

**APPLICATION OF UNSUPERVISED LEARNING
METHODS TO AN AUTHOR SEPARATION TASK**

By

Kevin Barlett

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
Major Subject: COMPUTER SCIENCE

Approved:

Boleslaw K. Szymanski, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

July 2008
(For Graduation August 2008)

CONTENTS

LIST OF TABLES	iii
LIST OF FIGURES	iv
Abstract	v
1. Introduction	1
1.1 Introduction to Unsupervised Learning	1
1.1.1 Data Mining and Text Mining	2
1.2 Historical Perspective	3
1.3 Unsupervised Learning as Applied to Text Mining	4
1.3.1 Feature Extraction	4
1.3.2 Feature Selection	4
1.3.3 Clustering	4
1.3.4 Cluster Evaluation	5
1.4 Author Separation Task	5
2. Methodology	7
2.1 Feature Extraction	7
2.1.1 Unigram	8
2.1.2 RDM	9
2.2 Feature Selection	10
2.2.1 Information Gain	10
2.2.2 Gain Ratio	11
2.2.3 Principle Component Analysis	12
2.3 Clustering	13
2.3.1 Expectation Maximization	13
2.3.2 KMeans	13
2.4 Cluster Evaluation	14
2.4.1 Scatter Separability	14
2.4.2 Silhouette Coefficient	14
2.5 Experimental Specifics	15

3. Results and Discussion	19
3.1 Feature Generation, Feature Selection, and Clustering	19
3.2 Cluster Evaluation	22
4. Conclusion and Future Work	25
Bibliography	
APPENDICES	
A. Literary Collection Details	28
B. Raw Results	29

LIST OF TABLES

3.1	Correlation between Cluster Evaluation and Accuracy	22
-----	---	----

LIST OF FIGURES

2.1	Experimentation Flow Chart	16
3.1	Overall Error per Method	19
	(a) Unigram - Overall Error per Method	19
	(b) RDM - Overall Error per Method	19
3.2	Error per Method	21
	(a) Unigram - Error per Method	21
	(b) RDM - Error per Method	21
3.3	Best Results per Dataset, per Method	22
	(a) Unigram - Best Results per Dataset, per Method	22
	(b) RDM - Best Results per Dataset, per Method	22
3.4	Cluster Error and Cluster Evaluation	23

Abstract

As the amount of information stored as text grows, being able to efficiently organize and sort the available information becomes crucial. Supervised learning methods provide a solution for some of these tasks, however they rely upon human interaction for the initial classification of texts. Unsupervised learning methods allow not only for a solution to some of these tasks, but by removing the human component increases the speed and efficiency of these methods, thereby decreasing the barriers for application of these methods. This work explores the similar structure taken by unsupervised learning methods when applied to text mining problems, followed by a brief overview of the four components: feature extraction, feature selection, clustering, and cluster evaluation. This framework is then applied to a problem involving author separation, where many excerpts of literary works are presented with the task of dividing the excerpts into groupings corresponding with individual authors. The applicability of various learning methods are then considered based upon their relative performance on the given task.

1. Introduction

1.1 Introduction to Unsupervised Learning

Learning is a focus of school children everywhere and comprises the search for a relationship between objects and groups of those objects. A study of these relationships results in the ability to accurately predict the properties of previously unobserved objects, allowing accurate conclusions to be made on limited information. Learning can occur in different environments based upon the properties and objects involved, and which properties are to be predicted for the objects encountered later.

It is generally the case that the set of properties to be predicted, the *class properties*, will be small in number when compared to the properties readily observable in the objects, the *training properties*. The set of objects presented to be learned from is the *training set*, and an object that we will wish to predict the class properties for is taken from the *testing set*.

If the class properties are a subset of the training properties then the learning task is a *Supervised* learning problem. Consider a children's game of determining the name for simple two dimensional figures. First, the child is shown many figures with their appropriate labels (e.g. circle, square, triangle). After all of the examples are presented, a figure is shown that the child has not seen before and she is asked to predict its label. In this case the example figures comprise the training set, and the class property will be the name of the shape present in the figure. The figures presented may differ in more properties than just the shape; for example the size might change, along with the color and orientation. These observable aspects make up the training properties. As the shape is defined for each training figure the child creates an appropriate mapping from figure to shape name, and then tries to apply that mapping when a testing object is presented. Confirmation of accuracy in predictions imply correctness of the mapping learned.

In some cases only a subset of the training examples have known values corresponding with the class properties, even though class properties are a subset of the

training properties. Expanding upon the previous example, these additions would be figures shown to the child without the name of the shape made known. The child gains some information about figures in general and can refine the mapping of figure to shape name based upon this fuller understanding of what a figure is. This additional information is helpful, however, not as much information is gained compared with knowing the name of the shape in every figure. This environment is considered as *Semi-Supervised*.

If there are no known values for the class properties in the training set then the environment is referred to as *Unsupervised*. In our example, determining the name for the shape corresponding to a figure would be all but impossible as the child wouldn't know the names of the valid shapes. Instead, it is best to formulate a game of grouping the figures together based upon their similarities. The child would determine which of the training properties, or combination thereof, correspond closest to the class properties, the idea known as 'shape' in this case, and group the figures accordingly. Note the difficulty faced by the child in determining which of the properties to consider and which to ignore. The child may cluster based solely upon color, and while this clustering is valid, it does not correspond with a notion of "shape." It may take many different attempts before the child begins to gain an understanding of what groupings correspond closely with the concept of "shape."

1.1.1 Data Mining and Text Mining

Data Mining refers to the application of specific methods of learning to real world problems. Consider predicting the stock market [12] as an example. Learning techniques are utilized to discover the trends in the historical data that may prove useful when attempting to predict the changes in the stock market in the coming weeks and months. Similarly, if we are attempting to forecast the crop yields in a specific region of the world [9] there will be some learning techniques that will be beneficial when attempting to determine the overall yield for the season based upon assessing the impact of rainfall, insect activity, temperature, etc.

Understanding the human language is a complex task, as shown by the intricacies of natural language research [14] [8]. It is possible, however, to utilize

observable qualities of a given text to perform some predictive tasks without having to understand the complete meaning of the text in question. If we were interested in determining whether a scientific article was published in a medical journal or an astronomy journal, it might be possible (given enough articles from medical and astronomy journals) to determine a set of words used only in medical contexts and a set of words used only in astronomy contexts, and then look for those words in the article in question. This would allow for the accurate prediction of the corresponding journal without requiring the computer to understand the content of the article in question. Methods useful in producing predictive models based upon inputs of text fall into the area of *Text Mining*.

1.2 Historical Perspective

Unsupervised learning methods when applied to text mining contexts have provided a great number of results. The grouping of text that result from the application of these methods revolve around observed patterns which allow for tasks such as determination of common thematic elements [2], ontology extraction [18], or text summarization [17].

