A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering

Evangelos E. Milios EEM@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

M. Mahdi Shafiei SHAFIEI@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

Singer Wang SWANG@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

Roger Zhang ROGER@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

Bin Tang BTANG@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

Jane Tougas TOUGAS@CS.DAL.CA
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5, Canada

Abstract

Increasingly large text datasets and the high dimensionality associated with natural language is a great challenge of text mining. In this research, a systematic study is conducted of application of three Dimension Reduction Techniques (DRT) on three different document representation methods in the context of the text clustering problem using several standard benchmark datasets. The dimensionality reduction methods considered include Independent Component Analysis (ICA), Latent Semantic Indexing (LSI) and one technique based on Document Frequency (DF). These three methods are applied on three Document representation methods based on the idea of Vector Space Model, namely word, term and N-Gram representations. Experiments with the k-means clustering algorithm show that ICA and LSI are clearly better than DF on all datasets. For word and N-Gram representation, ICA gives better results compared to LSI. Experiments also show that the word representation gives better clustering results compared to term and N-Gram representation. Finally, for N-Gram representation, it is shown that profile length equal to 2000 is

enough to capture the information and in most cases, 4-Gram representation gives better performance compared to 3-Gram representation.

Keywords: Latent Semantic Indexing, Independent Component Analysis, Dimension Reduction

1. Introduction

Today, constant and rapid changes in information and communication technologies offer ubiquitous access to vast amounts of information and make an exponential increase of the amount of documents available online. While more and more textual information is available electronically, effective retrieval and mining is getting more and more impossible without efficient organization, summarization and indexing of document content. Among different approaches tackled this problem, document clustering is one of the main and enabling approaches. In general, given a document collection, the task of text clustering is to group similar 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 different applications and algorithms, one can refer to two recent surveys on the topic (Jain et al. (1999) and Berkhin (2002)).

High dimensionality has always been a great challenge for all learning algorithm and "curse of dimensionality" has been studied for a long time. Traditional representation of documents known as bag-of-words considers every document as a vector in a very high dimensional space where each element of this vector represents one term appeared in the document collection. In a more general sense, this representation is based on the *Vector Space Model* (Salton and Buckley (1988)) where each document is represented as vector, where vector components represent certain feature weights.

Based on this idea, some other representations are proposed. The traditional representation as mentioned earlier, considers the components of vectors as unique words. Another approach is using N-Grams as the components of vectors. An N-gram is a sequence of symbols extracted from a long string (Cavnar (1994)). 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 N-gram representation has the advantage of being more robust and less sensitive to grammatical and typographical errors and requiring no linguistic preparations which makes it more language independent. Another approach for representing text documents is using multi-word terms as components of vectors. 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 words. Another advantage of using terms for representing a document is its lower dimensionality compared to traditional word representation or N-Gram representation.

Using 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. This will add a difficult characteristic to the document clustering problem. Moreover, a very small subset of all terms appeared in text

collection will appear in each document which results in having a very sparse but also very high-dimensional feature vector for describing that document.

Most learning algorithms use some kind of similarity measure to discriminate between two given training vector where in the case of document clustering, due to high dimensionality of feature vector, these similarity measures lose their discriminative power. In a sparse high dimensional space, feature vectors almost have equal distance to each other (K. Beyer and Shaft (1999)) which makes traditional similarity measures meaningless.

Many Researchers from different areas have tried to solve the high dimensionality problem and they have proposed various dimension reduction techniques (Fodor (2002)). In a general view of dimension reduction, one can think of two types of dimension reduction methods which are known as feature transformation and feature selection (Parsons et al. (2004)).

Feature transformation techniques try to reduce the dimensionality to a fewer new dimensions, which are linear or non-linear combinations of the original dimensions. In another word, the original high dimensional space is projected to a lower dimensional space in feature transformation method. These methods are believed to be very successful in uncovering 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 (Fodor (2002)) for more details. In feature selection methods, the objective is not extracting new features but rather removing features which seem irrelevant for modeling. This problem is a combinatorial optimization problem (Blum and Langley (1997)).

The focus of this research is to evaluate the relative effectiveness of dimension reduction techniques for document clustering problem when multiple document representation methods are used. This paper is organized as follows. Section 2 provides more details for the three Dimension Reduction Techniques used in this research. Section 3 presents some details about three different text representation methods used in experiments. Section 4 describes the general experimental procedure and evaluation methods, describes the characteristics of the datasets used and the pre-processing procedure followed and finally presents our experimental results and appropriate discussion notes. Finally, conclusions are drawn and future research directions identified in Section 5.

2. Dimension Reduction techniques

In mathematical terms, the problem of dimensionality reduction can be stated as follows: given the p-dimensional random variable $\mathrm{x} = (x_1, \ldots, x_p)^T$, the objective is finding a representation of data with lower dimensions, $\mathrm{s} = (s_1, \ldots, s_k)^T$ with $k \leq p$, which contains 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-goodness criterion threshold to eliminate some terms from the full vocabulary of the document corpus. In an unsupervised framework, some of these criteria are Document Frequency and Term Frequency Variance.

If we assume that we have n observations, 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 $k \leq p$ 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 $n \times p$ observation 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 (Yang and Pedersen (1997)).

In the following sections, we review one of mostly used feature selection methods for text and also two feature transformation techniques used for text dimension reduction.

2.1 Document Frequency based Method

Document Frequency of a term is the number of documents in which that term occurs. One can use Document Frequency as a criterion for selecting good terms. The basic intuition behind using document frequency as a criterion is that rare terms either don't capture much information about one category or they don't affect the global performance (Yang and Pedersen (1997)). It also may improve the performance if these low frequency terms happen to be noise terms. This method is often used after removing some very high frequent terms known as "stop words" and also stemming.

With using Document Frequency (DF) as a criterion for feature selection, only those dimensions with high Document Frequency values will appear in the feature vector. In spite of its simplicity, it has been believed to be as effective as more advanced techniques among feature selection methods (Yang and Pedersen (1997)). We are going to use this technique on different document representation methods based on Vector Space Model and see how effective this method will be on these different representation methods and in compare with other feature transformation methods for text clustering.

DF can be formally defined as follows. For a document collection in matrix notation, $A_{m\times n}$, with m terms and n documents, the DF value of term 't', DF_t, is defined as the number of documents in which t occurs at least once among the n documents. To reduce the dimensionality of A from m to k ($k < m$), we choose to use the k dimensions with the top k DF values. It is obvious that the DF takes $O(mn)$ to evaluate.

2.2 Latent Semantic Indexing

According to the mean-square error, Principal component analysis (PCA) is the best linear dimension reduction technique which is based on the covariance matrix of the variables and therefore is a second-order method (Fodor (2002)). Sometimes, in text mining, it is known as the singular value decomposition (SVD).

SVD takes a matrix $\mathbf{X}$ and represents it as $\hat{\mathbf{X}}$ in a lower dimensional space such that the distance between the two matrices as measured by the 2-norm is minimized:

$$\Delta = \|\mathbf{X} - \hat{\mathbf{X}}\|_2$$

The 2-norm for matrices is the equivalent of Euclidean distance for vectors.

Basically, PCA objective is reducing the dimension of data by finding a few new orthogonal dimension which are known as *principal components* (PC) and are linear combinations of the original variables with the largest variance. The first PC is the linear combination with the largest variance and the second PC is the linear combination with the second largest variance and orthogonal to the first PC, and so on. Theoretically, There are as many PCs as the number of the original variables, but for many datasets, the first several PCs explain most of the variance, so that the rest can be disregarded with minimal loss of information.

Latent Semantic Indexing is a technique which can be used to project documents into a space with 'latent' semantic dimensions. The latent semantic space that we project into has fewer dimensions than the original space (which has as many dimensions as terms).

Latent semantic indexing is the application of singular value decomposition (SVD), to a word-by-document matrix. Since SVD (and hence LSI) is a least-squares method, the projection into the latent semantic space is chosen such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences.

SVD project an n-dimensional space onto a k-dimensional space where $n \gg k$. In our application (word-document matrices), n is the number of word types in the collection. Values of k that are frequently chosen are 100 and 150. The projection transforms a document's vector in n-dimensional word space into a vector in the k-dimensional reduced space.

Latent Semantic Indexing (LSI) is closely related to Principal Component Analysis (PCA) with only this difference that PCA can only be applied to a square matrix whereas LSI can be applied to any matrix.

The SVD projection is computed by decomposing the document-by-term matrix $\mathbf{X}_{t\times d}$ into the product of three matrices, $\mathbf{T}_{t\times n}, \mathbf{S}_{n\times n}, \mathbf{D}_{d\times n}$:

$$\mathbf{X}_{t\times d} = \mathbf{T}_{t\times n}\mathbf{S}_{n\times n}(\mathbf{D}_{d\times n})^T$$

where t is the number of terms, d is the number of documents, $n = \min(t, d)$ and $\mathbf{T}$ and $\mathbf{D}$ have orthonormal columns.

