DTI5125: Data Science Applications

Text Clustering Group Assignment 2

Group Members:

Basma Reda Shaban Abd-Elsalam Abd-Elwahab

Amir Safwat Halim Youssef

Nada Ashraf Ismail AboElfetoh

Yasmine Ahmed Elsayed Mohamed

1- Overview:

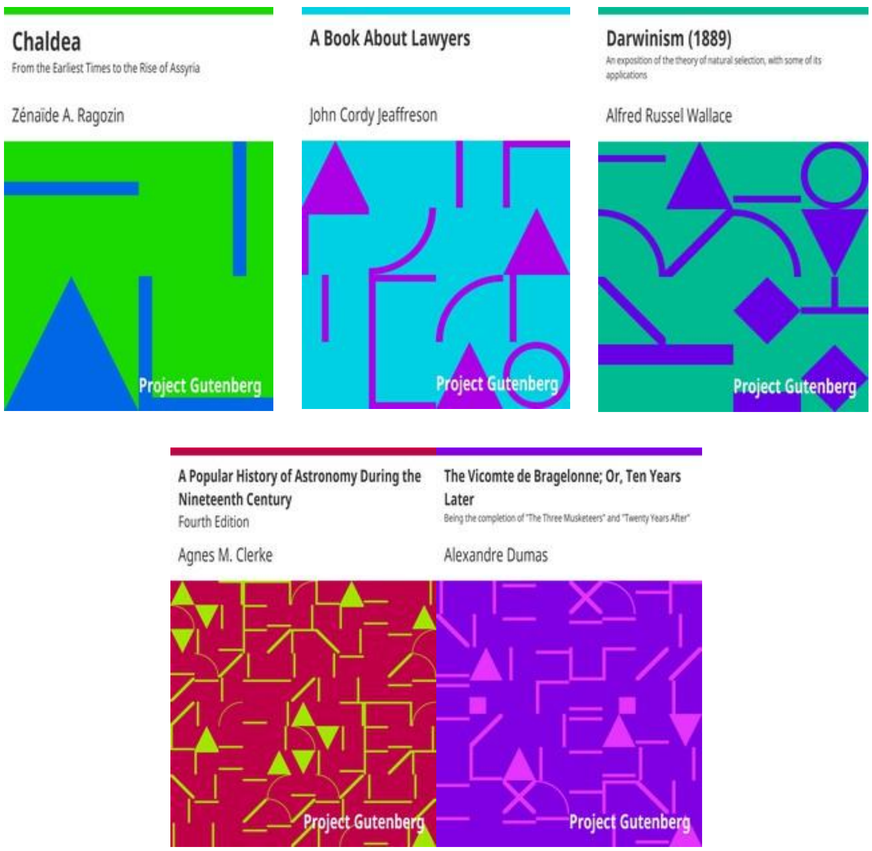
The main objective of this assignment is to produce similar clusters and compare them. Having five different books from five different categories, and five different authors, from Gutenberg website, and hide the label of the book.

This report discusses the different transformations and clustering techniques and a comparison between them, and which will be similar to the true label.

2- Methodology

We followed some defined steps to obtain the aimed results:

2.1 Preparing our different five books:



The books authors are: "Zénaïde A. Ragozin", "John Cordy", "Alfred Russel Wallace", "Alexandre Dumas, Père", "Agnes M. (Agnes Mary) Clerke".

2.2 Data preparation Step:

1- Removing stop words and garbage characters, converting all words to lower case, and performing lemmatization to return every word to it's origin:

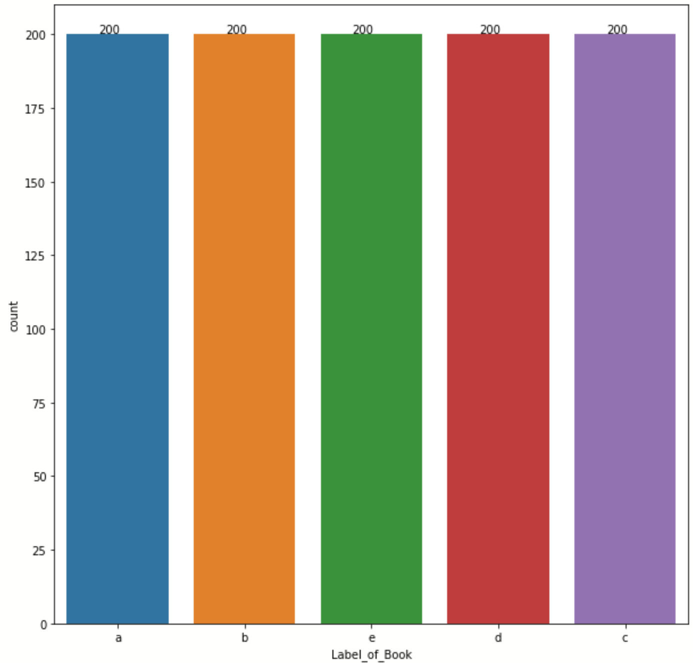
```
from urllib import request
#for loop to get every book in BooksURLs list
for URL  in BooksURLs :
  response = request.urlopen(URL)
  raw = response.read().decode('utf8' , errors = 'replace')
  wordsList= re.findall(r"[a-zA-Z]{3,}", raw)
  #perform lemmetization on the data
  lemmatizer = WordNetLemmatizer()
  lemmitizedWords =[]
  for i in wordsList:
    words = i.lower()
    word = lemmatizer.lemmatize(words)
    #check if the word not in stopwords set
    if word not in set(stopwords.words('english')):
      lemmitizedWords.append(str(word))
  Books.append(lemmitizedWords)
```

