Document Clustering using Feature Selection Technique

Md Rashadul Hasan Rakib, B00598853
Faculty of Computer Science
Dalhousie University

April 13, 2015

Abstract

Clustering documents is an old and interesting task in the area of Data Mining. The objective of document clustering is to cluster documents into different groups where the documents in same cluster are very similar to each other and dissimilar from the documents of other clusters. A document consists of words and the words are considered as its features. Not all the features of documents are equally important to find clusters among them. Some features are very common among the documents and some appear only in few documents. Hence, it is very crucial to select the important features from the documents so that the important features can only take part in clustering task so as to improve the clustering result. In order to select significant features from the documents we use different feature selection techniques such as unsupervised, supervised and Hybrid. Hybrid feature selection approach combines both unsupervised and supervised feature selection mechanisms. To cluster documents we adopt the K-Means clustering algorithm. The experimental results show that clustering documents using feature selection performs better than clustering without feature selection.

1 Introduction

Document clustering refers to organizing the documents into different groups. It is one of the central problems in the area of Data Mining [13]. With the rapid growth of internet, digital media, the amount of documents are increasing day by day at large scale. In order to organize these huge collection of documents, document clustering becomes a crucial task. However, selecting important features (e.g., words) from documents is more crucial in order to ensure the performance of clustering task. This is because, in this paper we delineate the document clustering algorithm with various feature selection techniques. Consider a document d="Showers continued throughout the week in the Bahia cocoa zone" where each individual word is taken into account as a feature. Figure 1 illustrates a sample document clusters where the documents are clustered into three different groups.

Document Clustering is an unsupervised approach that clusters documents without having prior knowledge about the meta information (e.g., categories, authors, create time) of documents. If we want to cluster documents based on their category information, then each category can imply a particular cluster. Then the question arises, how the clustering algorithm clusters documents into different groups without knowing the document category information. The answer is that it uses the similarity measure like Euclidean distance [5] by which it computes the distance between two documents that allows the clustering algorithm

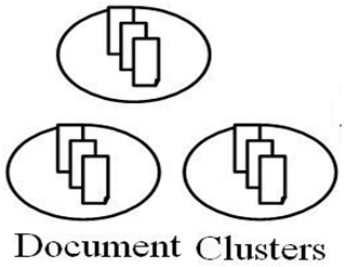

Figure 1: Sample Document Clusters.

to keep most similar documents into same cluster and dissimilar documents into other clusters. If we consider each document as a high dimensional vector and each feature (e.g., word) is considered as a single dimension, then the Euclidean distance between two vectors is computed using the distance between each dimension (e.g., feature). The general argument is that are all the features of a document are equally important for clustering. The answer is no. The important features are those that discriminate one document from another by which we can put the documents in their appropriate groups.

Since the document features are not numerical, we convert them into numerical (e.g., $tf - idf$ [14]) values to compute distance between two documents which in turn used for clustering. In order to convert the features into numerical values we use the term frequency (tf) and document frequency (df) of the features. The term frequency of a feature is the number of occurrences of that feature in a particular document. The document frequency implies that in how many documents a feature appears.

We use the K-Means [9] clustering algorithm along with or without various feature selection approaches [13] to investigate whether the clustering result improves or not after applying feature selection. There are three types of feature selection techniques [13] (e.g., unsupervised, supervised and hybrid) that we use. We adopt the most simplest unsupervised feature selection technique which is Document Frequency (DF). The supervised feature selection methods are Information Gain (IG) and Subset Evaluation (SE); and the hybrid approach combines both supervised and feature selections (e.g., DF+IG, DF+SE). From the experimental result we find that document clustering using DF performs better than clustering without any feature selection. After using supervised feature selection like IG and SE, we get better result than using DF. When we combine the unsupervised DF with supervised IG and SE the result improves more than just using unsupervised or supervised feature selection alone.

2 Related Work

Lots of works have been done about document clustering with various feature selection mechanisms. There are mainly three types feature selection methods [13]: unsupervised, supervised and hybrid. Some document clustering algorithms use unsupervised feature selection [12, 13, 15], some use supervised feature selection [13] and some use hybrid feature selection which is a combination of both supervised and unsupervised approach [13]. Other than these approaches, there are document clustering algorithms that use topic modelling, human interaction to select important features [4, 11].

The unsupervised feature selections do not rely on the document label or type. It is independent of the document meta information. Therefore for large data set, this approach

is very effective and scalable. There is a comparative study on document clustering using unsupervised feature selection [12] using various unsupervised feature selection mechanisms like Document Frequency (DF), Term Contribution (TC), Term variance quality (TVQ) and Term Variance (TV). They use a subset of Reuters21758 [10] dataset of 1657 documents and apply K-Means clustering algorithm along with the above mentioned four feature selection techniques to see which feature selection approach gives better clustering result. In their experimental result they show that, both Document Frequency (DF) and Term Contribution (TC) provide better clustering result than other feature selection techniques. There is another unsupervised feature selection method [15] for text clustering that adopts genetic algorithms to find most valuable groups of terms which in turn will be utilized to generate the final feature vector for document clustering.

