20 News groups

I Data

For 20 News groups, I use the email data cleanedup version provided in PIAZZA(20news18828clean.zip). There are 20 different news topics and approximately 1000 documents per topic which is shown in Table1.1. This dataset is perfect for experiments like data preprocessing and clustering in text applications of Machine Learning techniques due to its term-document format. Each newsgroup is stored in a subdirectory, with each article stored as a separate file. And this version doesn't contain cross-posts and includes only the "From" and "Subject" headers. We could select some of them from different categories to analyze the similarities or same category to explore the dissimilarities. The dataset is balanced in the number of documents per topic, while most of the topics have similar number of unique words too.

Number	News Topics	Documents	Unique Words
1	alt.atheism	1000	12152
2	comp.graphics	1000	15513
3	comp.os.ms-windows.misc	1000	31673
4	comp.sys.ibm.pc.hardware	1000	11858
5	comp.sys.mac.hardware	1000	11085
6	comp.windows.x	1000	19693
7	misc.forsale	1000	13216
8	rec.autos	1000	11982
9	rec.motorcycles	1000	11336
10	rec.sport.baseball	1000	10981
11	rec.sport.hockey	1000	12763
12	sci.crypt	1000	13766
13	sci.electronics	1000	12090
14	sci.med	1000	16703
15	sci.space	1000	15027
16	soc.religion.christian	997	12043
17	talk.politics.guns	1000	14128
18	talk.politics.mideast	1000	16427
19	talk.politics.misc	1000	15413
20	talk.religion.misc	1000	13940

Table 1.1

II Experiments

II.A Data Preprocessing

I try to process the full 20 NewsGroups data but is too slow due to my computer configuration(OS :Mac Processor: 2.3GHz Intel Core i5). So I select 6 topics shown as Table 2.1. The reason I select them is they have balanced documents number and almost come different subjects. As for topics comp.os.ms-windows.misc, I regard it as a different topic due to its unique words is much more than comp.graphic. So I expeact them should be different to each other in content features to general 6 good clusters, which could help me to analyze the dissimilarities from different subjects. If it's 5 clusters, that also prove the feature of unique words is not that important as subject.

News Topics	Documents	Unique Words
alt.atheism	1000	12152
comp.graphics	1000	15513
comp.os.ms-windows.misc	1000	31673
misc.forsale	1000	13216
rec.sport.baseball	1000	10981
talk.politics.guns	1000	14128

Table 2.1

I don't think clustering and LSA, LDA will perform well on your data just by analyzing the documents that I picked. Because all we know so far is just basic information, we still

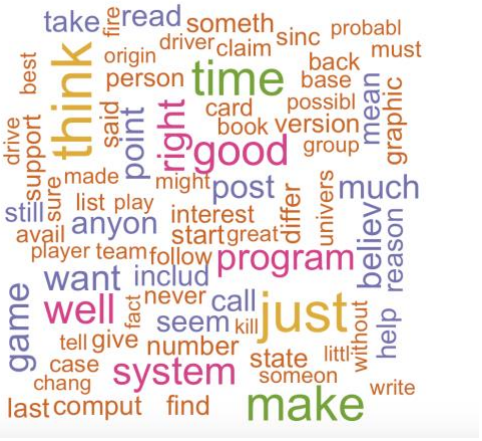
need the document-term matrices and more cleaning data to get a good result, which will be the next preprocessing section:

- Stopwords, whitespace, numbers and punctuations are removed using the tm_map() which cleans the raw data and prepare data for further analysis.
- Document-term matrix is created using the inbuilt DocumentTermMatrix(). This matrix describes the frequence of terms which occurs in a document.
- Stemming is performed on the cleaned data because the classifier doesn't understand verbs (i.e. organized & organizing) and treats them as different words.
- Pruning words by frequency: words that occurs in very few documents were removed. The vocabulary size after pruning decreased from 77622 to 33288, which is good because the entire focus is on the words which have high frequency. Therefore during LSA/LDA/Clustering, important data is analyzed. Both LSA and LDA are topic models.
- Tf-Idf weights are used on Document Term Matrix which is one of the popular term-weighting schemes. It is a statistical measure which evaluates the importance of word in the document.

```
> tfidf
<<DocumentTermMatrix (documents: 5548, terms: 203)>>
Non-/sparse entries: 102488/1023756
Sparsity           : 91%
Maximal term length: 10
Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

After preprocessing, Table 2.2 shows top ten frequent words in 6 newsgroups I select and Fig 2.1 shows the importance of each word via its frequency.

No	Word	Frequence
1	like	2815
2	file	2585
3	just	2360
4	think	2354
5	peopl	2331
6	time	2028
7	make	1993
8	window	1982
9	also	1765
10	good	1727



II.B Clustering Experiments.

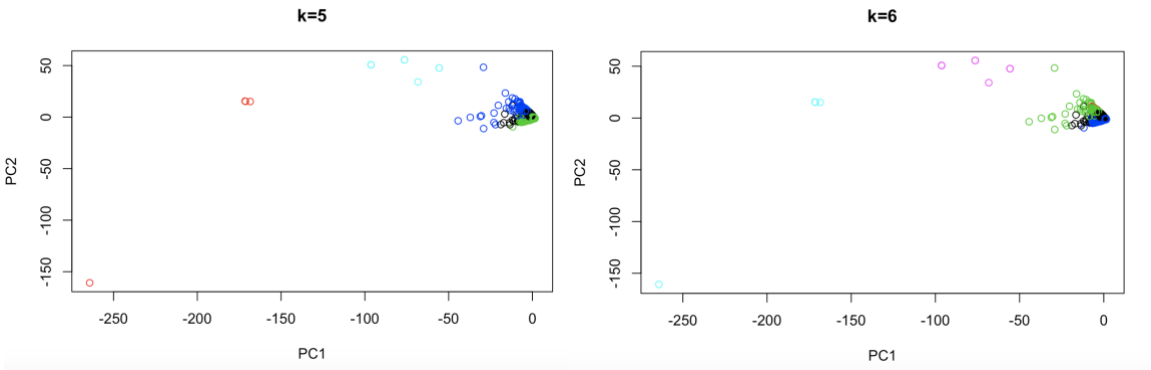
1. Cluster the document vectors from the tf-idf document-term matrix with kMeans to have a baseline for comparisons.
For the subjects I choose, it seem that K should be 5 or 6. I did both to compare the result shown as below:

```
> table(cl_5$cluster)

   1    2    3    4    5
 837    4 4664   99    7
```

```
> table(cl_6$cluster)

   1    2    3    4    5    6
 728  170   88 4614    4    7
```



Which are no too much difference.
So I try to use NbClust to determine the best number of clusters.

```
#nbclust
nbc<-NbClust(as.matrix(dtms),distance='euclidean',min.nc = 5,max.nc = 8,method = 'kmeans',index
```

The result shows that the best clusters number is 5.

Name	Type	Value
nbc	list [3]	List of length 3
All.index	double [4]	0.583 0.343 0.326 0.277
5	double [1]	0.583
6	double [1]	0.3429
7	double [1]	0.3262
8	double [1]	0.2767
Best.nc	double [2]	5.000 0.583
Number_clusters	double [1]	5
Value_Index	double [1]	0.583
Best.partition	integer [5548]	4 3 3 4 3 3 ...

2.LSA

• Compute the SVD of the document-term matrix
Dimensions :
SVD(Document-Term Matrix) = U*D* VT ; this is nothing but an interpretation of SVD for Document-Term matrix.
Here, U relates terms to topics, V relates documents to topics & D gives importance of topics. Thus, D*V T provides k dimensional LSA document vectors whereas D*U provides k dimensional LSA word vectors.
The result is :

svd_dtm	Large matrix (1139033 elements, 8.7 Mb)
um [1:5611, 1:203]	0.9033 0.2443 0.0328 1.255 0.0983 ...

R commands I use to create LSA document vectors that I use as input to clustering is shown as below:

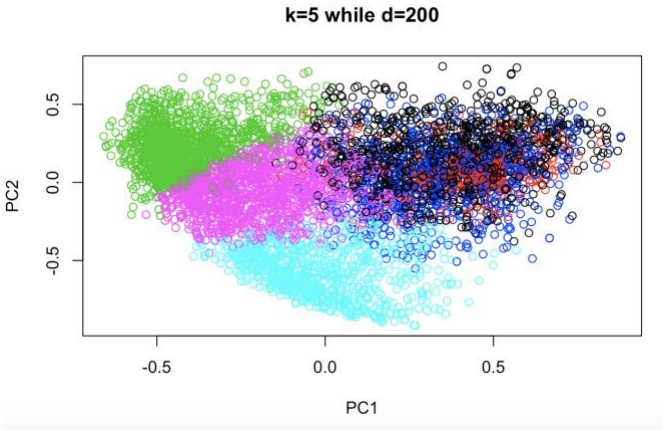
```
lsacl <- kmeans(norm_eucl(svd_dtm), 5)
lsaConfm <- table(category, lsa_cl$cluster)
```

• Compute the d=50, 100, 200 dimensional representation for the term-document matrix

Algorithm is as follows -

- S <- svd(dtm)
- Dd <- diag(S$d)[1:d , 1:d] gives the importance of topics where d is 50,100 or 200.
- Mat <- S%v[,1:d] %*% Dd %*% t(S%u[,1:d])

• Cluster the d-dimensional documents and the words with kMeans, using the same **K** as for the tf-idf vectors to be able to compare the clustering results.



So I think it do perform better.

• For each of the top 5 concepts report the most representative words.