The amount of information located in text is growing rapidly[4], due in large part to computers, digitization processes [16], and the World Wide Web. The ability to utilize automated techniques to distill limited information from the vast quantities of available text allow humans to have more information regarding the texts that may be useful when prioritizing the information [5] or extracting new facts [19]. In this environment, the unsupervised learning method is vital; it removes any need for collections of data to be tagged before the techniques can be applied. Without these methods, tagging becomes a costly task which relies on direct human interaction and judgments. With large amounts of raw information available, unsupervised techniques allow for the immediate processing of this information leading to results with minimal human interaction.

1.3 Unsupervised Learning as Applied to Text Mining

The process of unsupervised learning, especially in text mining applications, can be separated out into four distinct pieces that every unsupervised text mining application provides some mechanism to handle. These are feature extraction, feature selection, clustering, and cluster evaluation.

1.3.1 Feature Extraction

In text mining applications the raw text is provided, however, there is no meaningful way to group and cluster upon solely raw textual input. To link these two components *features*, quantitative properties of the text, need to be extracted. It is crucial when extracting features to find the ones that will be most useful in determining the class properties. For instance, the counts associated with the frequency of each letter may be extracted as features, while these might be of use in determining the language that is present, the features will probably not suffice when determining if the work in question was written before a specific date. Determining which features to extract from the text plays a crucial role in the overall success of the work.

1.3.2 Feature Selection

Once features have been extracted, it may be discovered that while in certain contexts the features are useful, in a specific text mining environment they are not. In this case, eliminating these features from our feature list will only simplify the clustering procedure. For instance, if we know there are two values for the class attribute present, meaning an accurate representation would result in two distinct clusters, and a feature exists that is the same for all input texts, then this uniform feature will not be of use when clustering is performed due to its lack of variation. This feature provides no new information so it can be safely discarded and not considered in the rest of the process.

1.3.3 Clustering

Once the set of important features has been selected, clustering is needed to be performed on the objects and their related features, grouping them into what

appears to be best combinations of objects. The cluster assignment that results will be able to map any data point to a cluster of those points most similar. These clusters may not have meaning to the outside observer, there is no inherent statement about the objects in a particular cluster other than they have more in common with the objects in their own cluster than those in different ones.

1.3.4 Cluster Evaluation

It is likely that many different clusterings are possible. For instance, if it is not clear how many clusters are appropriate, then different clusterings can be constructed with a different number of clusters present. Determining which among these many different clusterings is most accurate is achieved by using a cluster evaluation metric. When applied to a clustering this metric results in a numerical representation of the “goodness” of the particular clustering. These metrics should be robust enough to allow for the evaluation of many clusterings, followed by a comparison of the resulting metric values. This comparison allows for a ranking to be associated with each clustering, and then the selection of which clustering represents the most accurate representation.

1.4 Author Separation Task

In this work, the problem to be considered is the accurate author determination of sections of text taken from collections of literary works. These collections of literary works are taken from a pair of authors, the works are then separated into different sections of a few paragraphs, with the goal of accurately separating those excerpts of text into two distinct clusters, each corresponding to a different author.

This is similar to taking novels written by different authors, ripping off the bindings, and tossing the pages into the air. When the pages coalesce on the ground there is no distinction between the pages associated with one author or the other. The task is then to separate the pages into groups, each corresponding to an individual author. Just as unsupervised learning applied to the children’s game discussed earlier, there is no way to determine who the actual authors are (that would require a supervised or semi-supervised environment with other works written by the

authors), so the goal is to differentiate between the authors with as few errors as possible.

2. Methodology

To apply unsupervised text mining in this context, each of the four components must be considered and possible methods corresponding to each must be decided upon. Once specific values for each of the components is established then the details related to this specific inquiry are explored.

2.1 Feature Extraction

Feature Extraction is tasked with taking the inputted text and creating features that the clustering methods can handle. Clustering methods do not generally know how to cluster non-quantitative data, and so we wish to convert the various texts into a collection of numerical attributes which are derived from the written texts. These methods may be as simple as counting the frequency of a letter, in which case an attribute would correspond to the letter 'a' and its corresponding value would be the number of occurrences of the letter 'a' in the given text. A method could also be something as complex as an approximation of the author's age based upon sentence length and syllabic variation, as long as that approximation is based solely upon features of the text and the resulting value can be recorded as a numerical quantity, then it is a valid feature.

A common tool used in some feature extraction methods that focus on words as potential features is stemming. *Stemming* [10] corresponds with the practice of attempting to remove word endings and other grammatical requirements in order to limit the number of different words observed. For instance, a passage might contain the following simple sentences:

1. I run before work every morning.
2. Sally prefers to go running through this park.
3. Alice runs with friends on the weekend.

Each of these sentences contains a word which is closely associated with the verb "run." If a human were to read these sentences then the fact that words dealing

with running occur in each will be immediately obvious, however when a computer looks at the sentences since each occurrence has different spellings (“run” is different from “running” and “runs”) it will not discover this similarity. Instead, stemming removes this somewhat extraneous information, converting the sentences to:

1. I run befor work everi morn.
2. Salli prefer to go run through thi park.
3. Alice run with friend on the weekend.

Stemming allows for fewer words to be observed in the text, simply “run” instead of “run”, “running” and “runs” as previously observed. This benefit comes at a cost, however, as the information conveyed by the endings has been discarded and will not be a component of any feature based upon the stemmed text.

2.1.1 Unigram

The unigram model works by considering each word (or word stem) individually and counting the number of times each word is observed in a given text. The feature extraction method is solely concerned with word frequency and while that may not seem very powerful, in many problems it is sufficient for learning tasks. Consider the example provided earlier of determining which type of journal a particular article was published in. In that case, while there may be many common words that appear in both works, simultaneously there will be words that are associated only with each individual journal. Therefore a classification can be made based upon the presence, or lack thereof, of certain words.

This concept need not remain at the single word level. An approach similar to this is the bigram model, which focuses on the number of occurrences of each pair of words and its number of occurrences, instead of focusing on each single word. Take for instance the phrase “Data Mining”, both words have meaning on their own and may occur separately quite often depending upon the text corpus. When the words are used together as “Data Mining”, this conveys something more than just what a unigram model sees. A unigram model will observe one more occurrence of two different words, where a bigram model would see the phrase “Data Mining”

as occurring more often than the word “Data” followed by some other word. This increased expressive nature of bigram comes at a significant cost as there are many more pairs of words that will be observed in the text than just words themselves, and so will take more resources to compute the bigram representation of texts.

Similarly, the unigram and bigram concepts can be extended to n -grams, where n corresponds with the number of words that are considered for each resulting attribute. Just as bigram was more expressive than unigram, so is n -grams for larger values of n potentially more expressive than bigram, again at a further cost in resources and complexity.

2.1.2 RDM

Recursive Data Mining (RDM) [3] attempts to record not only the words and related frequencies, but also patterns and structure to be found in the text itself. This technique differs from a generic n -gram representation in that the size of patterns and structure need not be determined beforehand and are adjusted as the text is processed automatically.