The matrices T and D represent terms and documents in the new space. The diagonal matrix S contains the singular values of A in descending order. The i^{th} singular value indicates the amount of variation along the i^{th} axis. By restricting the matrixes $\mathbf{T}$, $\mathbf{S}$ and $\mathbf{D}$ to their first $k < n$ rows one obtains the matrixes $\mathbf{T}_{t\times k}, \mathbf{S}_{k\times k}, \mathbf{D}_{d\times k}$. Their product $\hat{\mathbf{X}}$

$$\hat{\mathbf{X}}_{t\times d} = \mathbf{T}_{t\times k}\mathbf{S}_{k\times k}(\mathbf{D}_{d\times k})^T$$

is the best square approximation of A by a matrix of rank k in the sense defined in the equation $\Delta = \|\mathbf{X} - \hat{\mathbf{X}}\|_2$.

In this representation the columns of $\mathbf{S}_k\mathbf{T}_k^T$ are identified as the "projected terms" and the columns of $\mathbf{D}_k$ are identified as the "projected documents". Note that the new representation of document j is $\mathbf{S}_k^{-1}\mathbf{T}_k^T\mathbf{X}(:;j)$ where $D(:;j)$ denotes the jth column of matrix X.

Choosing the number of dimensions k for $\hat{\mathbf{X}}$ is an interesting problem. While a reduction in k can remove much of the noise, keeping too few dimensions or factors may loose important information. As discussed in (Deerwester et al. (1990)) using a test database of medical abstracts, LSI performance can improve considerably after some very low dimensions, peaks around some bigger but still low dimensions, and then begins to diminish slowly. This pattern of performance (initial large increase and slow decrease to word-based performance) is observed with other datasets as well. Eventually performance must approach the level of performance attained by standard vector methods, since with $k = n$ factors $\hat{\mathbf{X}}$ will exactly reconstruct the original term by document matrix X. That LSI works well with a relatively small (compared to the number of unique terms) number of dimensions or factors k shows that these dimensions are, in fact, capturing a major portion of the meaningful structure.

The assumption in LSI (and similarly for other forms of dimensionality reduction like principal component analysis) is that the new dimensions are a better representation of documents and queries.

One objection to SVD is that, along with all other least-squares methods, it is really designed for normally-distributed data, but such a distribution is inappropriate for count data, and count data is what a term-by-document matrix consists of. One problematic feature of SVD is that, since the reconstruction $\hat{\mathbf{X}}$ of the term-by-document matrix A is based on a normal distribution, it can have negative entries, clearly an inappropriate approximation for counts.

It is hoped that these new dimensions represent meaningful underlying "topics" present in the collection. But the interpretation of the new dimensions can be difficult at times. Although they are uncorrelated variables constructed as linear combinations of the original variables, and have some desirable properties, they do not necessarily correspond to meaningful physical quantities.

2.3 Independent Component Analysis

In comparison to Principal Component Analysis (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. While the latter only involves the second-order statistics, the former depends on all the higher-order statistics. Independence always implies uncorrelatedness, but not vice versa in general.

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 (Jung (2001)) and there is no need to include any other information, for example from higher moments. This makes second-order methods very robust and computationally simple, since only classical matrix manipulations are used.

Independent Component Analysis (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 assumed to be linear or nonlinear mixtures of some unknown latent variables. The mixing coefficients are also unknown. 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 an extension to Principal Component Analysis. Actually, For Gaussian distributions, the Principle Components are Independent Components. ICA is a much richer technique, however, capable of finding the sources when these classical methods fail completely.

Let the observed mixture signals be denoted X, a matrix of size $T \times N$, where T is the number terms in the document collection and N is the number of documents. The noise free mixing model takes the form,

$$X = AS$$

where S is the source signal matrix (size $M \times N$, M is the number of sources) and A is the $T \times M$ mixing matrix.

In contrast with PCA and SVD, the objective of ICA is not necessarily dimension reduction. For dimensionality reduction, it is assumed that there are as many independent components as there are original variables, i.e. $k = p$. To find $k < p$ independent components, one needs to first reduce the dimension of the original data p to k, by a method such as PCA.

One problem of using ICA as a dimensionality reduction method is that there is no order among the Independent Components (ICs). 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 found that it may be a powerful tool for analyzing text document data as well, if the documents are presented in a suitable numerical form. ICA has been used for dimensionality reduction and representation of word histograms (Kolenda et al. (2000)).

3. Text Representation Methods

Due to inability of clustering algorithms for interpreting text documents directly, usually an indexing procedure that maps a text d_j into a compact representation of its content needs to be applied to documents. Selecting a representation for text depends on what one believes as 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) (Sebastiani (2002)) where the latter problem is usually neglected.

With this introduction, one of the widely used representation methods for text documents is based on the Vector-space model idea. In this model, each document is represented by a vector of weights of n "features" extracted from the document:

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

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

- different ways to understand what a feature is;
- different ways to compute feature weights.

In this research, we are partly interested in comparing three different ways of understanding what a feature is. If there are n documents in total, the corpus is represented by a $n \times m$ matrix X which is usually called term-document matrix. In this work, we study three different representation proposed and widely used by research community: words, terms and N-grams.

3.1 Word Representation

A typical choice for a feature for representing a text document is to identify features with words. This is often called either the "set of words" or the "bag of words" approach to document representation, depending on whether weights are binary or not.

3.2 Term Representation

The multi-word terms or sometimes called phrases can also be used as features in document vectors. Using term representation has the potential of reducing significantly the dimensionality and therefore believed by some researchers, giving better results than word representation in special text corpora (E. Milios Y. Zhang (2004)). But experimental results have not been uniformly encouraging (Fuhr et al. (1991), Schutze et al. (1995), Tzeras and Hartmann (1993)).

3.3 N-Gram Representation

N-Grams 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 (Cavnar (1994)). Hence, the number of distinct N-Grams in a text is less than or equal to $|A|^n$. This shows that the dimensionality of the N-Grams feature vector can be very high even for moderate values of n. However all these N-Grams are not present in a document, thus reducing the dimensionality substantially. For example there are 8727 unique trigrams (excluding stop words) in the Reuters dataset. Generally during N-Grams feature vector formation all the uppercase characters are converted into lowercase characters and space is assumed for punctuation. The feature vectors are then normalized.

Extracting character N-grams from a document is like moving a n character wide window across the document character by character. Each window position covers n characters, defining a single N-gram. In this process, 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 kind of 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 requiring no linguistic preparations which makes it more language independent. However, the n-grams representation is not so effective in reducing the number of dimensions (words) to be given to the text mining algorithm. This benefit is better achieved by stemming and stop word removal.

4. Experimental Results

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

4.1 Datasets and Data Preparation

We use 4 datasets for our test, including both unstructured newsgroup items and relatively more structured abstracts from scientific research papers. These datasets are 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 1466 per class. Below is a brief summarization of each dataset's characteristics and Table 1 summarizes the characteristics of the datasets.

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

Classic3 This is a long-existing dataset composed of 3893 abstracts from 3 disjoint research fields - 1400 aeronautical system papers (Cranfield), 1033 medical papers (Medline), and 1460 information retrieval papers (CISI). This dataset has been used by many researchers (Banerjee et al. (2004)) in text mining. We call this dataset "Classic3" hereafter.

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 test 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" hereafter.

A subset of Reuters 21578 Reuters 21578 is currently the most widely used test collection for text categorization research. The data was originally collected and labelled

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

Dataset	Dataset size	classes	class size range
Classic3	3891	3	???
NG	3500	7	???
RD-256	6519	10	???
RD-512	3948	10	???
URCS	528	4	???

Table 1: Summary of data sets used in experiments.

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 use two variants called hereafter: RD256 (all documents have at least 256 bytes) and RD512 (all documents have at least 512 bytes), in our test.

?????? Here comes the preprocessing part For N-Gram and Word representation that Singer and Roger have done. at most 1 page. ??????

The pre-processing of the datasets follows the most practiced procedures, including, removal of the tags and non-textual data, stop word removal , and stemming . Then we further remove the words with low document frequency.

In order to use term representation, we first pre-process each data set as follows:

- Remove noise characters such as ">" at the beginning of each line in some news documents
- Separate the sentences in each document by new-lines (exactly one sentence per line)
- Tokenize punctuations (insert a space before and after each punctuation mark)
- Apply POS (part of speech) tagging using the Brown Corpus [4] as a reference
- Remove stop words, numeric tokens, and punctuations
- Apply Porter stemmer
- Extract noun phrases (terms) using the automatic term extraction package developed by (Milios et al. (2003))

The extracted terms are then indexed, and store 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), and because of this, we do not perform any further term frequency based dimension reduction as we do for words.

3. http://www.daviddlewis.com/resources/testcollections/reuters21578/
4. 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 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

Notice one disadvantage of using term representation is that a small portion of each data set may get lost due to null document vectors, because the probability of a document's not having any of the extracted features is much higher with term representation than with words. In fact, this document loss may become quite significant in some cases, such as the Classic-3 and URCS data in our test. Table 3 shows the detailed information regarding this.

