierarchical manner, by making a local decision at each level of the hierarchy. In the case of HAC with complete linkage, clusters are merged based on the most dissimilar points in each pair of clusters. If there is an error in the estimate of similarity for this point, this will affect the rest of the cluster hierarchy. The HAC algorithm with average linkage is however less sensitive to errors in estimating similarity between documents, as it merges two clusters based on the average of similarities between all points in the two clusters. This means that in the case of HAC algorithms, the better the algorithm in estimating semantic similarity, the greater the improvement in performance of the hierarchial algorithms. This is why some latent models are better than explicit models for HAC algorithms, and why hybrid models achieve large improvements compared to latent models in the case of HAC algorithm with complete linkage.

Table 4 shows, for each data set, the values of F-measure for the output clustering obtained using different clustering algorithms and document representation models. For each data set and clustering algorithm (a column in the sub-table), the representation models are divided into groups according to the statistical significance between the distribution of their quality measures. The group of methods with the best values is highlighted in bold, while the group with the second best values is underlined. We can observe that the proposed hybrid models achieve the best performance for many data sets, especially when the hierarchical clustering with complete linkage is employed.

The improvement achieved by hybrid models with

Table 3: Comparison between different pairs of models (A, B) for each clustering algorithm based on statistical significance (using t-test). The symbols $\gg$, $\ll$, and $=$ indicate that A is significantly superior, inferior, and equivalent to B respectively.

Methods	**1,2**	**1,3**	**1,4**	**1,5**	**1,6**	**2,3**	**2,4**	**2,5**	**2,6**	**3,4**	**3,5**	**3,6**	**4,5**	**4,6**	**5,6**
Algorithms	**Relative F-Measure**														
Spherical k-means	$\ll$	$\ll$	$=$	$\ll$	$\ll$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$
HAC-Complete	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\gg$	$=$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$
HAC-Average	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$\ll$	$\ll$	$=$	$=$	$\ll$	$=$	$=$	$=$	$=$
Algorithms	**Relative Entropy**														
Spherical k-means	$\ll$	$\ll$	$\ll$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$
HAC-Complete	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$=$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$
HAC-Average	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$=$	$=$	$=$	$=$
Algorithms	**Relative Purity**														
Spherical k-means	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$
HAC-Complete	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$=$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$
HAC-Average	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$	$=$	$\ll$	$\ll$	$\ll$	$=$	$=$	$=$	$=$	$=$	$=$

agglomerative algorithms comes at the cost of additional computational complexity. Although the calculation of W can be done in an efficient way as discussed in section 5.1, the calculation of kernel K for HAC algorithms is more computationally demanding as the matrix W is non-sparse.

7 Conclusions and Future Work.

This paper proposes hybrid models for document representation that capture statistical similarity by applying dimension reduction techniques in a semantic space in which similarity between document vectors reflects how their terms are statistically correlated. The paper studies the effectiveness of the proposed models in enhancing document clustering and compares them to well-known models for document representation.

Results show that hybrid models are either statistically significantly better or equivalent to other representation models that capture semantic similarity between documents. Clustering algorithms that are based on making local decisions, such as hierarchical algorithms, are more sensitive to errors in estimating document similarity, and accordingly benefit more from hybrid models. The output of hierarchical algorithms is more informative for document data sets as it naturally represents the hierarchy of topics.

Future work in enhancing document clustering using semantic analysis includes studying the problem of determining the intrinsic dimensionality of the latent space, developing techniques for using the proposed models with clustering algorithms that require non-negative matrices, and reducing the computational complexity of semantic mapping and dimension reduction.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[2] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[3] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[4] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[5] S. K. M. Wong, W. Ziarko, and P. C. N. Wong, "Generalized vector spaces model in information retrieval," in *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1985, pp. 18–25.

[6] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci. Technol.*, vol. 41, no. 6, pp. 391–407, 1990.

[7] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Mach. Learn.*, vol. 42, no. 1/2, pp. 143–175, 2001.

[8] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Min. Knowl. Disc.*, vol. 10, no. 2, pp. 141–168, 2005.

Table 4: F-measures for different representation models. The best group of models for each data set and algorithm is highlighted in bold, the second best group is underlined.