The supervised feature selections rely on the document meta information. Using the document label, it selects feature from a collection of documents. In [13], supervised feature selection techniques such as Information Gain (IG), CHI-SQUARE (x^2) statistics are used to select features from documents in order to perform clustering on them. They also apply unsupervised feature selection techniques. Finally they come-up with two conclusions: i) Feature selection methods improve the performance of document clustering algorithm, ii) Iterative Feature Selection (IF) can solve the unavailability of document label problem. At first they use labels for small number of documents and select features using supervised feature selection method, then using those features, they iteratively select features from the documents whose labels are unknown.

Topic Modelling is another approach for feature selection. In [4], the most popular topic modelling technique called Latent Dirichlet Allocation (LDA) [1] is used to select the important features as top ranked topics where each topic is represented by a single word. Using the top ranked features, they cluster documents and get better result than clustering without feature selection. Human interaction based feature selection is used in [11] where a human iteratively selects important features which are then used to cluster documents. They show that this type of human interaction improves the clustering result.

3 Application Scenario and Solution Steps

The application scenario we focus for this project is to examine the performance of document clustering algorithm with or without feature selection technique on the Reuters21758 [10] dataset. In particular, if we apply feature selection technique on this document collection to select discriminative features which in turn are used for clustering, then we want to observe that the clustering result improves or not than clustering without any feature selection. Given a collection of documents from Reuters21758 [10] dataset where each document belongs to a particular category. Our aim is to cluster the documents with feature selection so that each document is placed into its appropriate category with higher accuracy rate. The detailed architecture of our system is depicted in Figure 2.

At first we extract documents and their corresponding categories from the XML files of Reuters21758 [10] dataset. Then we create a document collection by randomly selecting 1600 documents where each 400 documents belong to one of the four categories: EXCHANGES, PLACES, PEOPLE, ORGS. Each document is placed within a single XML $< BODY >$ tag, as shown in Figure 3. The category of a document is situated within one of the four category tags. The corresponding category of a document is determined by one of the non empty category tag among four categories. We can see from figure 3 that, this document

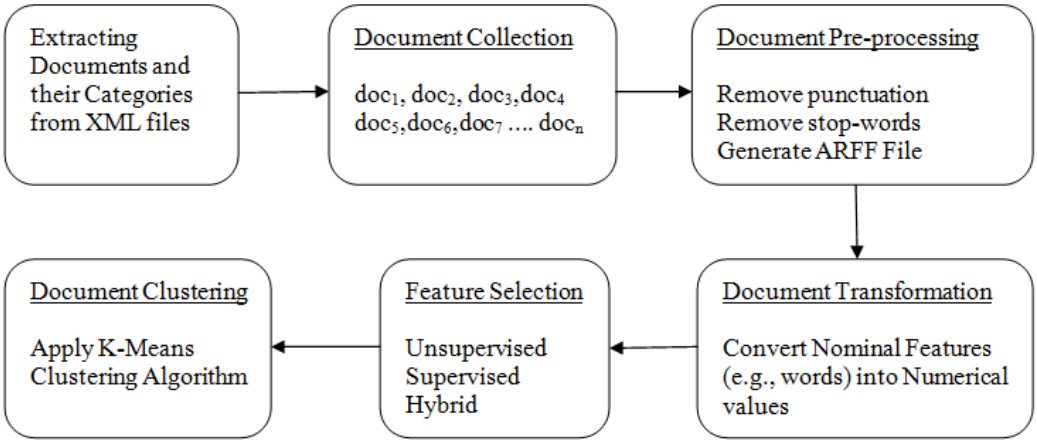

Figure 2: Detailed System Architecture.

belongs to PLACES category since PLACES category tag is non empty (e.g., it contains the value $< D > UK < D >$)

```
<REUTERS TOPICS="NO" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="18707" NEWID="2289">
    <DATE> 5-MAR-1987 13:14:47.61</DATE>
    <TOPICS></TOPICS>
    <PLACES>
        <D>uk</D>
    </PLACES>
    <PEOPLE></PEOPLE>
    <ORGS></ORGS>
    <EXCHANGES></EXCHANGES>

    <TEXT>
        <TITLE>CADBURY REQUESTS STOCK EXCHANGE ENQUIRY</TITLE>
        <DATELINE>    LONDON, March 5 - </DATELINE>
        <BODY>
            Cadbury-Schweppes Plc &lt;CADB.L> said it
            had asked the London Stock Exchange to launch a formal enquiry
            into dealings in the company's shares in recent months.
            It said it believed such a move was in the best interests
            of shareholders following recent charges being made under U.K.
            Insider dealing laws about the shares.
            Last week, former &lt;Morgan Grenfell Group Plc> executive
            Geoffrey Collier was charged with insider dealing in Cadbury
            shares.
        </BODY>
    </TEXT>
</REUTERS>
```

Figure 3: XML File format.