RDM initially receives sequences of *tokens*, generic entities, which are then processed. The meaning associated with each token is unimportant in RDM, and a token can be generated from text in many different ways. When in a text mining environment, tokens may be constructed from individual letters, syllables, words, or even larger selections.

Once the tokens have been created they are presented to RDM to be processed for sequences of tokens that occur frequently. Those sequences of tokens are then individually assigned new token identifiers, all occurrences of the sequence are replaced with the corresponding new token, and then this process is repeated until no sequences are found that are considered frequent. In this way, sequences can be built up out of other smaller sequences, which allow for a hierarchical model of the text, and the tokens are exported as the features with their related frequencies being computed for each text in question.

The calculation of the frequency of a given sequence of tokens in a given token series may also be expanded to include minor discrepancies (referred to as “gaps” in

the RDM literature), allowing for distinct, yet essentially similar, sequences to be considered the same sequence. This allows for a reduction in the number of overall sequences and also the sequences that remain are more numerous and thus have a better chance of crossing the frequency threshold to be included in other potential token sequences.

2.2 Feature Selection

Once features have been created to represent the text in question the method can immediately begin clustering, however this course of action is not always beneficial. The clustering algorithms treat each attribute that it is presented as equally important since there is no means to differentiate between the information represented by each attribute. In practice, however, this evenhandedness isn't always beneficial. Feature Selection is therefore responsible for reducing the impact of, or removing entirely, the features that contribute little meaningful information to the clustering process.

Consider the comparison of two texts where features have been extracted using a unigram model with a focus placed on just two attributes, one corresponding to the number of occurrences of “the”, and the number of occurrences of “galaxy.” If it were the case that the first document had 15 occurrences of “the” and 5 occurrences of “galaxy” and the second document had 5 occurrences of “the” and 15 occurrences of “galaxy” then this would be useful information. If these two attributes were then presented to a clusterer then they will be weighted equally; however, given an understanding of English, a large variation between the number of “the”'s present may not be significant where as a large variation of the word “galaxy” may be more significant. In this case, it is important to diminish the impact of the variation corresponding to the word “the” as spurious and inconsequential. It is this sifting that feature selection is tasked to accomplish.

2.2.1 Information Gain

Information gain provides a ranking of the information held by each of the attributes being considered. This ranking is done by computing the *entropy*, or

information, that each attribute provides, and those attributes contributing little to no information are disregarded as unimportant. The information gain is defined as

$$IG(Class, Attribute) = H(Class) - H(Class|Attribute)$$

where $H()$ corresponds to the entropy, or

$$H(X) = - \sum_{x \in X} Pr(x) \ln Pr(x)$$

. Once the information gain has been computed for each attribute, a threshold can be determined so that everything exceeding the threshold is selected for presentation to the clusterer, and everything else is ignored.

In a supervised setting the information gain can be computed quite easily, since a set of inputs and classes are provided. However in the unsupervised setting, this distinction will not be possible because the objects present in a particular *Class* are unknown. It still is useful to consider it in the context of unsupervised learning, however, as it results in a set of attributes that might be optimal for clustering. The accuracy of clustering based upon information gain represents a point of comparison. If this accuracy is found to be optimal, then other unsupervised feature selection methods, which can be applied in the unsupervised learning method, would be considered in hopes of finding an approximation of information gain without relying upon the class entropy.

2.2.2 Gain Ratio

Information gain is useful for determining which attributes provide more information about classes than others, however there are cases where information gain can be biased towards certain attributes [20]. If the attribute has a high entropy value then there are many different values associated with it, and the information gain measure will tend to favor that attribute, since large variations tends to correspond with large entropy values. This favoritism can be harmful, so to counteract

this bias the gain ratio corresponding to an attribute is computed as:

$$GR(Class, Attribute) = \frac{IG(Class, Attribute)}{H(Attribute)} = \frac{H(Class) - H(Class|Attribute)}{H(Attribute)}$$

In this way the entropy of the attribute inversely affects the gain ratio. Similar to information gain, a threshold is then determined to decide which attributes are to be passed to the clusterer and which aren't.

Since information gain is a component of the gain ratio measure, the same impossibility of using gain ratio in an unsupervised setting exists. While the method itself is not applicable in this case, its evaluation provides information about the performance of different attribute subsets that may be useful in determining which clusterings are accurate.

2.2.3 Principle Component Analysis

Principle Component Analysis (PCA) [7] operates by transforming the features that were selected into a different set of attributes entirely, unlike the attribute rankings provided by information gain or gain ratio. In PCA, a new feature corresponds with finding the vector corresponding to the largest variance in the data. The vector corresponding to the largest variance will not necessarily correspond perfectly with a single feature, but may instead correspond with a combination of different features. This vector is then considered a new attribute to be used by the clusterer and each data instance has a value associated with this new attribute determined by its projection onto the vector. The next attribute resulting from PCA is created by selecting a vector orthogonal to the previously chosen one that corresponds to the largest remaining variance, and then each data instance is projected onto this new vector. This process is then repeated, selecting a new vector orthogonal to all vectors selected previously and projecting the data instances down to determine their value, resulting in a set of new features and their related values.

Unlike the previous feature selection methods, PCA is an unsupervised method for feature selection as no class information is required. Therefore application of this method in the current unsupervised setting requires no modification whatsoever.

2.3 Clustering

Clustering is at the center of the unsupervised learning process. Up until now the focus of each component has been to produce the best selection of features to present to the clustering method. After this, the only concern involves choosing the clustering that is deemed “best”. To fulfill the role, a clusterer will take in the dataset comprised of the attributes produced by the feature extraction and selection components and then return the dataset divided up into some n clusters, where each point must be assigned to exactly one cluster and all the points in a cluster are more similar to the other points in the cluster than those all of the points of any other cluster.

2.3.1 Expectation Maximization

Expectation Maximization (EM) [15] works by attempting to determine which assignment of points to clusters is most probable by varying a probability distribution and its parameters. In this case each point has a corresponding probability distribution describing the probability of belonging to each of the n possible clusters, and then once a distribution has been settled upon assigning the point to the cluster which corresponds to the largest probability.

The distribution is settled upon through the convergence of an iterative process which occurs in two distinct steps. The first step attempts to find the best distribution while keeping the parameters static, while the second step then attempts to optimize with respect to the parameters while the distribution remains unchanged.

2.3.2 KMeans

KMeans [13] is also an iterative clustering method. To form n clusters, KMeans selects n arbitrary points, referred to as centroids, that correspond to the center of potential clusters. Each data point is then assigned to the cluster of the centroid it is closest to, according to some distance measure, which creates n clusters. The center of each of those n clusters is then computed and these centers are considered the new centroids, and again new clusters are defined according to these centroids, their membership is computed, and so on. This process continues until cluster

membership converges, and the resulting n clusters are returned.

2.4 Cluster Evaluation

Given different combinations of feature extraction, selection, and clustering methods, different cluster assignments will be created. It is the goal of unsupervised learning methods, however, to present only the single best clustering corresponding closest to the actual clustering. An evaluation criteria will be used to determine which cluster would make the best selection and is most likely to be closest to the actual clustering.

