Report on 20 Newsgroup Document –Classification & Clustering

|  |  |
| --- | --- |
| Version: | 0.1 |
| Date: | 21 Sept, 2016 |
| Author | Meenal Srivastava |
| Guide | Soumyakanti Chakraborty |

# Contents

[1 Contents 2](#_Toc462306916)

[1.1 List of Tables 3](#_Toc462306917)

[1.2 List of Figures 3](#_Toc462306918)

[2 Document Statements 6](#_Toc462306919)

[2.1 Purpose 6](#_Toc462306920)

[2.2 Assumptions and Dependencies 6](#_Toc462306921)

[2.3 In Scope 6](#_Toc462306922)

[2.4 Out of Scope 7](#_Toc462306923)

[2.5 Audience 7](#_Toc462306924)

[3 Introduction 8](#_Toc462306925)

[3.1 Project Overview 8](#_Toc462306926)

[3.2 Problem Statement 8](#_Toc462306927)

[3.3 Methodology 8](#_Toc462306928)

[3.4 Metrics 10](#_Toc462306929)

[4 Analysis 11](#_Toc462306930)

[4.1 The 20 Newsgroups Data Set 11](#_Toc462306931)

[4.1.1 Organization of dataset 11](#_Toc462306932)

[4.1.2 Data Exploration 11](#_Toc462306933)

[4.1.3 Data Pre-processing 22](#_Toc462306934)

[4.1.4 Category Wise - WordCloud 25](#_Toc462306935)

[4.1.5 Association 38](#_Toc462306936)

[4.2 Classification 43](#_Toc462306937)

[4.2.1 Multinomial Naïve Bayes 43](#_Toc462306938)

[4.2.2 Support Vector machines 46](#_Toc462306939)

[4.2.3 Max Entropy Model 49](#_Toc462306940)

[4.2.4 Decision Trees 50](#_Toc462306941)

[4.2.5 Random Forest 53](#_Toc462306942)

[4.2.6 KNN Model 55](#_Toc462306943)

[4.2.7 Neural Network Model 55](#_Toc462306944)

[4.2.8 Results 55](#_Toc462306945)

[4.2.9 Further work 56](#_Toc462306946)

[4.3 Clustering 57](#_Toc462306947)

[4.3.1 Evaluation and assessment 58](#_Toc462306948)

[4.3.2 Hierarchical Clustering 59](#_Toc462306949)

[4.3.3 K means Clustering 63](#_Toc462306950)

[4.3.4 SK means 66](#_Toc462306951)

[4.3.5 Conclusion 69](#_Toc462306952)

## List of Tables

[Table 1 : Vocabulary Size 9](#_Toc462306953)

[Table 2 : Vocab Size in diff sparsity levels 9](#_Toc462306954)

[Table 3 : Partitioned News Groups 15](#_Toc462306955)

[Table 4 : Train Documents 16](#_Toc462306956)

[Table 5 : Test Documents 18](#_Toc462306957)

[Table 6 : Naive Bayes at different Sparsity level for TF & TFIDF 43](#_Toc462306958)

[Table 7 : Naive Bayes Accuracy for TF IDF n gram DTM 45](#_Toc462306959)

[Table 8 : Linear SVM at 0.99 Sparsity 47](#_Toc462306960)

[Table 9 : Radial SVM at 0.99 Sparsity 48](#_Toc462306961)

[Table 10 : Maxent at 0.99 Sparsity 49](#_Toc462306962)

[Table 11 : Decision Tree at 0.99 Sparsity 51](#_Toc462306963)

[Table 12 : Random Forest at 0.99 Sparsity 53](#_Toc462306964)

[Table 13 : Comparison of Accuracy for all Classification Algorithms 55](#_Toc462306965)

[Table 14 : Hierarchical Clustering - 6 Clusters 60](#_Toc462306966)

[Table 15 : Hierarchical Clustering - 9 Clusters 61](#_Toc462306967)

[Table 16 : Hierarchical Clustering - 10 Clusters 62](#_Toc462306968)

[Table 17 : Hierarchical Clustering - 20 Clusters 63](#_Toc462306969)

[Table 18 : SK Means - 10 cluster 66](#_Toc462306970)

[Table 19 : SK Means PV 10 Cluster 67](#_Toc462306971)

[Table 20:SK Means PV 20 Cluster 1 69](#_Toc462306972)

## List of Figures

[Figure 1 : No. of Docs – Train Data 17](#_Toc462307018)

[Figure 2 : No. of Docs – Test Data 19](#_Toc462307019)

[Figure 3 : Lines per Doc - Train Dataset 20](#_Toc462307020)

[Figure 4 : Lines per doc - Test dataset 21](#_Toc462307021)

[Figure 5 : Category - Line per doc 22](#_Toc462307022)

[Figure 6 : Word cloud - Category alt.atheism 27](#_Toc462307023)

[Figure 7 : Word cloud - Category soc.religion.christian 28](#_Toc462307024)

[Figure 8 : Word cloud - talk.religion.misc comp.graphics 28](#_Toc462307025)

[Figure 9 : Word cloud - Category comp.graphics 29](#_Toc462307026)

[Figure 10 : Word Cloud - comp.os.ms-windows.misc 30](#_Toc462307027)

[Figure 11 : Word Cloud - comp.sys.ibm.pc.hardware 30](#_Toc462307028)

[Figure 12 : Word Cloud - comp.sys.mac.hardware 31](#_Toc462307029)

[Figure 13 : Word Cloud - comp.windows.x 31](#_Toc462307030)

[Figure 14 : Word Cloud - rec.autos 32](#_Toc462307031)

[Figure 15 : Word Cloud - rec.motorcycles 32](#_Toc462307032)

[Figure 16 : Word Cloud - rec.sport.baseball 33](#_Toc462307033)

[Figure 17 : Word Cloud - rec.sport.hockey 33](#_Toc462307034)

[Figure 18 : Word Cloud - sci.crypt 34](#_Toc462307035)

[Figure 19 : Word Cloud - sci.electronics 34](#_Toc462307036)

[Figure 20 : Word Cloud - sci.med 35](#_Toc462307037)

[Figure 21 : Word Cloud - sci.space 35](#_Toc462307038)

[Figure 22 : Word Cloud - talk.politics.guns 36](#_Toc462307039)

[Figure 23 : Word Cloud - talk.politics.mideast 36](#_Toc462307040)

[Figure 24: Word Cloud - talk.politics.misc 37](#_Toc462307041)

[Figure 25 : Word Cloud - misc.forsale 37](#_Toc462307042)

[Figure 26 : Association "Drive" 38](#_Toc462307043)

[Figure 27 : Association "File" 39](#_Toc462307044)

[Figure 28 : Association "God" 39](#_Toc462307045)

[Figure 29 : Association "Game" 40](#_Toc462307046)

[Figure 30 : Association "Jesus" 40](#_Toc462307047)

[Figure 31 : Association Player 41](#_Toc462307048)

[Figure 32 : Association Program 41](#_Toc462307049)

[Figure 33 : Association "Team" 42](#_Toc462307050)

[Figure 34 : Association "Christian" 42](#_Toc462307051)

[Figure 35 : Naive Bayes Accuracy for TF IDF DTM 44](#_Toc462307052)

[Figure 36 : Naive Bayes Accuracy for TF DTM 44](#_Toc462307053)

[Figure 37 : Accuracy/Precision/Recall for Linear SVM at 0.99 Sparse Matrix 47](#_Toc462307054)

[Figure 38 : Accuracy/Precision/Recall for Radial SVM at 0.99 Sparse Matrix 48](#_Toc462307055)

[Figure 39 : Accuracy/Precision/Recall for Maxent 50](#_Toc462307056)

[Figure 40 : Accuracy/Precision/Recall for Decision Tree at 0.99 Sparse Matrix 51](#_Toc462307057)

[Figure 41 : Accuracy/Precision/Recall for Random Forest at 0.99 Sparse Matrix 54](#_Toc462307058)

[Figure 42 : Comparison of Accuracy/Precision/Recall 56](#_Toc462307059)

[Figure 43 : Dendogram for Hierarchical Clustering 59](#_Toc462307060)

[Figure 44 : detrmining no of clusters in K Means Clustering 64](#_Toc462307061)

[Figure 45 : K Means Clusplot for 10 cluster kmeans 64](#_Toc462307062)

# Document Statements

The following sections describe the content of the document in terms of the following:

* Purpose. Explains why the document has been produced, providing a summary of the reasons and goals
* Assumption. Something held to be true to allow a project to proceed
* Dependency. A mandatory output from one project or piece of work that is required as an input for another project or piece of work
* In Scope. A summary of the areas covered by the document
* Out of Scope. Explicitly identifies any areas that are not covered by the document
* Audience. The type of reader expected to use this document
* Conventions. Typographical conventions used to clearly identify specific types of information

## Purpose

The purpose of this document is to demonstrate use of classification & clustering in text mining in predicting categories of the documents.

## Assumptions and Dependencies

The following assumptions have been made in the writing of this document:

* Dataset “ 20 Newsgroup” has been taken assuming that the data available in train dataset has been correctly labelled and can be used to train the models.
* It is assumed that the dataset “ 20 Newsgroup” can be clearly demarcated and grouped under 20 categories based on the contents in the documents.

Following dependencies have been identified in the writing of this document:

* Size of the dataset for test & train data had a dependency on RAM (memory) of the machine being used for project. So for most of the algorithms, I was forced to take a smaller dataset that can be accommodated in available RAM.

## In Scope

The scope of this document is:

* Analysis of the “20 Newsgroup” data
* Major classification algorithms and applying them on “20 Newsgroup” data
* Evaluation metrics involved in classification algorithms
* Clustering of “20 Newsgroup” data

## Out of Scope

The following items are considered outside the scope of this document:

* Due to limited time, fine tuning of the algorithms involved in classification which may further enhance the results, could not be done.

## Audience

As the document is the project report for “Certificate program in Data Analysis in Excel and R” from XLRI, Jamshedpur, so it can be consumed by the people involved in training program or all students to help them with knowledge on the topic.

# Introduction

## Project Overview

Document classification or document categorization is a problem of library science, information science as well as computer science. The task is to assign a document to a class. This may be done "manually" or algorithmically and the documents to be classified may be texts, images, music, etc. Each kind of document possesses its special classification problems.

Text classification is the process of classifying documents into predefined categories based on their content/subject or according to other attributes (such as document type, author, printing year etc.). It is the automated assignment of natural language texts to predefined categories. Text classification is the primary requirement of text retrieval systems, which retrieve texts in response to a user query, and text understanding systems, which transform text in some way such as producing summaries, answering questions or extracting data.

This project will focus both on

1. supervised machine learning algorithms for automatic classification of text documents in categories where some external mechanism (such as human feedback) provides information on the correct classification for documents and
2. Unsupervised document classification (document clustering) -hierarchical and non-hierarchical clustering of these documents. where the classification must be done entirely without reference to external information

I will apply these classification/clustering algorithms on 20 newsgroup dataset which is a collection of 20,000 messages, collected from 20 different newsgroups. The news will be classified according to their contents.

## Problem Statement

In the 20\_news group dataset, there are news of 20 categories, each piece of news belongs to one category, the goal is to extract proper features and build an effective model to assign each piece of news to the correct category.

## Methodology

I will explore the dataset in the beginning on the training part, then extract useful keywords and build vectors of features from the texts of news and based on those vectors I will use several classification methods to do classification, compare the efficiency of these classifiers on the testing data and choose one as final model.

For the supervised classification, I plan to use classification algorithms such as - Naive Bayes, Decision Tree, Maxent , SVM and Neural Networks (all of which have their advantages and disadvantages) and compare their performance.