After preparing the document collection, several document pre-processing steps are applied such as removing punctuation, stop-words, generating ARFF [6] (Attribute-Relation File Format) File. Then we apply WEKA StringToVector filter [6] to create Document-Feature matrix by converting the nominal features (e.g., words) into numerical values. Once the Document-Feature matrix is created, we apply different feature selection techniques to remove the unimportant features. Finally we apply K-Means Clustering algorithm [9] on the Document-Feature matrix before and after the feature selection to see how the feature selection improves the clustering result.

4 Document Preprocessing

Data Preprocessing is one of the major steps in Data Mining task. If the data is not preprocessed properly, then it is a regular case that the output will not be perfect regardless of the robustness of Data Mining algorithm. Hence, we pre-process the documents on which we are going to apply clustering algorithm with feature selection. The steps of Document Preprocessing are listed below.

- Removing punctuation
- Removing stop-words
- Generating ARFF File

4.1 Removing Punctuation

Punctuations are considered as noise in the Bag-Of-Word document model where the word orders are ignored. Hence we remove all types of punctuations by the regular expression $[\hat{}a-zA-Z]$ that removes all characters other than capital and small letters.

4.2 Removing stop-words

In the Data Mining Literature [8] it has been shown that most of the time stop-words are treated as noises in the documents since these are highly frequent and thus do not possess any discriminative characteristics by which we can differentiate one document from another. This is because we remove the stop-words from documents.

4.3 Generating ARFF File

ARFF means Attribute-Relation File Format which is mainly used by WEKA [6] data mining tool. Since we use WEKA, the document collection is converted into a single ARFF file. An ARFF [6](Attribute-Relation File Format) file is a plain text file that describes a list of instances with a set of attributes. An ARFF file has two principal sections. The first section is the Header information followed by the section about Data information. The Header of an ARFF file contains the name of the relation, a list of the attributes (e.g., features) and their associated types. The Data section of an ARFF file contains the attribute values maintaining the attribute orders. The ARFF file structure generated by our system is shown in Figure 4.

The first line is the name of the relation $reuters1600$, followed by two attributes $Document$ and $Category$ with types $String$ and $Nominal$, respectively. The values of the $Document$ and $Category$ attributes are different documents and their corresponding categories respectively.

5 Document Transformation

Document Transformation [6] refers to the conversion of nominal features (e.g., words) into numerical values. This step is very crucial for Document Clustering, because we need to compute distance between the documents which is measured based on the numerical values. The steps of Document Transformation accomplished by WEKA StringToVector filter [6] are given below.

@relation reuters1600

@attribute Document string
@attribute Category {EXCHANGES,PLACES,PEOPLE,ORGS}

@data
'word1 word2 .....', EXCHANGES
'word3 word4 .....', PLACES
'word5 word6 .....', PEOPLE
'word7 word8 .....', ORGS.
....
....

Figure 4: ARFF File structure.

- Convert Nominal Features (e.g., words) into Numerical values
- Construct Document-Feature matrix (e.g., Data Matrix)

5.1 Convert Nominal Features into Numerical Values

Each feature is converted into a numerical value (e.g., $tf - idf$ [14]) computed using the term frequency (tf) and document frequency (df). The mathematical representation of $tf - idf$ value of a feature within a particular document is given in Equation 1.

$$\begin{aligned} tf - idf(t,d) =& tf(t,d) \times idf(t,D) \\ =& tf(t,d) \times log(\frac{N}{df(t,D)+1}) \end{aligned} \tag{1}$$

$$\begin{aligned} D &= Set\ of\ documents\ d_1, d_2, ...d_n. \\ N &= Number\ of\ documents\ in\ D. \\ tf(t,d) &= Number\ of\ occurrences\ a\ feature\ (e.g., term),\ t\ in\ a\ document\ d. \\ df(t,D) &= In\ how\ many\ documents\ a\ feature,\ t\ appears. \\ idf(t,D) &= Inverse\ document\ frequency\ of\ a\ feature,\ t. \\ 1 &= Smoothing\ parameter\ to\ avoid\ zero\ idf\ (t,D). \end{aligned}$$

5.2 Construct Document-Feature Matrix

After calculating the $tf - idf$ value for all features, we construct a Document-Feature Matrix or data matrix called M_{DocFtr}, as shown below. The number of rows and columns of M_{DocFtr} are m and n respectively where m is the number of documents and n is the number of features. Each row of the matrix is a high dimensional vector to represent a single document in Vector Space Model (VSM) [8] and each column of the matrix represents a particular dimension (e.g., feature). Each cell, $\alpha_{i,j}$ stands for the $tf - idf$ value of a feature, ftr_j in document, doc_i.

$$M_{DocFtr} = \begin{array}{c} \\ doc_1 \\ doc_2 \\ \\ doc_i \\ \\ doc_m \end{array} \begin{array}{c} \begin{array}{cccccc} ftr_1 & ftr_2 & & ftr_j & & ftr_n \end{array} \\ \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,n} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,j} & \cdots & \alpha_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i,1} & \alpha_{i,2} & \cdots & \alpha_{i,j} & \cdots & \alpha_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{m,1} & \alpha_{m,2} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,n} \end{pmatrix} \end{array}$$

6 Feature Selection

Features of a document are the words, bi-grams, tri-grams and so on. For simplicity, we are interested only about the word features. Features of a document represent its context. In particular, the features exemplify the type or category of a document. However, all features of a document are not equally significant to symbolize its context. Some features are more important than others. Eventually, in document clustering, the selection of useful features plays a vital role in clustering result. In document clustering, we need to differentiate one group of documents from another group. This differentiation heavily depends on the discriminative features of the documents. Feature selection is a technique that selects features in a way such that we can easily differentiate one group of documents from another group. We adopt three types of feature selection mechanisms which are listed below.