```
[[1]]
 [1] "file"    "program" "version" "also"    "avail"   "softwar" "system"  "free"    "like"
[10] "graphic"

[[2]]
 [1] "peopl"   "also"    "system"  "like"    "think"   "time"    "graphic" "just"    "make"
[10] "program"

[[3]]
 [1] "graphic" "program" "system"  "avail"   "softwar" "send"    "mail"    "includ"  "also"
[10] "list"

[[4]]
 [1] "version" "program" "window"  "free"    "softwar" "better"  "current" "note"    "origin"
[10] "much"

[[5]]
 [1] "peopl"  "state"  "system" "right"  "govern" "read"   "kill"   "mean"   "claim"  "exist"
```

3. LDA

The document representation that I will get after processing the data with LDA in R is shown as below:

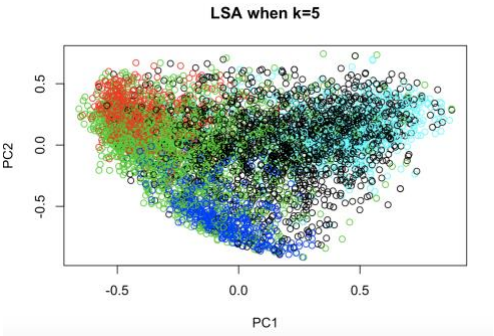
lda	Large LDA_Gibbs (2.7 Mb)
lda_cl	List of 9
lda_data	5548 obs. of 6 variables
lda_mat	Large matrix (33288 elements, 564.2 Kb)
lsa_cl	List of 9

LDA model will be estimated using Gibbs Sampling. The main parameters for LDA() are as follows:

- burnin - starting period, steps which does not reflect distribution property are removed.
- iter - number of iterations -
  thin - number of iteration at which correlation between samples is avoided
- seed - an integer for each starting point
- nstart - number of runs at different start points
- best - return result of best run
- LDA(dtm,4, method="Gibbs", control=list(nstart=nstart, seed = seed, best=best, burnin = burnin, iter = iter, thin=thin))

- Cluster LDA vectors with kMeans using the same **K** as for tf-idf

K=5, the result shown as below:



- For each of the top 5 concepts report the most representative words