Classification can be hierarchical and flat and also one or multi label. This is an example of **flat classification with one label** for each documents**.**

I also plan to cluster the documents both flat and hierarchical clustering (ward’s method).

Clustering algorithms may be classified as listed below

1. **Flat clustering** (Creates a set of clusters without any explicit structure that would relate clusters to each other; It’s also called exclusive clustering)
2. **Hierarchical clustering** (Creates a hierarchy of clusters)
3. **Hard clustering** (Assigns each document/object as a member of exactly one cluster)
4. **Soft clustering** (Distribute the document/object over all clusters)

**Algorithms I will use are :**Agglomerative (Hierarchical clustering) and K-Means, pamk, skmeans (Flat clustering, Hard clustering)

In this project, if I take the entire vocab into account, it will be a huge computing and memory demand for my computer, so as to speed computing, I have restricted the vocab size.

While generating the document term matrix, the vocab size if no word lengths or minimum number of documents in which the term is appearing is given, is 74920 unique words ( after stemming the words and removing the stop words).So I explored vocab size with various options and then after computing the time/memory required to run one algorithm, concluded a vocab size of around 2000 was OK. So I worked with 0.99 sparse matrix. i.e 1809 terms in the training set and 1766 terms in the test set

|  |  |  |
| --- | --- | --- |
| Min/Max Word Length | Min No of documents in which they appear | Vocab size |
| 3/18 | 2 | 35205 |
| 3/15 | 5 | 23367 |
| Not specified | 2 | 35239 |
| Not specified | Not specified | 74,920 |

Table 1 : Vocabulary Size

|  |  |
| --- | --- |
| Sparsity | Vocab Size |
| 0.99 | 1809 |
| 0.995 | 3039 |
| 0.999 | 8796 |

Table 2 : Vocab Size in diff sparsity levels

## Metrics

As it is a multi-class classification problem, I have use below metrics listed after creating a confusion matrix

1. Accuracy: the proportion of correct labels that we made if we apply our model to the training dataset. Ideal accuracy is 100%.

Accuracy= (TP+TN)/(TP+FN+FP+TN)

1. Precision :which indicates how many of the items that we identified were relevant,

Precision=TP/(TP+FP).

1. Recall: which indicates how many of the relevant items that we identified,

Recall=TP/(TP+FN)

1. The F-Measure (or F-Score), which combines the precision and recall to give a single score, is defined to be the harmonic mean of the precision and recall:

(2 × *Precision* × *Recall*) / (*Precision* + *Recall*).

# Analysis

## The 20 Newsgroups Data Set

### Organization of dataset

The data is organized into 20 different newsgroups, each corresponding to one of 20 different topic. One thousand messages from each of the twenty newsgroups were chosen at random and partitioned by newsgroup name. Some of the newsgroups are very closely related to each other (e.g. **comp.sys.ibm.pc.hardware / comp.sys.mac.hardware**), while others are highly unrelated (e.g **misc.forsale / soc.religion.christian**).

### Data Exploration

#### Importing the dataset in ‘R’

Firstly, I created two separate corpus for ‘Train’ and Test sets and also created a separate vector for the categories to which each of the document belongs.

#### Exploring one of the documents in the dataset randomly to get an idea of layout.

Below is one of the( ninth) documents of the first topic alt.atheism

From: keith@cco.caltech.edu (Keith Allan Schneider)

Subject: Re: <Political Atheists?

Organization: California Institute of Technology, Pasadena

Lines: 54

NNTP-Posting-Host: punisher.caltech.edu

(reference line trimmed)

livesey@solntze.wpd.sgi.com (Jon Livesey) writes:

[...]

>There is a good deal more confusion here. You started off with the

>assertion that there was some "objective" morality, and as you admit

>here, you finished up with a recursive definition. Murder is

>"objectively" immoral, but eactly what is murder and what is not itself

>requires an appeal to morality.

Yes.

>Now you have switch targets a little, but only a little. Now you are

>asking what is the "goal"? What do you mean by "goal?". Are you

>suggesting that there is some "objective" "goal" out there somewhere,

>and we form our morals to achieve it?

Well, for example, the goal of "natural" morality is the survival and

propogation of the species. Another example of a moral system is

presented within the Declaration of Independence, which states that we

should be guaranteed life liberty and the pursuit of happiness. You see,

to have a moral system, we must define the purpose of the system. That is,

we shall be moral unto what end?

>>Murder is certainly a violation of the golden rule. And, I thought I had

>>defined murder as an intentional killing of a non-murderer, against his will.

>>And you responded to this by asking whether or not the execution of an

>>innocent person under our system of capital punishment was a murder or not.

>>I fail to see what this has to do with anything. I never claimed that our

>>system of morality was an objective one.

>I thought that was your very first claim. That there was

>some kind of "objective" morality, and that an example of that was

>that murder is wrong. If you don't want to claim that any more,

>that's fine.

Well, murder violates the golen rule, which is certainly a pillar of most

every moral system. However, I am not assuming that our current system

and the manner of its implementation are objectively moral. I think that

it is a very good approximation, but we can't be perfect.

>And by the way, you don't seem to understand the difference between

>"arbitrary" and "objective". If Keith Schneider "defines" murder

>to be this that and the other, that's arbitrary. Jon Livesey may

>still say "Well, according to my personal system of morality, all

>killing of humans against their will is murder, and wrong, and what

>the legal definition of murder may be in the USA, Kuweit, Saudi

>Arabia, or the PRC may be matters not a whit to me".

Well, "objective" would assume a system based on clear and fundamental

concepts, while "arbitary" implies no clear line of reasoning.

Keith

**My observation:**

On exploring the dataset randomly for different classes, it is clear:

a) All docs are an email, there is a subject, an organization, an email address and a main body.

b) The subject and content can contain some useful words in terms of classification, for example, this article mentioned 'atheist', 'morality'.

c) There are many signs such as punctuation, digits in the content, which may be redundant for classification.

d) There are spelling mistakes, in this example ‘exactly’ is misspelled as ‘eactly’, ‘golden’ as ‘golen’ etc.

#### Dataset Partitioning

Below is a list of the 20 newsgroups, partitioned (more or less) according to subject matter:

|  |  |  |
| --- | --- | --- |
| comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x | rec.autos rec.motorcycles | sci.crypt sci.electronics sci.med sci.space |
| rec.sport.baseball rec.sport.hockey | talk.politics.misc talk.politics.guns talk.politics.mideast | talk.religion.misc alt.atheism soc.religion.christian |
| misc.forsale |  |  |

Table 3 : Partitioned News Groups

#### Partition volumes

List for number of documents per topics for training and test set are as below:

| S.No. | Train Category | No. of Documents |
| --- | --- | --- |
| 1 | alt.atheism | 480 |
| 2 | comp.graphics | 584 |
| 3 | comp.os.ms-windows.misc | 591 |
| 4 | comp.sys.ibm.pc.hardware | 590 |
| 5 | comp.sys.mac.hardware | 578 |
| 6 | comp.windows.x | 593 |
| 7 | misc.forsale | 585 |
| 8 | rec.autos | 594 |
| 9 | rec.motorcycles | 598 |
| 10 | rec.sport.baseball | 597 |
| 11 | rec.sport.hockey | 600 |
| 12 | sci.crypt | 595 |
| 13 | sci.electronics | 591 |
| 14 | sci.med | 594 |
| 15 | sci.space | 593 |
| 16 | soc.religion.christian | 599 |
| 17 | talk.politics.guns | 546 |
| 18 | talk.politics.mideast | 564 |
| 19 | talk.politics.misc | 465 |
| 20 | talk.religion.misc | 377 |

Table 4 : Train Documents

Following graph represents the number of documents category wise in train dataset

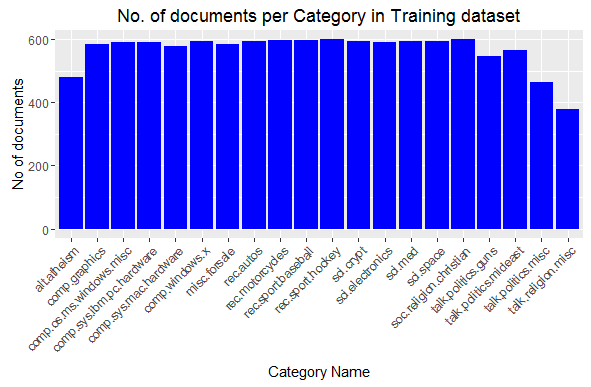


Figure 1 : No. of Docs – Train Data

| S.No. | Test Category | No. of Documents |
| --- | --- | --- |
| 1 | alt.atheism | 319 |
| 2 | comp.graphics | 389 |
| 3 | comp.os.ms-windows.misc | 394 |
| 4 | comp.sys.ibm.pc.hardware | 392 |
| 5 | comp.sys.mac.hardware | 385 |
| 6 | comp.windows.x | 395 |
| 7 | misc.forsale | 390 |
| 8 | rec.autos | 396 |
| 9 | rec.motorcycles | 398 |
| 10 | rec.sport.baseball | 397 |
| 11 | rec.sport.hockey | 399 |
| 12 | sci.crypt | 396 |
| 13 | sci.electronics | 393 |
| 14 | sci.med | 396 |
| 15 | sci.space | 394 |
| 16 | soc.religion.christian | 398 |
| 17 | talk.politics.guns | 364 |
| 18 | talk.politics.mideast | 376 |
| 19 | talk.politics.misc | 310 |
| 20 | talk.religion.misc | 251 |

Table 5 : Test Documents

Following graph represents the number of documents category wise in test dataset

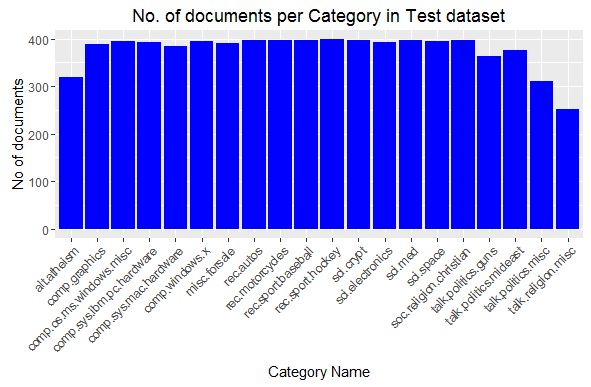


Figure 2 : No. of Docs – Test Data

Each category has almost the same number of articles, both the training and testing data are balanced. We need not reshuffle and split them.

#### Size of Documents

Based on the number of lines in the document, following graphs depicts number of documents in train data along with line count

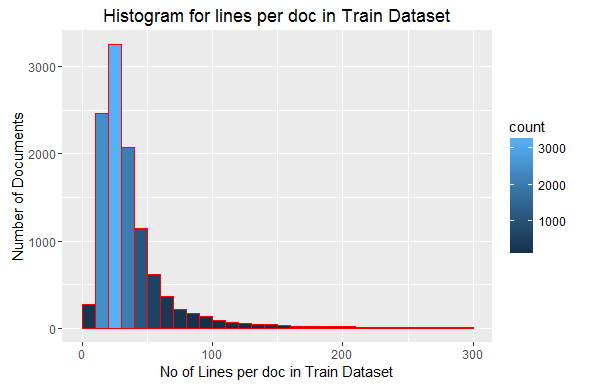


Figure 3 : Lines per Doc - Train Dataset

Above graph indicates that maximum number of documents are in the category of 12-50 lines per doc with a peak on around 25 lines per doc. After 50 lines per doc the count of documents decreases.