- Unsupervised
- Supervised
- Hybrid

6.1 Unsupervised Feature Selection

Unsupervised feature selection is independent of the document categories. It selects features without having prior knowledge of document types. We have implemented one of the simplest unsupervised feature selection technique called Document Frequency (DF). As we discussed before, DF of a feature is that in how many documents a feature appears. This statistics (e.g.,DF) is used to select the discriminative features of the documents within a document collection. We apply a very simple heuristic that, the DF of a discriminative feature is bounded between 2 and $SQRT(N)$ where N is the number of documents in the document collection that we want to cluster. The idea of $SQRT(N)$ is borrowed from [3] that selects features using Regularized Random Forest [2]. Then a simple question arises that why we do not use DF=1. The answer is that DF=1 will select all the features that might cause over-fitting problem. We do not select the features that have DF=N, because of two reasons: i) these features are not discriminative since they are shared among all the documents ii) there might be under-fitting problem since the number of features become very small. The algorithm for Unsupervised Feature Selection using Document Frequency (DF) is given in Algorithm 1. Using Document Frequency, we select 570 features out of 2255 features for 1600 documents. Each of the features among 570 has document frequency between 2 and $SQRT(1600) = 40$.

Algorithm 1 Unsupervised Feature Selection using Document Frequency

Input:
$listDocuments$ = Collection of documents
$docFrquencies$ = Document frequencies of the features
Output:
List of filtered features for all documents

```
1: function UFSDF(listDocuments,docFrquencies)
2:     N = Number of documents
3:     listOfFeaturesOfAllDocs = Empty List of filtered features for all documents
4:     for Each document, d in listDocuments do
5:         listOfFeaturesOfADoc = Empty list of features for a single document
6:         for Each feature, ftr in the list of Features of d do
7:             if docFrquencies(ftr)) >= 2 && docFrquencies(ftr) <= SQRT(N) then
8:                 listOfFeaturesOfADoc.add (ftr)
9:             end if
10:        end for
11:        listOfFeaturesOfAllDocs.add (listOfFeaturesOfADoc)
12:    end for
13: end function
```

6.2 Supervised Feature Selection

Supervised Feature Selection [13] is dependant on the document meta-information (e.g., category, author). Since our aim is to cluster documents into different categories, therefore we use the document category information for supervised feature selection task. Using this feature selection approach we select the important features so as to enhance the clustering accuracy. The supervised feature selection techniques that we apply are listed below.

- Information Gain (IG)
- Subset Evaluation (SE)

6.2.1 Information Gain (IG)

Information Gain feature selection evaluates the worth of an attribute by measuring the information gain [16] with respect to the category of the document [8, 13]. The feature that makes a good partition within the dataset, will produce higher information gain. Therefore we rank the features based on their information gains in descending order. Since we have a large number of features, we select the top SQRT(#total features). The idea of selecting top SQRT(#total features) features is borrowed from [3] with the aim of reducing the dimensionality to improve the scalability of the Clustering Algorithm. At first we have 2255 features for 1600 documents; then we select top $SQRT(2255) = 47$ features using WEKA, based on their information gains in descending order, as shown in Figure 5.

6.2.2 Subset Evaluation (SE)

Feature selection by subset evaluation (SE) [7] evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of

IG value	FtrNo	FtrName	Rank
0.26508	331	exchange	1
0.24969	969	stock	2
0.20467	1038	trading	3
0.19235	2075	ec	4
0.17516	327	european	5
0.17238	182	community	6
0.12364	222	countries	7
0.10655	1877	reagan	8
....	...	....	

Figure 5: Ranked features based on Information Gains in descending order.

redundancy between them. Subsets of features that are highly correlated with the category (or class) while having low inter-correlation are preferred. Using this feature selection approach we select 38 features out of 2255 features using WEKA for 1600 documents.

6.3 Hybrid Feature Selection

Hybrid feature selection [13] comprises both unsupervised and supervised feature selection approaches. The features selected by Hybrid approach outperforms the standalone unsupervised or supervised approach. At first, the feature are selected using Document Frequency (DF) that produces 570 features out of 2255. After that, we apply Information Gain or Subset Evaluation technique using WEKA on the selected 570 features that generate $SQRT(570) = 23$ and 29 features respectively. Using 23 features, selected by (DF+IG), we get better document clustering than the document clustering using only the selected features from DF or IG. Similarly, the performance of document clustering using 29 features selected by (DF+SE) is better then the document clustering using only the selected features from DF or SE.

