A Systematic Study of Document Representation and Dimension Reduction for Text Clustering

Mahdi Shafiei
Singer Wang
Roger Zhang
Evangelos Milios
Bin Tang
Jane Tougas
Ray Spiteri

Technical Report CS-2006-05

July 11, 2006

Faculty of Computer Science
6050 University Ave., Halifax, Nova Scotia, B3H 1W5, Canada

A Systematic Study of Document Representation and Dimension Reduction for Text Clustering

M. Mahdi Shafiei, Singer Wang, Roger Zhang, Evangelos E. Milios,
Bin Tang, Jane Tougas, Ray Spiteri
Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 1W5, Canada
http://www.cs.dal.ca/∼eem

July 11, 2006

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 Dimension Reduction Techniques (DRT) are applied on three different document representation methods in the context of the text clustering problem. Several standard benchmark datasets are used. The dimension reduction methods considered include independent component analysis (ICA), latent semantic indexing (LSI), and a technique based on Document Frequency (DF). These three methods are applied on three Document representation methods based on the vector space model; word, multi-word term, and character N-gram representations. 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 shown that a profile length of 2000 is enough 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 topic of clustering has been extensively studied in many scientific disciplines and over the last years a variety of different algorithms have been developed. For a comprehensive summary of the various applications and algorithms, one can refer to two recent surveys on the topic [13, 4].

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* [23], where vector components represent certain feature weights. A detailed comparison of various clustering algorithms applied to the Vector Space representation of documents is provided in [27]. Bisecting K-means and regular K-means are demonstrated to outperform other clustering methods, while they are significantly more efficient computationally, an important consideration with large datasets of high dimensionality.
Representations that are not based on the Vector Space Model have been proposed. The information bottleneck method focuses on the joint distribution of word clusters and documents, and first derives word clusters that maximally preserve the mutual information about the documents, and then document clusters, that preserve the information about word clusters [26]. Information-theoretic co-clustering was inspired by the information bottleneck method [9]. Frequent term-based text clustering uses frequent item (term) sets for clustering, derived using established algorithms for association rule mining [3]. Frequent term sets are sets of terms that co-occur in a significant percentage of documents in a collection.

Based on the Vector Space Model, some other representations have been proposed. The traditional representation, as already mentioned, considers the components of vectors as unique words. Another approach uses N-grams as the vector components. An N-gram is a sequence of symbols extracted from a long string [7]. These symbols can be a byte, character, or word. Extracting character N-grams from a document is like 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. These terms are usually extracted using automatic term extraction algorithms. This representation is based on the idea 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. Moreover, a very small subset of all terms that appear in a text collection as a whole will appear in each individual document. This results in a very sparse, but also very high-dimensional feature vector for describing a document.

Most learning algorithms use some kind of similarity measure to discriminate between two given training vectors. In the case of document clustering, due to high dimensionality of the feature vector, these similarity measures lose their discriminative power. In a sparse high-dimensional space, feature vectors have almost equal distance to each other [5] that makes traditional similarity measures meaningless. This is known as the "curse of dimensionality".

Many researchers from different areas have tried to solve the high-dimensionality problem, and they have proposed various dimension reduction techniques [10]. In a general view of dimension reduction, one can think of two types of dimension reduction methods, known as *feature transformation* and *feature selection* [22].

Feature transformation techniques try to reduce dimensionality to a smaller number of new dimensions, which are linear or non-linear combinations of the vector coordinates in the original dimensions. In other words, the original high-dimensional space is projected to a lower-dimensional space. These methods are believed to be very successful in uncovering the latent structure in datasets.

Various feature transformation techniques have been proposed which include principal component analysis, latent semantic analysis, independent component analysis, projection pursuit, and factor analysis. The reader can refer to [10] for more details. In feature selection methods, the objective is not constructing new features but rather removing features which seem irrelevant for modeling. This problem is a combinatorial optimization problem [6].

The focus of this research is to evaluate the relative effectiveness of dimension reduction techniques for the document clustering problem with various document representation methods. This paper is organized as follows. Section 2 describes the three dimension reduction techniques used in this research. Section 3 describes three different text representation methods used in the experiments. Section 4 describes the general experimental procedure and evaluation methods, the characteristics of the datasets used and the pre-processing procedure followed. It also presents and discusses our experimental results and appropriate discussion notes. Finally, conclusions are drawn and future research directions are identified in Section 5.

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 dimensions, $\mathrm{s} = (s_1, \ldots, s_k)^T$ with $k < p$, which preserves information content of the original data, as much as possible, according to some criterion.

Feature selection techniques remove non-informative terms according to corpus statistics and 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.

If we assume that we have n data items, 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:

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

$$\mathrm{s} = \mathbf{W}\mathrm{x},$$

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

$$\mathrm{x} = \mathbf{A}\mathrm{s},$$

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

$$\mathrm{S}_{i,j} = w_{i,1}\mathrm{X}_{1,j} + \ldots w_{i,p}\mathrm{X}_{p,j}, \quad \text{for} \quad i = 1, \ldots, k \quad \text{and} \quad j = 1, \ldots, n$$

where j indicates the jth realization, or, equivalently,

$$\mathbf{S}_{k\times n} = \mathbf{W}_{k\times p}\mathbf{X}_{p\times n},$$

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

Various dimension reduction techniques have been proposed for text data including both feature selection methods and feature transformation methods [30].

In the following sections, we review a widely used feature selection method for text, as well as two feature transformation techniques used for text dimension reduction.

2.1 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 the global performance [30]. This method is often combined with removal of some very high-frequency terms known as "stop words" and word stemming.

When using DF as a criterion for feature selection, only those dimensions with high values will appear in the feature vector. In spite of its simplicity, it is believed to be as effective as more advanced feature selection methods [30]. We use this technique on different document representation methods based on the vector space model and evaluate how effective it is in comparison with feature transformation methods for text clustering.

DF can be formally defined as follows. For a document collection in matrix notation, $\mathbf{A}_{t\times d}$, with t terms and d documents, the DF value of a term is defined as the number of documents in which the term occurs at least once among the d documents. To reduce the dimensionality of A from t to k ($k < t$), we choose to use the k dimensions with the top k DF values. It is obvious that the DF takes $O(td)$ to evaluate.

2.2 Latent Semantic Indexing

Latent semantic indexing (LSI) is a method of dimensionality reduction, originally proposed for information retrieval, that is based on the partial singular value decomposition (SVD) of the feature-document matrix representation of a dataset.

The SVD is a matrix factorization that can be used to elicit the salient features of a matrix by determining important vectors (i.e., directions) and quantifying their importance via weighting factors (called *singular values*). Given a matrix $\mathbf{A} \in \Re^{t\times d}$, its full SVD is written as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \Re^{t\times t}$, $\mathbf{V} \in \Re^{d\times d}$, and $\mathbf{\Sigma} \in \Re^{t\times d}$. $\mathbf{U}$ and $\mathbf{V}$ are *orthogonal matrices* containing the left and right *singular vectors* of $\mathbf{A}$ respectively. It is common to used a reduced SVD with $\mathbf{U} \in \Re^{t\times d}$, $\mathbf{V} \in \Re^{d\times d}$, and $\mathbf{\Sigma} \in \Re^{d\times d}$, where now $\mathbf{U}$ simply has orthonormal columns. When $\mathbf{A}$ is a feature-document matrix, $\mathbf{U}$ represents the feature vectors and $\mathbf{V}$ represents the document vectors. The matrix $\mathbf{\Sigma}$ can have non-zero entries only on the diagonal, and these entries must be non-negative. These diagonal entries, called the singular values of $\mathbf{A}$, denoted σ_j for $j = 1, 2, ..., d$, are arranged in non-increasing order. The relative size of the entries σ_j indicates the relative importance of the corresponding feature/document vectors. The number of non-zero singular values of a matrix is known as its *rank*, r.

The SVD can be used to interpret a matrix as equal to a weighted sum of r rank-1 matrices; i.e.,

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are the jth columns of matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. This interpretation of the SVD facilitates our understanding of the formation of lower-rank approximations of $\mathbf{A}$. Replacing r in this sum by any k with $0 \leq k < r$ gives the *partial* SVD of $\mathbf{A}$,

$$\mathbf{A}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \approx \mathbf{A}.$$

In matrix form, this is equivalent to taking $\mathbf{U}_k$ and $\mathbf{V}_k$ to be the first k columns of $\mathbf{U}$ and $\mathbf{V}$, and $\mathbf{\Sigma}_k$ to be the leading $k \times k$ submatrix of $\mathbf{\Sigma}$, yielding

$$\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T.$$

It can be shown that $\mathbf{A}_k$ is optimal in the following sense: it is the closest matrix of rank k to the original matrix $\mathbf{A}$ in terms of the 2-norm or the Frobenius norm. The 2-norm of a matrix is the norm induced by the vector 2-norm (or Euclidean norm); the Frobenius norm of a matrix is the square root of the sum of the squares of its elements.

This approximation is used to isolate the important directions of the feature-document matrix while extracting the underlying structure of the data. In LSI, typically $k \ll r$. The effect is a reduction in the noise caused by synonymy and an enhancement of the latent patterns that indicate semantically similar terms. This means that $\mathbf{A}_k$ may be a better representation of the data than the original feature-document matrix because $\mathbf{A}_k$ has less noise in it than $\mathbf{A}$ does. In other words, we would not choose $k = d$ even if this were computationally feasible.