```
> terms(lda,10)
      Topic 1   Topic 2  Topic 3   Topic 4    Topic 5
 [1,] "file"    "peopl"  "just"    "pleas"    "good"
 [2,] "window"  "right"  "think"   "anyon"    "year"
 [3,] "system"  "point"  "like"    "thank"    "game"
 [4,] "program" "believ" "thing"   "need"     "well"
 [5,] "also"    "make"   "problem" "post"     "time"
 [6,] "graphic" "mean"   "seem"    "work"     "first"
 [7,] "comput"  "state"  "someth"  "email"    "better"
 [8,] "version" "reason" "said"    "interest" "last"
 [9,] "softwar" "person" "time"    "card"     "look"
[10,] "avail"   "claim"  "never"   "book"     "team"
```

II.C Evaluation

Comparing Clustering results by evaluating SSE:
Within-SSE Ratio <- ((k$tot.withinss / k$totss) * 100)
Clustering with k-means:

1) SSE Measure:

Number of Clusters	Total Sum of Squares	Total Within Sum of Squares	Within-SSE Ratio(%)
5	633088.8	572409.7	90.41
6	633088.8	566890.9	89.54

2) Confusion Matrix:

Accuracy <- (sum(apply(c_matrix,1,max))/sum(k3$size))*100

Accuracy : 96.4%

	1	2	3	4	5	6
alt.atheism	955	26	10	5	4	1
comp.graphics	20	964	8	4	2	2
comp.os.ms-windows.misc	21	7	966	2	1	3
misc.forsale	3	5	12	961	7	12
rec.sport.baseball	1	30	1	2	958	8
talk.politics.guns	0	10	2	3	5	980

3) Precision, Recall &F1:

	Precison	Recall	F1
alt.atheism	0.87	0.98	0.99
comp.graphics	0.98	0.82	0.96
comp.os.ms-windows.misc	0.94	0.88	0.82
misc.forsale	0.84	0.87	0.85
rec.sport.baseball	0.92	0.91	0.98
talk.politics.guns	0.85	0.92	0.81

LSA:

1) SSE Measure:

Concepts	Total Sum of Squares	Total Within Sum of Squares	Within-SSE Ratio(%)
50	2078.017	1883.21	90.63
100	2193.872	2011.54	91.69
200	2256.234	2122.67	94.08

2) Confusion Matrix:

For d=200(same way it can be computed for d = 50,100)

Accuracy : 99.2%

	1	2	3	4	5	6
alt.atheism	989	2	4	0	4	1
comp.graphics	0	992	0	4	2	2
comp.os.ms-windows.misc	2	3	989	2	1	3
misc.forsale	1	2	1	993	1	2
rec.sport.baseball	1	0	1	0	998	0
talk.politics.guns	5	0	1	3	0	991

LDA:

1) SSE Measure:

Total Sum of Squares	Total Within Sum of Squares	Within-SSE Ratio(%)
308.1571	128.2058	58.39594

2) Confusion Matrix:

	1	2	3	4	5	6
alt.atheism	980	11	3	1	3	2
comp.graphics	24	968	2	2	1	3
comp.os.ms-windows.misc	1	14	978	2	2	3
misc.forsale	3	20	9	965	0	3
rec.sport.baseball	1	2	1	13	978	5
talk.politics.guns	2	3	12	10	10	963

Accuracy : 97.2%

3) Precision, Recall &F1:

	Precison	Recall	F1
alt.atheism	0.81	0.92	0.82
comp.graphics	0.89	0.88	0.81
comp.os.ms-windows.misc	0.82	0.91	0.82
misc.forsale	0.87	0.82	0.85
rec.sport.baseball	0.91	0.81	0.91
talk.politics.guns	0.82	0.82	0.84

Yelp Dataset

I Data (15 points)

Yelp dataset was released for academic challenges which is quite bigger when compared to 20 Newsgroup Dataset. The dataset downloaded from the website is 5.79Gb in size, with 6 files in JSON format. This dataset contains user, business, review, checkin, tip & photo information about local businesses in 12 metropolitan areas across 4 countries, with around 156639 businesses. Since, the yelp dataset is bulky, it is not a feasible plan to work on the entire dataset. Rather two of the json files are selected so that sub-problem infer-categories can be worked upon. Therefore, business.json and review.json are chosen. This requires preprocessing both the business and review json files. Jsonlite library can be used to work upon json files and extract what is required. There are total 1240 categories with various reviews. Out of these categories, three are selected to perform our analysis, which are "mobile phones", "real state" & "active life". The selected 3 categories are the reviews in the dataset. Since these 3 categories belong to different fields but the analysis performed on the reviews showed that there might have been a mix of topics. Example: reviews for mobile phones are mostly about electronics like where to buy from, repair shops etc. The raw data thus obtained contains many meaningless and redundant data which needs to be processed before performing any analysis.

II Experiments

II.A Data Preprocessing

I use the data provide in PIAZZA, which is .csv format.
The three categories which are decided in previous section will be used to provide better analysis and understanding. Since they belong to 3 different fields, clustering, LSA & LDA will perform well on this selected data.
Categories:

- Real Estate
- Active Life
- Mobile Phone

The data obtain contains whitespace, punctuation & stop words which are removed and is then stemmed to make the data available meaningful. The following table shows the most frequent words & figure provides the word cloud as shown below:

No	Word	Frequence
1	place	879
2	great	868
3	good	768
4	great	751
5	time	737
6	food	636
7	just	559
8	like	511
9	servic	509
10	back	490

Following tasks were completed after preprocessing:

- Document term matrix is created using the inbuilt DocumentTermMatrix(). This matrix describes the frequency of terms which occurs in a document.
- Stemming is performed on the cleaned data because the classifier doesn't understand verbs(i.e. organized & organizing) and treats them as different words.
- Pruning words by frequency: words that occurs in very few documents were removed. The vocabulary size after pruning decreased from 36452 to 1415, which is good because the entire focus is on the words which have high frequency. Therefore

during LSA/LDA/Clustering, important data is analyzed. Both LSA and LDA are topic models.

- Tf-Idf weights are used on Document Term Matrix which is one of the popular term-weighting schemes which will be used for document-term matrix. It is a statistical measure which evaluates the importance of word in the document.

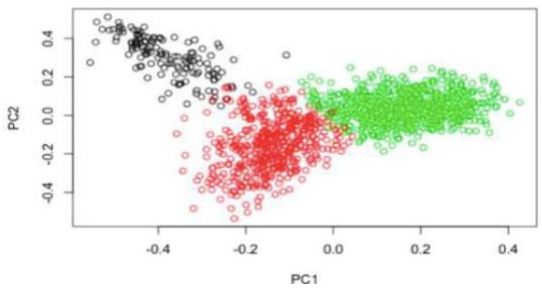
II.B Clustering Experiments.

1. K-Means & NbClust: This is same as what was done for 20 Newsgroup Dataset. Just like our previous dataset, k- means is plotted for k = 2,3,4. Again, Silhouette method is used to discuss and set the value of K. Following command is used to evaluate k-means.
k3 <- kmeans(dtm_tfidf, 3) plot(prcomp(tfidf)$x,col=k3$cluster)
The result shown as below:

```
#Optimal Clusters
avg_sil <- function(k) {
  km.res <- kmeans(tfidf, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(tfidf))
  mean(ss[, 3])
}

# Compute and plot wss for k = 2 to k = 9
k.values <- 2:9

# extract avg silhouette for 2-9 clusters
avg_sil_values <- map_dbl(k.values, avg_sil)
```



Average Silhouette method is used next to set K value.
Result: The graph plotted for silhouette method gives the optimal number of clusters as 3.
Confusion Matrix:
Accuracy: 79%

	1	2	3
Real Estate	476	9	128
Mobile Phone	23	349	1
Active Life	0	65	275

NbClust: Function NbClust() was used to determine the best numbers of cluster.
Parameters: distance = euclidean, minimum number of clusters, maximum number of clusters, method, index.

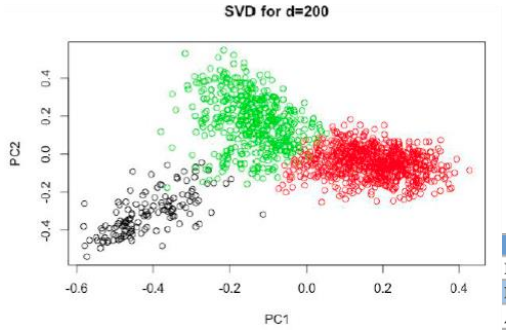
- NbClust(as.matrix(tfidf), distance = "euclidean", min.nc = 2, max.nc = 9, method = 'kmeans', index = 'silhouette')
- $Best.nc: Number_clusters = 3 ; Value_Index = 0.4678
- Result: Best number of clusters is 3.

2. LSA:

Most Frequent words for 5 concepts:

No.	Frequent Words
1	apart, manag, time, live, month
2	phone, repair, great, servc, like
3	life, golf, play, will, golf
4	store, help, nice, staff, leas
5	clean, cours, green, like, est

For d = 200: Cluster plot & Confusion matrix. Accuracy = 68% (same way it can be shown for d = 50,100)



	1	2	3
Real Estate	476	9	128
Mobile Phone	23	321	29
Active Life	29	12	299

3. LDA: Same with procedure as 20 Newsgroup dataset.
Most representative words SSE Measure