2.4.1 Scatter Separability

Scatter Separability [6] works by combining both the compactness of each cluster and the distance between clusters. Scatter separability assumes a good clustering, will be one where all of the points in a given cluster are close together while the clusters themselves are far apart. This makes intuitive sense, since we want the clusters to be oriented so that each point is closely related to the other points in its own cluster and radically different from the points in the other clusters.

The scatter separability measure is computed by first determining the intra-cluster scatter matrix, representing the scattering of the points within each cluster, and the inter-cluster scatter matrix, representing the scattering between the different clusters. Then to determine the scatter separability the inverse of the intra-cluster scatter matrix is combined with the inter-cluster scatter matrix, since we wish to favor small scatter matrices within clusters and large scatter matrices outside of clusters.

2.4.2 Silhouette Coefficient

The Silhouette Coefficient [11] represents a similar notion of good clusters as scatter separability, individual clusters should be compact however different clusters should be far apart, but measures it with a different mechanism. The silhouette coefficient of a point is computed by first calculating the average distance between that point and all other points in its cluster (let this be a), and then finding a point

that lies in another cluster that is closest to the point in question, and compute that distance (let this be b). The silhouette coefficient is then defined as the

$$SC = \frac{b - a}{\max(a, b)}$$

. With this metric we wish a to be as close to zero as possible, leading the cluster to be all on one point and therefore have little internal variation making predictions based upon the clustering accurate.

2.5 Experimental Specifics

Algorithm 1 Pseudocode of Experimental Procedure

Require: LW contains the Literary Works

for pagesize in [3,5,7] **do**

 dataset \leftarrow CREATE DATASET(pagesize)

for FEATUREEXTRACTOR in [UNIGRAM, RDM] **do**

 features_all \leftarrow FEATUREEXTRACTOR(dataset)

for FEATURESELECTOR in [INFORMATION GAIN, GAIN RATIO, PCA] **do**

 features_selected \leftarrow FEATURESELECTOR(features_all)

for CLUSTERER in [EM, KMEANS] **do**

 clustering \leftarrow CLUSTERER(features_selected)

 accuracy \leftarrow COMPUTEACCURACY(clustering, dataset)

 {Record accuracy as a function of pagesize, FEATUREEXTRACTOR, FEATURESELECTOR, and CLUSTERER}

for CLUSTEREVALUATOR in [SCATTER SEPERABILITY, SILHOUETTE COEFFICIENT] **do**

 eval \leftarrow CLUSTEREVALUATOR(clustering)

 {Record eval as a function of pagesize, FEATUREEXTRACTOR, FEATURESELECTOR, CLUSTERER, and CLUSTEREVALUATOR}

end for

end for

end for

end for

end for

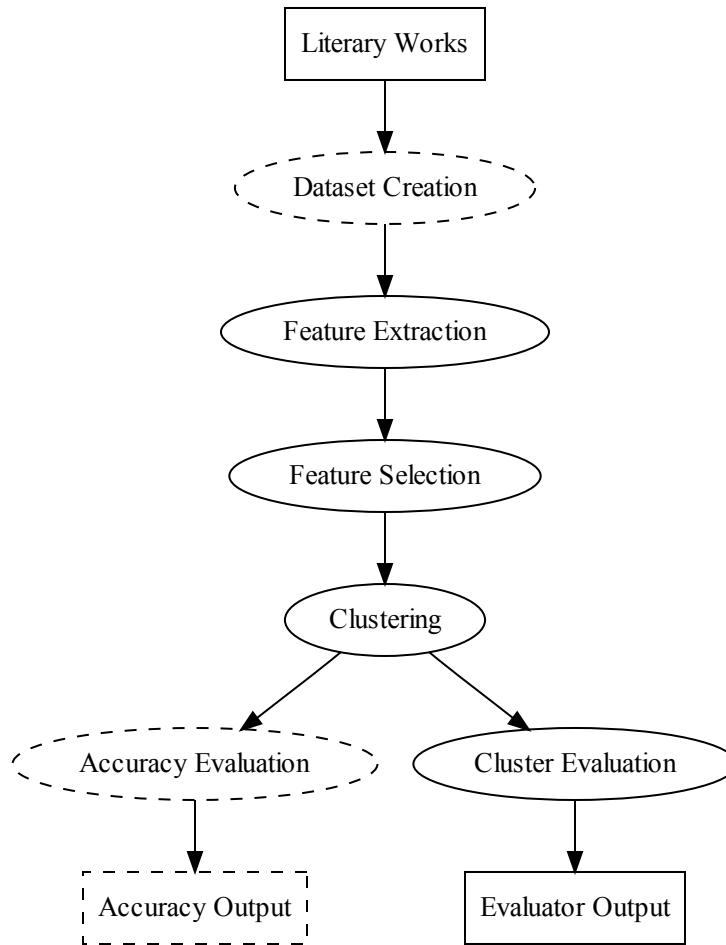


Figure 2.1: Experimentation Flow Chart

An overview of the experimental procedure can be found in Figure 2.1 with pseudocode available in Algorithm 1. The input to the entire system consists of literary works which then proceed through some preprocessing, followed by the different phases of unsupervised learning, with the end result being two quantities: the accuracy the clustering actually obtained, and the evaluator’s approximation of the relative accuracy of the clustering. If the datasets weren’t being generated from the literary collections as part of the experiment, then the **Accuracy Output** would not be obtainable, however it will always be possible to determine the **Evaluator Output** given a successful execution of the steps associated with unsupervised

learning. We now explore the specifics related to applying the previously discussed methods and metrics in this environment.

Datasets were created by taking collections of works from Project Gutenberg [1]. A detailed list of authors and works comprising each of the four collections can be found in the appendix. A literary work is prepared by first removing the non-literary components (license, credits to the transcribers, etc), and then separating the work into sections meant to approximate a page. As an appropriate length corresponding to a page is not an easily definable quantity, the sizes of three, five, and seven paragraphs were used as an approximation. Each dataset consists of the page approximations of a total of six works, with three works each being written by each of the two separate individuals.

Each dataset is then passed through the feature extraction phase which is provided by a modified version of RDM code. Unigram feature generation corresponds with the output of RDM before sequence replacement has taken place, and the output of RDM after sequence replacement has occurred yields the output of RDM in our experiments. RDM was modified to allow for the processing of unsupervised data by allowing for a single input class and removing the implicit requirement of a testing set. The RDM code was executed with a gap allowance of 0.25, lambda of 1, a window size of 3, and a prune level of 0.1 (See the RDM literature for a detailed description of these variables).