The number of dimensions k to keep in the reduced feature-document matrix when d is very large is still an open question, but experiments indicate that values of k between 100 and 300 typically give the best results [8]. Typically the performance of LSI improves dramatically as k increases from 1, peaks for some number that is still much less than d, 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 matrix $\mathbf{A}^T\mathbf{A}$:

$$\mathbf{A}^T\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1},$$

where $\mathbf{X}$ is the matrix of eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{A}^T\mathbf{A}$. It can be shown that if we have the SVD of $\mathbf{A}$, then

$$\mathbf{X} = \mathbf{V}, \quad \text{and} \quad \mathbf{\Lambda} = \mathbf{\Sigma}^2.$$

The issue of whether clustering in a vector space obtained from the first few PCA components is significantly better than the full set of dimensions was studied in [31]. 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 [18]. 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 [31] is in the low tens, while that of the document data is typically in the thousands.

2.3 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 [14], and there is no need to include any other information, for example from 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 database 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 $\mathbf{X}$ of size $t \times d$ be the observed mixture signals, where t is the number of features in the document collection, and d is the number of documents. The noise-free mixing model takes the form,

$$\mathbf{X} = \mathbf{AS}$$

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

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 $k \ll t$ 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 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 [17].

3 Text Representation Methods

Before a clustering algorithm can be applied to a document collection, a mapping of each text document d_j into a compact representation of its content needs to performed. Selecting a representation for text depends on what one believes to be the meaningful units of text (the problem of lexical semantics) and the meaningful natural language rules for the combination of these units (the problem of compositional semantics) [25]. The latter problem is usually neglected, by treating documents as bags of text units (words, terms, or character N-grams).

In the vector space model, each document is represented by a vector of weights of t features (words, terms, or character N-grams) extracted from the document:

$$d_j = (w_{1j}, w_{2j}, \ldots, w_{tj}),$$

where t is the number of features, and w_i is the weight of the ith feature. The weight of a feature represents how much that feature contributes to the semantics of document d_j. Differences among approaches are accounted for by

- different interpretations of a feature;
- different methods of computing feature weights.

In this research, we are interested in comparing three different feature types. If there are d documents in total, the corpus is represented by a $t \times d$ feature-document matrix $\mathbf{X}$.

As feature weights, we use the term-frequency, inverse-document frequency (TFIDF) scheme, which combines the term frequency and document frequency. Here TFIDF is used for all three feature types, words, terms, and N-grams. TFIDF is based on the idea that if a feature appears many times in a document, the feature is important to this document and should have higher weight. On the other hand, a feature that appears in many documents is not important since it is not very useful in distinguishing different documents; hence, it should have lower weight. The TFIDF variant we use is shown in Eq. 1:

$$w_{i,j} = (1 + \log f_{i,j}) \times \log \frac{d}{n_j} \tag{1}$$

where $w_{i,j}$ is the weight of feature i in document j, f_j is the frequency of the feature i in document j, n_j is the number of documents that contain feature j, and d is the total number of documents [1].

Word Representation A typical choice of a feature type for representing a text document is the "set of words" or the "bag of words". For each document and each word in it, a weight is assigned.

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, and it is therefore believed by some researchers to give better results than word representation in special text corpora [21]. However, experimental results have been mixed [11, 24, 29].

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 [7]. 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 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. However, the N-gram representation is not very effective in reducing the dimensionality of the document representation.

4 Experimental Results

We experimentally evaluated the performance of the different dimension reduction methods described previously on a number of different representations using several datasets. In this section we describe the various datasets and the pre-processing procedure, then our experimental methodology, followed by a description of the experimental results.

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. Below is a brief description of each dataset's characteristics, as well as the preprocessing applied to each. Table 1 summarizes the characteristics of the datasets.

University of Rochester Computer Science Technical Reports This dataset [1] consists of 528 abstracts from 4 categories - Artificial Intelligence (111), Systems (193), Theory (141), and Robotics (83). As all reports are from computer science, a fair amount of shared terminology between the categories is expected. This is the smallest dataset in our experiments. We call this dataset *URCS*.

The complete text of each abstract was used in the experiments with the exception of several short LaTeX formulas in several of the theory abstracts which were removed.

[1] http://www.cs.rochester.edu/trs/

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.

Classic3 This is a widely used dataset composed of 3891 abstracts from 3 disjoint research fields - 1398 aeronautical system papers (Cranfield), 1033 medical papers (Medline), and 1460 information retrieval papers (CISI). This dataset has been used by many researchers [2] in text mining. We call this dataset *Classic3*.

For the Cranfield and CISI documents, the title and the abstract were used. The Medline collection repeated the title in the abstract; thus for this collection only the abstracts were used.

A subset of 20 Newsgroups 20 Newsgroups is a collection of approximately 20000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. It has become a popular dataset for experiments in text applications of machine learning techniques[2]. The original dataset contains both closely related groups and highly disjoint ones. In our experiments we choose a subset of 7 relatively disjoint groups (comp.windows.x, rec.autos, sci.crypt, sci.med, talk.politics.guns, rec.sport.baseball, and soc.religion.christian), each with exactly 500 documents. We call this dataset *NG*.

Most experiments done previously with the newsgroup data tended to remove the newsgroup headers with the exception of the subject line, because the header contains the 'true' class of each document potentially biasing the clustering process. Thus this work took the same approach and removed the newsgroup headers with the exception of the subject line.

A subset of Reuters 21578 Reuters 21578 is currently the most widely used test collection for text categorization research. The data were originally collected and labelled by Carnegie Group, Inc. and Reuters, Ltd[3]. Because the dataset contains some noise, such as repeated documents, unlabelled documents, and nearly empty documents, we choose a subset of 10 relatively large groups (acq, coffee, crude, earn, interest, monet-fx, money-supply, ship, sugar, and trade), and we use two variants that we call: *RD256* (all documents have at least 256 bytes) and *RD512* (all documents have at least 512 bytes), in our experiments.

For each of the articles in the 10 categories that were used, the title and body of the article were extracted. It should be be noted that the word 'reuters' is technically a stopword for this dataset as every article ends with it.

The details of the preprocessing for each of the three representations used (words, terms, and N-grams) are presented next.

Word Representation. After the extraction the content of the documents in each dataset as described above, further processing is done to derive the word representation of each document.

[2]http://people.csail.mit.edu/jrennie/20Newsgroups/

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

For each of the datasets (URCS, Classic3, NG, RD256, RD512), the following processing steps were performed:

1. replaced all non-alphabetical characters with a space character
2. replaced all multispaces (tabs, spaces, or newlines) with a single space
3. converted all upper case characters to lower case
4. tokenized each document into a bag of words based on spaces
5. removed all stopwords from the standard van Rijsbergen stop word list
6. stemmed each word using using Porter's Stemmer [4] [1].
7. removed any word that appears in four documents or less in the dataset
8. calculated the weight of each word in each document using the standard TF/IDF (Term Frequency/Inverse Document Frequency).
9. normalized the feature vector for each document to unit length

Term representation. Documents were processed as follows:

- removed non-numeric characters such as at the beginning of each line (appear in some news documents)
- separated the sentences in each document by new-lines (exactly one sentence per line)
- tokenized punctuations (inserted a space before and after each punctuation mark)
- applied POS (part of speech) tagging using the Brown Corpus [5] as a reference
- removed numeric tokens, and punctuations
- applied Porter's stemmer
- extracted noun phrases (terms) using the automatic term extraction package developed by [20]

The extracted terms are then indexed, and stored in a file. Each of the terms represents a dimension of the feature space in the same way as when word features are used, except that the total dimension of the feature space is much smaller (as shown in table 2). Because of this, we do not perform any further term frequency-based dimension reduction as we do for words.

A disadvantage of using term representation is that a small subset of the document set may be represented by null document vectors, because the probability of a document not having any of the extracted features is much higher with term than word or N-gram representation. In fact, this document loss may become quite significant in some cases, such as the Classic-3 and URCS data in our experiments. Table 3 shows the detailed information regarding this issue.

[4]The software is available at `http://www.tartarus.org/~martin/PorterStemmer/`

[5]The Brown Corpus of Standard American English (or just Brown Corpus) was compiled by Henry Kucera and W. Nelson Francis at Brown University, Providence, RI

Feature representation	Initial dimension of feature space				
	Classic3	NG	RD256	RD512	URCS
Words	4420	9908	5396	4847	1430
Terms	786	1395	1325	1279	189

Table 2: Initial feature space dimensions (before reduction) using words and terms

	Classic3	NG	RD256	RD512	URCS
Actual documents	3893	3500	6519	3948	528
Null vectors	798	189	136	73	90
Percentage loss	20%	5%	2%	2%	17%

Table 3: Document loss when using term representation

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 [16, 19].

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 show that choosing N to be 3 or 4 tends to give optimal results, with little difference between the two [19]. Various profile lengths have been used by other researchers, with common lengths having been 1000, 2000, 3000, 4000, and 5000 [19, 16]. For the N-gram experiments, we used the N-gram software tool [15]. The preprocessing steps for N-grams are:

1. replace all non-alphabetical characters with a space character
2. replace all multispaces (tabs, spaces, or newlines) with a single space
3. convert all upper case characters to lower case
4. produce the N-gram representation for each document
5. derive the top k N-grams based on the number of documents each N-gram appears in (these set of k N-grams is called the N-gram profile)
6. remove any N-gram that does not appear in the N-gram profile from the N-gram representation of each document
7. calculate the weight of each N-gram in each document using the standard TF/IDF (Term Frequency/Inverse Document Frequency).
8. normalize the feature vector for each document 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 [32]. The overall purity of a clustering solution is defined as the weighted sum of individual cluster purities:

$$Purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r) \tag{2}$$

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.