We use $TFIDF$ feature weighting scheme which combines the term frequency and document frequency. $TFIDF$ 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 more weight. 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.

4.2 Experimental Design and Metrics

Several ways of measuring the quality of clustering and especially text clustering has been proposed by research community. Since for our datasets we have the class labels of each data item, we can use one group of measures which considers the class labels of the data for judging the quality of clustering results. Therefore, we have selected one of the mostly used quality measures in text clustering: Purity.

Purity measures the extend to which each cluster contained documents from primarily one class (Zhao and Karypis (2001)). The overall purity of clustering solution is defined as the weighted sum of individual cluster purities:

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

where $P(S_r)$ is the purity for a particular cluster of size n_r, k is the number of clusters and n is the total number of data items in the dataset. Obviously, the higher the Purity value, the purer the cluster in terms of the class labels of its members, the better the clustering results.

For having reliable and representative results and being able to generalize them to real situations, several points must be taken into account. First, we should pick several different

text collections which can represent different domains. Choosing the clustering algorithm is the second issue. It should be a commonly believed good clustering algorithm for text collections. 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 not always optimal and stable due to poor choices of 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.

Like suggested in (Husbands et al. (2001)) we have divided the whole dataset into training and testing set. It is not common to have two separate sets of training and test documents in text clustering and researchers, mostly like to use just clustering results on the training set. But as it is common in classification problems, in order to have results closer to actual performance of the clustering algorithm, we divide the whole text collection into the training part and test part. At the training part, we do usual clustering and extract the clusters for the dataset. Then for each cluster, we use it's mean as its representative. At the testing part, we simply use the nearest neighbor classification algorithm to assign a document to its cluster.

All our experiments are conducted under Matlab 7 environment. The svds procedure is taken directly from Matlab toolboxes. We also used the FastICA toolbox[5] for doing ICA. For the k_means algorithm, we used the GMeans Toolbox[6]. We have implemented the rest of the codes with matlab.

It is shown (B. Tang and Shepherd (2004)) 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-normalized projection matrix for ICA and LSI. Our experiments also show that using normalized version of projection matrix in these two methods doesn't give us any significant improvement in clustering quality.

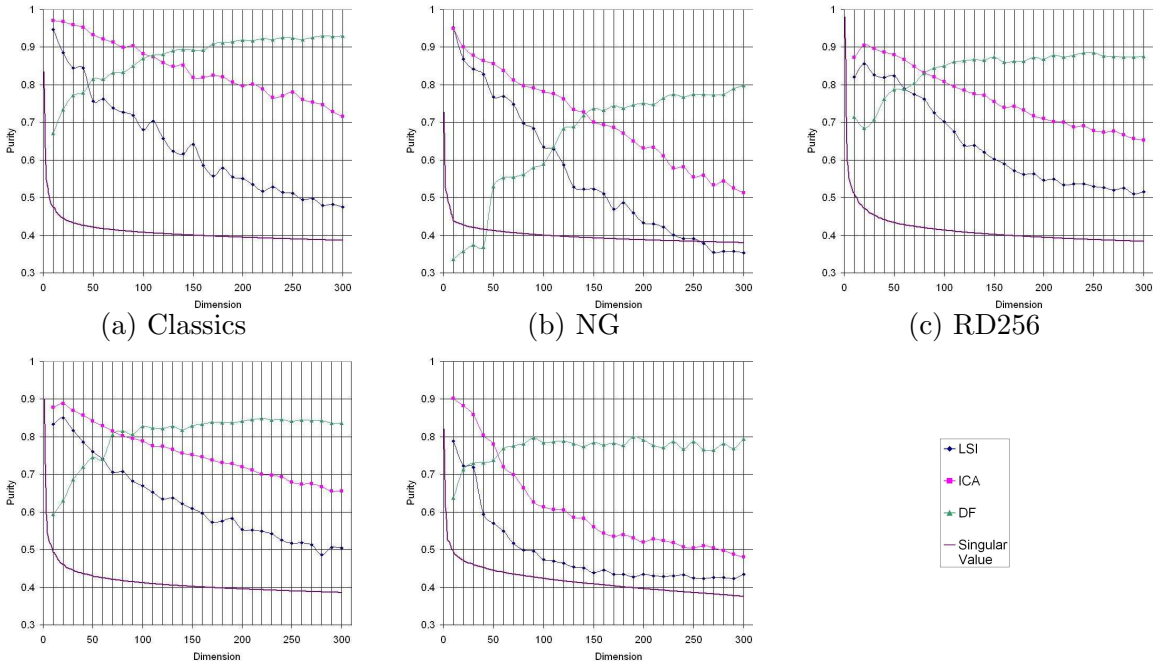
Authors in (B. Tang and Shepherd (2004)) suggest for detecting the "good range of reduced dimensions", plotting the dimensionality reduction methods performance against the singular values of LSI and eigenvalues used in the whitening step of ICA. They are hoping that this correlation may suggest how to determine the "good range of dimensions to reduced to" by ICA.

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 principle components are ordered based on their eigenvalues indicating their relative importance.

4.3 Experiments using Word Representation

in Figure 1, we summarize the performances of the three dimensionality reduction methods for word representation of all five datasets. For all datasets, we observe that, the singular values decrease very quickly for the first few tens of dimensions and after that, there is smooth and somewhat flat reduction. in (B. Tang and Shepherd (2004)), the authors refer

5. the FastICA Toolbox 2.1 for MATLAB is freely available from http://www.cis.hut.fi/projects/ica/fastica/
6. the GMeans Toolbox 2.1 for MATLAB is freely available from http://www.cs.utexas.edu/users/yguan/datamining/gmeans.html



(a) Classics (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. Also a Plot of Singular Values to Show the Correlation of its Transition Zone with the Performance Curves. x-axis represents dimensionality and y-axis represents purity value.

to this part of singular value curve that rapid transition for the reduction of siggular values happen as the transition zone. By looking at the purity performance diagrams, this transition zone seems to correspond to the best dimensionality where we can get best performance out of ICA or LSI.

A quick look at the results shows that for all datasets, clustering quality using ICA is better than that for LSI in the whole range of dimensionalities investigated. For low dimensions, especially for dimensions lower than 50, for all datasets, Document Frequency Based method has the worst performance amongst dimension reduction methods used. The following reviews these results in more detail for each dataset.

For Classic3 dataset, from Figure 1.a, we observe the following. The performance of DF peaks around dimension of 100 with purity of 0.80 and then flattens and settles around 0.78 and 0.77 with increasing dimensionality. ICA and LSI achieve their best results with lower dimensionality 10 that for LSI match with the best performance of DF but for ICA, it's better than the best performance of DF. LSI is inferior to ICA for the whole dimension range investigated.

We have the following observations based on Figure 1.b for NG dataset. For the range of dimensionalities between 10 and 100, both ICA and LSI are superior compared to DF. ICA provides its best performance at dimensionality of 10 and after that it still has the best

Dataset	Classic3	NG	RD256	RD512	URCS
p Value	0.5921	0.4965	0.0930	0.7566	0.7748
CI	$[-0.0018, 0.0031]$	$[-0.0053, 0.0026]$	$[-0.0004, 0.0048]$	$[-0.0016, 0.0022]$	$[-0.0077, 0.0058]$

Table 4: Paired student t test results for null hypothesis of means of purities for ICA and LSI being equal for term representation

results amongst two other dimensionality reduction methods examined. LSI also provides its best result at dimensionality of 10, 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.

Results for RD256 and RD512 in Figures 1.c and 1.d show that LSI is again inferior to ICA for the whole range of dimensionalities investigated. 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 performances of LSI/ICA and the transition zones of singular value curves.