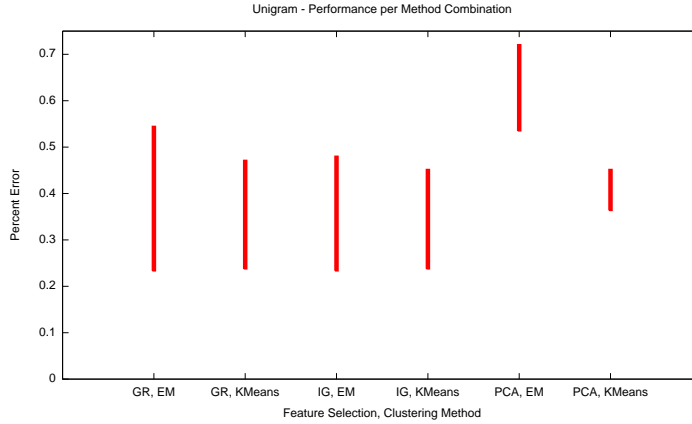
In order to limit the dimensionality of the clustering task it was assumed that preferred features would occur in a significant majority of one class' excerpts and in a small minority of the other's excerpts. If such a feature could be found, then that feature would serve as a good test for the two classes; therefore a bias is placed on finding these sorts of features over others. Due to the similar number of each author's excerpts included in a dataset, only those features present in between 25% and 75% of the excerpts are passed to the feature selection component. If a feature appears in more than 75% of the excerpts, then it must appear in a majority of both classes, making it relatively useless for author determination, similarly if a feature appears in less than 25% of the excerpts then it appears in a minority of excerpts, and determining which class that particular feature indicates will not be trivial.

The output of the feature generation was then passed to the feature selection code, which in turn was then used as the input for the clustering procedures. The implementations included in Weka [20] were used along with the default options for both of these steps. The Cluster Evaluation was accomplished through additions to Weka for each evaluator required, Weka's internal structure allowed for the rapid creation of these capabilities.

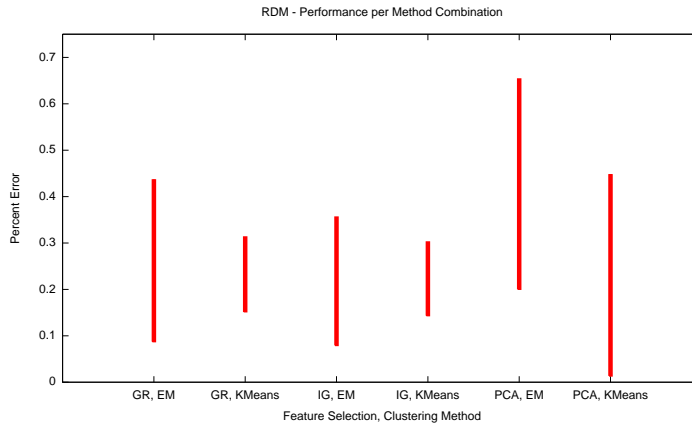
This entire process represents the evaluation of a single combination of available Feature Generation, Feature Selection, Clustering, and Cluster Evaluation methods. To accomplish a complete consideration of all available choices for each method, 12 distinct accuracy outputs for each dataset are required. Given that there are four literary collections, each with three distinct datasets, there will be a total of 144 distinct runs of the experiment required to obtain the accuracy results.

3. Results and Discussion

3.1 Feature Generation, Feature Selection, and Clustering



(a) Unigram - Overall Error per Method



(b) RDM - Overall Error per Method

Figure 3.1: Overall Error per Method

Figure 3.1(a) records the maximum and minimum accuracy values corresponding to the different combinations for feature selection and clustering after using the Unigram feature generation method. While this gives no information regarding distribution, it does make clear that no combination achieved an error rate less than 20%. It can also be observed that the combination of PCA and EM performs significantly worse than the other tested combinations.

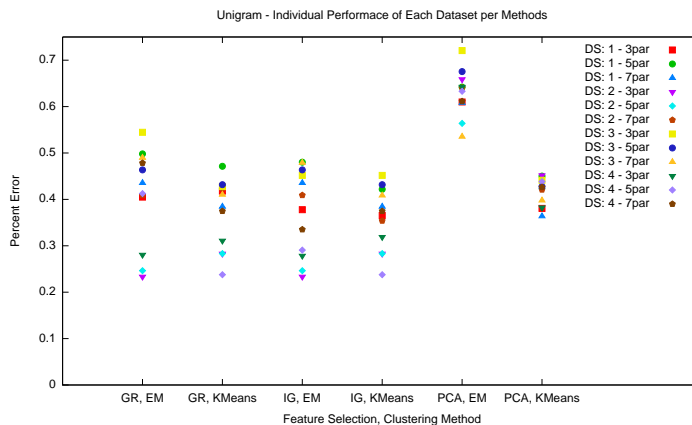
Figure 3.1(b) displays similar measures when the feature generation method is RDM. Here three different combinations break below 10% error, with one achieving less than 5% error on at least one dataset. These are significantly smaller error rates compared to the error rates produced with the use of unigram feature generation. Only the combination of PCA and KMeans was observed to perform worse under RDM instead of Unigram, the remaining combinations have smaller maximum errors associated with RDM as compared with Unigram.

The combination of PCA and KMeans achieving under 5% error is significant; however the large variance associated with that combination suggests caution is warranted. The possibility that the 5% error rate may be associated with a single outlier and the remaining datasets could have a corresponding error rate of approximately 45% demands a closer analysis of the error rates of each individual dataset.

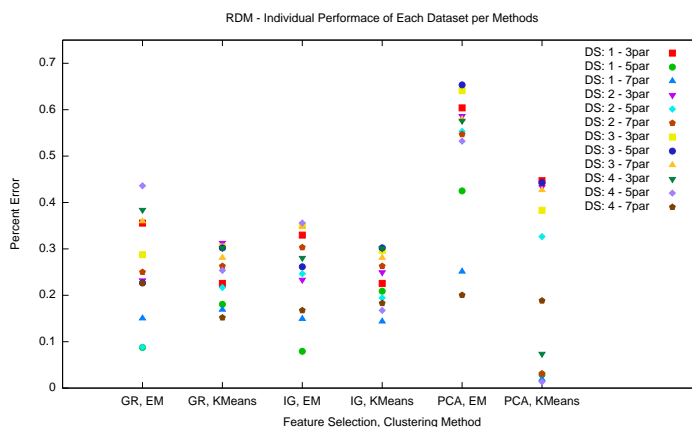
In Figures 3.2(a), 3.2(b) we observe the same data as before with each dataset's accuracy plotted individually, instead of solely the minimum and maximum. This allow for distributions to be observed corresponding to each feature extraction, feature selection and clustering combination. In 3.2(b), the three combinations discussed previously with sub-10% error rates can now be examined individually, and it is observed that two datasets achieve sub-10% with Gain Ratio and EM (Datasets 1-5par, and 2-5par) and only one dataset achieves that level under Information Gain and EM (Dataset 1-5par), while six datasets achieve sub-10% error with PCA and KMeans. The accuracy and the distribution associated with the RDM, PCA, KMeans combination is achieving a reliably small error rate.

The previous count of six datasets achieving a sub-10% error rate with PCA and KMeans may be misleading because it is difficult to determine which datasets are present and which are not. To counteract this Figures 3.3(a), 3.3(b) display the dataset from each collection of literary works that achieves the lowest error rate, for each feature generation, feature selection, and clustering combination. This allows for the consideration of the best case scenario under each method combination.