In order to have reliable and representative results, that generalize to real situations, several points must be taken into account. First, we should pick text collections from a variety of domains. The choice of clustering algorithm is the second issue. K-means or its variants are the most commonly used clustering algorithms used in text clustering, but it is a well-known problem that the clustering results of k-means are sensitive to its initialization. In order to reduce the negative effect of poor initialization, each result is computed from the average of 15 different runs of each experiment.

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 [12]. 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 7 environment. The `svds` procedure is a built-in Matlab function. We also used the FastICA toolbox[6] for performing ICA. For the k-means algorithm, we used the GMeans Toolbox[7]. We have implemented the rest of the code in Matlab. The 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.

It has been demonstrated [28] that it is not necessary to normalize the projection matrix computed in LSI or ICA, because there is no significant difference between using the normalized or non-

[6] http://www.cis.hut.fi/projects/ica/fastica/

[7] http://www.cs.utexas.edu/users/yguan/datamining/gmeans.html

normalized projection matrix. Our experiments also show that using a normalized version of projection matrix in these two methods does not give us any significant improvement in clustering quality.

The authors in [28] suggest plotting the singular values of LSI and eigenvalues used in the whitening step of ICA on the same horizontal scale as the clustering performance for detecting the "good" range of reduced dimensions. 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. This reflects the fact that dimension reduction in LSI is performed by taking the upper left submatrix of dimension k of the diagonal matrix of singular values. Increasing the dimension simply involves taking a larger upper left submatrix of the same diagonal matrix, where singular values are sorted in non-increasing order along the diagonal. The graph of eigenvalues in ICA is defined similarly.

In all figures, the dimensions are ordered as follows: for DF, the dimensions are ordered according to the DF values; for LSI the dimensions are ordered based on the singular values (which indicate the importance of the dimensions); similarly, for ICA, the number of dimensions are determined by the PCA preprocessing step, in which the principal components are ordered based on their eigenvalues indicating their relative importance.

4.3 Experiments using Word Representation

We summarize the performance of the three dimension reduction methods for the word representation of all five datasets in Fig. 1. For all datasets, we observe that the singular values decrease very quickly for the first few tens of dimensions and after that, there is a smooth and somewhat flat reduction. In [28], the authors refer to the part of the singular value curve, where a rapid reduction of singular values happens, as the *transition zone*. By looking at the purity diagrams, this transition zone seems to correspond to the dimensionality where we can get best performance out of ICA or LSI.

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 following reviews these results in more detail for each dataset.

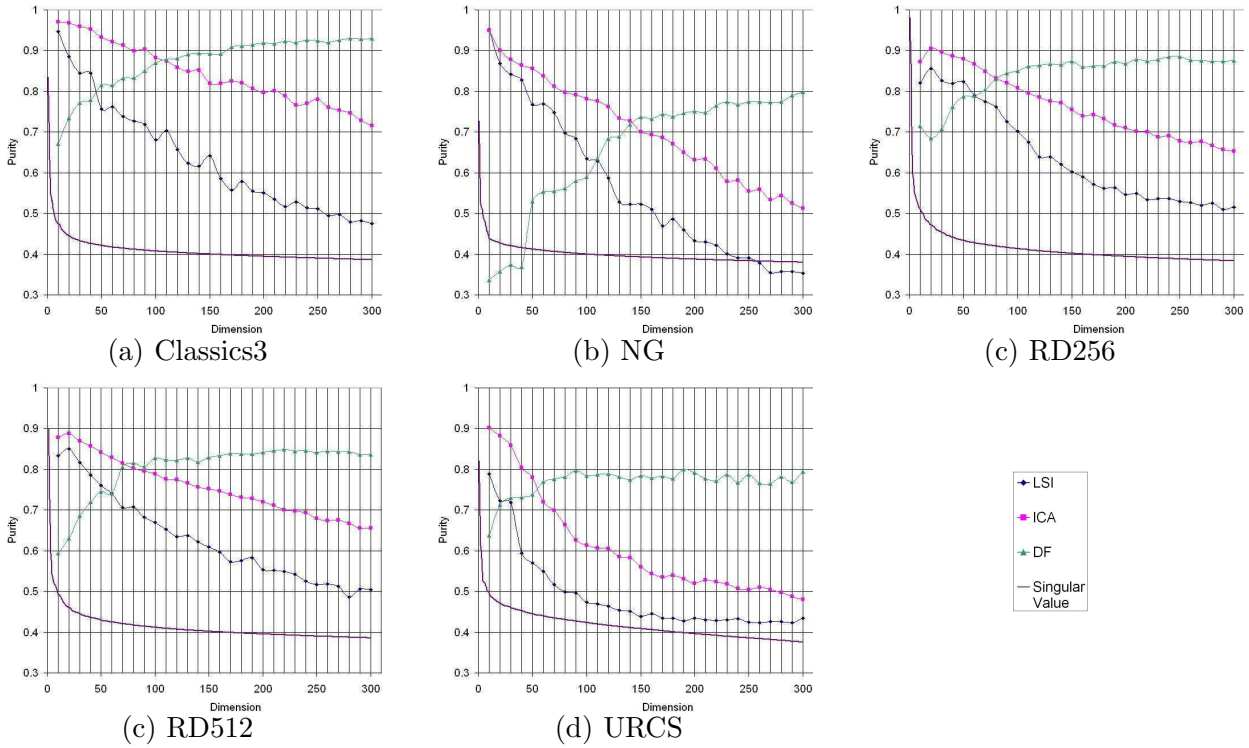
Classic3 dataset In the results shown in Fig. 1.a, we observe the following. The performance of DF peaks around dimensionality of 100 with purity of 0.80 and then flattens out and settles around 0.78 and 0.77 with increasing dimensionality. ICA and LSI achieve their best results with dimensionality equal to 10 (over the range examined). The best LSI performance matches the best performance of DF (achieved with a much higher dimensionality). The best ICA performance is better than the best performance of DF. LSI is inferior to ICA for the whole dimension range investigated.

NG dataset We observe the following in Fig. 1.b. For the range of dimensions between 10 and 100, both ICA and LSI are superior compared with DF. ICA provides its best performance with 10 dimensions and after that it still has the best results amongst the dimension reduction methods examined. LSI also provides its best result at 10 dimensions, but is inferior compared to ICA in terms of the best results and robustness. Similar to Classic3, the best results of LSI and ICA seem

to coincide with the transition zone of singular value curves.

RD256 and RD512 datasets LSI is again inferior to ICA for the whole range of dimensions investigated, as shown in Figs. 1.c and 1.d. For RD256, DF peaks at dimensions between 110 and 150 with purity around 0.87 and then flattens out and settles with a purity around 0.86. For RD512, it peaks again at dimensions between 110 and 150 with purity around 0.84 and then flattens out and settles with a purity around 0.83. LSI provides the best results at dimension 20 with purity of 0.85. ICA also provides its best result at dimension 20. Again, we observe a coincidence between good performance of LSI/ICA and the transition zones of singular value curves.

URCS dataset It can be seen in Fig. 1.e that LSI is again inferior compared to ICA. DF performance peaks at a dimension of 90 and then settles with a purity around 0.80. Both ICA and LSI provide better results over a range of [10, 40] compared with DF but for dimensionalities greater than 50, as the dimensionality increases, the performance of DF is getting better than both LSI and ICA. The best results of LSI and ICA are still better than that of DF, and they are achieved with very low dimensionalities. Again, the good performance of LSI/ICA coincides with the transition zones of the singular value curves in Fig. 1.e.



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 1: Comparing dimension reduction techniques, ICA, LSI, and DF on the word representation of five examined datasets. Purity is plotted as a function of the dimensions used. ICA outperforms LSI and DF in the low dimensions. A graph of the ordered and scaled sequence of singular values is also shown as a function of dimension, to display the correlation of its transition zone with the performance curves. The x-axis represents dimensionality, and the y-axis represents purity value.

Dataset	Classic3	NG	RD256	RD512	URCS
p-Value	0.5921	0.4965	0.0930	0.7566	0.7748
CI($\times$ 1000)	$[-1.8, 3.1]$	$[-5.3, 2.6]$	$[-0.4, 4.8]$	$[-1.6, 2.2]$	$[-7.7, 5.8]$

Table 4: Paired t-test results for the null hypothesis being that the means of purities for ICA and LSI are equal, for term representation. CI is the associated confidence interval.

4.4 Experiments using Term Representation

In the case of term representation, unlike word representation, there is no clear difference between LSI and ICA performance. Fig. 2 shows the performance results of LSI and ICA methods on all datasets.

To compare the performance of LSI and ICA within their investigated dimension range, the null hypothesis of the paired t-test assumes the means of purities for ICA and LSI for this range are equal. Table 4 shows the result of the paired t-test. The second row of this table contains the p values for the null hypothesis and the third row shows confidence interval for this hypothesis. The t-test does not reject the null hypothesis; therefore there is no statistically significant difference in the clustering quality of LSI and ICA methods.

