Document Representation and Dimension Reduction for Text Clustering

Mahdi Shafiei, Singer Wang, Roger Zhang, Evangelos Milios, Bin Tang, Jane Tougas, Ray Spiteri
Faculty of Computer Science, Dalhousie University
Halifax, Canada,
http://www.cs.dal.ca/∼shafiei

Abstract

Increasingly large text datasets and the high dimensionality associated with natural language create a great challenge in text mining. In this research, a systematic study is conducted, in which three different document representation methods for text are used, together with three Dimension Reduction Techniques (DRT), in the context of the text clustering problem. Several standard benchmark datasets are used. The three Document representation methods considered are based on the vector space model, and they include word, multi-word term, and character N-gram representations. The dimension reduction methods are independent component analysis (ICA), latent semantic indexing (LSI), and a feature selection technique based on Document Frequency (DF). Results are compared in terms of clustering performance, using the k-means clustering algorithm. Experiments show that ICA and LSI are clearly better than DF on all datasets. For word and N-gram representation, ICA generally gives better results compared with LSI. Experiments also show that the word representation gives better clustering results compared to term and N-gram representation. Finally, for the N-gram representation, it is demonstrated that a profile length (before dimensionality reduction) of 2000 is sufficient to capture the information and, in most cases, a 4-gram representation gives better performance than 3-gram representation.

1 Introduction

Advances in information and communication technologies offer ubiquitous access to vast amounts of information and are causing an exponential increase in the number of documents available online. While more and more textual information is available electronically, effective retrieval and mining is getting more and more difficult without the efficient organization, summarization, and indexing of document content. Among different approaches used to tackle this problem, document clustering is a principal one. In general, given a document collection, the task of text clustering is to group documents together in such a way that the documents within each cluster are similar to each other.

The traditional representation of documents known as *bag-of-words* considers every document as a vector in a very high-dimensional space; each element of this vector corresponds to one word (or, more generally, feature) in the document collection. This representation is based on the *Vector Space Model* [17], where vector components represent certain feature weights. Among clustering algorithms applied to the Vector Space representation of documents, Bisecting K-means and regular K-means have been found to outperform other clustering methods, while they are significantly more efficient computationally, an important consideration with large datasets of high dimensionality [20].

The traditional document representation considers unique words as the components of vectors. Another approach uses N-grams as the vector components. An N-gram is a sequence of symbols extracted from a long string [2]. These symbols can be a byte, character, or word. Extracting character N-grams from a document involves moving a n-character wide window across the document character by character. The character N-gram representation has the advantage of being more robust and less sensitive to grammatical and typographical errors, and it requires no linguistic preparation, making it more language independent than other representations. Another approach for representing text documents uses multi-word terms as vector components, which are noun phrases extracted using a combination of linguistic and statistical criteria. This representation is motivated by the notion that terms should contain more semantic information than individual words. Another advantage of using terms for representing a document is its lower dimensionality compared with the traditional word or N-gram representation.

Using any one of these representations, it is not surprising to find thousands or tens of thousands of different words, N-grams, or terms for even a relatively small sized text data collection of a few thousand documents, of which a very small subset appears in an individual document. This

1-4244-0832-6/07/$20.00 ©2007 IEEE.

results in a very sparse, but also very high-dimensional feature vector for describing a document. Due to high dimensionality of the feature vector, standard similarity measures between vectors lose their discriminative power, ("curse of dimensionality", [1]). To address this problem, various dimension reduction techniques have been proposed [5], generally of two types, *feature transformation* and *feature selection* [16, 24].

2 Dimension Reduction techniques

In mathematical terms, the problem of dimension reduction can be stated as follows: given the p-dimensional random variable $\mathrm{x} = (x_1, \ldots, x_p)^T$, the objective is to find a representation of lower dimensionality k, $\mathrm{s} = (s_1, \ldots, s_k)^T$ with $k < p$, which preserves the information content of the original data, as much as possible, according to some criterion.

Feature selection techniques use a term-usefulness criterion threshold to eliminate some terms from the full vocabulary of the document corpus. In an unsupervised framework, such criteria are document frequency and term frequency variance.

Assuming n documents, each being represented by a p-dimensional random variable $\mathrm{x} = (x_1, \ldots, x_p)^T$, there are two kinds of feature transformation techniques: linear and non-linear. In linear techniques, each of the $k < p$ components of the new transformed variable is a linear combination of the original variables:

$$s_i = w_{i,1}x_1 + \ldots w_{i,p}x_p, \quad \text{for} \quad i = 1, \ldots, k, \quad \text{or}$$

$$\mathrm{s} = \mathbf{W}_{k\times p}\mathrm{x},$$

where $\mathbf{W}_{k\times p}$ is the linear transformation weight matrix, and subscripts denote dimensions. Expressing the same relationship as

$$\mathrm{x} = \mathbf{A}_{p\times k}\mathrm{s},$$

we note that the new variables s are called the hidden or the latent variables. In terms of an $n \times p$ feature-document matrix $\mathbf{X}_{n\times p}$, we have

$$\mathrm{S}_{i,j} = w_{i,1}\mathrm{X}_{1,j} + \ldots w_{i,p}\mathrm{X}_{p,j}$$

for $i = 1, \ldots, k$ and $j = 1, \ldots, n$, and j indicates the jth document. Equivalently,

$$\mathbf{S}_{k\times n} = \mathbf{W}_{k\times p}\mathbf{X}^T_{n\times p}, \quad \text{and} \quad \mathbf{X}^T_{n\times p} = \mathbf{A}_{p\times k}\mathbf{S}_{k\times n}.$$