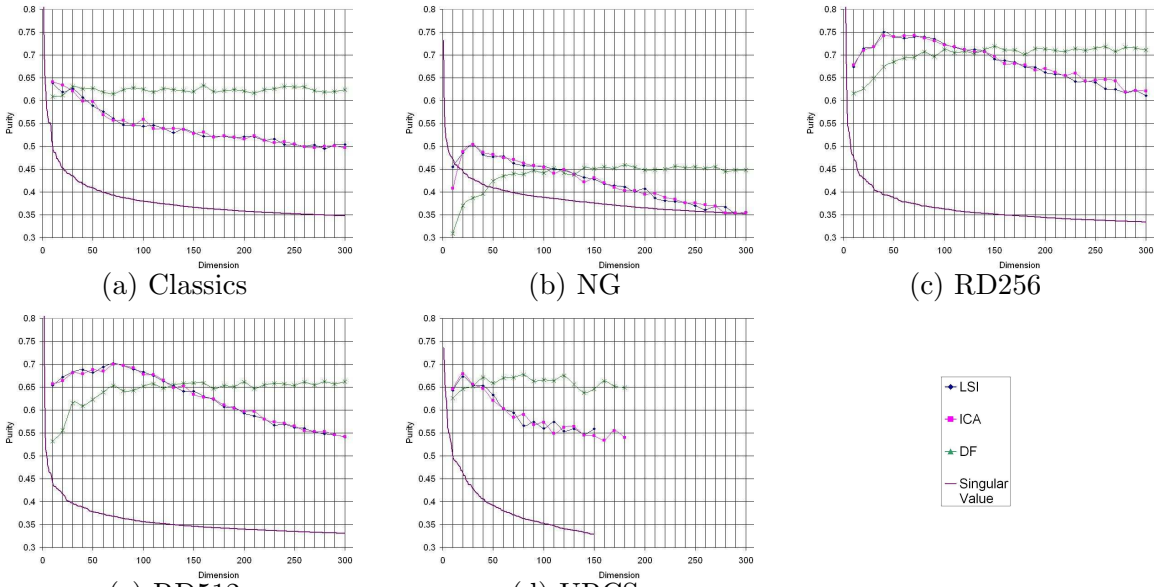
For URCS dataset, as it can be seen in Figure 1.e, LSI is again inferior compared to ICA. DF performance peaks at a dimension of 90 and then settles with a purity around 0.80. Both ICA and LSI provide better results over a range of [10, 40] compared to DF but for dimensionalities greater than 50, as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

4.4 Experiments using Term Representation

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

To compare the performance of ICA and LSI within their investigated dimension range, the null hypothesis of the paired student t test assumes the means of purities for ICA and LSI for this range are equal. Table 4 shows the result of paired student 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 result of the t-test doesn't rejects the null hypothesis. Therefore, it's not statistically reasonable to assume that clustering quality of LSI and ICA method are not equal.

It is also interesting to see that not like the word representation, the best results of LSI and ICA don't seem to coincide exactly with the transition zone of singular value curves,



(a) Classics (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. Also a Plot of Singular Values to Show the Correlation of its Transition Zone with the Performance Curves. x-axis represents dimensionality and y-axis represents purity value.

but the transition zone still can give some hints about the starting point of searching for the best dimensionality.

For the Document Frequency Based method, the overall performance pattern is like word representation. The clustering quality starts to increase as the dimensionality increases till some middle-range values for all datasets. After this middle-range value, the clustering quality settle 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 two other methods but here for the term representation, it's not very close to the maximum performance of other methods.

The trends of performance curves of LSI and ICA methods for all datasets are similar to corresponding curves in word representation experiment. In this case, like the word representation, performance of these two dimension representation methods reaches its maximum at some very low dimensions around 20 and then starts to degrade.

4.5 Experiments using N-Gram Representation

For N-grams experiments, we have used N-Gram software tool (Keselj (2004)).4-Gram representations with different profile length ranging from 500 to 5000. In this section, the objective of experiments is determining the effect of N-Gram profile length on clustering quality. We are also interested in identifying which of the two N-Gram representations achieves better performance when the three dimensionality reduction methods are applied.

Then for each dataset, we pick one N-Gram length and its corresponding profile length which achieved best clustering quality as the N-Gram representation representative when we compare different representation methods.

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 profile length used to generate N-Grams on the clustering quality. In these experiments, we apply ICA, LSI and Document Frequency Based 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 result amongst different profile lengths as the candidate profile length for the corresponding dataset and dimensionality reduction method.

Figure 3 shows the clustering performance for 3-Grams when ICA has been used as dimension reduction method. As it appears in these diagrams, for all datasets, clustering performance increases as the profile length increases. But for all datasets, it seems that profile length equal to 2000 is the shortest profile length enough to get the best performance out of 3-Grams and after this value, increasing the profile length doesn't necessarily increase the clustering performance too much.

We also use LSI as dimension reduction method for 3-Gram representation with different

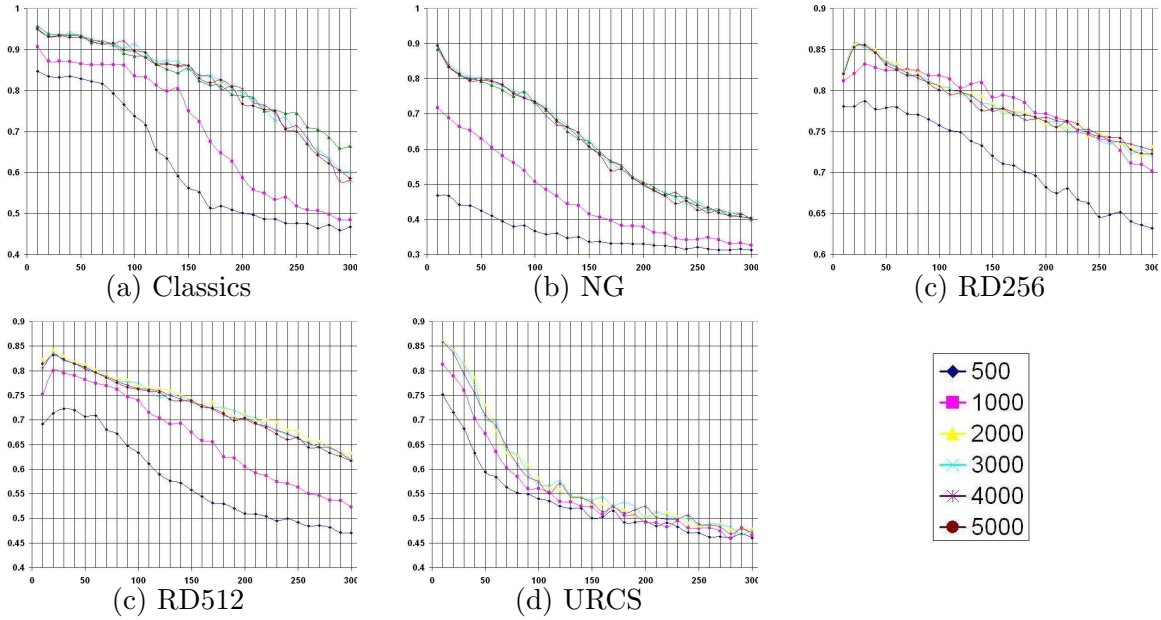


Figure 3: Impact of N-Gram profile length on clustering quality for 3Gram Representation when ICA Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

profile lengths. Figure 4 shows the clustering performance results. In spite of the previous case with ICA, with increasing the profile length, we don't get necessarily better clustering

results. For example, in datasets RD256 and RD512, after profile length 2000, with increasing the profile length, clustering quality gets worse. As it can be seen in this figure, the optimum profile length is different for each dataset, but still we don't need to go further than 4000 for profile length to get the best clustering result. Actually, profile length equal to 2000 is still one of the best profile lengths for 3-Gram when we use LSI as dimensionality reduction method.

Based on figure 5, increasing the profile length doesn't have much impact on clustering

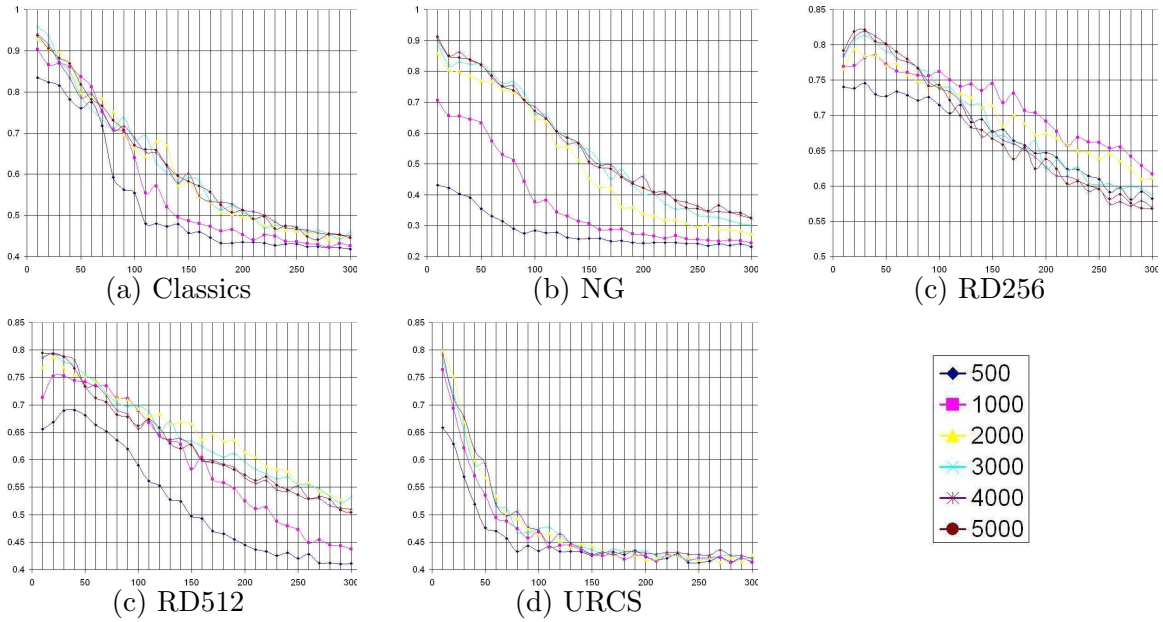


Figure 4: Impact of N-Gram profile length on clustering quality for 3Gram Representation when LSI Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

quality when Document Frequency is used as dimension reduction method. It is interesting that for all different profile lengths investigated, the clustering quality change pattern remains almost the same. Still we can see some small improvements in clustering quality with increasing profile length, but due to computational expenses, it doesn't seem reasonable to go further than 500 for getting better clustering quality using this dimensionality reduction method.