7 Document Clustering

We adopt the simplest partition based clustering algorithm, K-Means to cluster the documents. The input of the K-Means Clustering algorithm is the Document-Feature Matrix (e.g., data matrix) and the number of clusters which is 4. We choose 4 because we already know that there are 4 categories of documents. The pseudo code of K-Means clustering algorithm is shown in algorithm 2. We use WEKA clustering API [6] to cluster documents and apply the K-Means algorithm [9] in two different ways to cluster documents which are given below.

- Document Clustering without Feature Selection
- Document Clustering with Feature Selection

7.1 Document Clustering without Feature Selection

This is approach is fairly simple. We apply the K-Means clustering algorithm 2, on the document-feature matrix, M_{DocFtr} containing 1600 documents and 2255 features. This is

Algorithm 2 K-Means Clustering Algorithm

Input:
$M_{DocFtr} = m \times n$, Document-Feature Matrix
k = Number of clusters
Output:
K = Set of clusters
function $UFSDF(M_{DocFtr},k)$
  Randomly select k means c_1, c_2, ..., c_k ▷ Each mean is a randomly selected high-dimensional vector representing a document
  while Clusters in K are not stable **do** ▷ e.g., not met convergence criteria
    for Each document, doc_i in M_{DocFtr} **do**
      Assign doc_i to the cluster which has the closest mean ▷ Measured by Euclidean distance
    end for
    Calculate new mean of each cluster
  end while
  Output the discovered clusters stored in K
end function

considered as the baseline method for this project.

7.2 Document Clustering with Feature Selection

In this step, we apply the same K-Means clustering algorithm 2, on five different document-feature matrices, M_{DocFtr} containing 1600 documents with five different number of features selected by five different combinations of feature selection methods, as shown in Table 1.

Clustering Algorithm	#Documents	Combination of Feature Selection Methods	Total Features	Selected #Features
K-Means	1600	Document Frequency (DF)	2255	570
		Information Gain (IG)	2255	47
		Subset Evaluation (SE)	2255	38
		DF + IG	570	23
		DF + SE	570	29

Table 1: Number of features for different combinations of feature selection methods.

8 Implementation

In this section we talk about the detailed implementation process. The steps of detailed implementation are listed in the following.

- Pre-process documents and Generate ARFF file
- Transform Documents using WEKA
- Select Features

- Cluster Documents

8.1 Pre-process documents and Generate ARFF file

First of all, we parse documents from the Reuters21758 [10] dataset using the XML document API in $C\#$. At time of parsing we pre-process the documents (e.g., remove punctuation, stop words). We generate two ARFF files: i) One file is named reuters21578-big.arff, contains 2255 features, ii) The second file is named reuters21578-bigDF.arff, contains 570 features based on DF. Both files have 1600 documents where 400 documents are from each of the 4 categories: EXCHANGES, PLACES, PEOPLE, ORGS.

8.1.1 Generate reuters21578-big.arff File

A custom function, doParseAndLabel() has been written that parse each document and its corresponding category from the XML files and returns a list of documents with categories. Then the function, GenerateArffFile() is called to generate the ARFF document as shown in Feature 4. There are three classes with different functions, as shown in Table 2

Class Name	Methods	Description
SentenceUtil	loadStopWords()	load the stop words from file
	removeStopWords()	remove stop words (e.g., the, and, etc)
	removePunctuation()	remove non Alphabet characters
ARFFFile	doParseAndLabel()	parse documents and categories from XML
	GenerateArffFile()	generate the ARFF document
Program	main()	call GenerateArffFile()

Table 2: Class and Methods for ARFF file generation.

Build and Run Demo to generate reuters21578-big.arff:
This code is done in $C\#$. We press $Ctrl + Shif + B$ in Visual Studio 2012 to build the program. Visual Studio 2012 is an Integrated Development Environment developed by Microsoft. To run the program we press $Ctrl + F5$. The sample UI to run the program is shown in figure 6.

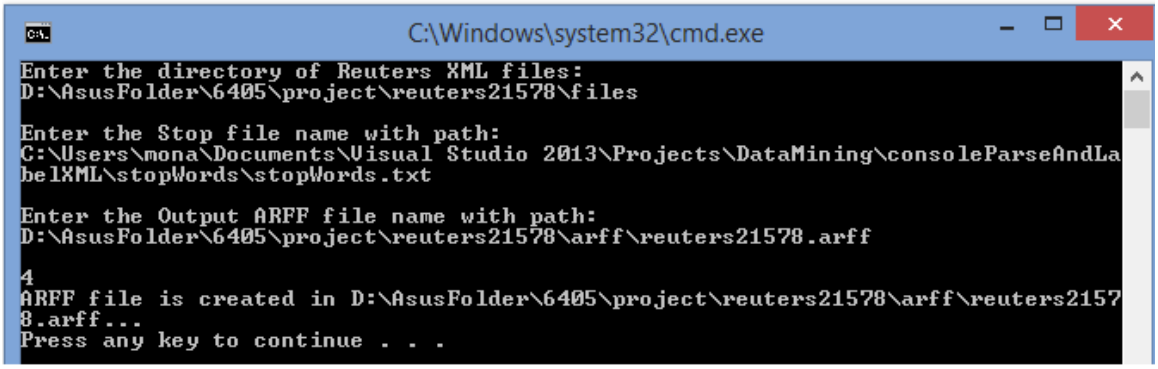

Figure 6: User Interface to generate ARFF file.