Document Frequency based Method The Document frequency (DF) of a term is the number of documents in which that term occurs. One can use DF as a criterion for selecting good terms. The basic intuition behind using document frequency as a criterion is that rare terms either do not capture much information about one category, or they do not affect global performance. In spite of its simplicity, it is believed to be as effective as more advanced feature selection methods [24].

Latent Semantic Indexing Latent semantic indexing (LSI) is a method of dimensionality reduction that is based on the singular value decomposition (SVD) of the feature-document matrix representation $\mathbf{X}_{n\times p}$ of a dataset. In the SVD, singular values are ranked in non-increasing order, and all but the first k singular values are set to zero, to obtain a partial SVD that corresponds to the reduced dimensionality.

The number of dimensions k to keep in the reduced feature-document matrix when the original dimensionality p is very large is still an open question, but experiments indicate that values of k of about 100 typically give the best results in an information retrieval context [3]. Typically the performance of LSI improves dramatically as k increases from 1, peaks for some number that is still much less than p, then slowly deteriorates from there.

The concept of the SVD also comes up in the context of Principal Component Analysis (PCA). PCA is a statistical analysis technique aimed at reducing the effective dimensionality of a dataset. The objective is to find a (linear) transformation of the original variables to a set of new uncorrelated variables (the principal components) such that a very high proportion of the variation of the old variables is captured by relatively few of the new ones. Besides offering an alternative viewpoint to some aspects of PCA theory, the SVD provides a robust computational method for determining the quantities involved in PCA. In PCA, one determines a spectral decomposition (i.e., in terms of eigenvalues and eigenvectors) of the $p \times p$ matrix $\mathbf{X}^T_{n\times p}\mathbf{X}_{n\times p}$:

$$\mathbf{X}^T_{n\times p}\mathbf{X}_{n\times p} = \mathbf{W}_{p\times p}\mathbf{\Lambda}_{p\times p}\mathbf{W}^{-1}_{p\times p},$$

where $\mathbf{W}_{p\times p}$ is the matrix of the p eigenvectors, and $\mathbf{\Lambda_{p\times p}}$ is a diagonal matrix with the eigenvalues of $\mathbf{X}^T_{n\times p}\mathbf{X}_{n\times p}$. The Principal Components z of a document vector x are given by the linear transformation $z = \mathbf{W}_{p\times p}x$. It can be shown that if we have the SVD of $\mathbf{X}^T_{n\times p} = \mathbf{V}_{p\times p}\Sigma_{p\times n}\mathbf{U}^T_{n\times n}$ then

$$\mathbf{W}_{p\times p} = \mathbf{V}_{p\times p} \quad \text{and} \quad \mathbf{\Lambda} = \mathbf{\Sigma}\mathbf{\Sigma}^T$$

The issue of whether clustering in a vector space obtained from the first few PCA components, corresponding to the largest eigenvalues, is significantly better than the full set of dimensions was studied in [25]. This study found that, for clustering gene expression profiles, there was no clear advantage to using the first few PCA components, and sometimes the results could be worse. Clustering performance depends on the clustering algorithm and similarity

metric used. On the other hand, Latent Semantic Indexing, which is related to PCA, has been demonstrated to give significant improvements in text clustering [13]. So it is possible that PCA-like dimensionality reduction behaves differently in the gene expression profile data than document data. It should be noted that the full dimensionality of the gene expression data used in [25] is in the low tens, while that of the document data is typically in the thousands.

Independent Component Analysis In comparison to PCA, Independent Component Analysis (ICA) is a higher-order method that seeks linear projections, not necessarily orthogonal to each other, that are as nearly statistically independent as possible. Statistical independence is a much stronger condition than uncorrelatedness. Whereas the latter only involves the second-order statistics, the former depends on all the higher-order statistics. Independence always implies uncorrelatedness, but the converse is not true.

With the classical assumption of Gaussianity, one can use a second-order technique like PCA because distribution of a normally distributed variable x can be completely described by second-order information [9], and there is no need to include any other information, for example higher moments. Because only classical matrix manipulations are used, this makes second-order methods very robust and computationally simple.

ICA is a computational technique for revealing hidden factors that underlie sets of measurements or signals. ICA assumes a statistical model whereby the observed multivariate data, typically given as a large set of samples, are linear or nonlinear mixtures of some unknown latent variables, with unknown mixing coefficients. The latent variables are nongaussian and mutually independent, and they are called the *independent components* of the observed data. Thus ICA can be seen as a generalization of PCA. In fact, for Gaussian distributions, the principal components are independent components. ICA is a much richer technique, however, capable of finding solutions to problems where classical methods fail.