Following graph represents the number of documents based on number of lines in test data

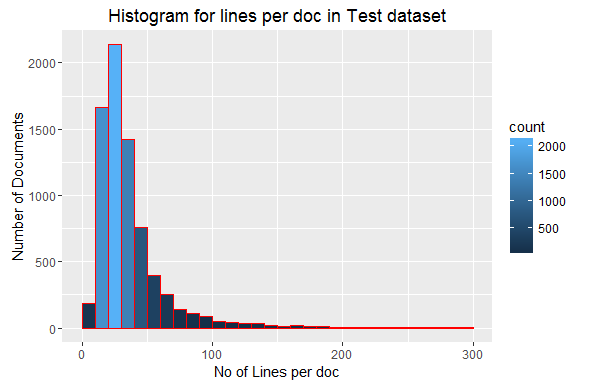


Figure 4 : Lines per doc - Test dataset

Similar to test data, here also maximum number of documents have 12-50 lines per doc.

Most articles have less than 50 lines, the median is 29, so these news consist of quite short articles. Mean and median are quite different which shows that few documents are very long (outliers)

#### Document Size – Category wise

Looking further into the data - category wise, following graph shows the number of documents with line count for each category. This graphs helps in understanding the similarity across the categories in terms of size of the documents. Each category has got maximum number of documents in 0-50 line count category after which it decreases.



Figure 5 : Category - Line per doc

### Data Pre-processing

#### Cleaning the train and test set corpus

As this is a large text document, the first step is to create a vocabulary list or a ‘dictionary’ that contains all the words in the documents for both train and test set. To do this:

1. I removed all characters except alpha numeric and then converted the text to lower case.
2. I removed punctuations and numbers from the text.
3. Then I removed the stop words (common words in English language) and also added a few of my own words (eg. "com","edu","nntp","etc","lines") which I did not find useful so that only the useful words useful for classification will be left.
4. I used two approaches: one with stemming and the other analysis is without stemming the words. Another option was to use lemmatisation instead of stemming.

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. *Stemming* usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma* .

1. The words left after the above pre-processing were compiled in a single text document one for training set and the second for test set.

Inspecting the same document 9 for the first category (which we inspected earlier) after processing the above steps

keith cco caltech keith allan schneider

subject political atheists

organization california institute technology pasadena

posting host punisher caltech

reference line trimmed

livesey solntze wpd sgi jon livesey writes

good deal confusion started

assertion objective morality admit

finished recursive definition murder

objectively immoral eactly murder

requires appeal morality

yes

now switch targets little little now

asking goal mean goal

suggesting objective goal somewhere

form morals achieve

well example goal natural morality survival

propogation species another example moral system

presented within declaration independence states

guaranteed life liberty pursuit happiness see

moral system must define purpose system

shall moral unto end

murder certainly violation golden rule thought

defined murder intentional killing non murderer will

responded asking whether execution

innocent person system capital punishment murder

fail see anything never claimed

system morality objective one

thought first claim

kind objective morality example

murder wrong don t want claim

s fine

well murder violates golen rule certainly pillar

every moral system however assuming current system

manner implementation objectively moral think

good approximation can t perfect

way don t seem understand difference

arbitrary objective keith schneider defines murd

s arbitrary jon livesey may

still say well according personal system morality

killing humans will murder wrong

legal definition murder may usa kuweit saudi

arabia prc may matters whit

well objective assume system based clear fundament

concepts arbitary implies clear line reasoning

keith

#### Generating Document Term Matrix

For the classification/clustering of documents, we need to create a document term matrix

1. To avoid spelling mistakes and also to reduce the size, I considered only words which appear in minimum of 2 documents.
2. Next was to select a method to count the terms in each document
3. First approach is Term Frequency-TF : A frequency count is performed for all the words in all the documents to form a dictionary (Bag of Words) for the whole set.
4. Second approach is TFIDF: Tf-idf stands for term frequency-inverse document frequency. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

The default Unique words after performing either of the above process is 74,920 words.

#### Feature Selection

Due to high dimension of features, I faced computation memory problems. So,I had to reduce the dictionary size by removing sparse terms (0.99) to appx 1805 words and used it on all algorithms.

For, SVM and Maxent, I also tested the accuracy on 23,000 most important words to check accuracy on increasing the vocab size. For the other algorithms, I tested on a larger vocab size of 8769 words with the best performing parameter from the lower vocab size, to asses the performance of the algorithms on increasing the size.For Naive Bayes, I will be testing for different sizes of BOW as there are no other parameters that can be tuned in the algorithms.

### Category Wise - WordCloud

Word cloud for each of the category showing the important words and occurrence frequency (size) is shown in following subsection.

Based on the word cloud, five key words for each category are as following

| S.No. | Category | Keywords |
| --- | --- | --- |
| 1 | alt.atheism | Atheist, God, Keith, Islam, Caltech |
| 2 | soc.religion.christian | Christian, God, Jesus, Church, Bible |
| 3 | talk.religion.misc | Christian, God, Jesus, moral, kent |
| 4 | comp.graphics | graphic, image, file, program, format |
| 5 | comp.os.ms-windows.misc | window, file, driver, dos, max |
| 6 | comp.sys.ibm.pc.hardware | SCSI, Drive, Ide, Card, motherboard, |
| 7 | comp.sys.mac.hardware | apple, mac, drive, monitor, modem |
| 8 | comp.windows.x | window, motif, server, display, program |
| 9 | rec.autos | car, dealer, auto, engine, oil |
| 10 | rec.motorcycles | bike, ride, motorcycle, dod, bmw |
| 11 | rec.sport.baseball | baseball, team, game, player, pitch |
| 12 | rec.sport.hockey | hockey, game, team, play, player |
| 13 | sci.crypt | key, encrypt, clipper, escrow, chip |
| 14 | sci.electronics | circuit, wire, use, electron, power |
| 15 | sci.med | pitt, gordon, bank, doctor, medic |
| 16 | sci.space | space, orbit, nasa, henri, moon |
| 17 | talk.politics.guns | gun, firearm, weapon, fbi, crime |
| 18 | talk.politics.mideast | israel, armenian, arab, turkish, jew, isra |
| 19 | talk.politics.misc | homosexu, tax, clinton, state, govern |
| 20 | misc.forsale | sale, offer, ship, price, sell |

Common keywords across categories again establishes inter-relationships between related categories and the fact that they lie in the same partition. Most occurring words in above keywords will be considered to analyse association covered in next section.

In the next section association of the most occurring words in above keywords will be picked which will be later used to analyse and depict association