The next three experiments shows impact of profile length for 4-Gram representation when one of three dimension reduction methods is used. The first experiment tries to show this impact for ICA method and the results can be seen in Figure 6. For all datasets, clustering performance increases as the profile length increases like what we have seen for 3-Grams. In this case again, profile length equal to 2000 is the shortest length enough to get the best performance out of 4-Grams and after this value, increasing the profile length doesn't necessarily increase the clustering performance too much.

The next experiment which its results is shown in Figure 7 uses LSI as dimension reduction method. In spite of what we saw in the similar experiment for 3-Gram representation, with

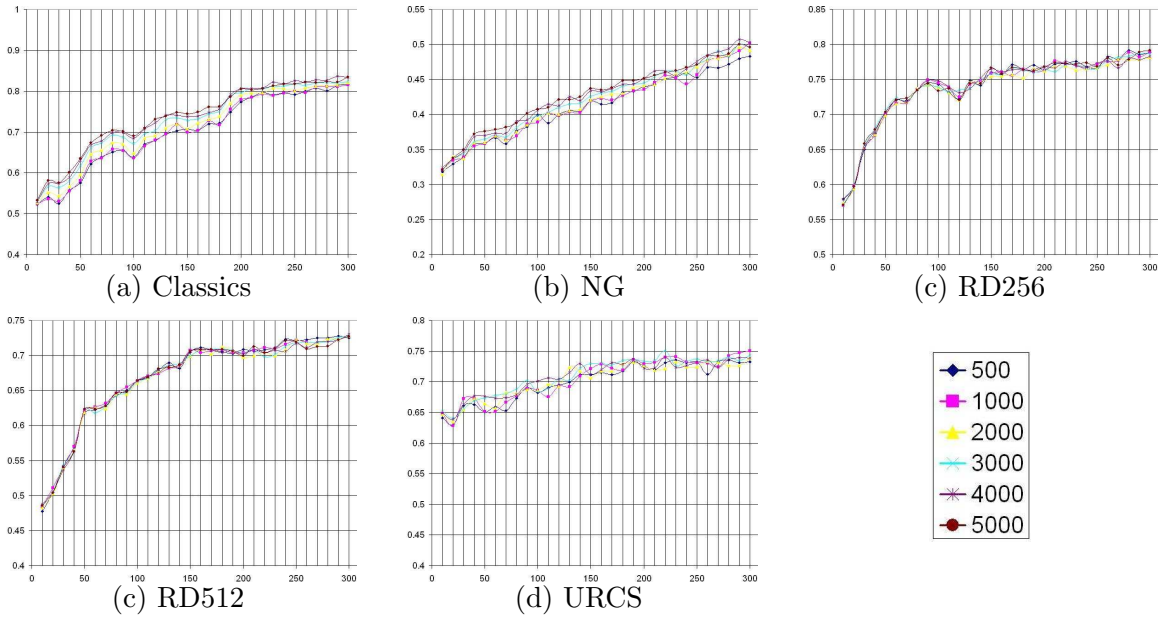


Figure 5: Impact of N-Gram profile length on clustering quality for 3Gram Representation when Document Frequency Based Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

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 dimensionality reduction method

Based on figure 8, like for 3-Gram representation, increasing the profile length doesn't 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 expenses, it doesn't seem reasonable to go further than 500 for getting better clustering quality using this dimensionality reduction method.

Based on the experiments in this section, increasing the profile length doesn't change the clustering quality considerably when Document Frequency based method is used for dimensionality reduction. Due to computational costs incurred from having longer profile length, It seems that for this dimensionality reduction method, profile length equal to 500 is good enough to get the optimum clustering quality. For ICA dimensionality reduction method, for all datasets, after dimensionality equal to 2000, we don't get reasonably enough increase in clustering quality to select longer profile length. But for LSI method, for most datasets, increasing profile length seems to have impact on clustering quality. 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.

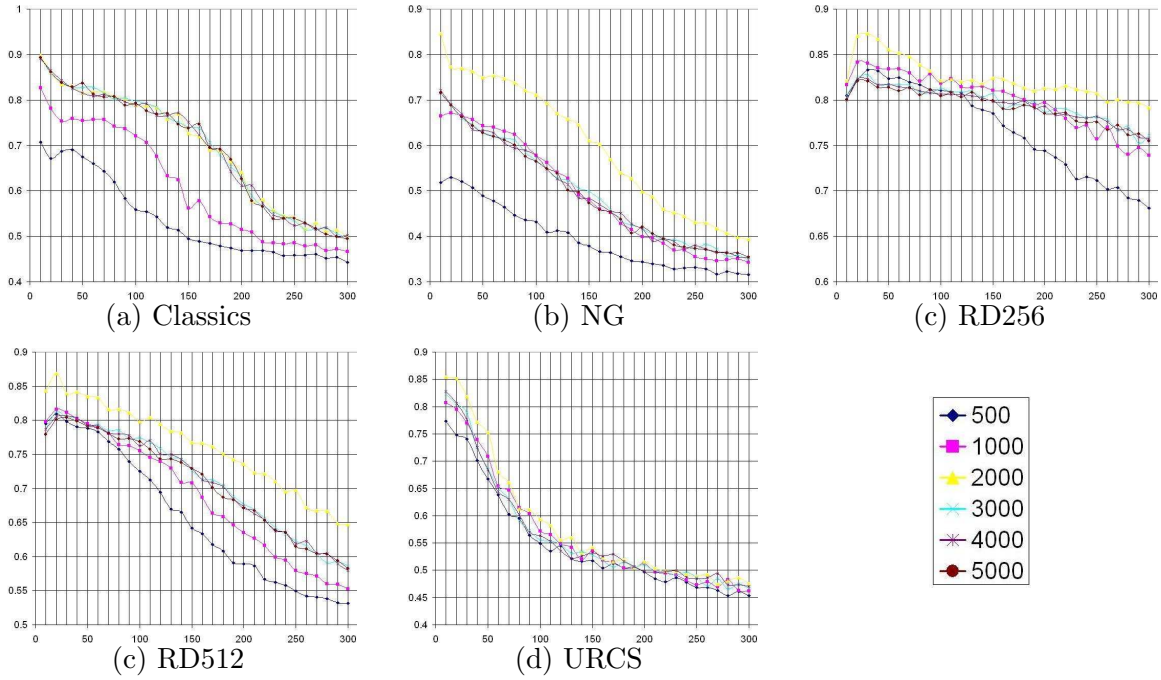


Figure 6: Impact of N-Gram profile length on clustering quality for 4Gram Representation when ICA Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

4.5.2 N-Gram Length and Clustering Quality

In order to investigate the N-Gram length effect at clustering quality, for 3-Grams and 4-Grams, we choose the best profile length based on the results shown in the previous section. Table 5 shows these best profile length for 3-Grams and 4-Grams when one of the three dimension reduction methods - LSI, ICA or DF - is applied.

Dimensionality and sparsity increase with increasing N-Gram size. Therefore, we expect by increasing the N-Gram size, we should have longer profile in order to capture the same amount of information. We see this fact in table 5 for LSI method. As it can be seen, for 4-Gram representation we need to have longer profile to get the best clustering quality compared to 3-Gram representation.

Figure 9 shows the comparison between the best performance achieved with 3-Grams and 4-Grams with different profile lengths when ICA is used as dimensionality reduction method on all datasets.