8.1.2 Generate reuters21578-bigDF.arff File

The file, reuters21578-bigDF.arff is used to store the features based on document frequency DF. To generate this file we create a function called filterByDF() that takes reuters21578-big.arff as input and produces reuters21578-bigDF.arff as output containing the features that are frequent between 2 and $SQRT(1600) = 40$ documents where 1600 is the total documents in our collection. We have to do small processing on reuters21578-big.arff file which are removing the tags (e.g., @relation ,@attribute, @data) with values.

There are two classes with two major functions, as shown in Table 3.

Class Name	Methods	Description
FilterDF	filterByDF()	filter features based on document frequency
TestDocumentFrequency	main()	Call filterByDF()

Table 3: Class and Methods for filtered ARFF file generation using document frequency.

Build and Run Demo to generate reuters21578-bigDF.arff:
This code is accomplished JAVA. To build this program, we need to go to the root directory (e.g., src) of the program, then run "javac dal/*.java" command where dal is a package folder. To run the program, we need to execute the command "java dal.TestDocumentFrequency". How to run this program is shown in figure 7.

```
rakib@bluenose:~/project6405/src$ javac dal/*.java
rakib@bluenose:~/project6405/src$ java dal.TestDocumentFrequency
Enter the reuters21578-big.arff file name with path as input:
reuters21578-big.arff
Enter the reuters21578-bigDF.arff file name with path as output:
reuters21578-bigDF.arff

reuters21578-bigDF.arff is stroted in reuters21578-bigDF.arff
rakib@bluenose:~/project6405/src$
```

Figure 7: User Interface to generate filtered ARFF file using Document Frequency.

8.2 Transform Documents using WEKA

We transform the ARRF file into Document-Feature matrix using WEKA StringToVEctor filter, as shown in Figure 8. The detail body of the functions are attached with this report. We can also transform using the WEKA UI as shown in Figure 9. In both cases, we set different properties like "Lowercase=true", I"DFTransform=true" and son on.

```
Instances insts = new LoadArffFile(arffFile).load();
NGramTokenizer ngtokenizer = new NGTokenize(1).getNGTokenizer();
StringToWordVector filterngVectors = new StringToVectorization(ngtokenizer,insts).convertStringToVector();
```

Figure 8: Java code to convert ARFF files into Document-Feature matrix.

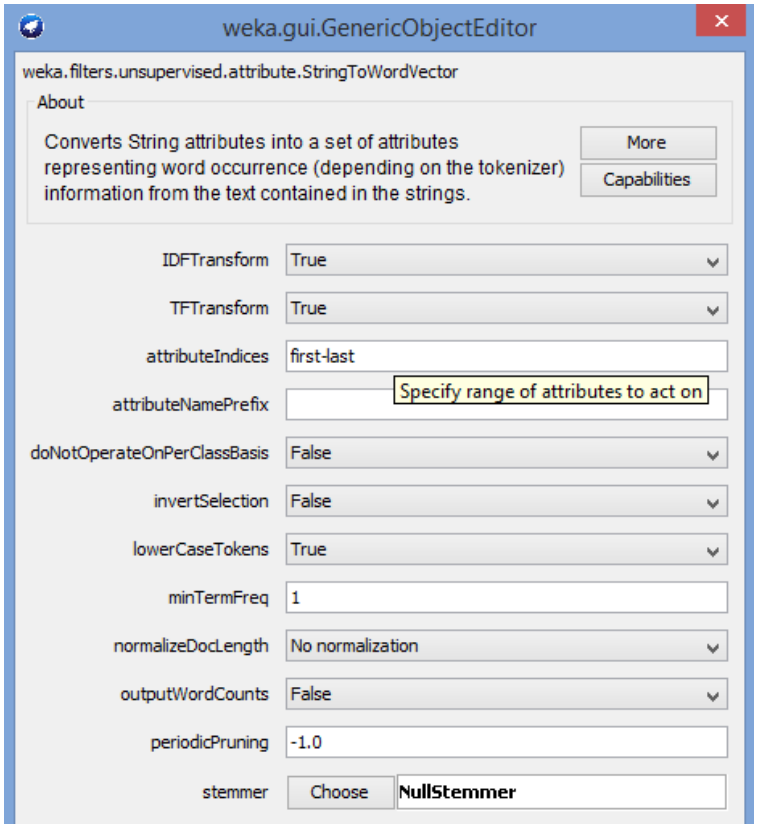

Figure 9: Convert ARFF files into Document-Feature matrix by WEKA UI.