Let $\mathbf{X}_{p\times n}$ be the matrix of the observed mixture signals, where p is the number of features in the document collection, and n is the number of documents. The noise-free mixing model takes the form,

$$\mathbf{X}_{p\times n} = \mathbf{A}_{p\times m}\mathbf{S}_{m\times n}$$

where $\mathbf{S}$ is the source signal matrix of size $m \times n$, where m is the number of sources, and $\mathbf{A}$ is the $p \times m$ mixing matrix.

The first step in the computation of ICA is to whiten the data, which can be accomplished by PCA. The second step is the solution of an optimization problem to maximize the non-Gaussianity of the independent components [8].

In contrast with PCA, the objective of ICA is not necessarily dimension reduction, but rather only *identification* of independent components. For dimensionality reduction, it is assumed that it is possible to find $m \ll p$ components that effectively capture the variability of the original data.

One problem of using ICA as a dimensionality reduction method is that there is no natural order or ranking of the Independent Components. One solution to this is ordering them according to the norms of the columns of the mixing matrix (similar to the ordering in PCA) once they are estimated.

Although ICA was originally developed for digital signal processing applications, it has recently been suggested that it may be a powerful tool for analyzing text document data as well, provided that the documents are presented in a suitable numerical form. ICA has been used for dimensionality reduction and representation of word histograms [12].

3 Text Representation Methods

In this research, we are interested in comparing three different feature types, words, multi-word terms and character N-grams. As feature weights, we use the term-frequency, inverse-document frequency (TFIDF) scheme, which combines the term frequency and document frequency.

Word Representation A typical choice of a feature type for representing a text document is the "set of words" or the "bag of words". Following established practice, stopwords are removed, the remaining words are stemmed, and word stems appearing in less than five documents are removed.

Term Representation Multi-word terms, sometimes called phrases, can also be used as features in document vectors. Term representation has the potential to reduce significantly the dimensionality, compared to the word representation, while retaining the meaning [15]. However, experimental results on this issue have been mixed [18], [22].

N-gram Representation Traditional N-gram work has focused on word bi-grams, that is pairs of words, but there is a recent trend toward using character N-grams and byte N-grams. Character N-gram is a language independent text representation technique. It transforms documents into high-dimensional feature vectors where each feature corresponds to a contiguous substring. N-grams are N adjacent characters (substring) from the alphabet A [2]. Hence, the number of distinct N-grams in a text can be, in principle, as high as $|A|^N$. This shows that the dimensionality of the N-grams feature vector can be very high even for moderate values of N. However, in practice, only a small fraction of all possible N-grams are present in a given document collection, thus reducing the dimensionality substantially. For example, there are 8727 unique trigrams in the

Dataset	Dataset size	classes	class size range
Classic3	3891	3	1033 - 1460
NG	3500	7	500
RD-256	6519	10	105 - 2778
RD-512	3948	10	89 - 1449
URCS	528	4	83 - 193

Table 1. Summary of data sets used in experiments.

Reuters dataset, less than half of the $27^3 = 19683$ possible N-grams. During N-gram feature vector formation, all the upper case characters are converted into lower case, and space is substituted for punctuation. The feature vectors are then normalized.

In extracting character N-grams from a document, any non-letter character is replaced by a space and two or more consecutive spaces are treated as a single one. The byte N-grams are N-grams retrieved from the sequence of the raw bytes as they appear in data files, without any preprocessing.

In comparison with stemming and stop word removal, the N-gram representation has the advantage of being more robust and less sensitive to grammatical and typographical errors, and it requires no linguistic preprocessing; therefore it is language independent.

4 Experimental Results

We experimentally evaluated the performance of different document representations under the dimension reduction methods described previously on a number of standard datasets. In this section we describe the datasets and the pre-processing procedures, our experimental methodology, followed by experimental results. We present selected results from one dataset. For more details on the experiments and results on other datasets, the reader is referred to [19].

4.1 Datasets and Data Preparation

We use four datasets for our experiments, including both unstructured newsgroup items and relatively more structured abstracts from scientific research papers. These datasets have been widely used in the research of information retrieval and text mining. The number of classes ranges from 3 to 10 and the number of documents ranges between 83 and 2778 per class. Table 1 summarizes the characteristics of the datasets.

N-gram Processing In both character and byte N-grams, multispaces (new lines, tabs, and space) are converted into a single space. In addition, for character N-grams, any non-alphabetical character is also converted into a space. Compared to word representation, the standard processing procedures like stop word removal, removal of low frequency terms, and stemming are not applied [11, 14].