For Classic3 dataset, 3-Gram representation clearly achieves better clustering quality compared to 4-Gram representation. For NG dataset, 3-Gram representation is slightly better than 4-Gram representation in the whole range of dimensions investigated. In spite of Classic3 and NG datasets, for both versions of Reuters dataset, 4-Grams achieves better performance compared to 3-Gram representation. For URCS dataset, the difference between performance of these two representations is not clear like other datasets, but a t student

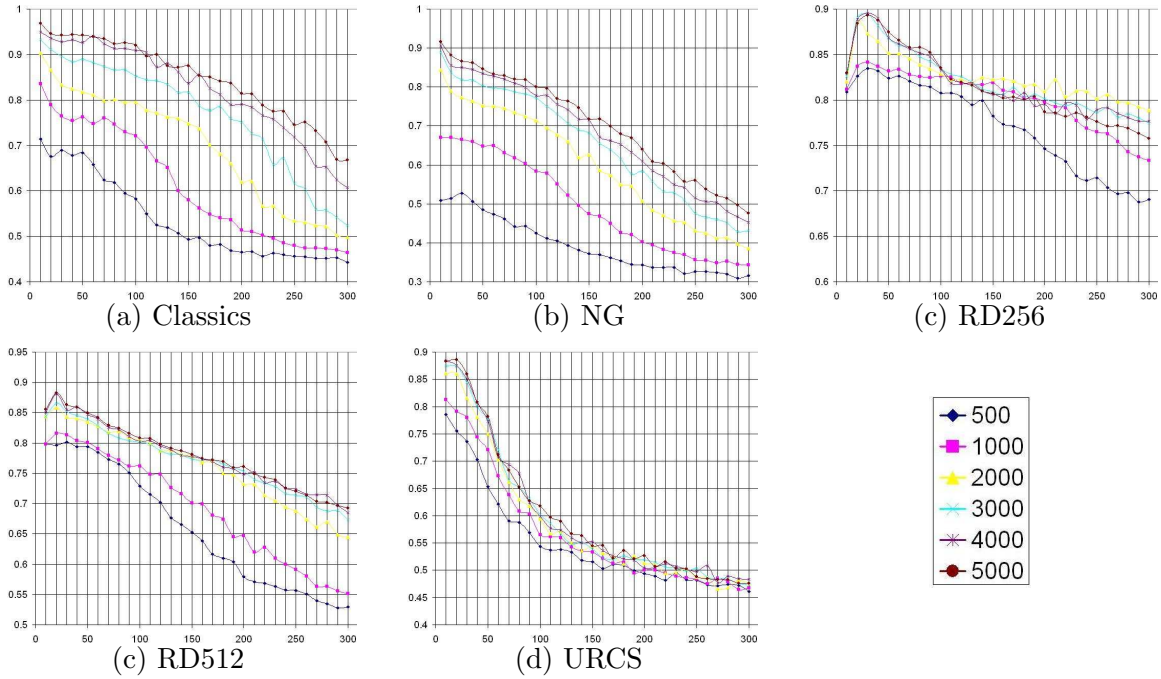


Figure 7: Impact of N-Gram profile length on clustering quality for 4Gram Representation when LSI Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

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

Table 5: Best profile lengths for 3-Grams and 4-Grams when one of listed dimensionality reduction methods is applied

test for comparing the means of two performance curves shows that 4-Gram representation is slightly better than 3-Gram representation.

Figure 10 shows the comparison between the best performance achieved with 3-Grams and 4-Grams with different profile lengths when LSI is used as dimensionality reduction method on all datasets. For this case, as it is clear from the figure, 4-Grams achieve better clustering performance compared to 3-Grams. For all datasets, 4-Gram representation seems to a better representation than 3-Gram representation when LSI LSI is used as dimensionality reduction method on all datasets.

Figure 11 shows the comparison between the best performance achieved with 3-Grams and

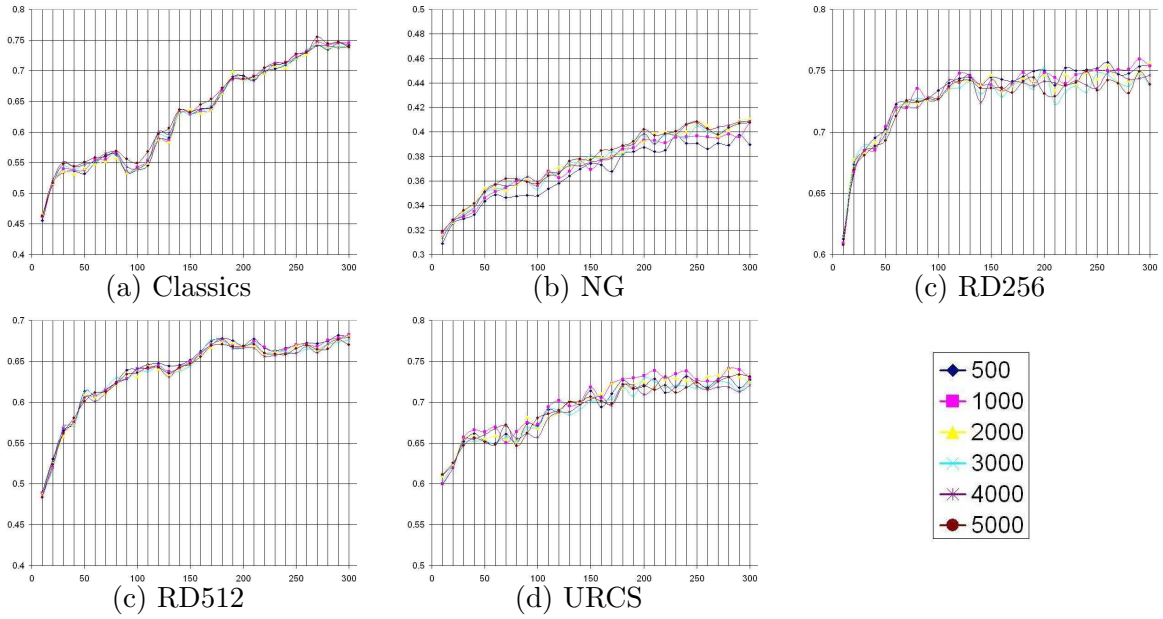


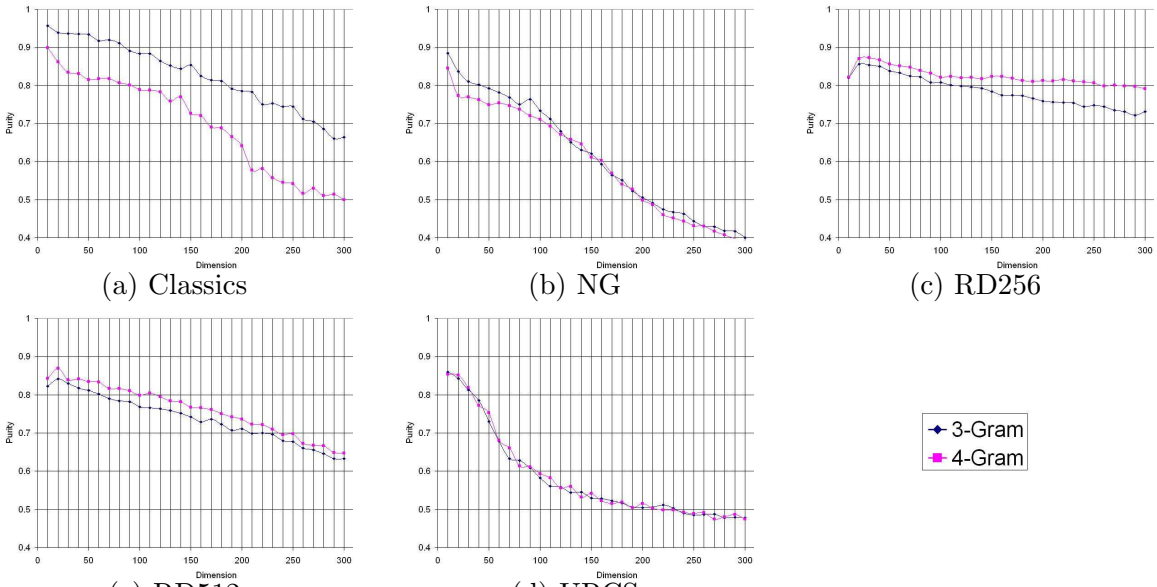
Figure 8: Impact of N-Gram profile length on clustering quality for 4Gram Representation when Document Frequency Based Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

4-Grams with different profile lengths when Document Frequency Based Method is used as dimensionality reduction algorithm on all datasets.

For Classic3 dataset, 3-Gram representation clearly achieves better clustering quality compared to 4-Gram representation. For NG dataset, 3-Gram representation is slightly better than 4-Gram representation in the whole range of dimensions investigated. In spite of Classic3 and NG datasets, for both versions of Reuters dataset, 4-Grams achieves better performance compared to 3-Gram representation. For URCS dataset, the difference between performance of these two representations is not clear like other datasets, but a t student test for comparing the means of two performance curves shows that 4-Gram representation is slightly better than 3-Gram representation.

4.5.3 Best N-Gram Parameters

In this section, we try to select a sufficient profile length and dimension reduction for 3-Gram and 4-Gram representation. Figure 12 shows that for almost all datasets ICA method achieves better clustering quality compared to LSI method. Only for NG dataset, for dimensionality less than 80, LSI seems to be better. But notice that for only this dataset, we compare profile lengths of 4000 and 2000 for LSI and ICA respectively. If we compare equal profile lengths for this representation, we will notice that in this case, the ICA method works better than LSI.