Data sets	**D01**	**D02**	**D03**	**D04**	**D05**	**D06**	**D07**	**D08**	**D09**	**D10**	**D11**	**D12**	**D13**
Models	**Spherical k-means**												
VSM	0.41	0.67	0.58	0.48	0.74	**0.72**	0.67	0.67	0.45	0.47	**0.52**	0.72	0.50
GVSM-COV	0.54	**0.72**	0.58	0.50	0.73	**0.73**	0.70	0.67	0.45	0.48	**0.57**	0.75	**0.57**
LSI	0.55	0.66	0.58	0.51	0.74	**0.73**	0.69	**0.72**	0.46	**0.50**	**0.56**	**0.78**	**0.56**
PCA	0.56	0.70	0.57	0.51	0.71	**0.72**	0.68	0.71	0.47	0.49	**0.56**	0.69	**0.55**
LSI-COV	**0.56**	0.62	0.59	**0.52**	**0.77**	**0.73**	**0.70**	0.68	0.46	0.49	**0.58**	0.77	**0.58**
PCA-COV	0.55	0.69	**0.59**	0.48	0.69	**0.73**	0.69	0.70	**0.49**	0.48	**0.58**	0.71	**0.58**
Models	**HAC with complete linkage**												
VSM	0.16	0.45	0.55	0.33	0.41	0.32	0.73	0.59	0.41	0.32	0.46	0.48	0.52
GVSM-COV	0.42	0.70	0.61	**0.49**	0.59	0.63	0.76	0.68	0.46	0.52	0.61	0.71	0.59
LSI	0.49	0.53	0.53	0.44	0.61	0.55	0.60	0.67	0.48	0.47	0.58	0.66	0.57
PCA	0.49	0.65	0.53	**0.48**	0.71	0.63	0.65	0.70	0.48	0.48	0.57	0.66	0.57
LSI-COV	**0.50**	0.72	**0.64**	**0.50**	**0.78**	**0.69**	**0.80**	0.70	0.47	**0.55**	0.62	**0.80**	0.62
PCA-COV	**0.50**	**0.75**	0.59	**0.50**	0.70	0.66	**0.79**	**0.73**	**0.51**	0.53	**0.64**	0.74	**0.64**
Models	**HAC with average linkage**												
VSM	0.10	0.45	0.61	0.33	0.41	0.33	0.72	0.65	0.41	0.54	0.52	0.81	0.53
GVSM-COV	0.22	0.63	0.64	0.53	0.63	0.59	0.81	0.62	**0.50**	0.59	0.54	0.86	0.54
LSI	0.48	0.49	0.65	0.48	0.54	0.71	0.81	**0.76**	**0.49**	0.62	0.60	**0.90**	0.60
PCA	0.52	**0.69**	0.64	0.51	**0.71**	0.73	0.84	**0.76**	**0.51**	0.55	**0.63**	0.90	**0.63**
LSI-COV	0.47	0.62	**0.67**	**0.55**	0.63	**0.76**	**0.86**	0.70	**0.52**	**0.64**	0.61	0.89	0.61
PCA-COV	**0.52**	0.66	0.67	0.50	0.57	**0.76**	0.80	0.73	**0.51**	0.60	0.62	0.89	**0.63**

[9] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[10] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[11] A. Hotho, S. Staab, and G. Stumme, "Wordnet improves text document clustering," in *Proceedings of the SIGIR 2003 Semantic Web Workshop*. New York, NY, USA: ACM, 2003, pp. 541–544.

[12] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 6–12.

[13] J. Carbonell, Y. Yang, R. Frederking, R. Brown, Y. Geng, and D. Lee, "Translingual information retrieval: A comparative evaluation," in *International Joint Conference on Artificial Intelligence*, vol. 15, 1997, pp. 708–715.

[14] A. K. Farahat and M. S. Kamel, "Document clustering using semantic kernels based on term-term correlations," in *ICDMW '09: Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 459–464.

[15] I. Jolliffe, *Principal component analysis*. Springer verlag, 2002.

[16] N. Cristianini, J. Shawe-Taylor, and H. Lodhi, "Latent semantic kernels," *J. Intell. Inf. Syst.*, vol. 18, no. 2, pp. 127–152, 2002.

[17] B. Scholkopf, A. Smola, and K. Muller, "Kernel principal component analysis," *Lecture notes in computer science*, vol. 1327, pp. 583–588, 1997.

[18] I. Dhillon, J. Kogan, and C. Nicholas, *Feature selection and document clustering*. Springer-Verlag New York Inc, 2003, ch. 4, pp. 73–100.

[19] D. Lewis, "Reuters-21578 text categorization test collection distribution 1.0," 1999.

[20] E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "WebACE: A web agent for document categorization and exploration," in *Proc. of the 2nd International Conference on Autonomous Agents*, 1998, pp. 408–415.

[21] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, 2004.

[22] G. Karypis, "CLUTO - a clustering toolkit," University of Minnesota, Department of Computer Science, Tech. Rep. #02-017, 2003.

[23] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, 2005.