The two most important decisions in working with N-grams are the choice of the value N and the profile length. The profile length is the number of N-grams used in the feature vector. Typically the most frequent N-grams are chosen. Theoretically the maximum profile length, or the dimensionality of the feature vector, for character N-grams is 27^N. However, the actual number will be significantly less as character sequences like 'QQQ' are not likely to appear in normal documents. Obviously choosing a large N will lead to high dimensionality. Prior work has shown that choosing N to be 3 or 4 tends to give optimal results, with little difference between the two [14]. Various profile lengths have been used by other researchers, with common lengths having been 1000, 2000, 3000, 4000, and 5000 [14, 11]. For the N-gram experiments, we used the N-gram software tool [10]. The preprocessing steps for N-grams are: replacement of non-alphabetical characters with a space character, multispaces (tabs, spaces, or newlines) with a single space, and conversion of all upper case characters to lower case. The N-grams extracted from each document are ranked according to the number of documents each N-gram appears in, and the top k N-grams are selected to form the *N-gram profile*. A vector is formed for each document using TFIDF, and normalized to unit length.

In this work, both 3-grams and 4-grams with profile lengths of 500, 1000, 2000, 3000, 4000, and 5000 were used. The limited size of the URCS dataset resulted in the total number of 3-grams being less than 5000; thus it was not possible to produce a 3-gram profile of 5000.

4.2 Experimental Design and Metrics

Several ways of measuring the quality of clustering, especially text clustering, have been proposed. For our datasets, we have class labels for each data item, and therefore we can use a group of measures which considers the degree of agreement or overlap between the classes and the computed clusters. Accordingly, we have selected one of the most highly used quality measures in text clustering: *purity*.

Purity measures the extent to which each cluster contains documents from primarily one class [26]. The overall purity of a clustering solution is defined as the weighted sum of individual cluster purities:

$$Purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r) \quad (1)$$

where $P(S_r)$ is the purity for a particular cluster of size n_r, k is the number of clusters and d is the total number of data items in the dataset. Purity of a single cluster is defined

by $P(S_r) = \frac{n_d}{n_r}$, where n_d is the number of documents in cluster r that belong to the dominant (majority) class in r; i.e., the class with the most documents in r. Obviously, the higher the purity value, the purer the cluster in terms of the class labels of its members, and the better the clustering results.

It is not common to have two separate sets of training and test documents in text clustering, as most researchers prefer to report clustering results on the training set. However, as is common in classification problems, in order to have results closer to the actual performance of the clustering algorithm, we divide the whole text collection into a training and a test part [7]. We then perform the usual clustering on the training part, and, for each cluster, we use its mean as its representative. For the testing part, we use the nearest neighbor classification algorithm to assign a test document to its cluster.

All our experiments were conducted in the Matlab R14 environment. The `svds` procedure is a built-in Matlab function. We also used the FastICA toolbox[1] for performing ICA. For the k-means algorithm, we used the GMeans Toolbox[2]. The computed number of clusters k is five times the number of known classes in the dataset used, the aim being to obtain clusters with high purity, even if a single class is split among several clusters. To reduce the effect of random initialization of the k-means algorithm, results are the averages of 15 different runs of each experiment. Normalization of the projection matrix computed in LSI or ICA has been demonstrated not to be necessary, because no significant difference was found between using the normalized or non-normalized projection matrix [21]. Our experiments confirm these findings.

In order to detect the "good" range of reduced dimensions and correlate it with the corresponding singular values in LSI or eigenvalues of the whitening step in ICA, it is instructive to plot these values against dimensionality on the same horizontal scale as clustering performance [21]. For a given dimension k in the singular value graph, the k-th singular value in the sorted list of singular values in decreasing order is plotted.

4.3 Word Representation Experiments

For all datasets, we observe that the singular values, ranked by non-increasing order, decrease very quickly for the first few tens of dimensions and after that, there is a smooth and somewhat flat drop. The part of the singular value curve, where a rapid drop of singular values happens, has been referred to as the *transition zone* [21]. The transition zone seems to correspond to the dimensionality where we can get best performance out of ICA or LSI (Fig. 1). The x-axis represents dimensionality, and the y-axis represents purity value, or singular values / eigenvalues sorted in non-increasing order.

The results show that for all datasets, clustering quality using ICA is better than using LSI in the whole range of dimensionalities investigated. For low dimensionalities, especially lower than 50, for all datasets, the DF based method has the worst performance among the dimension reduction methods used. The best performance of DF is comparable to the best performance of LSI and ICA, but at much higher dimensionalities.

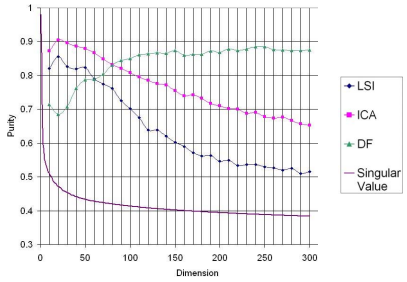

Figure 1. Comparing dimension reduction techniques, ICA, LSI, and DF on the word representation of a typical dataset (RD-256).