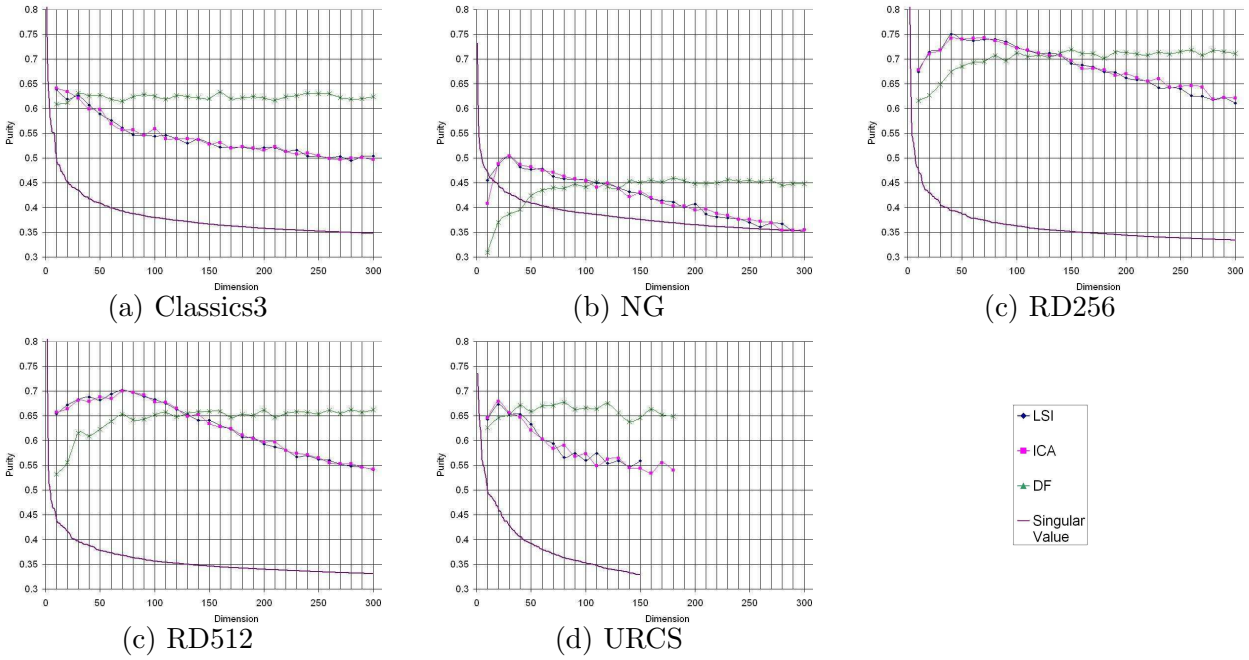
It is also interesting to see that, unlike the word representation, the best results for LSI and ICA do not seem to coincide exactly with the transition zone of singular value curves, but the transition zone still can give some hints about the starting point of searching for the best dimensionality.

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 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.

4.5 Experiments using N-gram Representation

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.



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 2: Comparing dimension reduction techniques, ICA, LSI and DF, on the term representation of five examined Datasets. The x-axis represents dimensionality, and the y-axis represents purity value. ICA and LSI have no significant performance difference, and they generally outperform DF in the low dimensions. In addition, a plot of Singular Values is included to show the Correlation of the Transition Zone with the Performance Curves.

4.5.1 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 is shown in Fig. 3. As it appears in these diagrams, for all datasets, clustering performance increases as the profile length increases. But for all datasets, it seems that a profile length equal to 2000 is the best-case length as defined above.

Clustering performance with LSI as the dimension reduction method for 3-gram representation with different profile lengths is shown in Fig. 4. Unlike the previous case with ICA, increasing the profile length does not necessarily provide better clustering results. 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.

Increasing the profile length does not have much impact on clustering quality when DF is used as dimension reduction method on 3-grams, as shown in Fig. 5. It is interesting that for all

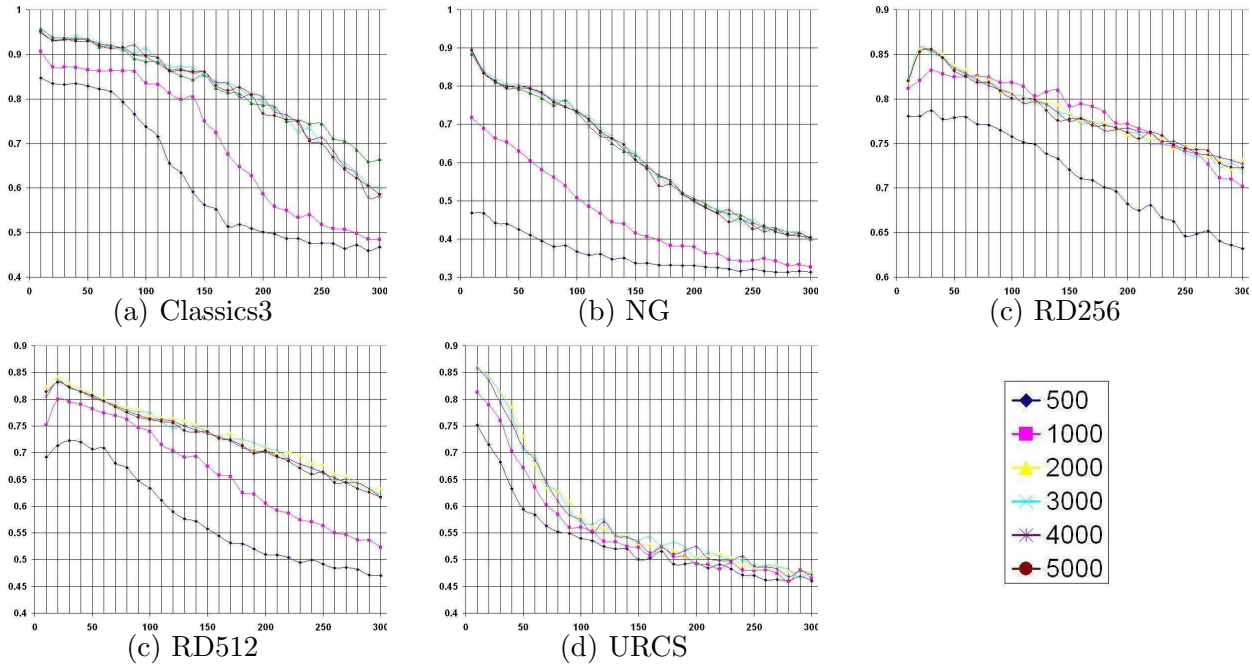


Figure 3: Purity as a function of dimension, parameterized by profile length for 3-gram representation when ICA dimension reduction is used. Profile length of 2000 is the shortest necessary to get the best performance. x-axis represents dimension and y-axis represents purity value.

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. The first experiment shows this impact for ICA method and the results can be seen in Fig. 6. 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.

The next experiment uses LSI as dimension reduction method, and its results are shown in Fig. 7. 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 5 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, as shown in Fig. 8. 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

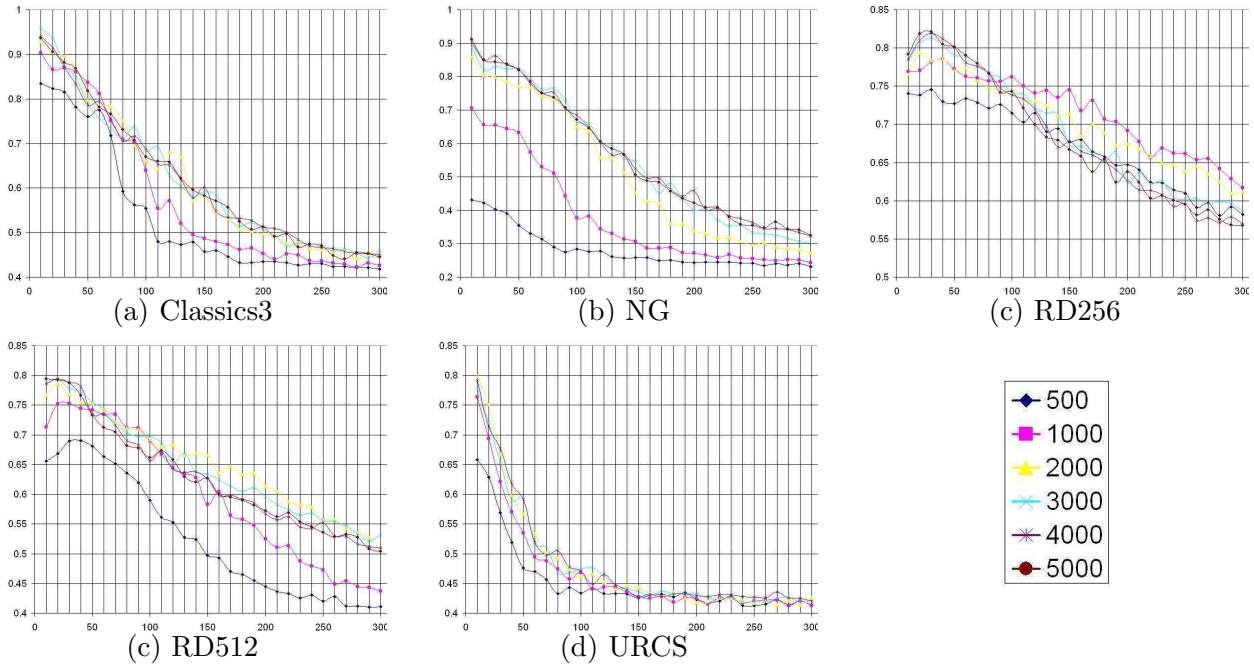


Figure 4: Purity as a function of dimension, parameterized by profile length for 3-gram representation when LSI dimension reduction is used. Profile length between 2000 and 3000 is necessary to get the best performance. x-axis represents dimension and y-axis represents purity value.

