2014 Fourth International Conference on Communication Systems and Network Technologies

Pattern and Cluster Mining on Text Data

Deepak Agnihotri
Department of Computer Applications, NIT Raipur
Raipur, India
agnihotrideepak@hotmail.com

Kesari Verma
Department of Computer Applications, NIT Raipur
Raipur, India
kverma.mca@nitrr.ac.in

Priyanka Tripathi
Department of Computer Engineering and Applications, NITTTR Bhopal,
Bhopal, India
ptripathi@nitttrbpl.ac.in

Abstract— **Due to heavy use of electronics devices nowadays most of the information is available in electronic format and a substantial portion of information is stored as text such as in news articles, technical papers, books, digital libraries, email messages, blogs, and web pages. Mining the knowledge like pattern finding or clustering of similar kind of words is one of the important issues nowadays. This paper focuses on mining the important information from the text data. This paper uses the stories data set from project Guttenberg's William Shakespeare stories dataset for experimental study. R is used as Text Mining and statistical analysis tool in Ubuntu 12.04 LTS Linux Operating System. Frequent pattern mining is used to find the frequent terms, appeared in the documents and word Association among two or more words is measured at a given threshold value. Our algorithm uses cosine similarity in order to measure the distance between the words before clustering. The algorithm may be used to find the similarity between stories, news, emails. In this paper k-means and hierarchical agglomerative clustering algorithm is used to form the cluster.**

Keywords- text mining, stop words, stemming, TF – IDF, Clustering, Word Association

I. INTRODUCTION

Approximately 2% words of Corpus is used for text analysis other words (e.g. Stop words, White Spaces, Header, and footer etc.) are not required for frequent pattern analysis, and clustering of the documents. Therefore lot of preprocessing task is required before text analysis and extracting knowledge from these texts. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the structured data, and finally evaluation and interpretation of the output. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities) [1, 4]. Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The overarching goal is, essentially, to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods [1, 2, 3, 4, 5].

This paper is organized as follows- Second section describes the related works in this area. Third Section describe the preprocessing of unstructured text contents, Fourth Section gives brief introduction of text data mining task, Fifth Section explains experimental works and discussions. The paper is concluded in sixth section. The figures used in this paper are shown at the end of paper.

II. RELATED WORKS

There are various works in the field of Text mining carried out by the researchers. The Delany et. al. [6] studied the text mining technique on spam data set of mobile messages. The R statistical analysis tool has been used to form the cluster of spam messages and for finding associations among words of messages. Andrew et. al. [11] have applied text mining on "Complete Works of William Shakespeare" stories data set downloaded from [14]. This work shows text mining like clustering, word association, TF - IDF calculations, and plotting of results in figures etc. The web page [13] shows an example on text mining of Twitter data which shows clustering, word association, *word cloud formation etc*. There are other works [7,8,9,10,12,13,15,16,17] shows examples on text mining which provides function for text mining like, stop words removal, stemming, whitespace removal, TF - IDF calculations, clustering, word association, frequent terms, word cloud and plotting of results in figures etc.

III. PREPROCESSING OF TEXT DATA SET

Text Data Mining deals with unstructured data, a gigantic amount of data is stored as unstructured or semi-structured text like in books, newspaper articles, research papers, e-mail messages, Web pages, and XML documents. Retrieval of patterns from such sources is the task of text-mining which requires text filtering as stop-words removal, stemming, and white space removal as a pre-processing task. Often there is some structure implicit in the documents in that they have titles, sections, paragraphs, etc. For many pattern discovery tasks, each document is treated as a bag of words-that is the set of all words with the frequency of the words appearing in that

978-1-4799-3070-8/14 $31.00 © 2014 IEEE
DOI 10.1109/CSNT.2014.92

document .In Vector-Space model, the *dimensions* of vector-space consist of all the documents in the collection. Each document is then represented as a vector in this space containing the frequency of all the words in that document. For example if the dimension are ('and', 'the', 'machine', 'learning'), then a document may be represented as (2, 3, 2, 0) the number indicates the frequency of the corresponding words in that document [3].