4.4 Term Representation Experiments

In the case of term representation, unlike word representation, there is no statistically significant difference between LSI and ICA performance. Performance results of LSI and ICA methods on all datasets are shown in Fig. 2. It is interesting to see that, unlike the word representation, the best results for LSI and ICA do not seem to coincide exactly with the transition zone of singular value curves, but the transition zone still can give some hints about the starting point of searching for the best dimensionality.

For the DF based method, the overall performance pattern is like that of word representation. The clustering quality starts to increase as the dimensionality increases until some middle-range values for all datasets. After this mid-range value, the clustering quality settles down around some maximum clustering performance. In word representation, this maximum performance, which is achieved at higher dimensions, was very close to the maximum performance of the other two methods; however, for the term representation, this is not the case.

The trends of performance curves of LSI and ICA methods for all datasets are similar to the word representation

[1] http://www.cis.hut.fi/projects/ica/fastica/
[2] http://www.cs.utexas.edu/users/yguan/datamining/gmeans.html

experiment. In this case, as with the word representation, performance of these two dimension reduction methods reaches its maximum at some very low dimension greater than 20, and then starts to degrade.

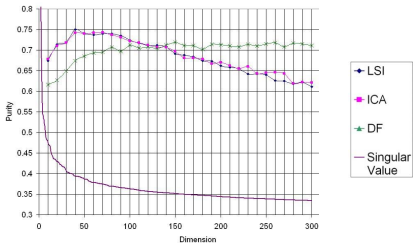

Figure 2. Comparing dimension reduction techniques, ICA, LSI and DF, on the term representation of a typical dataset (RD-256).

4.5 N-gram Representation Experiments

For each dataset, we generate its 3-gram and 4-gram representations with different profile lengths ranging from 500 to 5000. In this section, the objective of experiments is to determine the effect of N-gram profile length on the clustering quality. We are also interested in identifying which of the two N-gram representations achieves better performance when the three dimension reduction methods are applied. Then for each dataset, we select the N-gram length and the corresponding profile length which achieved the best clustering quality.

N-gram Profile Length and Clustering Quality In the first set of experiments with N-grams, we are interested in investigating the impact of the N-gram profile length on clustering quality. In these experiments, we apply the ICA, LSI, and DF dimension reduction methods on 3-grams and 4-grams with different profile lengths ranging from 500 to 5000. In each case, we select the shortest profile length which gives results close to the best over different profile lengths as the "best-case" profile length for the corresponding dataset and dimension reduction method.

The clustering performance for 3-grams when ICA has been used as dimension reduction method increases as the profile length increases. But for all datasets, it seems that a profile length equal to 2000 is the best-case length (Fig. 3). In the Figure, x-axis represents dimension and y-axis represents purity value.

The clustering performance with LSI as the dimension reduction method for 3-gram representation does not improve with increasing profile length (Fig. 4). For example, in datasets RD256 and RD512, further increasing the profile length beyond 2000 makes clustering quality worse. The optimum profile length is different for each dataset, but overall we do not need to go further than profile length of 4000 to get the best clustering result. Profile length equal to 2000 is still one of the best profile lengths for 3-gram when we use LSI as the dimension reduction method.

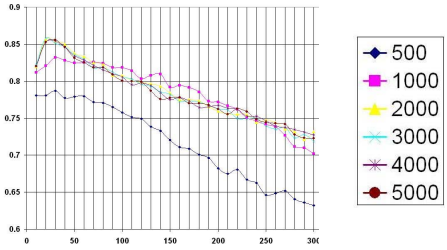

Figure 3. Purity as a function of dimension, parameterized by profile length for 3-gram representation when ICA dimension reduction is used on a typical dataset (RD-256).

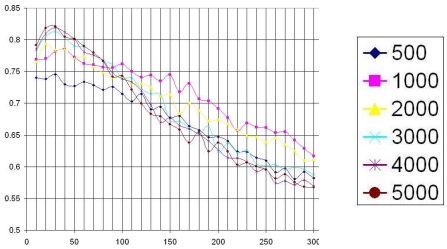

Figure 4. Purity as a function of dimension, parameterized by profile length for 3-gram representation when LSI dimension reduction is used on a typical dataset (RD-256).

Increasing the profile length does not have much impact on clustering quality when DF is used as dimension reduction method on 3-grams (Fig. 5). It is interesting that for all profile lengths investigated, the clustering quality change pattern remains almost the same. We can still see some small improvements in clustering quality with increasing profile length, but due to computational expense, it does not seem reasonable to go further than 500 for getting better clustering quality using this dimension reduction method.

The next three experiments show the impact of profile length for 4-gram representation when each of the three dimension reduction methods is used.

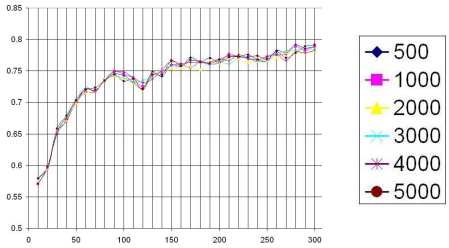

Figure 5. Purity as a function of dimension, parameterized by profile length for 3-gram representation when DF based dimension reduction is used on a typical dataset (RD-256).