#### Word Cloud - alt.atheism



Figure 6 : Word cloud - Category alt.atheism

#### Word Cloud - soc.religion.christian

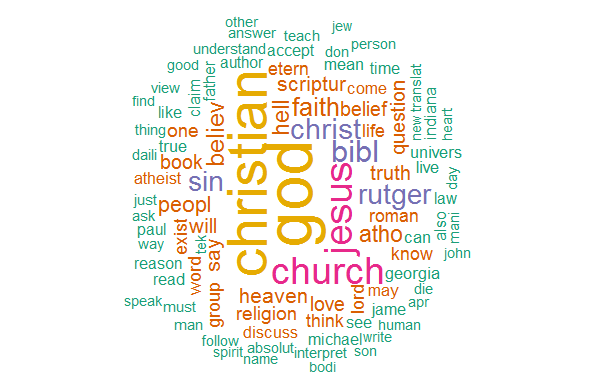


Figure 7 : Word cloud - Category soc.religion.christian

#### Word Cloud - talk.religion.misc comp.graphics



Figure 8 : Word cloud - talk.religion.misc comp.graphics

#### Word Cloud - comp.graphics

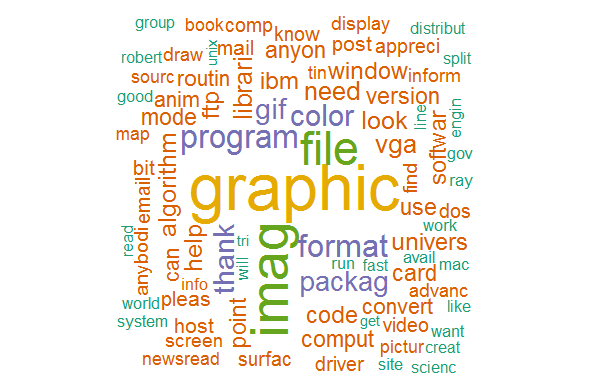


Figure 9 : Word cloud - Category comp.graphics

#### Word Cloud - comp.os.ms-windows.misc

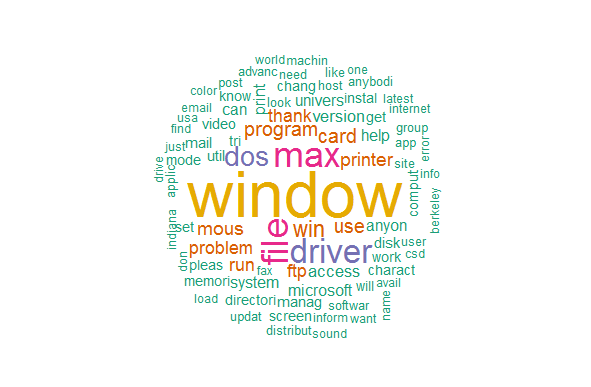


Figure 10 : Word Cloud - comp.os.ms-windows.misc

#### Word Cloud - comp.sys.ibm.pc.hardware



Figure 11 : Word Cloud - comp.sys.ibm.pc.hardware

#### Word Cloud - comp.sys.mac.hardware

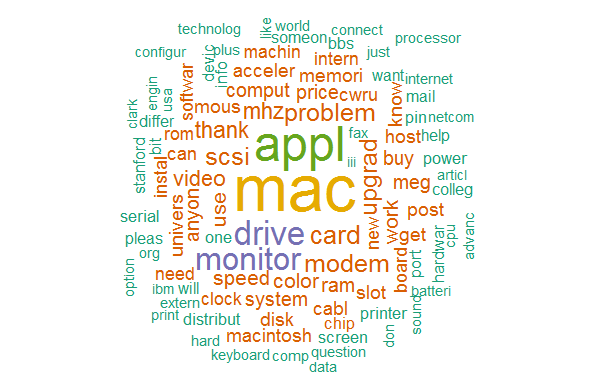


Figure 12 : Word Cloud - comp.sys.mac.hardware

#### Word Cloud - comp.windows.x

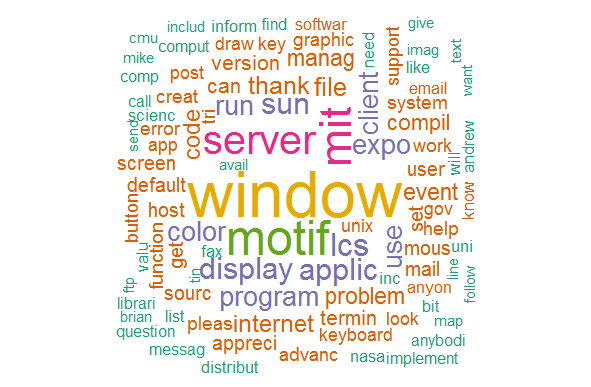


Figure 13 : Word Cloud - comp.windows.x

#### Word Cloud - rec.autos

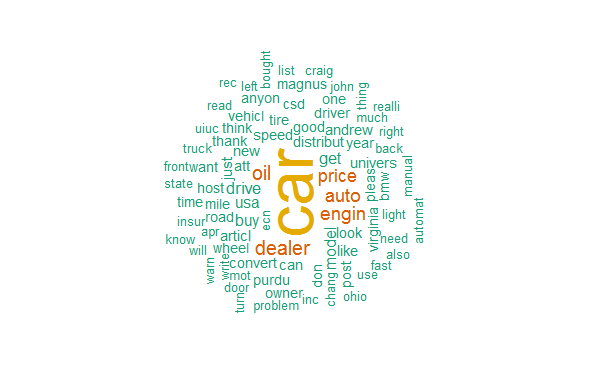


Figure 14 : Word Cloud - rec.autos

#### Word Cloud - rec.motorcycles

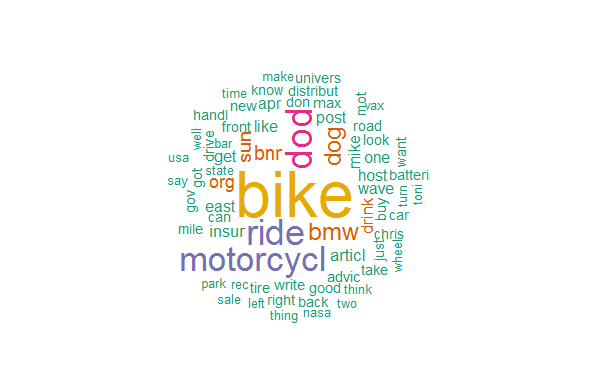


Figure 15 : Word Cloud - rec.motorcycles

#### Word Cloud - rec.sport.baseball

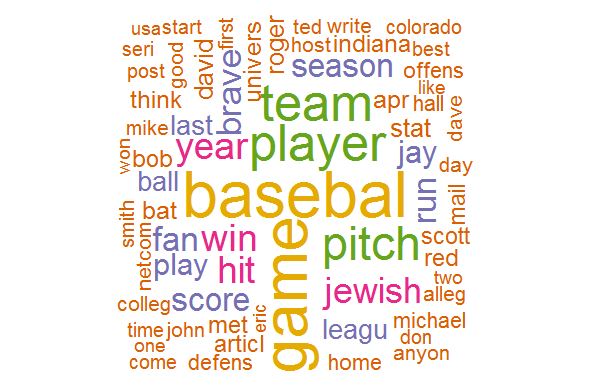


Figure 16 : Word Cloud - rec.sport.baseball

#### Word Cloud - rec.sport.hockey



Figure 17 : Word Cloud - rec.sport.hockey

#### Word Cloud - sci.crypt

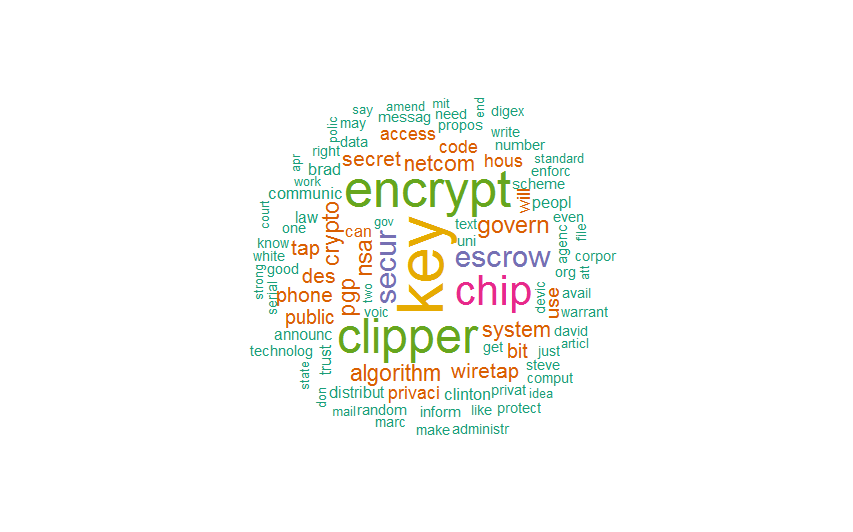


Figure 18 : Word Cloud - sci.crypt

#### Word Cloud - sci.electronics

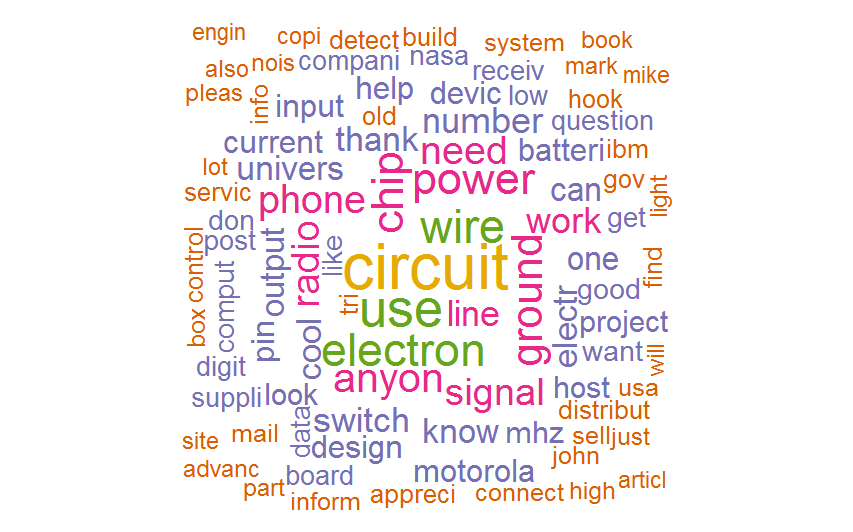


Figure 19 : Word Cloud - sci.electronics

#### Word Cloud - sci.med

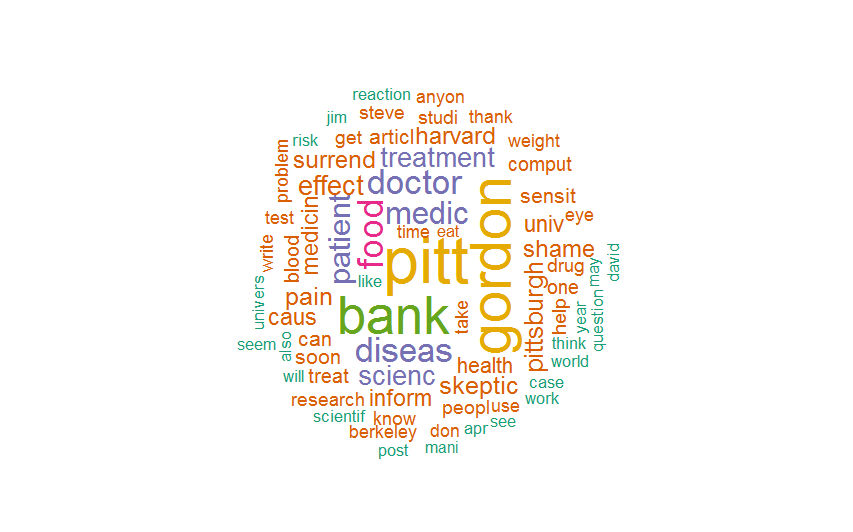


Figure 20 : Word Cloud - sci.med

#### Word Cloud - sci.space

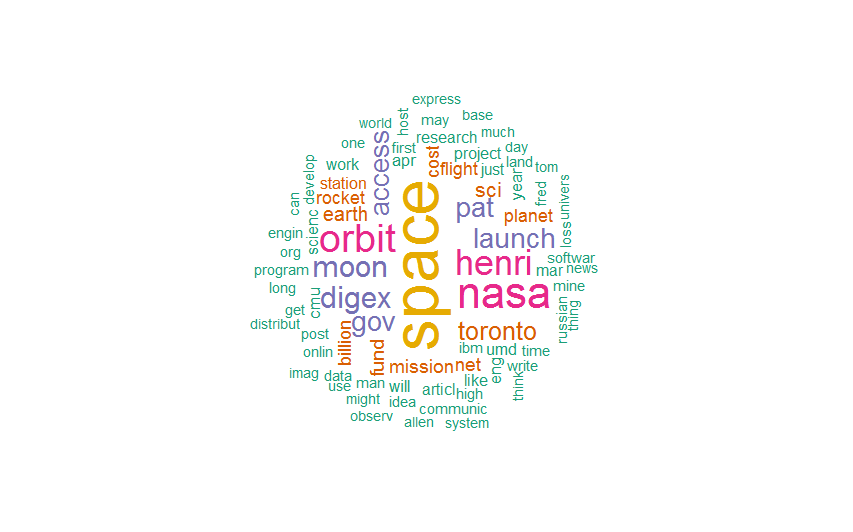


Figure 21 : Word Cloud - sci.space

#### Word Cloud - talk.politics.guns

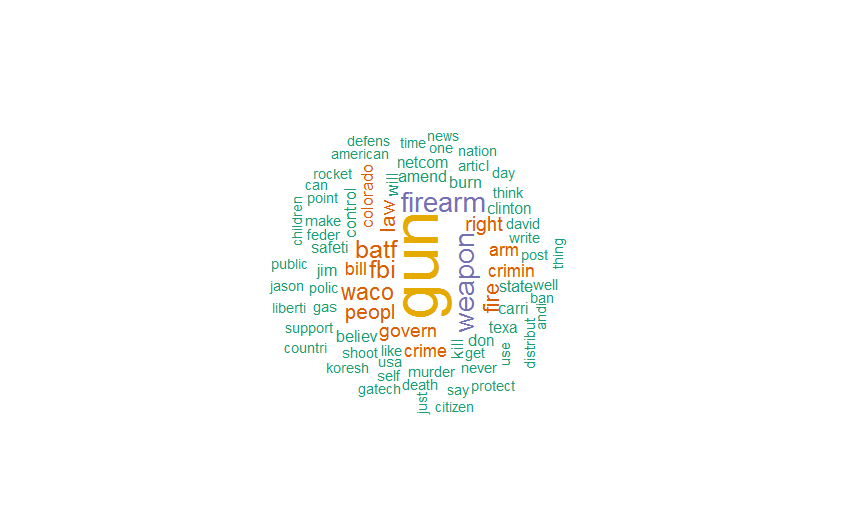


Figure 22 : Word Cloud - talk.politics.guns

#### Word Cloud - talk.politics.mideast

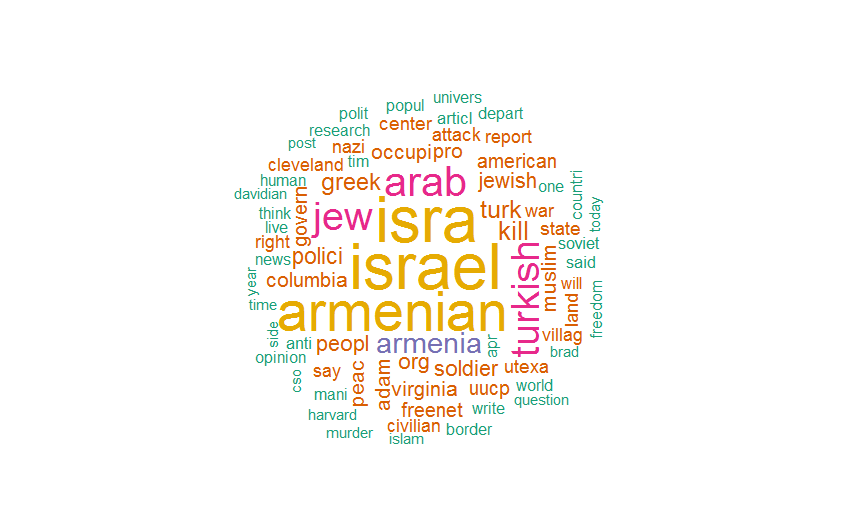


Figure 23 : Word Cloud - talk.politics.mideast

#### Word Cloud - talk.politics.misc

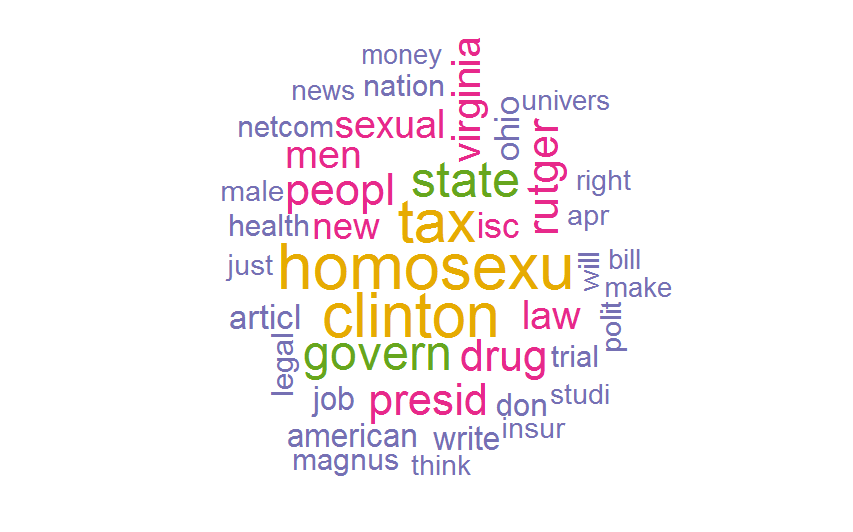


Figure 24: Word Cloud - talk.politics.misc

#### Word Cloud - misc.forsale



Figure 25 : Word Cloud - misc.forsale

### Association

As described in section 4.1.4, most occurring words in keywords of word clouds were picked and association plot was done for each one of them as following