8.3 Select Features

For unsupervised feature selection, we use the reuters21578-bigDF.arff file because all features in this files are selected based on their document frequency. For supervised feature selection, we use WEKA user interface as depicted in Figure 10.

8.4 Cluster Documents

We implement the k-means clustering algorithm using WEKA API [6] for both with feature selection (only by DF) and without feature selection.

8.4.1 Classes and Methods for KMeans algorithm

The implemented classes and principal methods for clustering are described in Table 4.

8.4.2 Build and Run Demo for Document Clustering

We use the WEKA K-Means clustering API to cluster 1600 documents. The input of the K-Means algorithm is the ARFF file as shown in Figure 4 and the number of clusters should be between 1 to 1600 because we cannot cluster 1600 documents more than 1600 groups. The steps of document clustering are given in the following. The method ClusterAndEvaluate() performs the clustering task as well as it evaluate the clustering result by counting the number of instances which are placed in the wrong clusters.

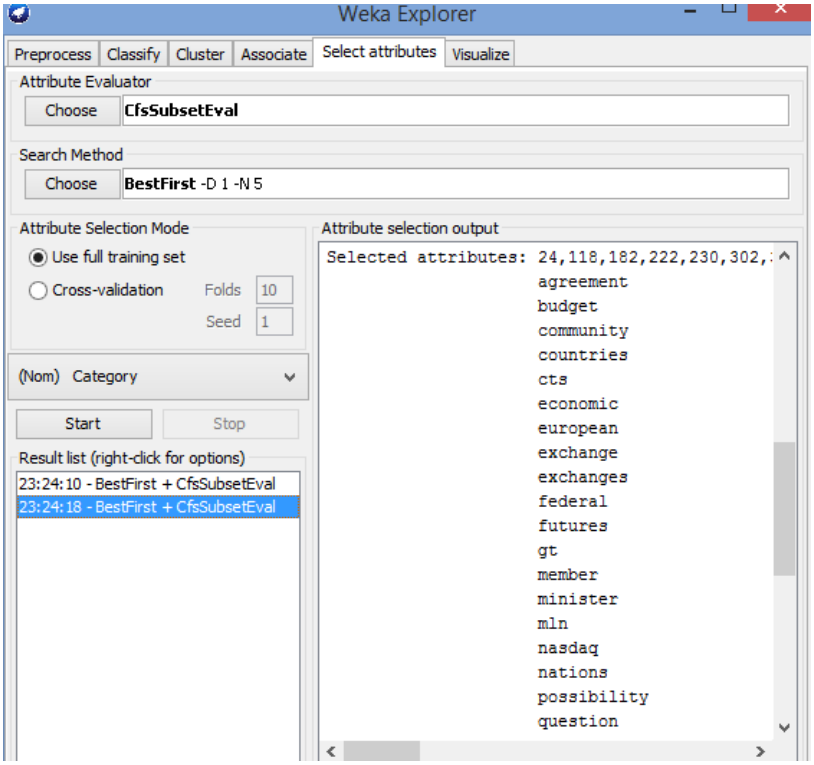

Figure 10: Supervised Feature Selection using WEKA UI.

Class Name	Methods	Description
Utils	getWordCount()	return number of features in a document
	getDocFrequency()	return document frequencies of features (e.g., the, and, etc)
StringToVectorization	convertStringToVector()	convert document collection into document-feature matrix
NGTokenize	getNGTokenizer()	tokenize features from document
LoadArffFile	load()	load the ARFF file into memory
KMeansClustering	ClusterAndEvaluate()	perform k-means clustering on document-feature matrix and evaluate performance
TestClustering	main()	call ClusterAndEvaluate()

Table 4: Class and Methods for KMeans Clustering algorithm.

- Load the ARFF file into memory by load()
- Tokenize the ARFF file by getNGTokenizer() (e.g., split the document into word features)
- Transform the document collection into document-feature matrix by convertString-ToVector()
- Perform clustering and evaluation by ClusterAndEvaluate()

To build and run the K-Means clustering algorithm we need an external jar file named weka.jar. All the java source files are in the package folder *dal*. To build and run this

program, we need to go to the root directory (e.g., src) of the program, then execute the commands "javac -cp .:weka.jar dal/*.java" and "java -cp .:weka.jar dal.TestClustering" respectively as shown in Figure 11.

```
rakib@bluenose:~/project6405/src$ javac -cp .:weka.jar dal/*.java
rakib@bluenose:~/project6405/src$ java -cp .:weka.jar dal.TestClustering
Enter the ARFF file name with path as input for k-means clustering:
reuters21578-bigDF.arff

Enter the number of clusters:
4

Result:
Scheme:weka.clusterers.SimpleKMeans -N 4 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:     Reuters21578FilteredDF-weka.filters.unsupervised.attribute.StringToWordVector-R1-
W1000-prune-rate-1.0-T-I-N0-L-S-stemmerweka.core.stemmers.NullStemmer-M1-
tokenizerweka.core.tokenizers.NGramTokenizer -delimiters " \r\n\t.,;:\'\"()?!" -max 1 -min 1
Instances:    1600
Attributes:   570

kMeans
======
Class attribute: Category
Classes to Clusters:

  0   1   2   3 <-- assigned to cluster
 30  14 355   1 | EXCHANGES
 24 261 115   0 | PLACES
112 273  15   0 | PEOPLE
117 280   3   0 | ORGS

Cluster 0 <-- PEOPLE
Cluster 1 <-- ORGS
Cluster 2 <-- EXCHANGES
Cluster 3 <-- No class

Incorrectly clustered instances : 853.0     53.3125 %
```