When the 4-gram representation is used in the ICA method, for all datasets, clustering performance increases as the profile length increases, as we have seen for 3-grams. In this case again, profile length equal to 2000 is the shortest length required to get the best performance out of 4-grams, and after this value, increasing the profile length does not increase the clustering performance by much.

In the next experiment, which uses LSI as dimension reduction method, despite what we saw in the similar experiment for 3-gram representation, with increasing the profile length, we do get better clustering results. As table 2 shows, in most cases profile lengths equal to 4000 and 5000 are the best for 4-gram representation when LSI is used as dimension reduction method.

For the DF dimension reduction method, as with the 3-gram representation, increasing the profile length in the 4-gram representation does not have much impact on clustering quality. Similar to the case of 3-gram representation, the clustering quality change pattern remains almost the same for all profile lengths. Increasing profile length makes some small improvements in clustering quality, but due to computational expense, it does not seem reasonable to go further than 500 for getting better clustering quality using this dimension reduction method.

Based on the experiments in this section, increasing the profile length does not change the clustering quality considerably when DF is used for dimension reduction. Due to computational costs incurred from having longer profile length, it seems that for this dimension reduction method, profile length equal to 500 is good enough to get the optimum clustering quality. For ICA and for profile length above 2000, we do not get enough increase in clustering quality to justify a longer profile length for all datasets. However, for LSI, increasing profile length seems to have positive impact on clustering quality for most datasets. This fact is much clearer for the 4-gram representation; as we have seen in this case, the optimum profile length tends to be greater than 4000 for almost all datasets.

Dataset	Classic3	NG	RD256	RD512	URCS
Representation	3-gram				
ICA	2000	2000	2000	2000	2000
LSI	2000	4000	2000	2000	3000
DF	2000	2000	2000	2000	2000
Representation	4-gram				
ICA	2000	2000	2000	2000	2000
LSI	5000	5000	3000	4000	4000
DF	2000	2000	2000	2000	2000

Table 2. Optimal profile lengths for 3-grams and 4-grams when each of the listed dimension reduction methods is applied.

	3-Gram					
Dataset	500	1000	2000	3000	4000	5000
Classic3	0.53	0.67	0.79	0.86	0.89	0.91
NG	0.35	0.51	0.68	0.77	0.82	0.86
RD256	0.61	0.72	0.83	0.88	0.91	0.93
RD512	0.49	0.64	0.78	0.84	0.88	0.90
URCS	0.45	0.61	0.76	0.83	0.87	NA
	4-Gram					
Classic3	0.65	0.74	0.82	0.86	0.88	0.90
NG	0.51	0.63	0.73	0.79	0.82	0.84
RD256	0.72	0.79	0.86	0.89	0.91	0.92
RD512	0.64	0.73	0.81	0.85	0.87	0.89
URCS	0.60	0.69	0.78	0.83	0.86	0.88

Table 3. Each value shows the fraction of zero elements in the corresponding matrix

N and Clustering Quality In order to investigate the effect of N-gram length (N) on clustering quality, we choose the best profile length for 3-grams and 4-grams based on the results shown in the previous section. Table 2 shows these best profile length for 3-grams and 4-grams when each of the three dimension reduction methods, LSI, ICA, or DF, is applied.

Dimensionality and sparsity increase with N, as can be seen in Table 3. Therefore, we intuitively expect that by increasing N, we need a longer profile in order to capture the same amount of information. We observe this in table 2 for the LSI method: for the 4-gram representation, we need to have longer profile to get the best clustering quality compared to 3-gram representation. However, this intuition does not hold for ICA in most cases.

The comparison between the best performance achieved with 3-grams and 4-grams with different profile lengths, when ICA and DF is used as dimension reduction method, shows that 3-grams and 4-grams perform close, and it is dataset-dependent which one performs better.

The comparison between the best performance achieved with 3-grams and 4-grams over all profile lengths considered when LSI is used as dimension reduction method shows that 4-grams achieve better clustering performance

compared to 3-grams.

Best N-gram Parameters Overall, we are interested in selecting a good profile length and reduced dimensionality for 3-gram and 4-gram representations. The general result is that for each of the 3-gram and 4-gram representation, and for a mid-range profile length (around 2000), ICA is the method of choice. But choosing between the 3-gram and 4-gram representations seems to depend on the data set. Typical results are shown in Fig. 6.

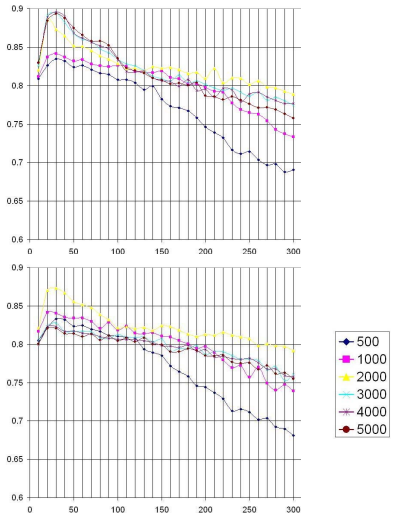

Figure 6. Purity as a function of dimension, parameterized by profile length for 4-gram represen- tation when LSI / ICA dimension reduction is used (top/bottom) on a typical dataset (RD-256).