(a) Classics (b) NG (c) RD256

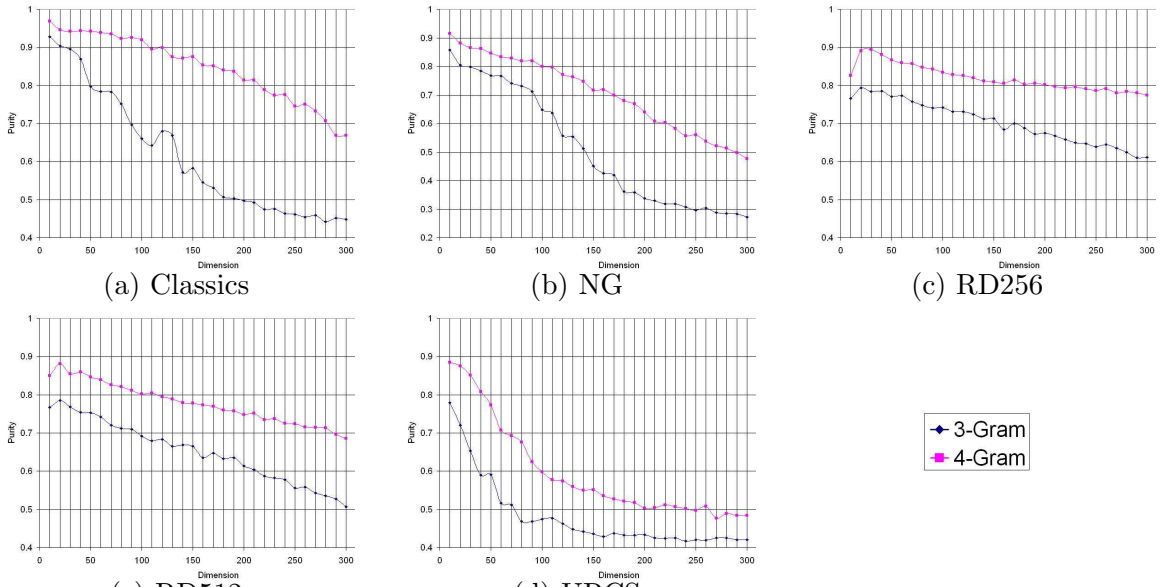
(c) RD512 (d) URCS

Figure 9: Comparing best clustering results for 3Gram and 4-Gram Representation when ICA Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

Document Frequency Based method achieves better clustering quality for dimensions in the second half of the investigated range of dimensions, but in the first half and for lower dimensions, this method seems to be worse than the other two methods. But it is interesting 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 Document Frequency Based method is not the best selection amongst these three method.

In spite of the results for 3-Gram representation, Figure 13 shows that for all datasets LSI method achieves better clustering quality compared to ICA method in the range of dimensionalities we are more interested. But with a precise 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, therefore 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.

Document Frequency Based method, like in 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 this representation compared to 3-Gram representation in the previous figure. Again, like in 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 bigger compared to 3-Gram representation. Since we are interested in lower dimensions, it seems that Docu-



(a) Classics (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 10: Comparing best clustering results for 3Gram and 4-Gram Representation when LSI Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

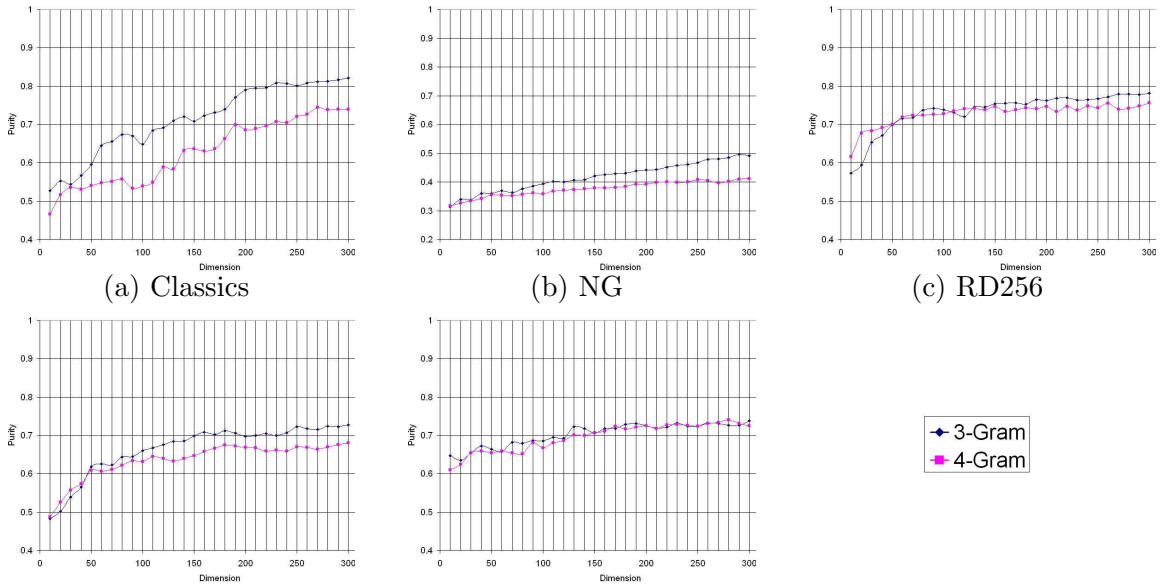
ment Frequency Based method is not the best selection amongst these three method.

These experiments shows that given equal profile lengths, for both 3-Gram and 4-Gram representation, ICA method achieves better results than LSI method. In 3-Gram case, this superiority is very clear but for 4-Gram representation, the difference is not large and the ICA method is very slightly better than LSI. The interesting point is for 3-Gram representation, even the result of 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 4-Gram representation, with choosing precisely good profile length for LSI method, we can expect that LSI method achieves better results compared to ICA method when mid-range profile length is used.

As a general result for all experiments done for N-Gram representation, it seems that for each of 3-Gram and 4-Gram representation, the best configuration is using ICA as dimensionality reduction method with mid-range profile length (around 2000). But choosing among 3-Gram and 4-Gram representation seems to depend on the dataset as we have seen each of these two representations achieved better results on some datasets compared to the another one.

4.6 Comparing DR Techniques

As it is clear in most experiments, the performance of Document Frequency Based method often reaches its maximum at some middle range dimensions (much higher than that of ICA/LSI's best dimensions), and then the performance remains somewhat the same as the number of dimensions increases. It is also interesting that the best performance of this



(a) Classics (b) NG (c) RD256

(c) RD512 (d) URCS

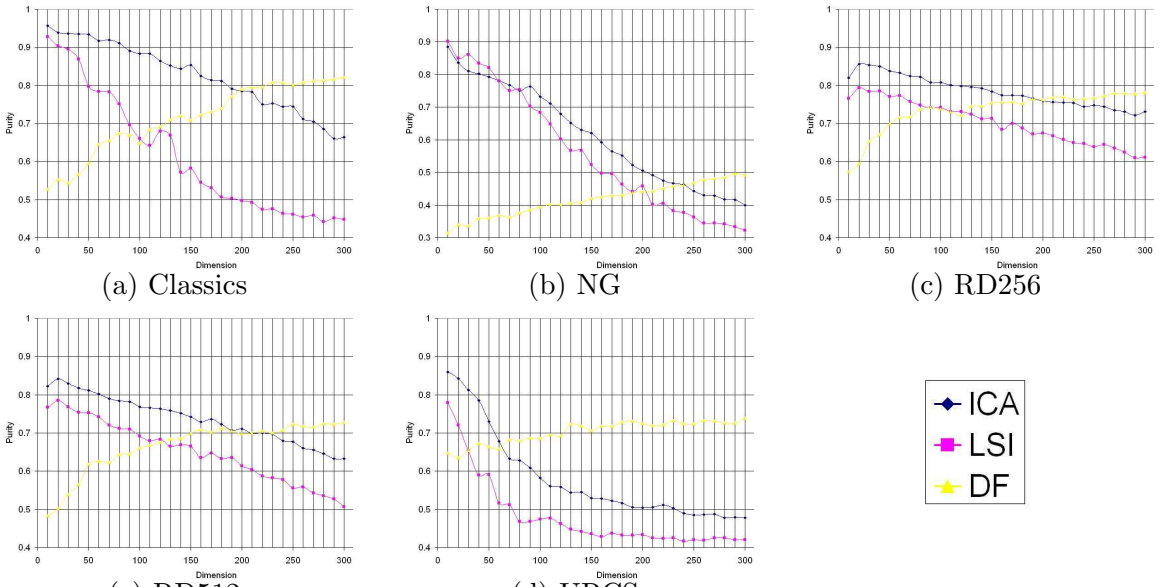
Figure 11: Comparing best clustering results for 3Gram and 4-Gram Representation when Document Frequency Based Dimension Reduction is used. x-axis represents dimensionality and y-axis represents purity value.

method at these middle range dimensions often is equal to the best performances of ICA/LSI which is achieved at much lower dimensions. This suggest that it might be possible to use Document Frequency Based method as a preprocessing task to pre-select some dimensions to be used for ICA/LSI instead of using the full set of dimensions. This can help especially when the original dimensionality is too high and very expensive to compute ICA or LSI. In the case of ICA, sometimes this is the only way because the input matrix for ICA in spite of the one for LSI is dense and so, sometimes it's impossible to do SVDS with memory limitations.