In computing, stop words such as a, an, *the*, *is*, *at*, *which*, and *on etc.* are words which are filtered out prior to, or after, processing of natural language data (text). These words occur most frequently in the text document but they can't make sense in the topic of documents [1, 2, 3, 5]. In order to improve performance algorithm for removal of stop words from phrase.

Input: D Database of phrase,
CD clean database = {},
S = {Dictionary of stop word}
1. S = {Dictionary of stop word e.g. that, this is, a, an, the, etc.}
2. ***Foreach*** *word € database D delimited by space till EOF*
3. If w_length (word) < 2 remove the word from D
4. ***Else If*** *word € {S} continue*
5. ***Else*** *CD ⇓ word*
6. End

A. TF - IDF Calculation

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take. One such scheme is TF - IDF. They are useful in the field of natural language processing [1]. In Mathematical Terms, Corpus Term Frequency=The fraction of all words in the corpus that are term t. Sentence Term Frequency=The fraction of all the sentences in the corpus containing term t, Document Term Frequency=The fraction of all the documents in the corpus containing term t, Given a Corpus of Documents =C, Let the number of documents in C=Nd, number of Sentences in C=Ns, number of distinct terms in C= Nt. For, a set of terms T= {t1, t2, t3, …, tk}, where $k \geq 1$. nd (t1, t2, …, tk)=number of documents in C, that contain all the terms in T. ns(t1, t2, …, tk) = number of sentences in C that contain all the terms in T. Document Frequency of T = df(t1, t2, …, tk) = nd (t1, t2, …, tk) / Nd. Sentence Frequency of T = sf(t1, t2, …, tk) = ns(t1, t2, …, tk) / Ns. If k = 2, then dpf=Document Pair Frequency and spf=Sentence Pair Frequency. For a Single Term t, we define the inverse document frequency of t as, IDF(t)=log(Nd / nd (t)). Inverse Sentence Frequency as, ISF(t)= log(Ns / ns (t)). TF - IDF weight=TF(t) * IDF(t). For terms t1, t2, …, tk the measure for the association{t1, t2, …, tk} is mk(t1, t2, …, tk) = SF(t1, t2, …, tk) X $\prod$ idf (tj). Where j=1, 2, …, k. If k=2 then, m(t1, t2)= SF(t1, t2) * IDF(t1) * IDF(t2). These tf- idf values are computed for every term. Only terms that have a tf- idf value more than some threshold are considered worth of inclusion into the vector space model. Other terms are neglected [18, 19]. This greatly reduces the number of dimensions that one must deal with [3]. TF– IDF is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF - IDF value 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, which helps to control for the fact that some words are generally more common than others [1, 2, 3]. Suppose we have a set of English text documents and wish to determine which document is most relevant to the query "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document and sum them all together; the number of times a term occurs in a document is called its term frequency [1]. However, because the term "the" is so common, this will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less common words "brown" and "cow". Hence an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely [1].

IV. TEXT DATA MINING

Text Data Mining defined as - "The nontrivial extraction of implicit, previously unknown, and potentially useful information from textual data" (Strict def.), or as "The science of extracting useful information from large text data sets" (Loose def.)[20].Text Data Mining is related with document clustering, finding of frequent terms in document's, association among words etc. Broadly it can be categorized as Text Clustering and Text words Association.

A. Document Clustering

Documents can be clustered into hierarchical structure, which is suitable for browsing [19]. However, such an algorithm usually suffers from efficiency problems. The other algorithm is developed using the K-means algorithm and its variants. Usually, it is of greater efficiency, but less accurate than the hierarchical algorithm [1]. K-means clustering is one of many techniques within unsupervised learning that can be used for text analysis. In order to use k-means clustering with text data, we need to do some text-to-numeric transformation of our text data. K-Means algorithm for text document clustering [18].