4.6 Comparing Dimension Reduction Techniques

As it is clear in most experiments, the performance of DF based clustering reaches its maximum at some middle range of dimensions (much higher than the best dimensions of ICA/LSI), and then the performance remains stable as the number of dimensions increases. It is also interesting that the best performance of this method at these middle range dimensions is often equal to the best performance of ICA/LSI, which is achieved at much lower dimensions. This suggests that it might be possible to use the DF based method as a preprocessing step to pre-select a subset of dimensions to be used for ICA/LSI instead of using the full set of dimensions for these methods. This can help especially when the original dimensionality is too high, making it too expensive to compute ICA or LSI. In the case of ICA, sometimes this may be necessary, because the input matrix for ICA, unlike LSI, is dense and therefore performing SVD on it is extremely expensive.

In summary, for all cases except 4-grams, the performance of ICA is clearly better than LSI. In the case of 4-grams, ICA is at least as good as LSI for equal profile lengths, but LSI is better than ICA for the best profile length of each method.

4.7 Comparing Text Representation Methods

In order to decide on the best text representation method, we compare their best clustering performance over every dimension reduction method considered. For each text representation method, we first select the dimension reduction method, with which it has achieved its best performance. For example, ICA is the best dimension reduction method for word representation, and LSI is the best for the 4-gram representation. Actually, except for the 4-gram representation, for all other representations ICA had the best performance. We then compare the clustering performance of each representation applying its best dimension reduction method. The results of this comparison are summarized next.

The term representation has the worst clustering quality amongst the four different representations. Both 3-gram and 4-gram representations achieve impressive results, but it is worth noticing that these represent best-case results, achieved after a careful investigation of different parameter values (including N-gram length and its corresponding profile length). Without these optimized parameters, N-gram representation is likely to have lower clustering performance. Our experiments indicate certain appropriate values for these parameters. For the 3-gram representation, profile length equal to 2000 is sufficient to obtain close to the highest clustering quality. For the 4-gram representation, this value is around 3000. We also observed that, in most cases, the 4-gram representation achieves better clustering quality than the 3-gram representation using these suggested values for profile length.

4.8 Computational Considerations

The run time of the experiments consists of two components, dimensionality reduction and K-means clustering. The run time of dimensionality reduction in the case of ICA and LSI increases slightly faster than linear with the chosen

	df		ICA		LSI	
	Terms	Words	Terms	Words	Terms	Words
Classic3	0.17	2.18	96	316	22	72
NG	0.36	7.25	171	282	29	106
RD256	0.69	4.20	311	442	46	113
RD512	0.40	2.70	168	277	29	73
URCS	0.01	0.11	5	35	2	12

Table 4. Run time of dimensionality reduction in seconds for dimensionality of 100. Run time for LSI and ICA is slightly faster than linear function of dimensionality for the range tested (up to 300). Run time for DF is constant.

Data set	Run time (s)
Classic3	5.50
NG	8.70
RD512	28.70
URCS	0.28

Table 5. Run time in seconds for clustering for dimensionality of 100.

dimensionality for the range of dimensionalities tested (up to 300). The run time of DF is practically a constant function of dimensionality, as the main computation is sorting the features according to their DF value. For comparing the run time of DF, ICA and LSI, the run time in seconds for dimensionality of 100 is shown in Table 4, on a Linux Pentium 4, 1.6GHz, with 1GB of RAM. The differences among the data sets are explained by the different initial dimensionalities, as shown in Section 4.1.

The run time of clustering for dimensionality equal to 100 for the word representation in seconds is shown in table 5. Run time increases linearly with dimensionality for the range tested (up to 300).

5 Discussion

In this research, we have studied three well-known dimension reduction techniques, DF, LSI, and ICA, for the document clustering task. We applied these methods to five benchmark datasets in order to compare their relative performance. We have also compared three different representation methods based on the Vector Space Model, and we applied the dimension reduction methods to find the best combination of representation method and dimension reduction algorithm for text clustering.

From the experimental results, several general conclusions can be drawn. In general, we can rank the three dimension reduction techniques in the order of ICA, LSI, DF, with ICA being the best. ICA demonstrates good performance and superior stability compared to LSI in almost all configurations. Both ICA and LSI can effectively reduce the dimensionality from a few thousands to a range between 10 and 100. The best performance of ICA/LSI seems to correspond well with the transition zone of the singular value curve. In the case of N-gram representation, ICA performs well with mid-range profile lengths, whereas LSI performs better on longer profile lengths. The DF based technique can get close to optimal performance of two other methods, but at much higher dimensions. At lower dimensions, its performance is much worse than the other two methods.