#### Association plot for word Drive

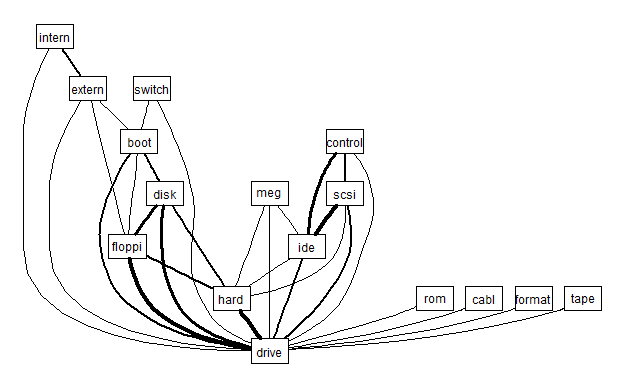


Figure 26 : Association "Drive"

#### Association plot for word File

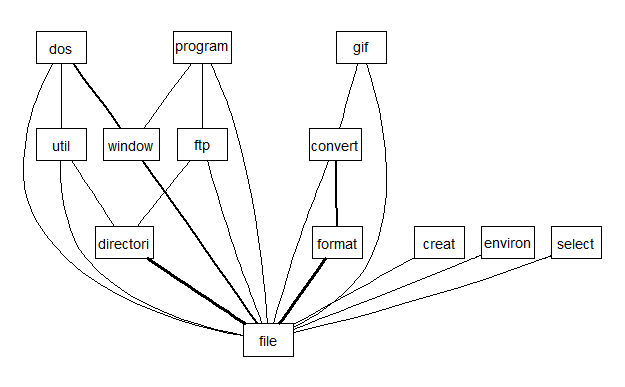


Figure 27 : Association "File"

#### Association plot for word God

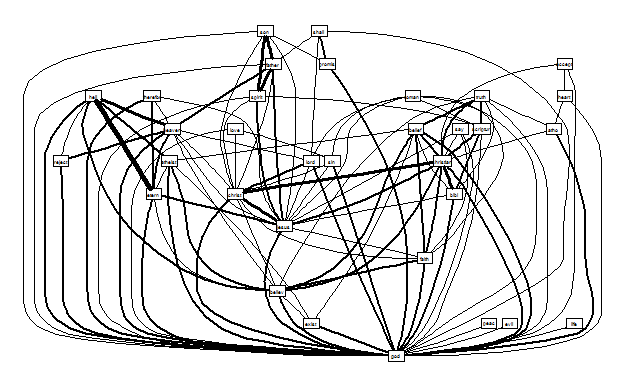


Figure 28 : Association "God"

#### Association plot for word Game

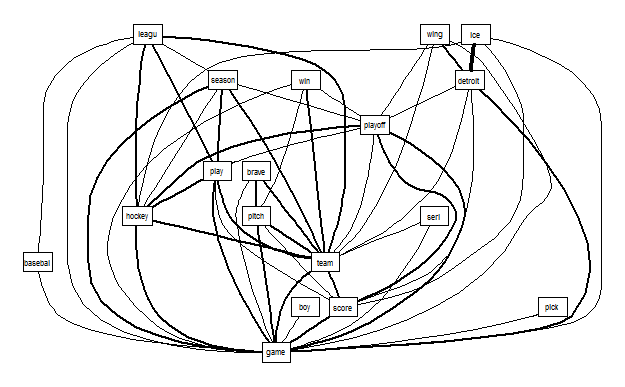


Figure 29 : Association "Game"

#### Association plot for word Jesus

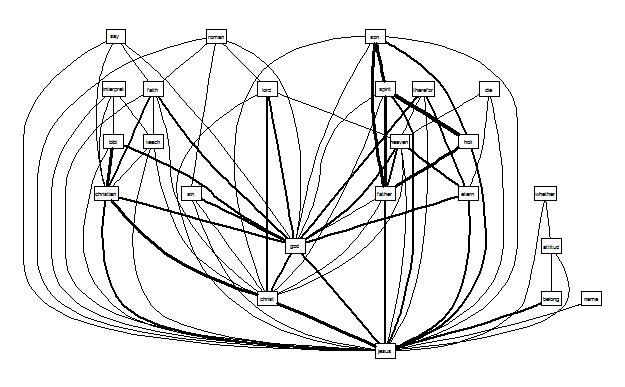


Figure 30 : Association "Jesus"

#### Association plot for word Player

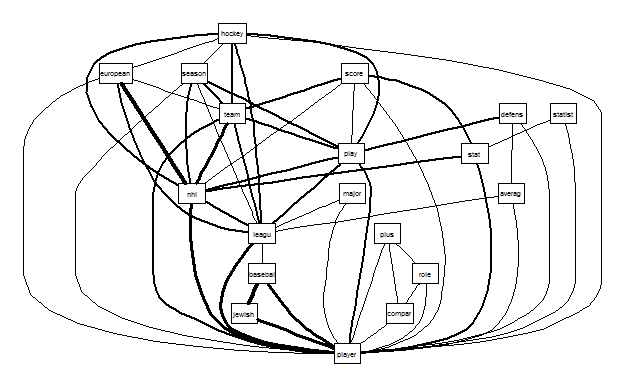


Figure 31 : Association Player

#### Association plot for word Program

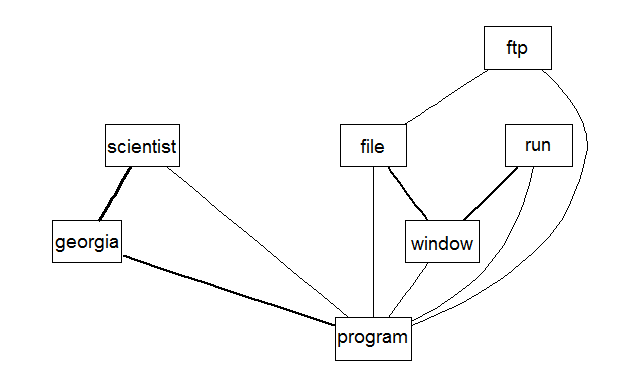


Figure 32 : Association Program

#### Association plot for word Team

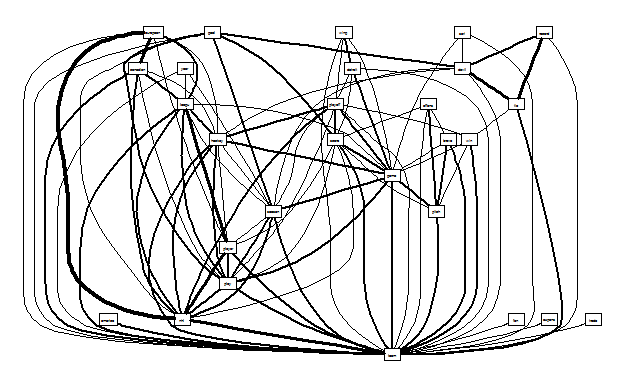


Figure 33 : Association "Team"

#### Association plot for word Christian

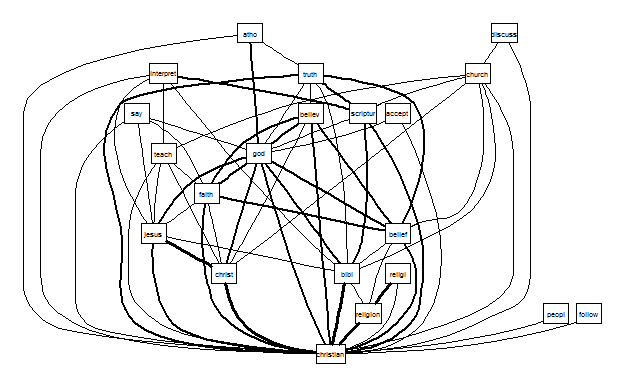


Figure 34 : Association "Christian"

## Classification

Different algorithms were tried for classifying the documents. I trained & tested all of these on a smaller vocab size (referred as “Small Vocab”) of 1805 words and then retested with the large vocab size (referred as “Large Vocab”) of 8796 words (with the algorithm parameters kept as the highest performing parameters of 1805 vocab size) and then compared.

All the algorithms were tested with frequency calculated as per TFIDF on Unigrams except for Naïve Bayes where frequency was calculated with both TF and TFIDF methods and n-grams were also tested.

I tested the SVM, Maxent and neural network algorithms through the R text tools package and others were tested through their respective packages.

### Multinomial Naïve Bayes

For Naïve Bayes, I tested with different vocab sizes as there are no parameters to tune. Also I tried with both unigrams and n -grams and also with both TFIDF and TF approaches.

Following steps were executed on the cleaned corpus :

* Created document term matrix with both TFIDF and TF for the training set for unigrams with different sparsity levels.
* Model was trained with Laplace smoothing of 1, hence ‘Terms’ for train and test set need not be same.
* Created document term matrix for unigrams with both TFIDF and TF approaches for test set as well.
* Predictions were generated by applying the trained model on the test dataset.
* Confusion matrix was generated and evaluation metrics were recorded
* The above process was repeated for generating and evaluating models with n grams as well. Models with 2, 3, 4 and 5 grams were trained and tested.