K-Means Method:

Input: D= {d1,d2, …,dn},Corpus of documents
K=number of clusters
Steps:

1. *Select K document vectors as the initial centroids of K clusters.*
2. *Repeat*
   For i=1,2,…, n
3. *Compute similarities between di and K centroids.*
4. *Put di in the closest cluster*
5. *END For*
6. *Recompute the centroid of the cluster Until the centroid don't change.*

Output: K clusters of Text Documents

Next we may use Hierarchical Agglomerative Clustering Algorithm for text document clustering [18].
Hierarchical Agglomerative Clustering Method:
Input: D= {d1,d2,…, dn},Corpus of documents
d1= {t11, t12,…, t1n}, d2={t21, t22,…, t2n}…documents containing terms.
Steps:
1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose ijth entry gives the similarity between the ith and jth clusters.

$Cos(d1,d2)=d1.d2/|d1|X|d2|$
$d1.d2=(t11*t21+t12*t22+…+t1n*t2n)$
$|d1|=\sqrt{[(t11)^2+(t12)^2+…+(t1n)^2]}$
$|d2|=\sqrt{[(t21)^2+(t22)^2+…+(t2n)^2]}$

2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pair wise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.
Output: Dendogram of Documents clusters

B. Words Association

If two terms co-occur within the same paragraph, they constitute an association e.g. <term1, term2, associative frequency>. For terms t1, t2, … , tk the measure for the association{t1, t2, …, tk} is as

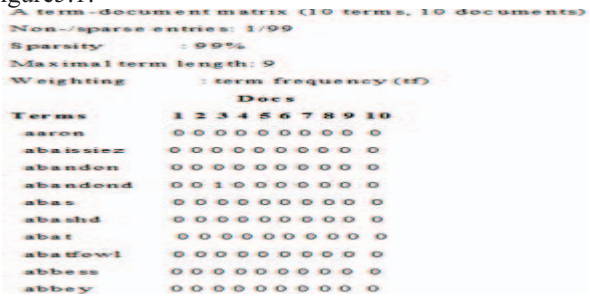
$$mk(t1,t2,\ldots,tk) = Sentence\ Frequency(t1,t2,\ldots,tk) \times \prod idf(tj)$$

$$Where\ j = 1,2,\ldots k.$$

The most frequently occurring terms in different-different stories or documents, these words are associated with other words at specified threshold value .e.g. word "love" is associated with two words "beauti" and "eye" at 0.8 threshold value.

V. EXPERIMENTAL WORKS AND DISCUSSIONS

R statistical analysis tool is used here for text mining. First of all we have to remove stop words and whitespaces from text document, then stemming of words. Now TF - IDF weight is calculated, and then based on these weights we will find cosine similarity among documents. Then we may find association among words in a document, form clusters by using K-Means and Hierarchical Agglomerative Clustering Algorithms. We may plot figures for better analysis of our result on text documents. A term-document matrix is shown in figure5.1:



```
A term-document matrix (10 terms, 10 documents)
Non-/sparse entries: 1/99
Sparsity           : 99%
Maximal term length: 9
Weighting          : term frequency (tf)
                 Docs
Terms             1 2 3 4 5 6 7 8 9 10
  aaron           0 0 0 0 0 0 0 0 0  0
  abaissiez       0 0 0 0 0 0 0 0 0  0
  abandon         0 0 0 0 0 0 0 0 0  0
  abandond        0 0 1 0 0 0 0 0 0  0
  abas            0 0 0 0 0 0 0 0 0  0
  abashd          0 0 0 0 0 0 0 0 0  0
  abat            0 0 0 0 0 0 0 0 0  0
  abatfowl        0 0 0 0 0 0 0 0 0  0
  abbess          0 0 0 0 0 0 0 0 0  0
  abbey           0 0 0 0 0 0 0 0 0  0
```