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.

4.5.2 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 5 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 6. Therefore, we intuitively expect that by increasing the N-gram size, we need a longer profile in order to capture the same amount of information. We observe this in table 5 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 is used as dimension reduction method on all datasets is shown in Fig. 9.

For the Classic3 dataset, the 3-gram representation clearly achieves better clustering quality com-

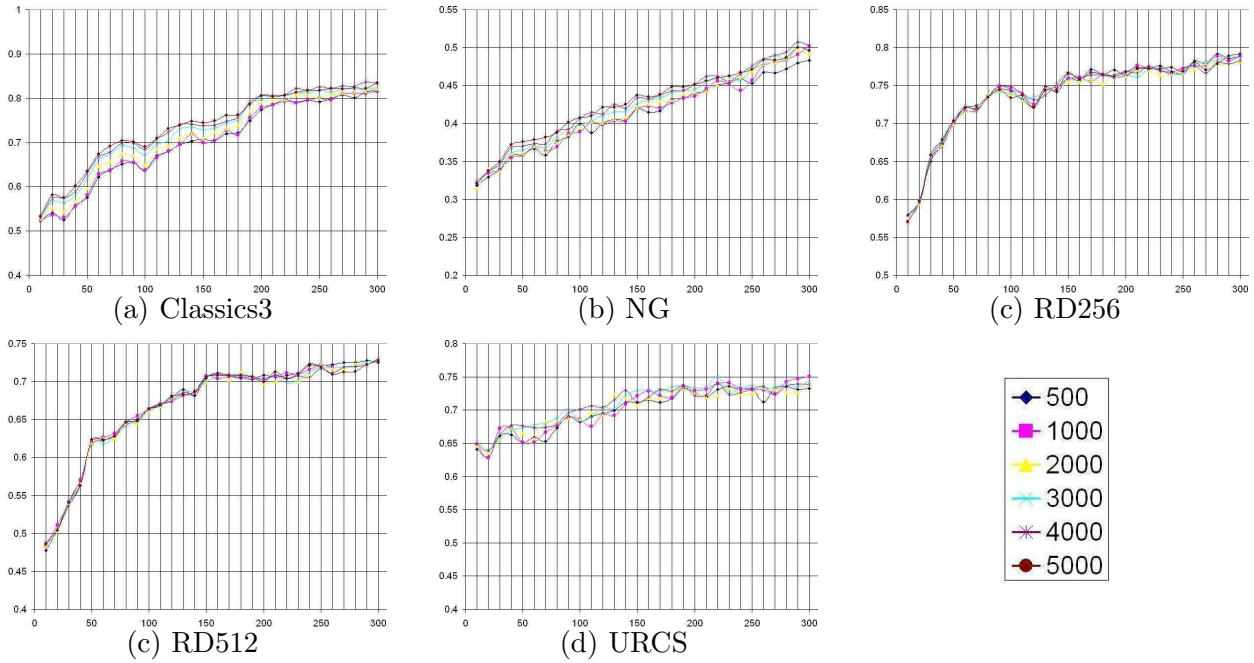


Figure 5: Purity as a function of dimension, parameterized by profile length for 3-gram representation when DF based dimension reduction is used. Increasing the profile length does not necessarily improve the clustering performance significantly. The x-axis represents dimensionality, and the y-axis represents purity value.

pared with 4-gram representation. For the NG dataset, 3-gram representation is slightly better than 4-gram representation in the whole range of dimensions investigated. Unlike the Classic3 and NG datasets, for both versions of the Reuters dataset, 4-gram achieves better performance compared to 3-gram representation. For the URCS dataset, the difference in performance between these two representations is not as pronounced as for other datasets, but a t-test for comparing the means of two performance curves shows that 4-gram representation is marginally better than 3-gram representation (p-values are approximately 0.1).

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 on all datasets is shown in Fig. 10. We observe that 4-grams achieve better clustering performance compared to 3-grams.

The comparison between the best performance achieved with 3-grams and 4-grams with different profile lengths when DF is used as dimension reduction algorithm on all datasets is shown in Fig. 11.

For the Classic3 dataset, the 3-gram representation clearly achieves better clustering quality compared to the 4-gram representation. For the NG dataset, the 3-gram representation is slightly better than the 4-gram representation in the whole range of dimensions investigated. Unlike the Classic3 and NG datasets, for both versions of Reuters dataset, 4-grams achieve better performance compared with 3-grams. For the URCS dataset, the difference between the performance of these two representations is not as pronounced as for other datasets, but a t-test for comparing the means of two performance curves shows that 4-gram representation is marginally better than 3-gram representation (with the p-value approximately 0.1).

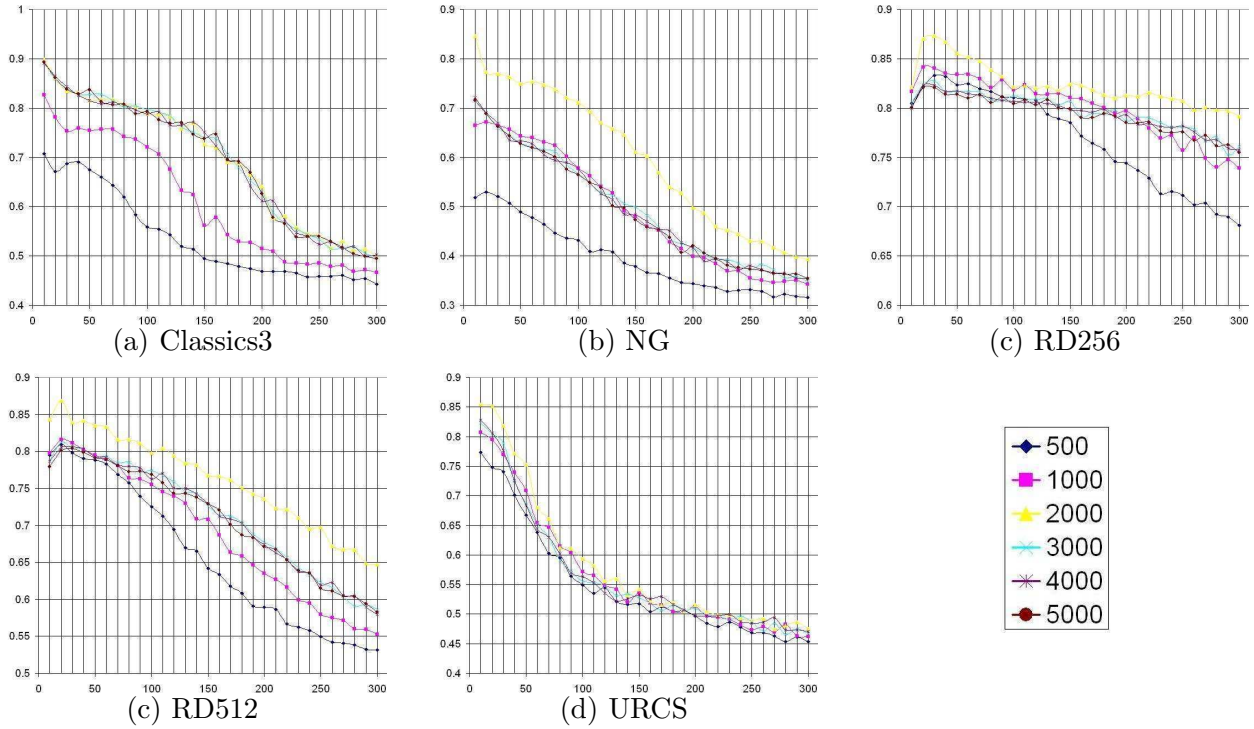


Figure 6: Purity as a function of dimension, parameterized by profile length for 4-gram representation when ICA dimension reduction is used. Profile length of 2000 is the shortest necessary to get the best performance. The x-axis represents dimensionality, and the y-axis represents purity value.

4.5.3 Best N-gram Parameters

In this section, we try to select a good profile length and reduced dimensionality for 3-gram and 4-gram representations. We have observed that, for almost all datasets, ICA achieves better clustering quality compared with LSI, as shown in Fig. 12. Only for the NG dataset, and for dimensionality less than 80, does LSI seem to be better. We notice, however, that for this dataset only, we compare performance with profile lengths of 4000 and 2000 for LSI and ICA respectively. If we compare equal profile lengths for this representation, we will notice that ICA performs better than LSI.

The DF based method with the 3-gram representation achieves better clustering quality for dimensions in the second half of the investigated range of dimensions, while in the first half and for lower dimensions, this method seems to be worse than the other two methods. It is interesting, however, that even the best performance of this method is still considerably worse than the best performance of the two other methods. Since we are interested in lower dimensions, it seems that DF is not the best selection amongst these three methods.

Contrary to the results for 3-gram representation, for all datasets, LSI with the 4-gram representation achieves better clustering quality compared to ICA in the range of dimensions in which we are interested, as shown in Fig. 13. However, upon a careful look at the profile lengths which are used for this comparison, we notice that because we are comparing the best results of ICA and LSI method, the profile lengths used are not necessarily equal. If we use equal profile lengths for comparison, then ICA method is slightly better than LSI method.