Table 6 : Naive Bayes at different Sparsity level for TF & TFIDF

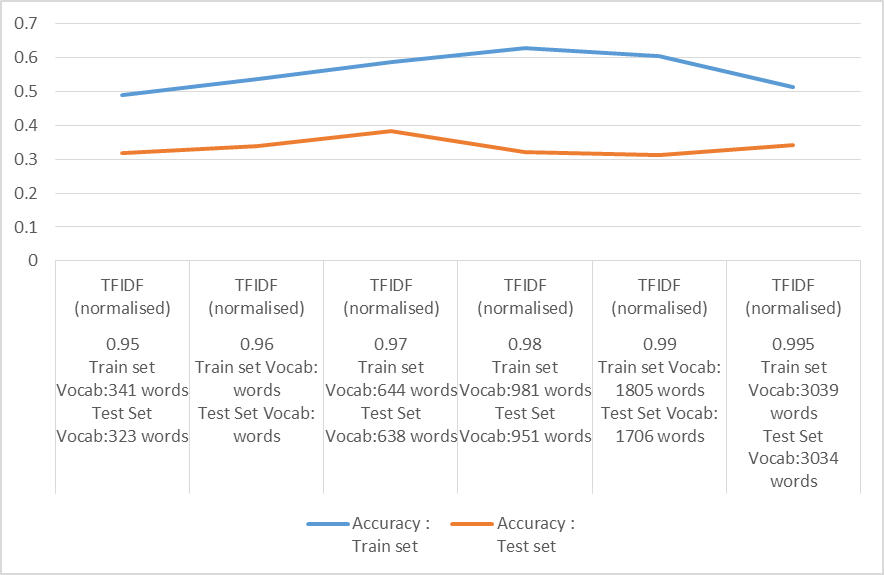


Figure 35 : Naive Bayes Accuracy for TF IDF DTM

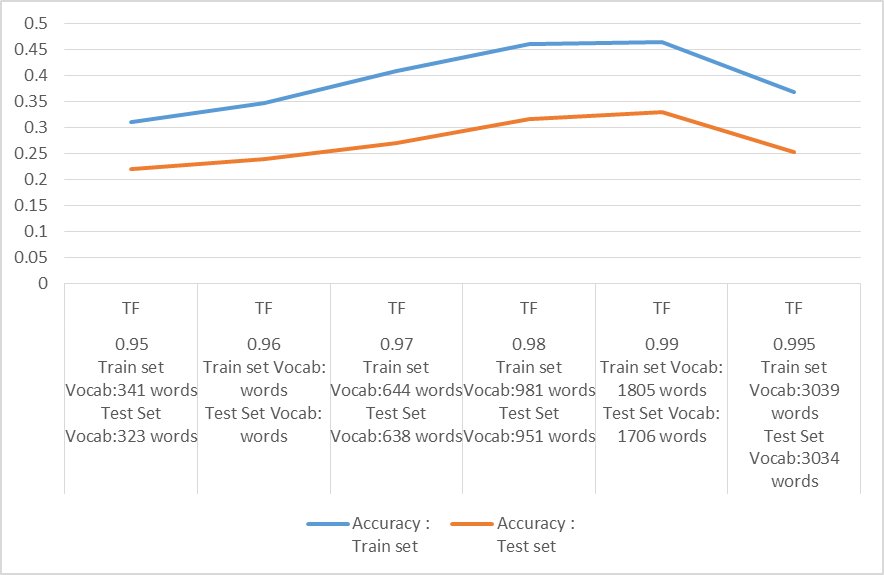


Figure 36 : Naive Bayes Accuracy for TF DTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Sparsity | train vocab size | test vocab size | Accuracy Train | Accuracy Test |
| 2 grams | 0.997 | 1910 | 1945 | 0.124 | 0.2373 |
| 3 grams | 0.998 | 952 | 933 | 0.0527 | 0.0652 |
| 4 grams | 0.9992 | 968 | 1176 | 0.0459 | 0.0521 |
| 5 grams | 0.9998 | 2255 | 2370 | 0.056 | 0.1373 |

Table 7 : Naive Bayes Accuracy for TF IDF n gram DTM

The overall accuracy of Naïve Bayes was low as compared to other algorithms like SVM and Maxent models. The algorithm was not able to distinguish between similar categories for eg. alt.atheism, talk.religion and soc.christian which have overlapping subject materials and misclassified them leading to a low accuracy.

For Naïve Bayes, the accuracy was higher for TFIDF models then the corresponding TF models. Also unigrams had a higher accuracy rate than the n grams.

Next I tried the Naïve Bayes on “large Vocab” of 8796 words with best accuracy combination of the ‘small vocab’



Attaching below excel with the confusion matrix for all the above variations for all the vocab sizes.



Overall Analysis of NaiveBayes:

1) Overall accuracy for Naïve bayes was not impressive.Highest accuracy on test data was 0.38 on 0.97 sparse matrix. The algorithm was not able to distinguish between similar categories for eg. alt.atheism, talk.religion and soc.christian which have overlapping subject materials and misclassified them leading to a low accuracy.

2) Accuracy was higher for TFIDF models then the corresponding TF models. Also unigrams had a higher accuracy rate than the n grams.

3) Also the algorithm was relatively slow to train as compared to SVM and Maxent models.

Methods which might enhance the performance:

Highest accuracy I could achieve on the test set with this algorithm was 0.38.This was mainly due to misclassification between related categories. Hence the model need to be fine-tuned to distinguish between related topics. This needs to be explored further.

I feel, accuracy can be increased by switching to ‘one vs all’ classification process where we classify only a single category as 1 and all the balance categories as 0.This is repeated for all the categories respectively.

Alternatively, we can follow some sort of hierarchical classification, where we first classify the documents as majorly 6-7 topics and then reclassify within each of these 6-7 topics.

Another method can be during feature selection process. During generation of vocabulary set (DTM) for training the algorithm, some sort of weighing of the terms need to be done. Some advanced statistic methods (e.g. Information Gain, Mutual Information, χ2 statistics) need to be applied so that better key words representative of the documents, are generated.

### Support Vector machines

A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. If the data is not linearly separable, the finite-dimensional space is mapped into a much higher-dimensional space, presumably making the separation easier in that space(kernels)

For SVM, I developed models with both linear and radial kernels.

Following steps were executed on “Small Vocab” for developing a linear SVM model

* Created container of training dataset
* Model was trained on cost varying from 0.1 to 100
* Created container of testing dataset
* Test data set vocab was made same as the train vocab terms.
* Results were created by applying model on the test dataset
* Confusion matrix was generated and evaluation metrics were recorded

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cost:0.1 | Cost:1 | Cost:10 | Cost:100 |
| Accuracy | 0.6871 | **0.7382** | 0.7328 | 0.7195 |
| Precision | 0.6909 | **0.7388** | 0.7356 | 0.7202 |
| Recall | 0.6818 | **0.7298** | 0.7297 | 0.7131 |
| F Score | 0.6863 | **0.7343** | 0.7326 | 0.7166 |

Table 8 : Linear SVM at 0.99 Sparsity

**\*Best Accuracy**

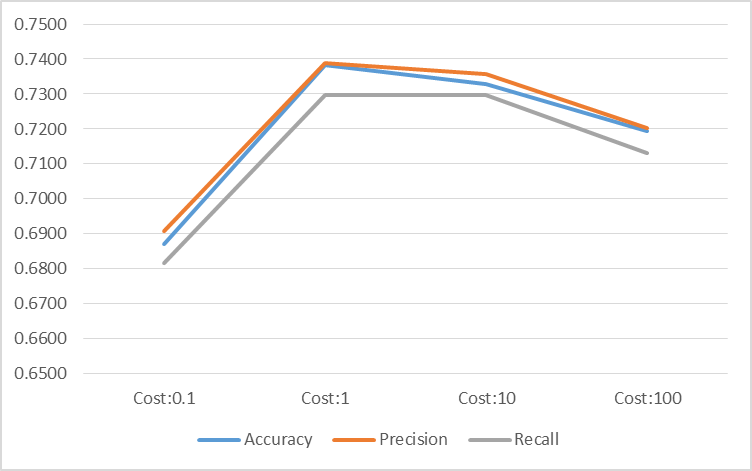


Figure 37 : Accuracy/Precision/Recall for Linear SVM at 0.99 Sparse Matrix

The best accuracy with the linear kernel was 0.738 (cost of 1)

Following steps were executed on “Small Vocab” for developing a radial SVM model

* Created container of training dataset
* Model was trained on cost varying from 0.1 to 100000 and Gamma varying from 10^-5 to 0.2.
* Created container of testing dataset.
* Test data set vocab was made same as the train vocab terms.
* Results were created by applying model on the test dataset
* Confusion matrix was generated and evaluation metrics were recorded

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cost | Gamma | Accuracy | Precision | Recall | F Score |
| 0.1 | Not specified | 0.2408 | 0.2460 | 0.2597 | 0.2527 |
| 1 | Not specified | 0.2172 | 0.2200 | 0.2363 | 0.2279 |
| 1 | 0.2 | 0.1489 | 0.2933 | 0.2841 | 0.2886 |
| 1 | 0.0001 | 0.2228 | 0.2201 | 0.2427 | 0.2308 |
| 10 | Not specified | 0.2448 | 0.2273 | 0.2626 | 0.2437 |
| 100 | Not specified | 0.6895 | 0.7061 | 0.6838 | 0.6948 |
| 100 | 0.001 | 0.6899 | 0.7060 | 0.6842 | 0.6949 |
| 1000 | 0.001 | 0.7398 | 0.7374 | 0.7314 | 0.7344 |
| 1000 | 0.005 | 0.7394 | 0.7369 | 0.7310 | 0.7339 |
| 1000 | 0.02 | 0.7394 | 0.7368 | 0.7310 | 0.7339 |
| 1000 | **0.0001** | **0.7400** | **0.7375** | **0.7316** | **0.7345** |
| 1000 | 0.00001 | 0.7398 | 0.7374 | 0.7315 | 0.7344 |
| 100000 | 0.001 | 0.7203 | 0.7231 | 0.7203 | 0.7217 |

Table 9 : Radial SVM at 0.99 Sparsity

**\*Best Accuracy**

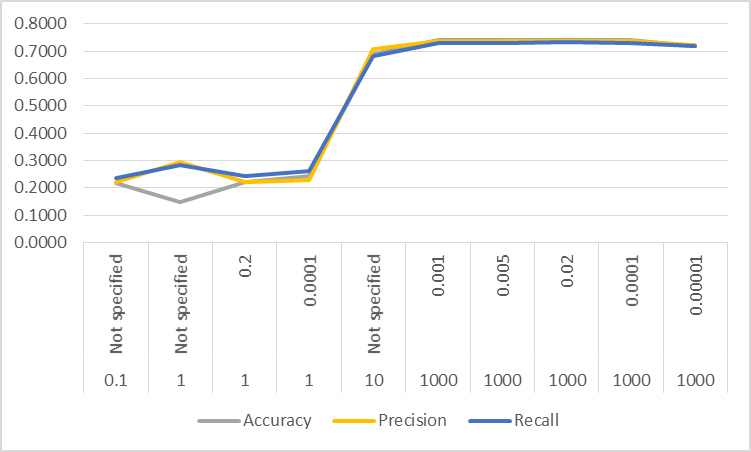


Figure 38 : Accuracy/Precision/Recall for Radial SVM at 0.99 Sparse Matrix

The best accuracy with radial kernel was 74% (cost as 1000, Gamma=.0001).

Next I tried the SVM on “large Vocab” of 8796 words with best accuracy combination of the ‘small vocab’

For linear SVM, taken cost as 1 and accuracy achieved was 81.81%

|  |  |
| --- | --- |
| Accuracy | 0.8181 |
| Precision | 0.7481 |
| Recall | 0.6818 |
| F Score | 0.7134 |

For Radial SVM, taken cost as 1000 and Gamma as 10^-4 and accuracy achieved was 80.05%

|  |  |
| --- | --- |
| Accuracy | 0.8005 |
| Precision | 0.7423 |
| Recall | 0.6818 |
| F Score | 0.7108 |

Attaching below the excel with the confusion matrix for all the above liner and radial variations for both the vocab sizes.



Overall Analysis of SVM:

1) Performance of SVM was quite impressive taking into account the small vocab size I had taken for training. Both the linear and radial models were able to achieve an accuracy of about 81% on vocab size of just 8000 words.

2) Also we can see that the linear model is performing as well as the more complex radial kernel.

3) The time required for training the SVM and Maxent models was the least of all the algorithms I tested.