Table 5.1: illustrates TDM for TF- IDF measure

The table 5.1 shows, that the word "abandond" occurred once in document number 2 but was not present in any of the other first ten documents. We could have generated the transpose of the DTM as well. After stemming and white space removal, now we can find the most frequently occurring terms.

```
[1] "can"   "come"  "duke"  "enter"  "exeunt" "eye"   "father" "first"
 [9] "give"  "god"   "good"  "hand"   "hath"   "heart"  "ill"   "king"
[17] "know"  "ladi"  "let"   "like"   "look"   "lord"   "love"  "make"
[25] "man"   "may"   "mine"  "much"   "must"   "never"  "now"   "one"
[33] "say"   "see"   "shall" "sir"    "speak"  "take"   "tell"  "thee"
[41] "thi"   "think" "thou"  "time"   "tis"    "upon"   "well"  "will"
[49] "yet"
```

Table 5.2: illustrates terms which occurred 1000 times in documents

friend	give	may	shall	let	one	leav	make
0.62	0.59	0.59	0.59	0.58	0.57	0.55	0.55
part	take	yet	hear	must	show	upon	well
0.55	0.55	0.55	0.54	0.54	0.54	0.54	0.54
can	much	prayer	find	half	learn	littl	made
0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52
therefor	true	will	everi	flesh	good	heart	howsom
0.52	0.52	0.52	0.51	0.51	0.51	0.51	0.51
law	live	none	now	pray	prepar	answer	best
0.51	0.51	0.51	0.51	0.51	0.51	0.50	0.50
ever	mind	monarch	thus	tis	word	world	
0.50	0.50	0.50	0.50	0.50	0.50	0.50	

Table 5.3: illustrates word "wish" is associated with other words at 0.5 thresh hold value.

```
Within cluster sum of squares by cluster:
 [1]  291.83594   58.63775  145.75827   88.43234
674.78924   87.93055
 [7]  100.57605 1979.84960  396.64951 13675.42813
 (between_SS / total_SS =  4.6 %)
Available components:
[1] "cluster"     "centers"     "totss"       "withinss"
"tot.withinss"
[6] "betweenss"   "size"
There are 10 clusters clusters.
   1    2       3    4      5    6      7     8      9     10
 371  183  212  135  778  182  211 2357  463 13847
```

Table5.4: illustrates output of k-means clustering using cosine similarity as distance measure technique among words.

Let's take a small example to explain text mining and analysis:

```
e1<-"i like apple and bannanas"
e2<-"the cost of apple and bannanas are very high in these days"
e3<-"the cost of grapes and oranges is also very high"
e4<-"apple,bannanas,grapes,and oranges are fruits"
e5<-"the cost of vegitables are also very high during these days"
```

Terms

Docs	also	apple	bannanas	cost	days	fruits	grapes	high	like	oranges	vegitables
1	0	1	1	0	0	0	0	0	1	0	0
2	0	1	1	1	1	0	0	1	0	0	0
3	1	0	0	1	0	0	1	1	0	1	0
4	0	0	0	0	0	1	0	0	0	1	0
5	1	0	0	1	1	0	0	1	0	0	1

Table 5.5: illustrates small text data mining

Figure 5.1 illustrates hierarchical agglomerative clustering using average linkage; there are other ways like single linkage, complete linkage, and wards method for hierarchical agglomerative clustering.

VI. CONCLUSIONS

Text-mining is an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics. Typical text-mining tasks include text categorization, text clustering, word association etc. An important goal is to derive high-quality information from text. Text-mining usually requires-First, structuring the unstructured input text by parsing which requires deep knowledge of regular expressions. Second, the addition of some derived linguistic features, and Third removal of other like white space, new line, tab characters etc and subsequent insertion into the database. R as statistical analysis tool might be helpful for all these types of work associated with text-mining. A substantial portion of information is stored as text such as in news articles, technical papers, books, digital libraries, email messages, blogs, and web pages. Hence, research in text mining has been very active. Further, I am trying to implement all text mining tasks for opinion data and review data. I am also trying to implement the various probabilistic and statistical models for selection of feature vectors from text corpus before applying text mining.