In terms of computational requirements, we see that LSI and ICA require one to two orders of magnitude more computation compared to DF, and that ICA requires 2-5 times more computation compared to LSI. Clustering requires the same order of magnitude of computation as the DF dimensionality reduction. This means that dimensionality reduction is the dominant computation in clustering using reduced dimension by LSI and ICA. This is true with the k-means clustering algorithm for the low dimensionalities tested. Investing in dimensionality reduction of a document collection is worthwhile when multiple clusterings of the same collection are required, for example hierarchical clustering. Another case is interactive clustering for visualization, where the user can change clustering parameters and view the resulting clusters interactively. In interactive clustering, it is important for the clustering computation to be fast, while preserving quality, and therefore using the lowest possible dimensionality is desirable.

Among the three representation methods, traditional word representation seems to achieve better results in most cases, especially for lower dimensions. The N-gram representation can be considered to be a replacement for word representation because its performance result is close to word representation. However, it needs careful and precise determination of its two input parameters, which are the N-gram length and the profile length. If these parameters are selected carefully, then the performance of the N-gram representation performance can be very close to word representation, and, for higher dimensions, even better. Term representation performance is significantly worse than the two other representations. Even if we use default parameters for N-gram representation, its worst performance is still better than best performance of term representation.

For clustering unlabelled datasets, the recommended process is to use words as features and ICA for dimension reduction. Terms are an inferior representation. N-grams can provide equal or better performance than words, if carefully tuned. N-grams need no language-specific preprocessing (e.g., stemming, stop word removal), and they are thought to be better than words in handling noisy text, for example text obtained through optical character recog-

nition of printed documents or the output of a speech recognizer.

In ongoing research we are investigating the use of other clustering methods, such as information theoretic co-clustering [4], and other clustering evaluation metrics [6]. Further investigation into the interaction between clustering algorithms and evaluation metrics is required [23].

Acknowledgements. We would like to acknowledge support for this project from the Natural Sciences and Engineering Research Council of Canada, the MITACS NCE, GINIus, and IT Interactive Services Inc.

References

[1] K. Beyer, J. Goldstein, R. Ramakrishnan, , and U. Shaft. When is the nearest neighbour meaningful? In *Proceedings of the 7th International Conference on Database Theory*, pages 217–235, 1999.

[2] W. B. Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. In *TREC*, pages 269–278, 1994.

[3] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[4] I. Dhillon, S. Mallela, and D. Modha. Information theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, pages 89–98, Washington, DC, August 2003.

[5] I. K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, June 2002.

[6] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems,*, 17(2/3):107–145, 2001.

[7] P. Husbands, H. Simon, and C. H. Q. Ding. On the use of the singular value decomposition for text retrieval. In *Computational information retrieval*, pages 145–156. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[8] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[9] A. Jung. An introduction to a new data analysis tool: Independent component analysis. In *Proceedings of Workshop GK "Nonlinearity"*, Regensburg, October 2001.

[10] V. Keselj. Perl package text::ngrams, last accessed on Jan. 11, 2007. http://users.cs.dal.ca/∼vlado/srcperl/Ngrams/.

[11] V. Keselj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics*, August 2003.

[12] T. Kolenda, L. Hansen, and S. Sigurdsson. Independent components in text. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 229–250. Springer-Verlag, 2000.

[13] K. Lerman. Document clustering in reduced dimension vector space. http://www.isi.edu/∼lerman/papers/papers.html, 1999, last accessed Jan. 11, 2007.

[14] Y. Mao, V. Keselj, and E. E. Milios. Comparing document clusteing using n-grams, words, and terms. Master's thesis, Dalhousie University, 2004.

[15] E. Milios, Y. Zhang, and N. Zincir-Heywood. Term-based clustering and summarization of web page collections. In *the Seventeenth Conference of the Canadian Society for Computational Studies of Intelligence (AI04)*, pages 60–74, London, ON, May 2004.

[16] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2004.

[17] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[18] H. Schutze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Research and Development in Information Retrieval*, pages 229–237, 1995.

[19] M. Shafiei, S. Wang, R. Zhang, E. Milios, B. Tang, J. Tougas, and R. Spiteri. A systematic study of document representation and dimension reduction for text clustering. Technical Report Technical Report CS-2006-05, Faculty of Computer Science, Dalhousie University, Halifax, Canada, July 2006. http://www.cs.dal.ca/research/techreports/2006/.

[20] M. Steinbach, G. Karypis, and V. Kumar. A comparison of common document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[21] B. Tang, X. Luo, M. I. Heywood, and M. Shepherd. Comparative study of dimension reduction techniques for document clustering. Technical Report CS-2004-14, Faculty of Computer Science, Dalhousie University, December 2004.

[22] K. Tzeras and S. Hartmann. Automatic indexing based on bayesian inference networks. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 22–35, New York, NY, USA, 1993. ACM Press.

[23] H. Xiong, J. Wu, and J. Chen. Kmeans clustering versus validation measures a data distribution perspective. In *KDD*, Philadelphia, PA, USA, Aug. 20-23 2006. ACM.

[24] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[25] K. Yeung and W. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, September 2001.

[26] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001.