The observed trend continues in relation to RDM, PCA, and KMeans, as three of the four literary collections have datasets achieving sub-10% error rates. When compared to two out of four corresponding to Gain Ratio and EM and one of the four



(a) Unigram - Error per Method

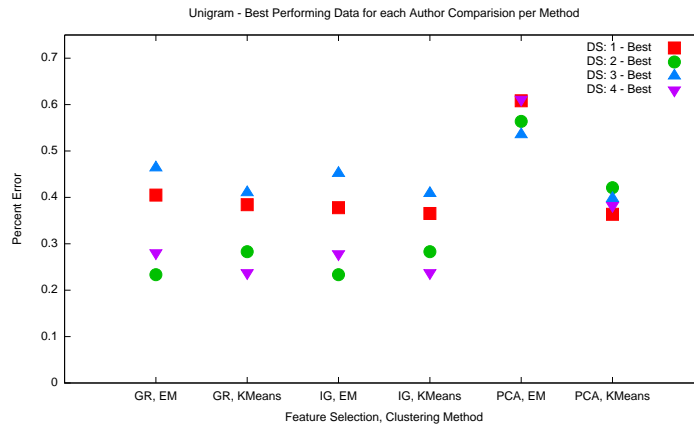


(b) RDM - Error per Method

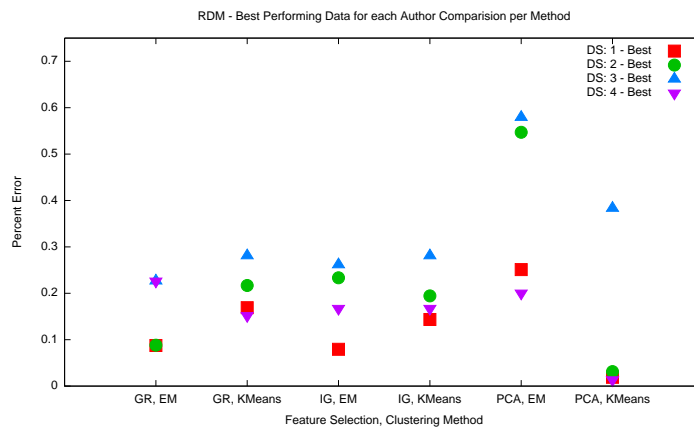
Figure 3.2: Error per Method

resulting from Information Gain and EM the previously noted preferred outcome of RDM, PCA, and KMeans continues.

The only literary collection that RDM, PCA, and KMeans didn't achieve sub-10% error rate was literary collection 3; however this literary collection also performed the worst under all other combinations involving RDM, along with the majority of the combinations involving Unigram feature generation. This leads to the conclusion that the distinction between the authors in literary collection 3 is fundamentally different from the distinction necessary for the other literary collections. If literary collection 3 were to then be considered an outlier, as its removal would increase the performance of a significant majority of the feature generation, feature selection, clusterer combinations, then the selection of RDM, PCA, and KMeans is



(a) Unigram - Best Results per Dataset, per Method



(b) RDM - Best Results per Dataset, per Method

Figure 3.3: Best Results per Dataset, per Method

further solidified as the preferred choice providing the possibility of smallest error.

3.2 Cluster Evaluation

Cluster Evaluation	Correlation with Accuracy
Scatter Seperability	-0.89966632
Silhouette Coefficient	-0.18618025

Table 3.1: Correlation between Cluster Evaluation and Accuracy

As RDM, PCA, and KMeans was determined to be the most likely combination to provide the smallest error, significant progress had been made. There are, however, many possible datasets corresponding to the each literacy collection and a

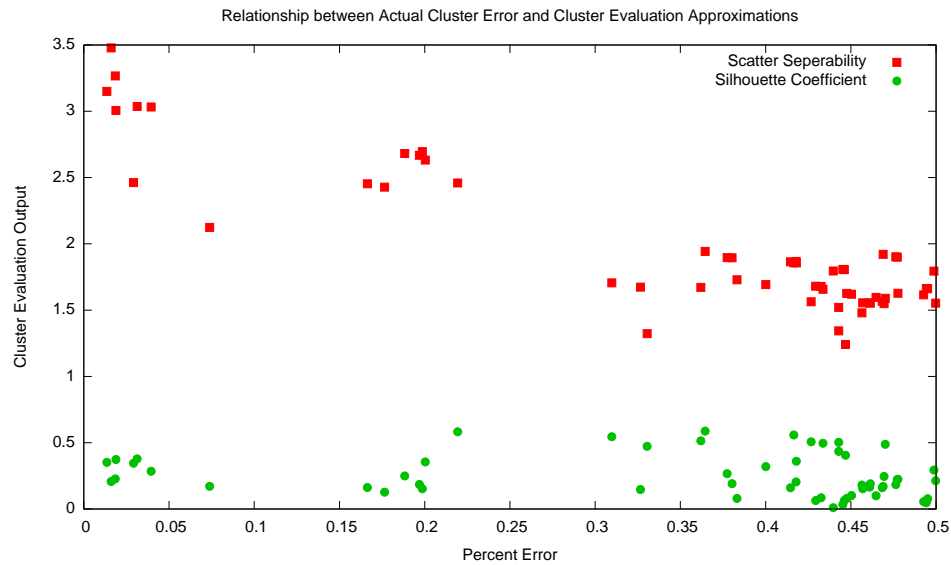


Figure 3.4: Cluster Error and Cluster Evaluation

determination of which of the possible clusterings is most desirable is still needed. This is an unsupervised setting, therefore up to this point the accuracy has been used to determine which of the various combinations results in a more desirable clustering, in order to accomplish the task of selecting a clustering must eventually be done using only cluster evaluation metrics.

For a cluster evaluation metric to be preferred, the metric should be highly correlated (either positively or negatively) with the actual error rate. This correlation should be reasonably robust and therefore useful in comparing various resulting clusters across datasets generated by the same literary collection, and also across literary collections if possible. The ability to compare across datasets is fundamental, it allow for conclusions to be drawn regarding which dataset derived from a given literary collection provides the best clustering for that particular literary collection. A metric able to compare clusterings across literary collections would also allow for statements regarding which pairs of authors were confidently distinguished.

Figure 3.4 displays the comparison between the percent error and the value of each of the two potential cluster evaluation metrics. A high correlation would be noticeable here as a consistent increase in the actual error rate would result in a consistent shift in the cluster evaluation metric result. While an increase in error

rate seems to have little impact on the values corresponding with the Silhouette Coefficient metric, there is a marked general downward trend present in the value produced by Scatter Separability. This downward trend suggests a negative correlation, and when the correlations are computed explicitly (Table 3.1) for every dataset generated from the use of PCA and KMeans, the correlation between error rate and Scatter Separability is markedly larger than the correlation between error rate and Silhouette Coefficient. This large correlation leads to the conclusion that Scatter Separability is a better choice as a Cluster Evaluation Metric in these circumstances, and can be used as a reasonably accurate approximation of the relative error rates between clusters not only between datasets of the same literary collection, but also datasets created from different literary collections.