```
      Topic 1  Topic 2  Topic 3
 [1,] "time"   "place"  "apart"
 [2,] "just"   "like"   "live"
 [3,] "work"   "great"  "move"
 [4,] "call"   "staff"  "manag"
 [5,] "need"   "year"   "month"
 [6,] "even"   "nice"   "offic"
 [7,] "come"   "look"   "rent"
 [8,] "back"   "cours"  "leas"
 [9,] "servic" "good"   "peopl"
[10,] "never"  "friend" "complex"
```

II.D Results Summary:

So far, 3 types of document representations were evaluated, which are tf-idf, LDA & LSA. Few useful observations are as follows:
- Tf-idf used word frequency counts as document vector's feature.
- LSA & LDA represents documents as vectors in space.
- LDA's results were not satisfactory in both the cases.
- LSA's result was much better than tf-idf. When 200 dimensional LSA representation is used, best clustering is obtained.

Below table summarizes the results so far:

20 NewsGroup	Accuracy	Yelp	Accuracy
Tf-idf	94%	Tf-idf	79%
LSA(d=50)	89%	LSA(d=50)	73%
LSA(d=100)	97%	LSA(d=100)	70%
LSA(d=200)	99%	LSA(d=200)	68%
LDA	85%	LDA	51%

III Analysis:
• The analysis performed here was designed in a way to make it easier to grasp the concepts behind mining of the data.
• Datasets used for our analysis were 20 Newsgroup Dataset and Yelp Dataset, with different characteristics. The raw data was then preprocessed at initial stage.
• Pruning words by i.e. removing words which occur in very less documents is a better way to prepare data for analysis so that focus remains on the most important data.
• Tf-idf weights were then used on document term matrix to reflect how important a word is to a document in a collection.
• The K value obtained via k-means was further supported by Silhouette method and NbClust in both the datasets.
• LSA, depending upon the dataset can handle synonymy problems. It has better runtime since it only involves decomposing document term matrix. It becomes less efficient in front of deep neural networks.
• LSA was computed for different dimensions, but a word of caution, computation of LSA for different dimensions will affect the distance between vectors of document.

Hence, the increase in SSE for clustering LSA. Although, LSA with 200 dimension gave highest accuracy, meaning clustering was not that worse.
• LSA concepts when more in number gives better result, this was proved when high accuracy was achieved for 200
dimensional LSA.
• The clustering plots shows that the documents from each of the groups were nicely separated.
• The most frequent words obtained as a result of LDA representation in 20 Newsgroup shows that LDA successfully discovered semantic topics of 3 newsgroups. Topics discovered were related to Christian, sports and technology.
• Yelp data's result were not that satisfactory when compared with 20 newsgroup.
• This homework gave me an opportunity to work on real data, data obtained from newspaper and yelp. I learned to: Work on data right from the scratch, from when the data is raw and meaningless to meaningful data. Perform cluster experiments on the data and find the word that belongs to the same group.LSA & LDA, which really help me to learn a lot.

IV) LSA Derivation:

A is a n*m term document matrix.

$$\text{SVD(A)} = \text{A} = \text{UD}V^T$$

this is the SVD of A.

Since $A = A^T, AA^T = A^TA = A^2$, U - n*n orthogonal matrix , V - m*m orthogonal matrix , D - n*m matrix with other entries apart from diagonal are 0.

If A is a positive definite and symmetric, SVD and Eigen decomposition coincide.

We know, $A^T = (UDV^T)^T = \text{VD}U^T$

Then we have,

$$\text{A}A^T = \text{UD}V^T\text{VD}U^T = \text{U}D^2U^T = A^2$$
$$A^T\text{A} = \text{VD}U^T\text{UD}V^T = \text{V}D^2V^T = A^2$$

So, for these matrices, SVD(A) can be used to compute their SVD and also they are singular positive definite then this will be their Eigen decomposition($\text{D} = D^2$)

LSA document representation for clustering experiment:
SVD matrices are:
U : is orthogonal and relates terms to topics
D : gives importance of topics with only diagonal entries, rest are 0
V : is orthogonal and relates documents to topics
Thus, $\text{D} \times \text{V}^\text{T}$ provides k dimensional LSA document vectors whereas D*U provides k dimensional LSA word vectors.
The following commands were used to get LSA document representation for clustering experiment. This representation depends on SVD matrices where the documents are re-represented in a reduced vector space in which they have higher similarity.

```
#LSA/SVD
s<-svd(dtms)
D <- diag(s$d)
D50 <- D[1:50,1:50]
u <- as.matrix(s$u[,1:50])
v <- as.matrix(s$v[,1:50])
d <- as.matrix(diag(s$d)[1:50,1:50])
mat50 <- as.matrix(u%*%d%*%t(v))
```