REFERENCES

[1] http://en.wikipedia.org/wiki/Text_mining

[2] G.K.Gupta, Introduction to Data Mining with Case Studies, PHI 2006,

[3] Vikram Paudi, P. Radha Krishna,Data Mining,Oxford University Press, First Edition, 2009

[4] *Jiawei Han, Micheline Kamber, Jian Pei, DATA MINING Concepts and Techniques, Elsevier, Third Edition, 2012*

[5] *Arun K Pujari, DATA MINING TECHNIQUES, Universities Press, Second Edition, 2009*

[6] *S J. Delany, M. Buckley & D. Greene (2012) "SMS spam filtering: methods and data" Expert Systems With Applications 39, p 9899-9908, http://www.elsevier.com/*

[7] *http://www.dit.ie/computing/research/resources/smsdata/*

[8] http://michael.hahsler.net/SMU/7337/install/tm.R

[9] http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

[10] http://faculty.washington.edu/jwilker/tft/Stewart.LabHandout.pdf

[11] Andrew, Text Mining the Complete Works of William Shakespeare, R-blogs, Sep 5 2013, http://www.r-bloggers.com/text-mining-the-complete-works-of-william-shakespeare/

[12] Yanchang Zho, R and Data Mining: Examples and Case Studies, Elsevier, December 2012, http://www.rdatamining.com/

[13] RdataMining.com: R and Data Mining, http://www.rdatamining.com/examples/text-mining

[14] Project Guttenberg, http://www.gutenberg.org/

[15] http://www.r-bloggers.com/clustering-search-keywords-using-k-means-clustering/

[16] Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Data Set, Cambridge, 2011, DOI: http://dx.doi.org/10.1017/CBO9781139058452.002

[17] *nptel.iitm.ac.in/courses/106104021/pdf_lecture/lecture30.pdf*

[18] C. A. Murthy, Text Document Clustering, ACM *Text Mining Workshop TMW-2014 at ISI Kolkata,*

[19] http://www.isical.ac.in/~acmsc/TMW2014/TMW2014.html

[20] Mandar Mitra, An Introduction to Text Mining, http://www.isical.ac.in/~scc/DInK%2710/ studymaterial/textmining.pdf