4. Conclusion and Future Work

In this work the focus was on determining which unsupervised methods would provide the best separation corresponding with two distinct author's work. Based upon the results obtained, the combination of RDM, PCA, K-Means, and Scatter Separability provide an unsupervised learning method capable of distinguishing between some combinations of two authors to within an acceptable margin of error. The performance of RDM over Unigram as a feature generation method suggests that the features most useful in determining the separation were based not upon single word frequency but included multiple word phrases and similar complex repetitions. The gains from the inclusion of PCA suggests that many distinct repeated phrases were required to complete the task, because the combinations of features presented by PCA resulted in higher accuracy than when features were considered individually. The overall unsupervised learning method focuses on determining the collection of words and phrases that allow for the adequate separation of literary excerpts. Furthermore, because these resulting collections of words and phrases allowing for the adequate separation of excerpts, they serve as an approximate representation of the two authors' comparative style and therefore may be useful in other contexts.

Potential further work based upon this endeavor includes an exploration of the validity of the hypothesis regarding the extraction of the authors' comparative style and related applications. The generalization of this work to the separation of more than two authors' works is not trivial and is deserving of further study, as well as a closer look at the tuning of the methods mentioned along with other methods that can be used in any of the unsupervised learning categories.

BIBLIOGRAPHY

- [1] Gutenberg project. *Accessible at <http://www.gutenberg.org>.*
- [2] M. Caillet, J. Pessiot, M. Amini, and P. Gallinari. Unsupervised learning with term clustering for thematic segmentation of texts. *Proceedings of RIAO*, pages 648–660, 2004.
- [3] V. Chaoji, A. Hoonlar, and B. Szymanski. Recursive data mining for author and role identification. *Cyber Security and Critical Infrastructure Coordination*, 2008.
- [4] K. Coffman and A. Odlyzko. The size and growth rate of the Internet. *First Monday*, 3(10):1–25, 1998.
- [5] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.
- [6] K. Fukunaga. *Statistical Pattern Recognition (2nd Ed)*. Academic Press, 1990.
- [7] Z. Ghahramani. Unsupervised Learning. *Advanced Lectures on Machine Learning*, 3176:72–112, 2004.
- [8] W. Goshawke, I. Kelly, and J. Wigg. *Computer translation of natural language*. Halsted Press New York, NY, USA, 1988.
- [9] J. Hansen and M. Indeje. Linking dynamic seasonal climate forecasts with crop simulation for maize yield prediction in semi-arid Kenya. *Agricultural and Forest Meteorology*, 125(1-2):143–157, 2004.
- [10] D. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

- [11] L. Kaufman and P. Rousseeuw. Finding groups in data. an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990*, 1990.
- [12] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 1–6, 1990.
- [13] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(281-297):14, 1967.
- [14] C. Manning et al. *Foundations of statistical natural language processing*. MIT Press, 1999.
- [15] T. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60, 1996.
- [16] S. Mori, C. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.
- [17] T. Nomoto and Y. Matsumoto. A new approach to unsupervised text summarization. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–34, 2001.
- [18] M. Reinberger and P. Spyns. Unsupervised text mining for the learning of dogma-inspired ontologies. *Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press. to appear*, 2005.
- [19] P. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning*, pages 491–502, 2001.
- [20] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques (2nd Ed)*. Morgan Kaufmann, San Francisco, 2005, 2005.

APPENDIX A
Literary Collection Details

LC	Author	Works
1	Jane Austen	“Emma”, “Pride and Prejudice”, “Sense and Sensibility”
1	Joseph Conrad	“Arrow of Gold”, “Change”, “Secret Agent”
2	Sir Walter Scott	“The Abbot”, “Black Dwarf”, “Waverley”
2	Virginia Woolf	“Jacobs Room”, “Night and Day”, Voyage Out“
3	D. H. Lawrence	”Aaron’s Rod“, ”Lost Girl“, ”Sons and Lovers“
3	Jack London	”Jerry of the Islands“, ”Martin Eden“, ”Valley of the Moon“
4	Mary Shelly	”Frankenstein“, ”Last Man“, ”Mathilda“
4	H. G. Wells	”Island of Doctor Moreau“, ”New Machiavelli“, ”World Set Free“

APPENDIX B

Raw Results

All accuracy measurements for each combination of feature extraction, feature selection, and clustering combination for each of the 12 data sets. Error rates under 10% are in bold, error rates under 5% are denoted by a start (*).

Methods	Data Set	Error Rate
Unigram,GR,EM	1-3par	0.40492531287848205
Unigram,GR,EM	1-5par	0.4979757085020243
Unigram,GR,EM	1-7par	0.43531633616619453
Unigram,GR,EM	2-3par	0.23337268791814247
Unigram,GR,EM	2-5par	0.2463863337713535
Unigram,GR,EM	2-7par	0.40900735294117646
Unigram,GR,EM	3-3par	0.5447125621007807
Unigram,GR,EM	3-5par	0.4636309875813128
Unigram,GR,EM	3-7par	0.4896437448218724
Unigram,GR,EM	4-3par	0.28064992614475626
Unigram,GR,EM	4-5par	0.4125615763546798
Unigram,GR,EM	4-7par	0.4784110535405872
Unigram,GR,KMeans	1-3par	0.4178441663302382
Unigram,GR,KMeans	1-5par	0.47125506072874496
Unigram,GR,KMeans	1-7par	0.3843248347497639
Unigram,GR,KMeans	2-3par	0.2829594647776466
Unigram,GR,KMeans	2-5par	0.2831800262812089
Unigram,GR,KMeans	2-7par	0.42371323529411764
Unigram,GR,KMeans	3-3par	0.42760823278921223
Unigram,GR,KMeans	3-5par	0.43169722057953874
Unigram,GR,KMeans	3-7par	0.4101077050538525
Unigram,GR,KMeans	4-3par	0.310930576070901
Unigram,GR,KMeans	4-5par	0.2376847290640394
Unigram,GR,KMeans	4-7par	0.3747841105354059

Methods	Data Set	Error Rate
Unigram,IG,EM	1-3par	0.3778764634638676
Unigram,IG,EM	1-5par	0.4801619433198381
Unigram,IG,EM	1-7par	0.43531633616619453
Unigram,IG,EM	2-3par	0.23337268791814247
Unigram,IG,EM	2-5par	0.2463863337713535
Unigram,IG,EM	2-7par	0.40900735294117646
Unigram,IG,EM	3-3par	0.4517388218594748
Unigram,IG,EM	3-5par	0.4636309875813128
Unigram,IG,EM	3-7par	0.4788732394366197
Unigram,IG,EM	4-3par	0.27843426883308714
Unigram,IG,EM	4-5par	0.29064039408866993
Unigram,IG,EM	4-7par	0.33506044905008636
Unigram,IG,KMeans	1-3par	0.3653613241824788
Unigram,IG,KMeans	1-5par	0.42186234817813767
Unigram,IG,KMeans	1-7par	0.3843248347497639
Unigram,IG,KMeans	2-3par	0.2829594647776466
Unigram,IG,KMeans	2-5par	0.2831800262812089
Unigram,IG,KMeans	2-7par	0.3538602941176471
Unigram,IG,KMeans	3-3par	0.4517388218594748
Unigram,IG,KMeans	3-5par	0.43169722057953874
Unigram,IG,KMeans	3-7par	0.4084507042253521
Unigram,IG,KMeans	4-3par	0.3190546528803545
Unigram,IG,KMeans	4-5par	0.2376847290640394
Unigram,IG,KMeans	4-7par	0.3747841105354059