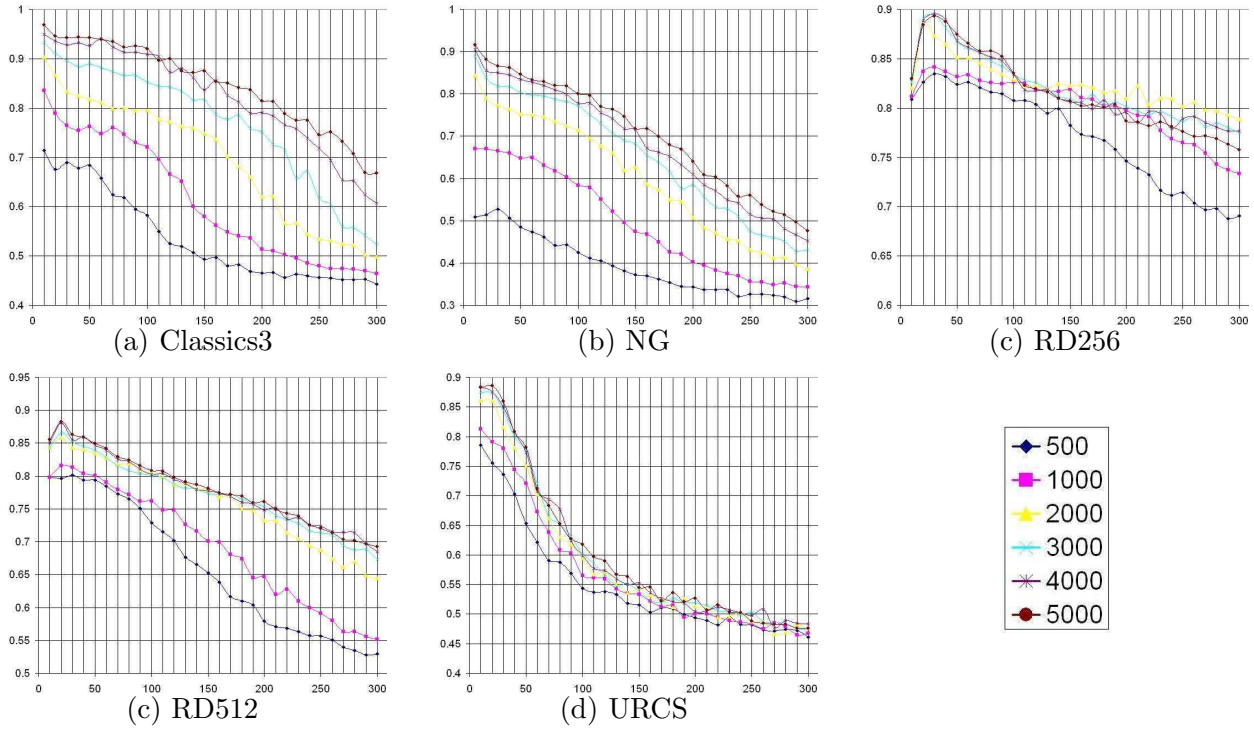


Figure 7: Purity as a function of dimension, parameterized by profile length for 4-gram representation when LSI dimension reduction is used. Profile length between 3000 and 5000 is necessary to get the best performance. x-axis represents dimensionality and y-axis represents purity value.

The DF based method, as with the 3-gram case, achieves better clustering quality for dimensions in the second half of the investigated range of dimensions, but the performance of ICA and LSI method is clearly much better than DF method for the 4-gram representation, compared with the 3-gram representation in Fig. 12. Again, as with the 3-gram case, it is interesting that even the best performance of this method is still considerably worse than the best performance of the two other methods. In this case, due to better clustering quality of 4-grams for some datasets, this difference is much larger compared to 3-gram representation. Since we are interested in lower dimensions, it seems that the DF based method is not the best choice amongst these three methods.

These experiments show that, given equal profile lengths, for both 3-gram and 4-gram representation, ICA achieves superior results to LSI. In the 3-gram case, this superiority is very clear, but for the 4-gram representation, the difference is not large, and the ICA method is only slightly better than LSI. The interesting point is that for the 3-gram representation, even the result of the best profile length when LSI is used is worse than a mid-range profile length when ICA is used as dimension reduction method. But for the 4-gram representation, when choosing the good profile length for LSI, we can expect LSI to achieve better results compared to ICA when the mid-range profile length is used.

As a general result for all experiments performed on the N-gram representation, it seems 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.

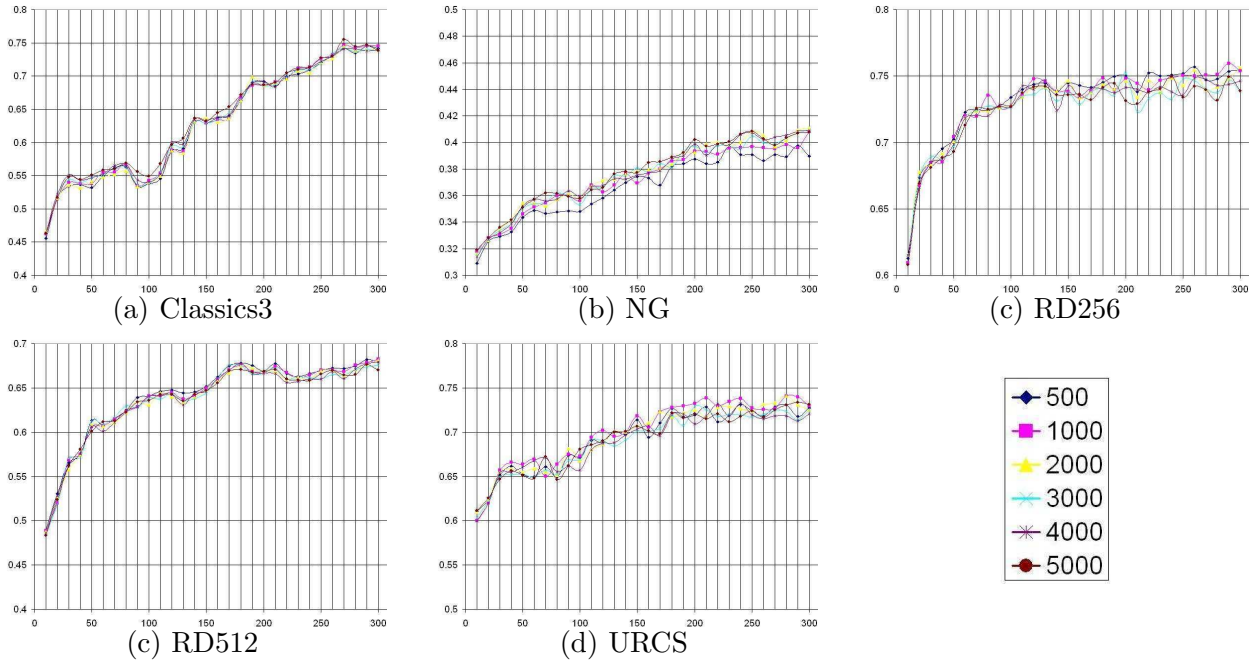


Figure 8: Purity as a function of dimension, parameterized by profile length for 4-gram representation when DF based dimension reduction is used. Increasing the profile length does not necessarily increase the clustering performance significantly. The x-axis represents dimensionality, and the y-axis represents purity value.

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

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 5: 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	0.5277	0.6665	0.7940	0.8561	0.8908	0.9124
NG	0.3503	0.5140	0.6840	0.7720	0.8242	0.8577
RD256	0.6028	0.7203	0.8289	0.8805	0.9092	0.9270
RD512	0.4930	0.6363	0.7743	0.8418	0.8798	0.9033
URCS	0.4517	0.6061	0.7578	0.8320	0.8730	NA
	4-Gram					
Classic3	0.6526	0.7432	0.8210	0.8602	0.8850	0.9022
NG	0.5089	0.6264	0.7316	0.7854	0.8193	0.8435
RD256	0.7226	0.7947	0.8559	0.8865	0.9059	0.9196
RD512	0.6385	0.7278	0.8067	0.8465	0.8721	0.8905
URCS	0.5965	0.6937	0.7822	0.8286	0.8587	0.8799