[21] www.comp.lancs.ac.uk

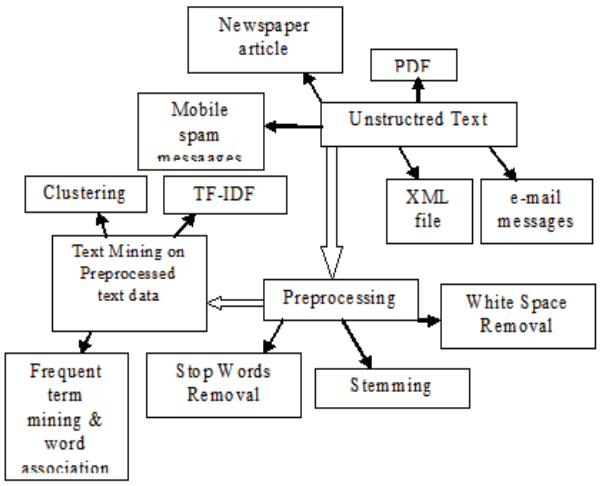


Figure 1. Block Diagram of Text Mining Process

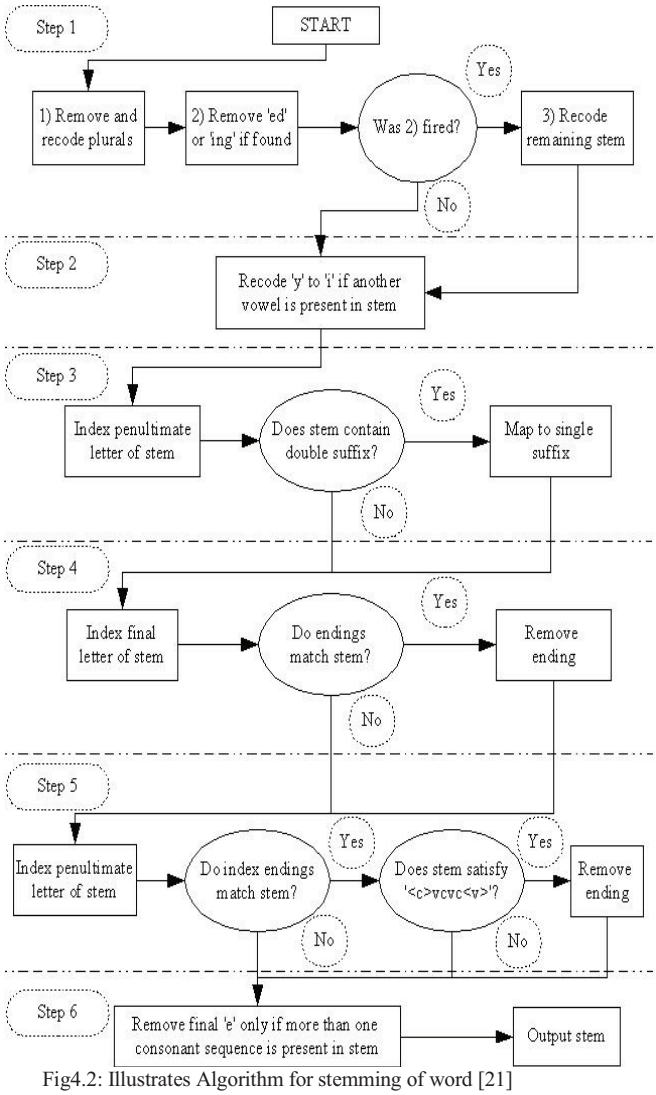


Fig4.2: Illustrates Algorithm for stemming of word [21]

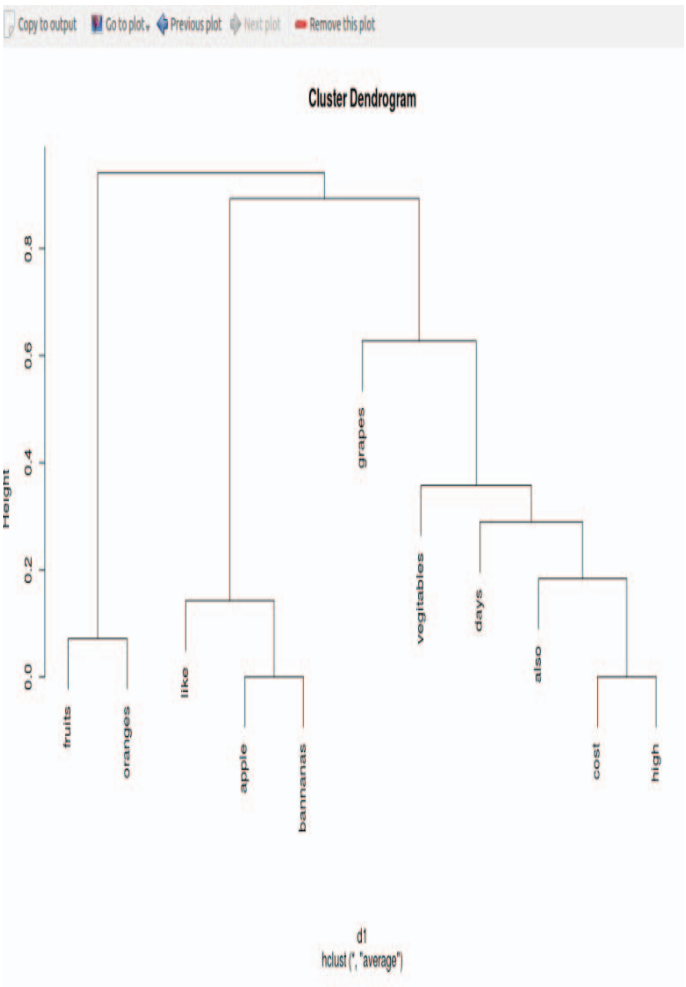


Figure5.1: illustrates hierarchical agglomerative clustering using average linkage

****