Methods	Data Set	Error Rate
Unigram,PCA,EM	1-3par	0.610012111425111
Unigram,PCA,EM	1-5par	0.6421052631578947
Unigram,PCA,EM	1-7par	0.6081208687440982
Unigram,PCA,EM	2-3par	0.6587957497048406
Unigram,PCA,EM	2-5par	0.5637319316688568
Unigram,PCA,EM	2-7par	0.640625
Unigram,PCA,EM	3-3par	0.7207239176721079
Unigram,PCA,EM	3-5par	0.6753400354819633
Unigram,PCA,EM	3-7par	0.5352112676056338
Unigram,PCA,EM	4-3par	0.6425406203840472
Unigram,PCA,EM	4-5par	0.6330049261083743
Unigram,PCA,EM	4-7par	0.6113989637305699
Unigram,PCA,KMeans	1-3par	0.38070246265643926
Unigram,PCA,KMeans	1-5par	0.4502024291497976
Unigram,PCA,KMeans	1-7par	0.3635505193578848
Unigram,PCA,KMeans	2-3par	0.45179063360881544
Unigram,PCA,KMeans	2-5par	0.4231274638633377
Unigram,PCA,KMeans	2-7par	0.4209558823529412
Unigram,PCA,KMeans	3-3par	0.4410929737402413
Unigram,PCA,KMeans	3-5par	0.42755765819041985
Unigram,PCA,KMeans	3-7par	0.3976801988400994
Unigram,PCA,KMeans	4-3par	0.3825701624815362
Unigram,PCA,KMeans	4-5par	0.43842364532019706
Unigram,PCA,KMeans	4-7par	0.4265975820379965

Methods	Data Set	Error Rate
RDM,GR,EM	1-3par	0.3556721840936617
RDM,GR,EM	1-5par	0.0874493927125506
RDM,GR,EM	1-7par	0.1501416430594901
RDM,GR,EM	2-3par	0.23219205037386856
RDM,GR,EM	2-5par	0.08804204993429698
RDM,GR,EM	2-7par	0.25
RDM,GR,EM	3-3par	0.28743789921930446
RDM,GR,EM	3-5par	0.22649319929036074
RDM,GR,EM	3-7par	0.3595691797845899
RDM,GR,EM	4-3par	0.38404726735598227
RDM,GR,EM	4-5par	0.43596059113300495
RDM,GR,EM	4-7par	0.22625215889464595
RDM,GR,KMeans	1-3par	0.22567622123536535
RDM,GR,KMeans	1-5par	0.1805668016194332
RDM,GR,KMeans	1-7par	0.16902738432483475
RDM,GR,KMeans	2-3par	0.31286894923258557
RDM,GR,KMeans	2-5par	0.21681997371879105
RDM,GR,KMeans	2-7par	0.26286764705882354
RDM,GR,KMeans	3-3par	0.3041163946061036
RDM,GR,KMeans	3-5par	0.3021880544056771
RDM,GR,KMeans	3-7par	0.28086164043082024
RDM,GR,KMeans	4-3par	0.3020679468242245
RDM,GR,KMeans	4-5par	0.2536945812807882
RDM,GR,KMeans	4-7par	0.15198618307426598

Methods	Data Set	Error Rate
RDM,IG,EM	1-3par	0.32983447719014936
RDM,IG,EM	1-5par	0.07935222672064778
RDM,IG,EM	1-7par	0.14919735599622286
RDM,IG,EM	2-3par	0.23337268791814247
RDM,IG,EM	2-5par	0.2463863337713535
RDM,IG,EM	2-7par	0.30330882352941174
RDM,IG,EM	3-3par	0.34989354151880764
RDM,IG,EM	3-5par	0.2613837965700769
RDM,IG,EM	3-7par	0.347970173985087
RDM,IG,EM	4-3par	0.28064992614475626
RDM,IG,EM	4-5par	0.35591133004926107
RDM,IG,EM	4-7par	0.16753022452504318
RDM,IG,KMeans	1-3par	0.22567622123536535
RDM,IG,KMeans	1-5par	0.2089068825910931
RDM,IG,KMeans	1-7par	0.14353163361661944
RDM,IG,KMeans	2-3par	0.24950806768988587
RDM,IG,KMeans	2-5par	0.19448094612352168
RDM,IG,KMeans	2-7par	0.26286764705882354
RDM,IG,KMeans	3-3par	0.29701916252661464
RDM,IG,KMeans	3-5par	0.3021880544056771
RDM,IG,KMeans	3-7par	0.28086164043082024
RDM,IG,KMeans	4-3par	0.30132939438700146
RDM,IG,KMeans	4-5par	0.16748768472906403
RDM,IG,KMeans	4-7par	0.18307426597582038

Methods	Data Set	Error Rate
RDM,PCA,EM	1-3par	0.6039563988696003
RDM,PCA,EM	1-5par	0.4251012145748988
RDM,PCA,EM	1-7par	0.251180358829084
RDM,PCA,EM	2-3par	0.5863833136560409
RDM,PCA,EM	2-5par	0.5532194480946123
RDM,PCA,EM	2-7par	0.546875
RDM,PCA,EM	3-3par	0.6408800567778566
RDM,PCA,EM	3-5par	0.6534594914251922
RDM,PCA,EM	3-7par	0.5791217895608948
RDM,PCA,EM	4-3par	0.5760709010339734
RDM,PCA,EM	4-5par	0.5320197044334976
RDM,PCA,EM	4-7par	0.2003454231433506
RDM,PCA,KMeans	1-3par	0.44691158659668956
RDM,PCA,KMeans	1-5par	0.029149797570850202 (*)
RDM,PCA,KMeans	1-7par	0.018885741265344664 (*)
RDM,PCA,KMeans	2-3par	0.4336875245966155
RDM,PCA,KMeans	2-5par	0.32654402102496716
RDM,PCA,KMeans	2-7par	0.03125 (*)
RDM,PCA,KMeans	3-3par	0.38325053229240597
RDM,PCA,KMeans	3-5par	0.4429331756357185
RDM,PCA,KMeans	3-7par	0.4266777133388567
RDM,PCA,KMeans	4-3par	0.07385524372230429
RDM,PCA,KMeans	4-5par	0.013546798029556651 (*)
RDM,PCA,KMeans	4-7par	0.18825561312607944