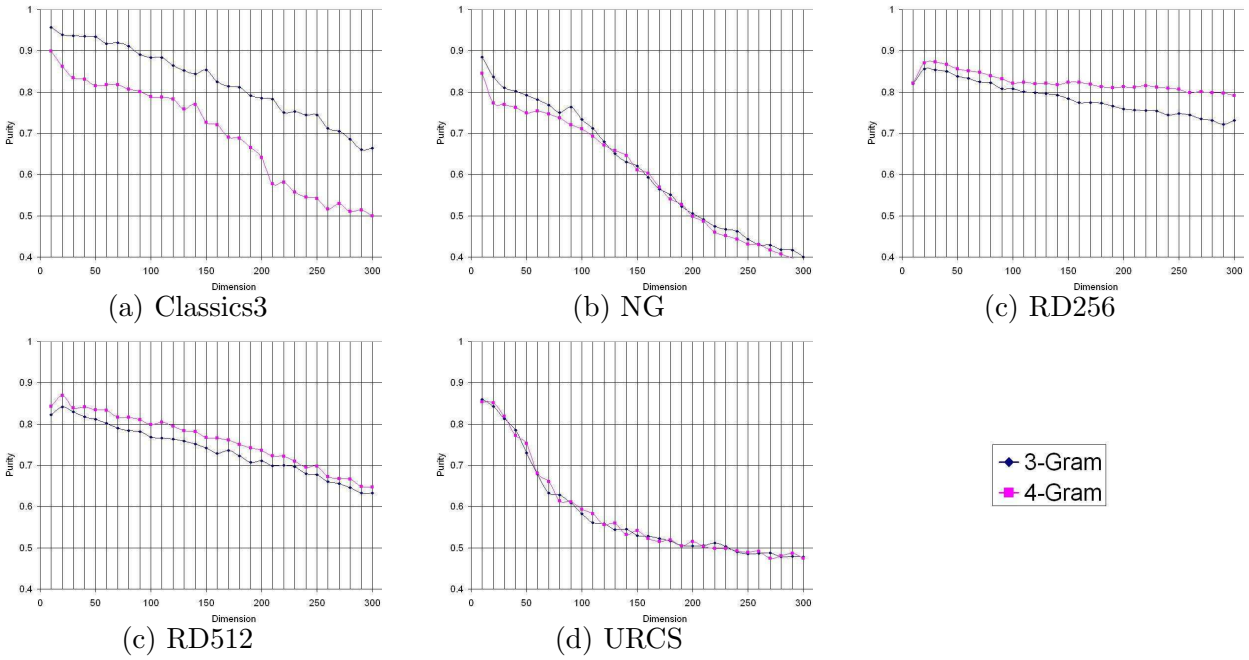
Table 6: Each value shows the fraction of zero elements in the corresponding matrix

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 shown in Fig. 14. We see that 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 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 7, on a Linux Pentium 4, 2.8GHz, with 1GB of RAM. The differences among the data sets are explained by the



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 9: Comparing best clustering results over all profile lengths considered for 3-gram and 4-gram representation, when ICA dimension reduction is used. No clear winning N-gram length can be seen. The x-axis represents dimensionality, and the y-axis represents purity value.

different initial dimensionalities, as shown in Section 4.1.

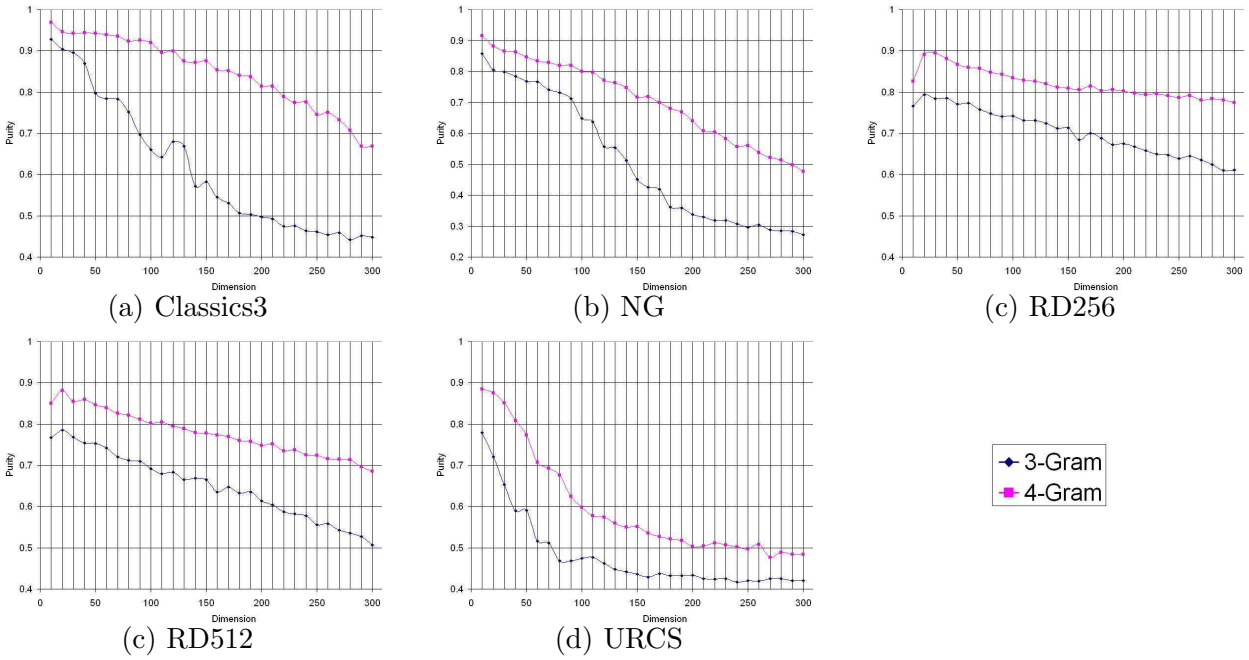
The run time of clustering for dimensionality equal to 100 for the word representation in seconds is shown in table 8. Run time increases linearly with dimensionality for the range tested (up to 300).

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

	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	0.007	0.11	5	35	2	12

Table 7: Run time of dimensionality reduction in seconds for dimensionality of 100. Run time for LSI and ICA is a slightly faster than linear function of dimensionality for the range tested (up to 300). Run time for DF is constant.



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 10: Comparing best clustering results over all profile lengths considered for 3-gram and 4-gram representation when LSI dimension reduction is used. 4-grams appear to outperform 3-grams. The x-axis represents dimensionality, and the y-axis represents purity value.

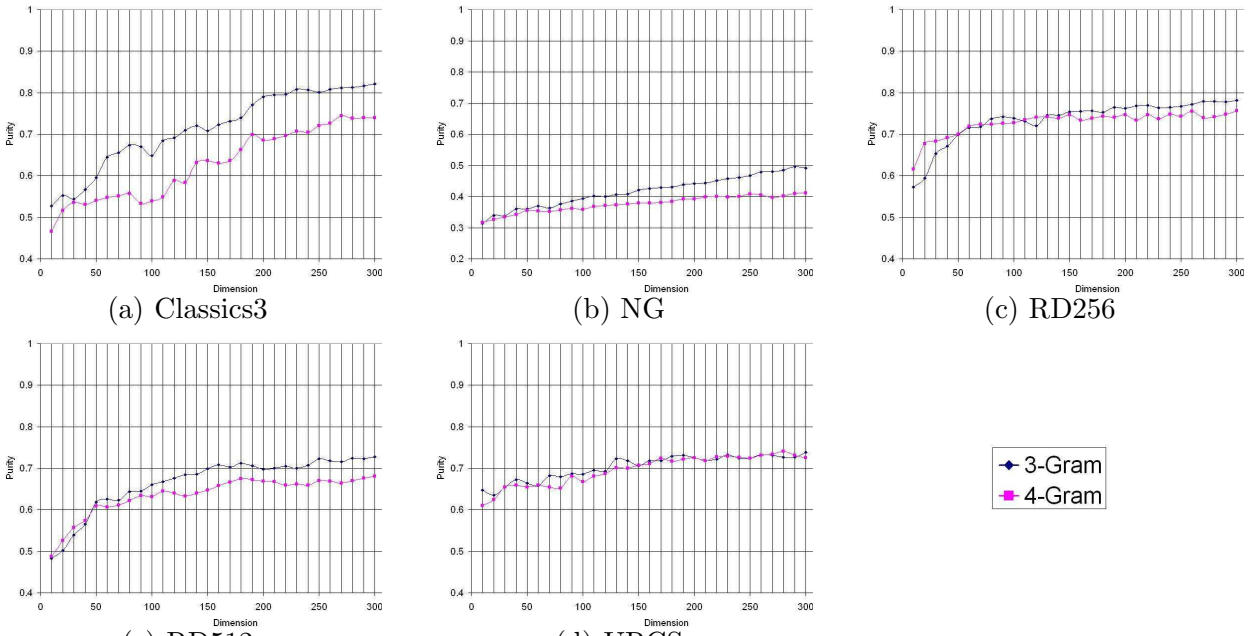
Data set	Run time (s)
Classic3	5.50
NG	8.70
RD512	28.70
URCS	0.28

Table 8: Run time in seconds for clustering for dimensionality of 100.

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.