### Max Entropy Model

Following steps were executed on “Small Vocab” for developing max entropy model

* Created container of training dataset
* Model was trained
* Created container of testing dataset.
* Test data set vocab was made same as the train vocab terms.
* Results were created by applying model on the test dataset
* Confusion matrix was generated and evaluation metrics were recorded

|  |  |  |
| --- | --- | --- |
|  | Vocab:  1805 words | Vocab:  23367 words |
| Accuracy | 0.6942 | 0.8159 |
| Precision | 0.6933 | 0.8151 |
| Recall | 0.6885 | 0.8094 |
| F Score | 0.690891663 | 0.81224 |

Table 10 : Maxent at 0.99 Sparsity

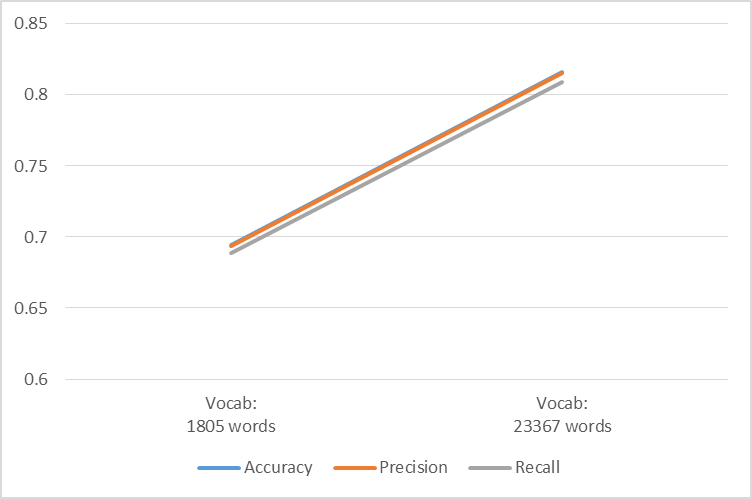


Figure 39 : Accuracy/Precision/Recall for Maxent

Attaching below excel with the confusion matrix for Maxent.



Overall Analysis of Maxent:

1) Performance of SVM was quite impressive taking into account the small vocab size I had taken for training. Both the linear and radial models were able to achieve an accuracy of about 81% on vocab size of just 8000 words.

2) Also we can see that the linear model is performing as well as the more complex radial kernel.

3) The time required for training the SVM and Maxent models was the least of all the algorithms I tested.

### Decision Trees

For Decision tree following steps were executed

* Train dataset was split into training and validation set
* Model was trained for different values of MinSplit, MinBucket & cp parameters on training data set and results were tested on validation set
* Tuning the parameters was also done through the caret package with 3 folds cross validation (cp = .001 was returned as the optimum parameter) which further validated the result achieved by manual tuning.
* Test data set vocab was made same as the train vocab terms.
* Results were generated by applying the model on test data set
* Confusion matrix was generated and evaluation metrics were recorded



Table 11 : Decision Tree at 0.99 Sparsity

**\*Best Accuracy**

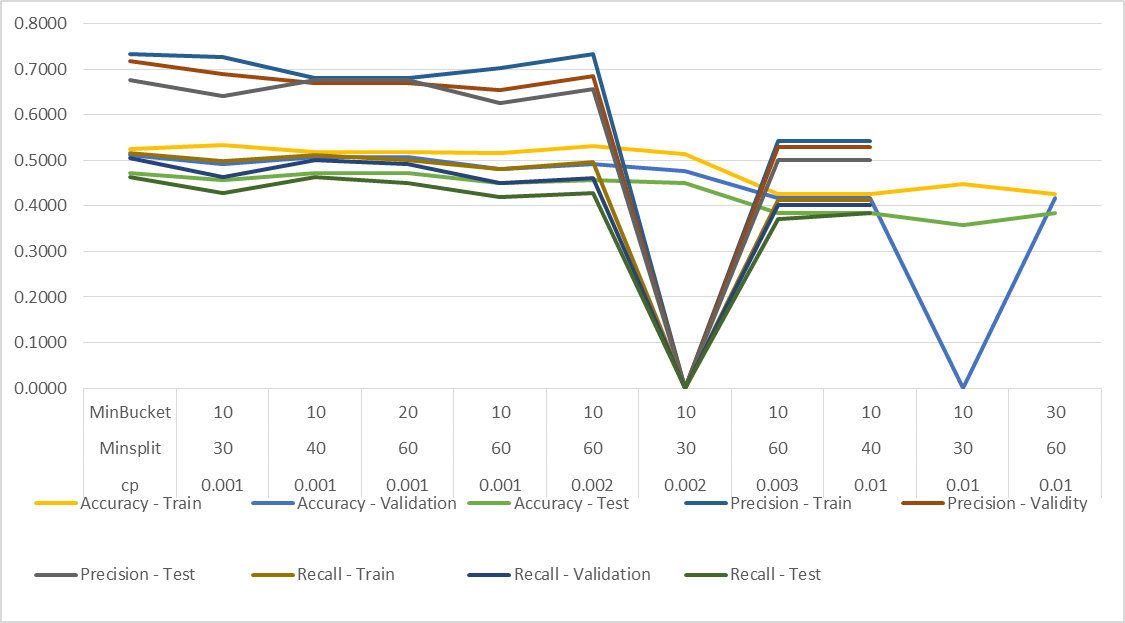


Figure 40 : Accuracy/Precision/Recall for Decision Tree at 0.99 Sparse Matrix

Decision trees did not give a satisfactory result .The highest accuracy I could achieve on the dataset with vocab size of 1805 words was about 0.53 on the train set and 0.47 on the test set. Most of the incorrectly classified documents were labelled in sci.electronics category by the algorithm.

Next I tried decision tree on “large Vocab” of 8796 words with best accuracy combination of the ‘small vocab’

For decision tree, taken cp as 0.001 and accuracy achieved was 47.24%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precison | Recall | F-Score |
| Train | 0.5400 | 0.7487 | 0.5335 | 0.6230 |
| Validation | 0.5489 | 0.7398 | 0.5318 | 0.6188 |
| Test | 0.4724 | 0.6811 | 0.4649 | 0.5526 |

There was no major increase in the accuracy and incorrectly classified documents were again mainly labelled as sci.electronics.

This might be because decision tree optimises purity of node at every step instead of optimising the overall results and needs to be further investigated.

Attaching below excel with the confusion matrix for all the above variations for both the vocab sizes.



Overall Analysis of Decision tree

1. Overall accuracy of the decision tree was not impressive. It showed a bias towards classifying terms in sci.med category for ‘small vocab’ and sci.electronics for ‘large vocab’.
2. There were certain categories like comp.sys.ibm.pc.hardware where it was not able to predict even a single document correctly for ‘small vocab’
3. Training time was higher than SVM and Maxent models.
4. The accuracy did not increase when we increase the vocab size as expected.

Although decision trees are quite simple to understand and interpret as it give us business rules wherein we can classify the test documents fast once the model has been trained, I feel decision trees might not be best algorithm to choose for text classification. Decision-tree learning algorithms are based on heuristics such as the greedy algorithms where locally-optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally-optimal decision tree.

### Random Forest

**Random forests** or random decision forests are an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) Random decision forests correct for decision trees' habit of overfitting to their training set.

For random forest following steps were executed

* Train dataset was split into training and validation set.
* Test data set vocab was made same as the train vocab terms.
* Model was trained for different values of Node size and number of trees on training data set and results were tested on validation set.
* Results were generated by applying the model on test data set
* Confusion matrix was generated and evaluation metrics were recorded.



Table 12 : Random Forest at 0.99 Sparsity

**\*Best Accuracy**

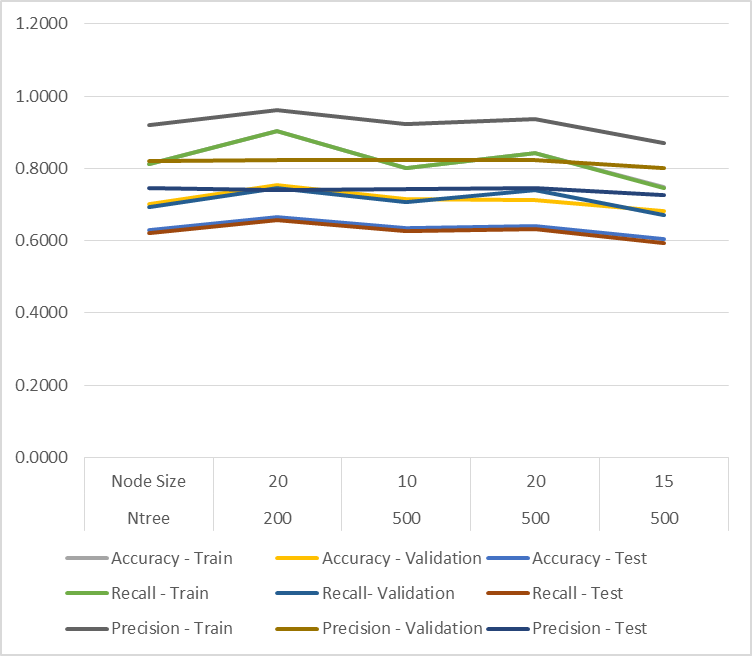


Figure 41 : Accuracy/Precision/Recall for Random Forest at 0.99 Sparse Matrix

Random forest performed better than the decision trees and were able to achieve an accuracy of 0.67.However,training on ‘large vocab’ of 8769 words with highest performing parameters of ‘small vocab (ie node size of 100 and no of trees as 500),could not be done as it was very slow.

Attaching below excel with the confusion matrix for all the above variations.



Overall Analysis of Random Forest Ensemble

1) Random forest performed better than the decision trees but the accuracy can still not be compared with SVM and Maxent models.

2) Training time for these algorithm was very long and very slow as compared to SVM and Maxent.

3) Decision-tree learners can create over-complex trees that do not generalize well from the training data (overfitting).For node size of 10 and no of trees 500, we can see this overfitting. Mechanisms such as pruning of trees can be done to avoid this problem. But in this case pruning was not possible as some categories would have been pruned altogether resulting in low accuracy.

4)We can see high variance in the algorithm for node size of 10 and no of trees 500 which is giving a testing accuracy of 67%.But a node size of 15 and no of trees 500 gives an accuracy of 64% and is more acceptable than the earlier one which is overfitting on the training data.

### KNN Model

I also tried training the KNN model but the processing time was too long and had to stop. The accuracy on was with 1 neighbours on small vocab??

### Neural Network Model

Tried training the neural network but due to time constraint could not explore on parameter tuning and the best accuracy, I could achieve was 0.42 with 1 hidden layer of 3 and 10 units and ‘rang’ of 0.0005. But as I was not able to tune the parameters with caret pkg in R or manually by exploring various combinations, hence it cannot be deduced that the algorithm is not suited for the purpose.

### Results

Testing and training Accuracy of various algorithms (line chart) at diff parameters

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Precision | Recall |
| Naïve Bayes - TF | 0.331 | 0.3012 | 0.328 |
| Naïve Bayes - TF-IDF | 0.385 | 0.3672 | 0.3814 |
| Decision tree | 0.471 | 0.6759 | 0.463 |
| Random Forest | 0.6666 | 0.7398 | 0.6574 |
| Max Entropy Model\* | 0.8159 | 0.8151 | 0.8094 |
| SVM(linear)  SVM(Radial) | 0.8181  0.8005 | 0.7481  0.7423 | 0.6818  0.6818 |

Table 13 : Comparison of Accuracy for all Classification Algorithms

\*Max entropy results were generated on vocab size of 23000 words while others are calculated at 8769 words.

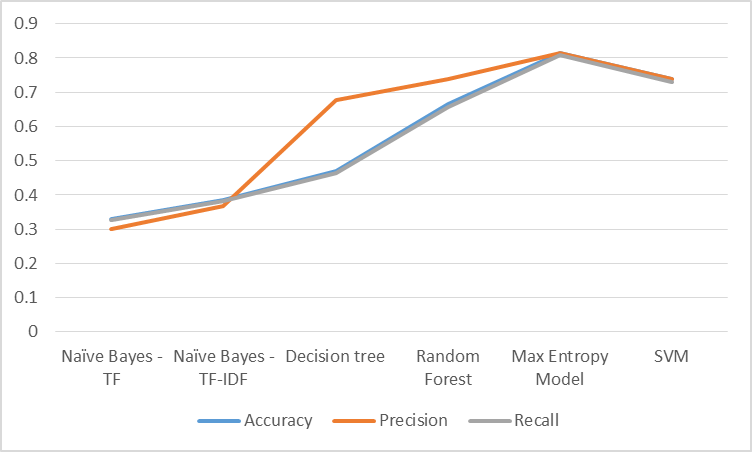


Figure 42 : Comparison of Accuracy/Precision/Recall

**Conclusion**

From the evaluation made on the different classification algorithms in this project, it looks like Maxent and SVM(linear and radial) outperform all the rest algorithms.All the three algorithms took quite less time to train. Linear SVM is preferred over radial for text classification as it is simpler with less parameters and is less prone to overfitting.