Figure 11: User interface to build and run K-Means Clustering Algorithm.

9 Experimental Result and Interpretation

In this section, we discuss the experimental result of document clustering with or without feature selection and explain the reason of it.

9.1 Experimental Result

We apply K-Means clustering algorithm to cluster 1600 documents of Reuters21758 [10] dataset into 4 categories which are EXCHANGES, PLACES, PEOPLE, ORGS. We choose the number of clusters = 4 because we already know that the 1600 documents are distributed among 4 categories. Since our aim is to observe the performance of document clustering with or without feature selection, therefore we are strict to the number of clusters = 4. Experimental result of Document clustering for different combinations of feature selection method is shown in Table 5. We measure the performance of K-Means algorithm [9] in terms of accuracy of placing a document in its appropriate category, as defined in Equation 2.

$$Accuracy = 1 - \frac{\#documents,\ placed\ in\ wrong\ clusters}{\#total\ documents} \tag{2}$$

Clustering Algorithm	Combination of Feature Selection Methods	#Wrongly clustered documents out of 1600	Accuracy (%)
K-Means	All Features	1040	35.00
	Document Frequency (DF)	853	46.6875
	Information Gain (IG)	561	64.9375
	Subset Evaluation (SE)	556	65.25
	DF + IG	547	65.8125
	DF + SE	554	65.375

Table 5: Accuracy of Document Clustering using different combinations of feature selection methods.

9.2 Result Interpretation

From the experimental result in Table 5, we observe that feature selection [13] plays an important role for document clustering task because the accuracy of document clustering improves after using feature selection techniques. The reason for this improvement is that, the feature selection techniques selects the features that are discriminative by which we can easily differentiate one group of documents from another groups.

By using all features, the accuracy of document clustering is only 35% whereas after applying different feature selection techniques the accuracy improves. We implement the unsupervised feature selection named document frequency (DF) and by using that the accuracy improves by almost 12%. The clustering accuracy improves almost twice after applying supervised feature selections such as Information Gain (IG) and Subset Evaluation (SE) which are already implemented in WEKA [6].

Since DF is unsupervised and DF does not take into account the document category information, the clustering accuracy does not improve at a higher scale. On the other hand, Both IG and SE select features based on the document category information. We also combine the unsupervised feature selection technique with supervised ones that enhances the accuracy of document clustering than just using unsupervised and supervised techniques alone.

10 Conclusion and Future Work

Feature Selection Plays an important role for document clustering. It reduces the dimensionality of the documents by selecting only significant features. Since the number of features are reduced by feature selection technique, therefore it augments the scalability of clustering algorithms as well as improves their accuracy.

In the future, we plan to apply other feature selection techniques such as Term Strength, Entropy-based Ranking and Term Contribution in various clustering algorithms like Hierarchical and Spectral to see, whether the performance of this clustering algorithms improves or not on document clustering task.

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[2] Houtao Deng. Guided random forest in the rrf package. *arXiv:1306.0237*, 2013.

[3] Houtao Deng and George Runger. Feature selection via regularized trees. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012.

[4] Anna Drummond, Chris Jermaine, and Zografoula Vagena. Topic models for feature selection in document clustering. In *SDM*, pages 521–529. SIAM, 2013.

[5] Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44, February 2008.

[6] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[7] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[8] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[9] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[10] Philip J. Hayes, Peggy M. Anderson, Irene B. Nirenburg, and Linda M. Schmandt. TCS: A shell for content-based text categorization. In *IEEE Conference on Artificial Intelligence Applications*, 1990.

[11] Yeming Hu, Evangelos E. Milios, and James Blustein. Interactive feature selection for document clustering. In *In the 26th Symposium On Applied Computing*, pages 1148–1155, 2011.

[12] Luying Liu, Jianchu Kang, Jing Yu, and Zhongliang Wang. A comparative study on unsupervised feature selection methods for text clustering. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 597–601. IEEE, 2005.

[13] Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. An evaluation on feature selection for text clustering. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 488–495. AAAI Press, 2003.

[14] Georgios Paltoglou and Mike Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1386–1395, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[15] Pirooz Shamsinejadbabki and Mohammad Saraee. A new unsupervised feature selection method for text clustering based on genetic algorithms. *J. Intell. Inf. Syst.*, 38(3):669–684, June 2012.

[16] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.