As results show, for all cases except for the 4-Grams, the performance of ICA is clearly better than LSI. Even in the case of 4-Grams, we compared the best performance of ICA with the best performance of LSI achieved for different profile lenghts. If we compare the performance of corresponding profile lengths for 4-Grams, we see that for equal profile lengths, in some cases, ICA achieves better performance than LSI. Therefore, in general, ICA seems to achieve clearly better performance than LSI for all three different representations we have investigated.

4.7 Comparing Text Representation Methods

In order to pick the best text representation method, one way is comparing their clustering results for every dimensionality reduction method we investigated separately. But for having a very explicit argument about their performance, for each of them, first we pick the corresponding dimensionality reduction method with which the text representation method



(a) Classics (b) NG (c) RD256

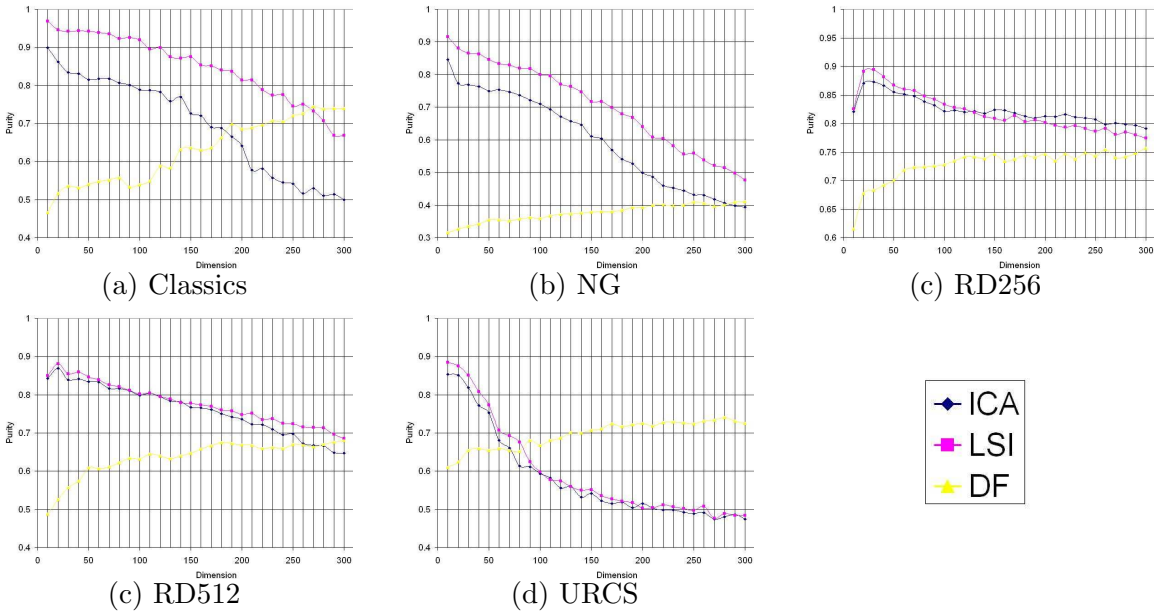
(c) RD512 (d) URCS

Figure 12: 3Gram Representation - Comparing Different Dimension Reduction Methods. x-axis represents dimensionality and y-axis represents purity value.

has achieved its best performance. For example, as we saw in the previous section, ICA is the best dimensionality reduction method for word representation and LSI is the best one for 4-Gram representation (Actually except for the 4-Gram representation, for all other representations ICA had the best performance). After having determined for each representation, its best appropriate dimensionality reduction method, we compare clustering quality of representation applying these best dimensionality reduction method.

Figure 14 shows the results of this comparison. It shows that Term representation has the worst clustering quality amongst four different representations.

Both 3-Gram and 4-Gram representations achieve very good and impressing results. But it's worth noticing that these clustering qualities are not actual clustering quality that using N-Gram representation may have. We achieved these results after a careful investigation of different parameter values (including N-Gram length and its corresponding profile length) for this representation and without this optimized parameters, N-Gram representation can have lower clustering qualities. Although we found some general patterns for initial appropriate values for these parameters. As experiments show for 3-Gram representation, profile length equal to 2000 is quite enough for getting approximately the highest clustering quality and for 4-Gram representation, this value is around 3000. We also observed that in most cases 4-Gram representation achieves better clustering quality than 3-Gram representation using these suggested values for profile length.



(a) Classics (b) NG (c) RD256

(c) RD512 (d) URCS

Figure 13: 4Gram Representation - Comparing Different Dimension Reduction Methods. x-axis represents dimensionality and y-axis represents purity value.

5. Conclusions and Future Work

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 mentioned dimension reduction methods on them to find out the best configuration of representation method and dimension reduction algorithm for text clustering.

From the experiment results, several general behaviors can be identified. In general, we can rank the three dimension reduction techniques in the order of $ICA > LSI > DF$. 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 the range of 10 to 100. The best performances of ICA/LSI seem correspond well with the transition zone of the singular value curve. The Document Frequency Based technique can get close to best performance of two other methods at very higher dimensions but at lower dimensions, its performance is much lower than two others.

Amongst three representation methods, traditional word representation seems to achieve better results in most cases and especially for lower dimensions. N-Gram representation can be considered as a replacement for word representation because its performance result is close to word representation. But it needs careful and precise determination of its two parameters which are N-Gram length and its profile length. If these parameters are selected in somehow carefully, then N-Gram representation performance can be very close

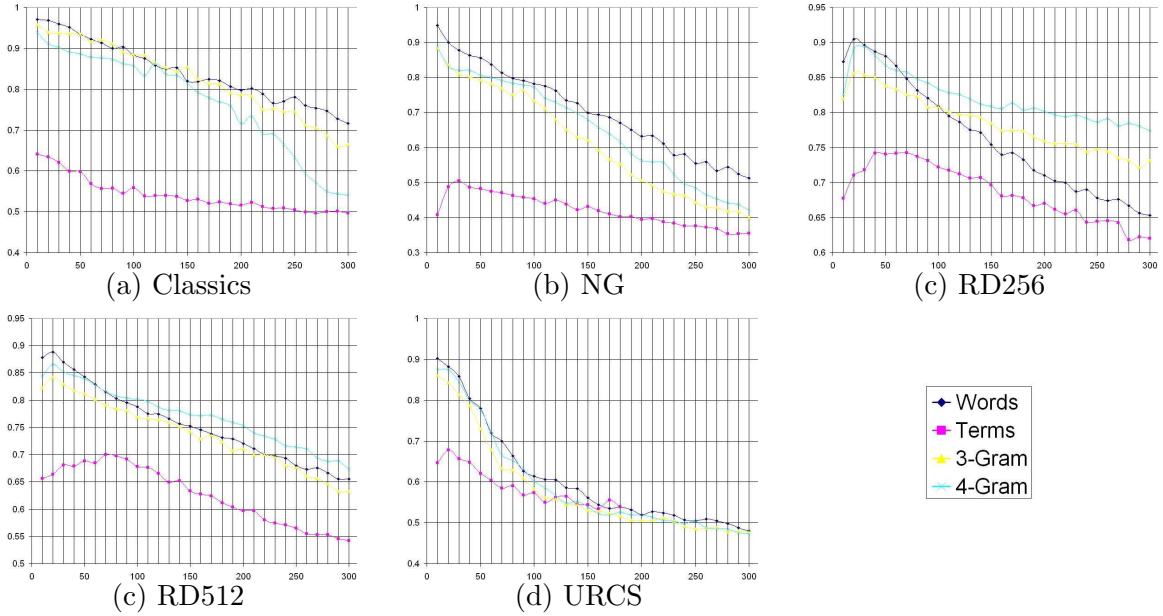


Figure 14: Comparing Different Text Representation Methods. x-axis represents dimensionality and y-axis represents purity value.

to word representation and for higher dimensions, even better than word representation performance.

Term representation performance is much worse than the two other representations. Even if we use some default parameters for N-Gram representation, its worst performance is still better than best performance of term representation.

Acknowledgments

We would like to acknowledge support for this project from

References

M. I. Heywood B. Tang, X. Luo 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.

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. ISBN 1-58113-888-9.

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

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

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

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.

N. Zincir-Heywood E. Milios Y. Zhang. 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.

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.

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.

Parry Husbands, Horst Simon, and Chris H. Q. Ding. On the use of the singular value decomposition for text retrieval. pages 145–156, 2001.

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

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

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

Vlado Keselj. Perl package text::ngrams, 2004.

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.

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.

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.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988. ISSN 0306-4573.

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.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002. ISSN 0360-0300.

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. ISBN 0-89791-605-0.

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.

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.