All the algorithms tested have their strength and weakness, but Linear Support Vector Machine and Maxent produced the most accurate and highly optimized result followed by Random forest (though it had a very high training time) and lastly the Naïve Bayes.

SVM and Maxent models achieved the highest accuracy and were able to achieve an accuracy of about 73% on a vocab size of 1805 words which is quite good taking into consideration that the articles in the dataset have overlapping categories.

Also the training time for both these algorithms was the least of all the algorithms tested on the dataset.

The accuracy might further increase on increasing the training vocab size as we saw an increase in accuracy from 73% to 81 % on increasing vocab from 1800 to 8769 words.

Due to time constraints and also because I had largely reduced the vocab size due high dimension of the dataset I had to deal with, it is important to note that the current results might not be best obtainable using each of the algorithms highlighted (as far as the evaluation metrics are concerned) but the results will be in line with the highly optimized results.

### Further work

1) Due to limited memory size of my laptop,I could not work high vocab sizes which inturn might increase the classification accuracy of the algorithms.

2) In this project, I stemmed the words to reduce them to their base form. There is a more sophisticated procedure called lemmatisation that takes grammatical context into account when reducing the words to their base form which can be explored to give better results. Also classification accuracy, without stemming or lemmatisation ca be studied.

3) I followed ‘all vs all ‘approach to classify the documents in their respective categories. There is another approach ‘One Vs all Classification’ approach. It treats a multiclass classification as a binary classification. When we want to perform multiclass classification, it select one class (one) and makes the others (rest) zero as we will do in binary classification. This can also be explored to further enhance the classification accuracy.

5) All algorithms, except Naïve Bayes, were trained and tested on document term matrix generated with TFIDF frequency. Effect of DTM generated with TF on classification accuracy can also be studied further.

6) All algorithms, except Naïve Bayes, were trained and tested on document term matrix generated with unigrams. Effect of ngrams on classification accuracy can also be studied further.

7) Feature selection: TFIDF was used in this project. But we can further explore ‘SMART’ method for generating the DTM and using other statistical methods like mutual gain, chi square etc)

7) I had also planned to implement topic modelling on the dataset but due to time constraints, was not able to do the same. This can also be worked on.

8) I did not do stem completion of the words before generating the word cloud due to the memory limitation of my laptop which is evident when we analyze word clouds of some categories.

9) Neural networks can also be explored further.

## Clustering

Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. Examples of document clustering include web document clustering for search users.

In general, there are two common algorithms. The first one is the hierarchical based algorithm, which includes single link, complete linkage, group average and Ward's method. By aggregating or dividing, documents can be clustered into hierarchical structure, which is suitable for browsing. However, such an algorithm usually suffers from efficiency problems. The other algorithm is developed using the K means and its variants. Generally hierarchical algorithms produce more in-depth information for detailed analyses, while algorithms based around variants of the K means algorithm are more efficient and provide sufficient information for most purposes.

**Clustering Vs Classification**

Clustering algorithms in computational text analysis groups documents into what are called subsets or *clusters* where the algorithm's goal is to create internally coherent clusters that are distinct from one another.Classification on the other hand, is a form of supervised learning where the features of the documents are used to predict the "type" of documents.

### Evaluation and assessment

Evaluation of clustering results sometimes is referred to as cluster validation.

#### Internal evaluation

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example, k-Means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.

* Silhouette coefficient: The silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering, and is also used to determine the optimal number of clusters.

#### External evaluation

In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by (expert) humans. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes.

* The F-measure can be used to balance the contribution of false negatives by weighting.β ≥ 0 {\displaystyle \beta \geq 0} F β = ( β 2 + 1 ) ⋅ P ⋅ R β 2 ⋅ P + R {\displaystyle F\_{\beta }={\frac {(\beta ^{2}+1)\cdot P\cdot R}{\beta ^{2}\cdot P+R}}}
* The mutual information is a measure of how much information is shared between a clustering and a ground-truth classification that can detect a non-linear similarity between two clusterings. Adjusted mutual information is the corrected-for-chance variant of this that has a reduced bias for varying cluster numbers.

### Hierarchical Clustering

Following steps were executed on vocab size of 253 terms to cluster the training documents.

* Created a dtm with 0.94 sparsity (253 terms)
* Computed Euclidean distance between the TFIDF of Terms for all documents
* Used ward.d method.
* Plotted the dendogram to determine optimum number of clusters.
* 6,9,10 clusters look feasible. Clustering was performed for 6, 910 and 20 clusters. 20 clusters were chosen as we know that there are 20 categories for this dataset.

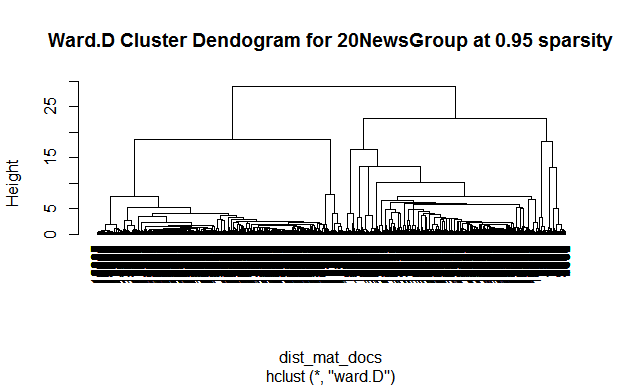


Figure 43 : Dendogram for Hierarchical Clustering

On analysis, none of the clustering generated were good. Two of the clusters had most of the documents clustered into them and they cannot be identified into well separated categories.



Table 14 : Hierarchical Clustering - 6 Clusters



Table 15 : Hierarchical Clustering - 9 Clusters



Table 16 : Hierarchical Clustering - 10 Clusters



Table 17 : Hierarchical Clustering - 20 Clusters

Here again , the categories are not well defined between clusters.

Cluster 1 and Cluster 4: Religion

Cluster 9/18: Sports

Cluster 10:autos/motorcycles

Cluster 11:sci crypt

Cluster 14/15/16/19:computers

Cluster 17:sci space

Cluster 3 & 5 are majorly cluttered with documents of different categories and also have 50% of the documents clustered in them.

So, the clustering cannot be considered.Average silhoute width of 20 clusers, is also very poor

### K means Clustering

Following steps were executed on vocab size of 253 terms to cluster the training docments.

* Created a dtm with 0.94 sparsity (253 terms)
* Calculated within cluster sum of squares (tot $ withins) clusters to determine optimum number of clusters.
* 2, 8, 10/11, 17 clusters look feasible. Clustering was performed for 10 and 20 clusters.
* Results were not quite good. Clus plot and silhouette values were very poor.

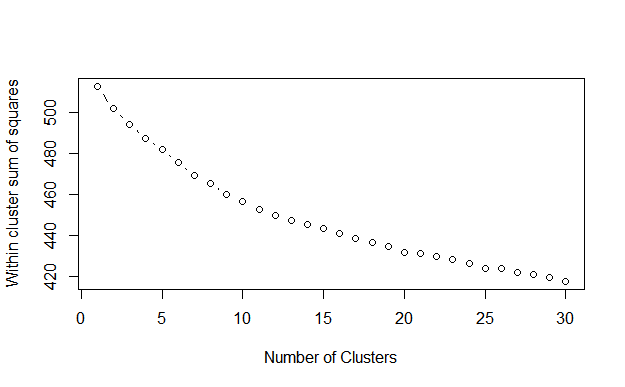


Figure 44 : detrmining no of clusters in K Means Clustering

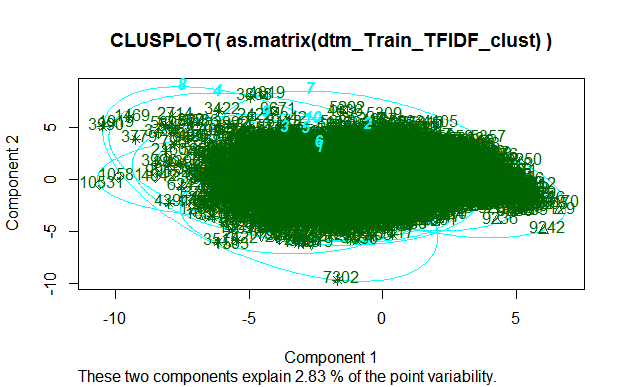


Figure 45 : K Means Clusplot for 10 cluster kmeans

Cluster membership for 10 clusters.

1 2 3 4 5 6 7 8 9 10

213 219 526 7534 219 278 1451 340 153 381



Table 18 : K Means - 10 Cluster

Cluster 7 and 8 have 8000 of the 11314 documents and they do not have well defined categories in them



Table 19: K Means - 20 Cluster

### SK means

I also clustered the documents with SK means method.



Table 20 : SK Means - 10 cluster

Results of 10 cluster of skmeans is the best amongst all the clustering done.

Clusters are better defined than other methods and none of the clusters are cluttered with majority of the documents of different categories.

Cluster 1:Religion

Cluster 2/6:Computers

Cluster 3: Sports

Cluster 4:science/computers

Cluster 5: computers/sci.electronics

Cluster 7:Hardware(computers)/Autos/Motorcycles

Cluster 8:Politics and sci.crypt documents

Cluster 9:Not well defined

Cluster 10:sci space and computers/science



Table 21 : SK Means PV 10 Cluster

Clusters are better defined than other methods and none of the clusters are cluttered with majority of the documents of different categories.

Cluster 1:Religion

Cluster 2:Computers

Cluster 3:majorly science and computers but a bit cluttered with different categories

Cluster 4:sports

Cluster 5: computers/sci.electronics

Cluster 7:sci.space/computers

Cluster 6/8:Science/Computers

Cluster 9:Autos/motorcycles

Cluster 10: computers

Silhouette of 11314 units in 10 clusters

Cluster sizes and average silhouette widths:

767 1153 451 628 3880 1532 1212 636

0.15 0.073 0.10 0.16 0.007 -0.01 0.022 0.07

515 540

0.18 0.09

Individual silhouette widths:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.109200 0.004673 0.024890 0.051500 0.080860 0.410200



Table 22: SK Means PV 20 Cluster 1

Silhouette of 11314 units in 20 clusters

Cluster sizes and average silhouette widths:

538 440 342 375 354 413 271 337

0.17 0.07 0.11 0.20 0.10 0.19 0.07 0.083

1016 741 303 465 679 375 571 322

-0.007 0.15 0.10 0.17 0.05 0.13 0.04 0.23

367 485 353 2567

0.14 0.042 0.09 -0.0084

Individual silhouette widths:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.150500 0.004233 0.047350 0.075970 0.125700 0.484300

### Conclusion

As is very evident, from the above clustering, none of the algorithms were able to cluster documents in good homogeneous groups. SK means was the best but still it did not cluster the documents in a well defined manner.

Perhaps, better feature selection methods for generating key words should be used for applying clustering to text documents. Key words generated at present might not be representative of the respective categories. Another approach that can be used when generating DTM for clustering is Synonym expansion(where similar words are replaced with synonyms to better identify similarities between documents) and lemmatisation which can help in generating better key words Also the categories are overlapping.This might also be the reason for the documents not been clustered in a well defined manner.