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



(a) Classics3 (b) NG (c) RD256

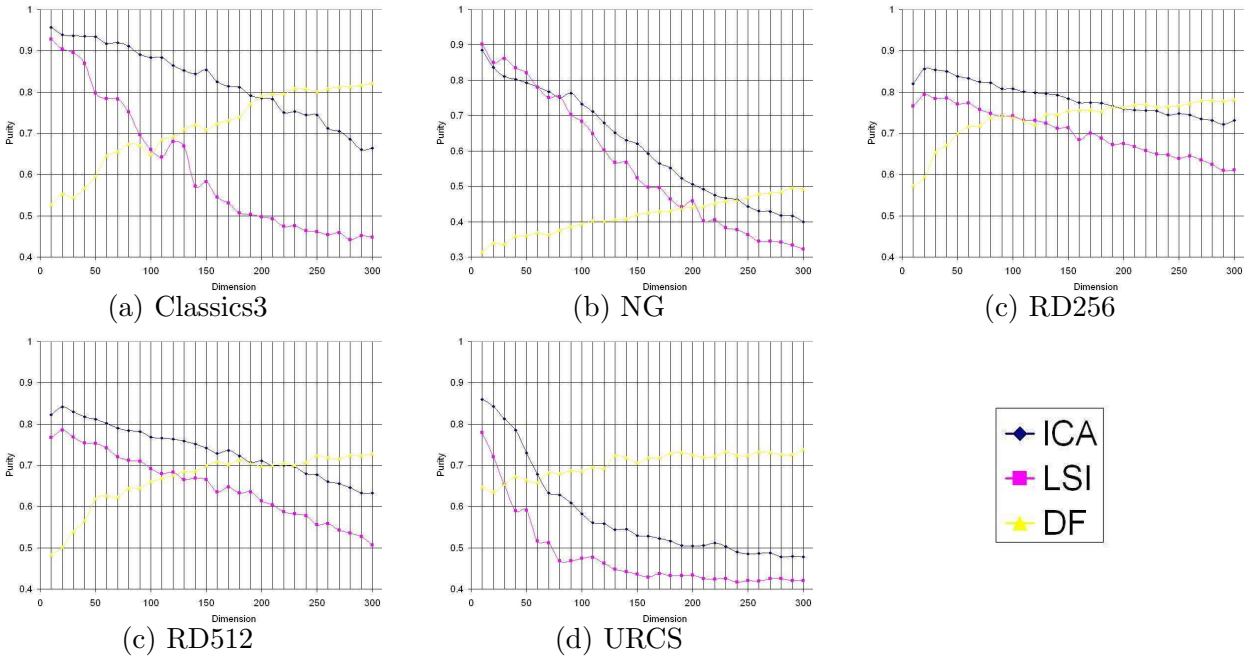
(c) RD512 (d) URCS

Figure 11: Comparing best clustering results over all profile lengths considered for 3-gram and 4-gram representation when DF based dimension reduction is used. No clear winning profile length is seen. The x-axis represents dimensionality, and the y-axis represents purity value.

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



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 12: 3-gram representation - comparing different dimension reduction methods. ICA appears to outperform LSI and DF in low dimensions. For each dimension reduction method, we select the profile length which gives the best result. The x-axis represents dimensionality and y-axis represents purity value.

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

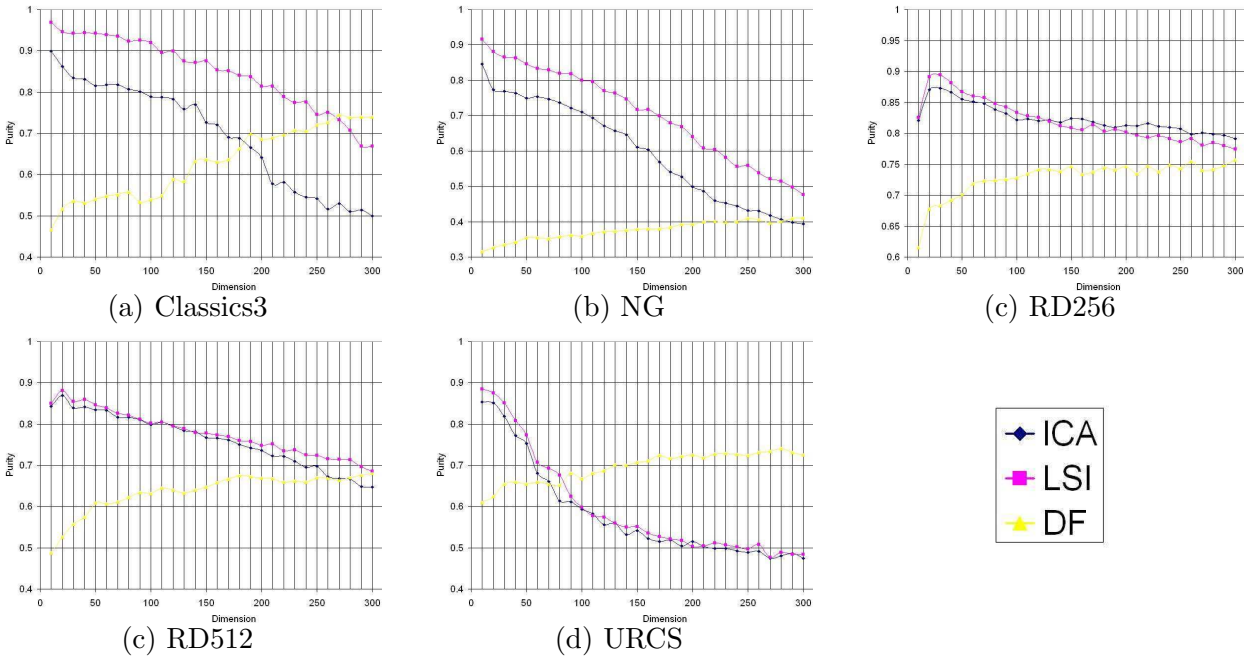
[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, NY, 1999.

[2] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD '04: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 509–514, New York, NY, USA, 2004. ACM Press.

[3] F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proc. 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Edmonton, Alberta, 2002.

[4] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[5] 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.



(a) Classics3 (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 13: 4-gram representation - comparing different dimension reduction methods. LSI appears to come ahead of ICA when the best profile length for each method is used. The x-axis represents dimensionality, and the y-axis represents purity value.

[6] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

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

[8] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[9] 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.

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

[11] Norbert Fuhr, Stephan Hartmann, Gerhard Knorz, Gerhard Lustig, Michael Schwantner, and Konstadinos Tzeras. AIR/X – a rule-based multistage indexing system for large subject fields. In André Lichnerowicz, editor, *Proceedings of RIAO-91, 3rd International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 606–623, Barcelona, ES, 1991. Elsevier Science Publishers, Amsterdam, NL.

[12] Parry Husbands, Horst Simon, and Chris 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.

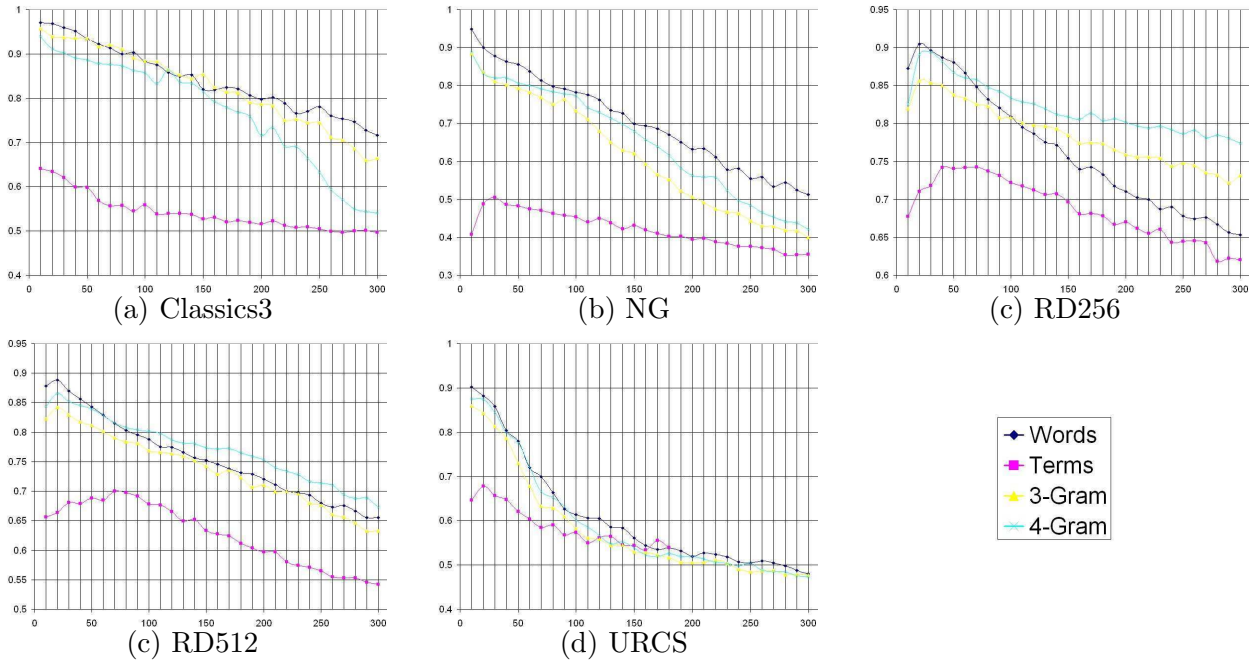


Figure 14: Comparing text representation methods using the best dimension reduction technique for each dataset. Words and N-grams outperform terms. N-gram computation needs tuning, but it is language independent. The x-axis represents dimensionality, and the y-axis represents purity value.

[13] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

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

[15] Vlado Keselj. Perl package text::ngrams, 2004.

[16] Vlado Keselj, Funchen Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics*, August 2003.

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

[18] Kristina Lerman. Document clustering in reduced dimension vector space. http://www.isi.edu/∼lerman/papers/papers.html, 1999.

[19] Yingbo Mao, Vlado Keselj, and Evangelos E. Milios. Comparing document clusteing using n-grams, words, and terms. Master's thesis, Dalhousie University, 2004.

[20] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic term extraction and document similarity in special text corpora. In *Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics (PACLing'03)*, pages 275–284, Halifax, Nova Scotia, Canada, August 22-25 2003.

[21] 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.

[22] Lance Parsons, Ehtesham Haque, and Huan 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.

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

[24] Hinrich Schutze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Research and Development in Information Retrieval*, pages 229–237, 1995.

[25] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[26] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *23rd Annual International ACM SIGIR Conference*, 2000.

[27] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of common document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[28] 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.

[29] Kostas Tzeras and Stephan 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.

[30] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In Douglas 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.

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

[32] 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.