2- Split every book into 200 partitions, and every partition have 150 words, and labeling books as [a, b, c, d, e]

```
[14] #print head of data
     result.head(5)
```

	index	Author_of_Book	Title_of_Book	Label_of_Book	PartitionsList
19	0	Zénaïde A. Ragozin	Chaldea	a	utility road irrigation encouragement commerce...
171	1	John Cordy	A Book About Lawyers	b	cake consequence custom required new judge sen...
19	4	Agnes M. (Agnes Mary) Clerke	A Popular History of Astronomy During the Nine...	e	upon modern astronomy associative character el...
155	3	Alexandre Dumas, Père	The Vicomte de Bragelonne	d	artagnan musketeer tolerably bad humor desired...
123	1	John Cordy	A Book About Lawyers	b	displeasure seldom care discriminate blameless...

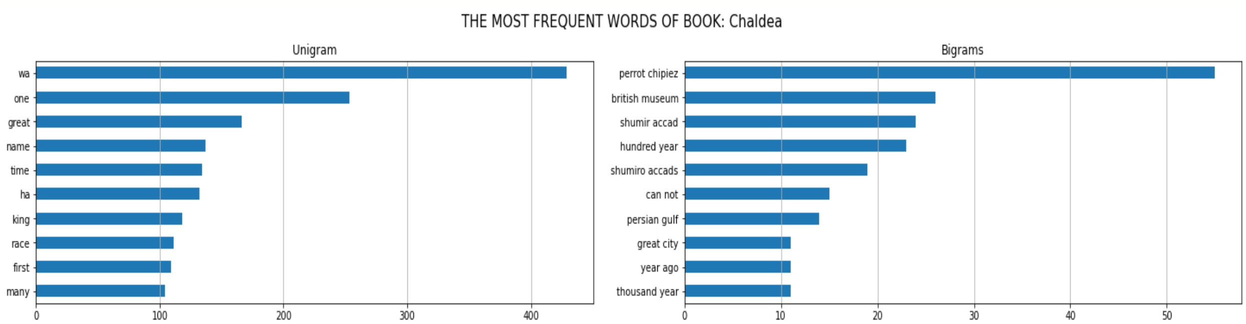
3- Ensure that every book have 200 partitions:



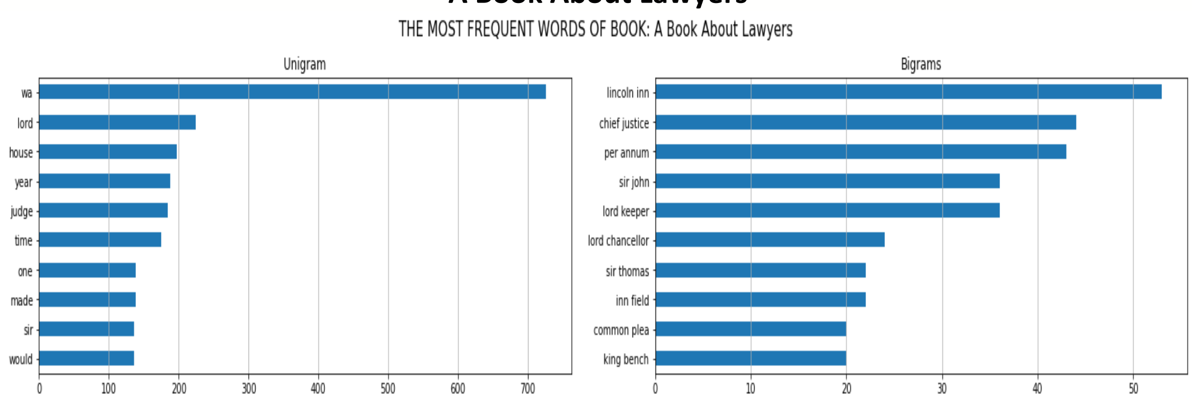
4- Showing the most frequent words in every book using two techniques:

1- Unigram and Bigram:

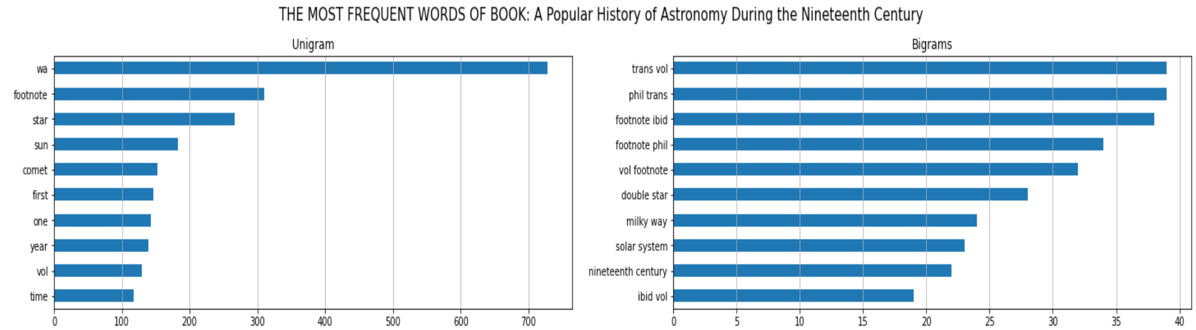
Chaldea book



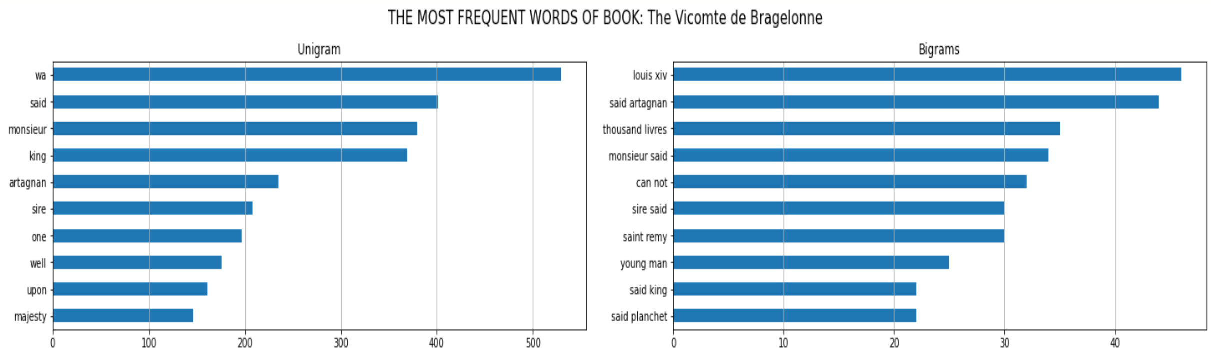
A Book About Lawyers



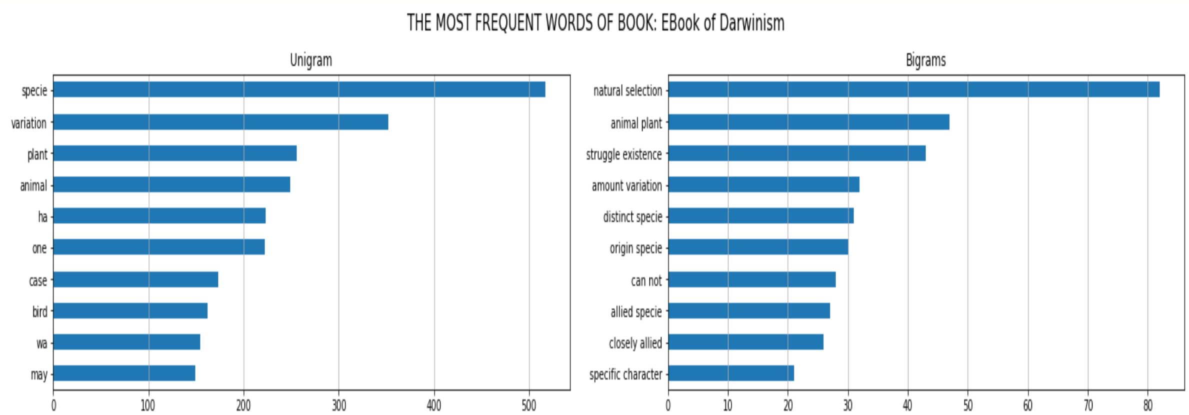
A Popular History of Astronomy During the Nineteenth Century



The Vicomte de Bragelonne



EBook of Darwinism



2- Using wordCloud:

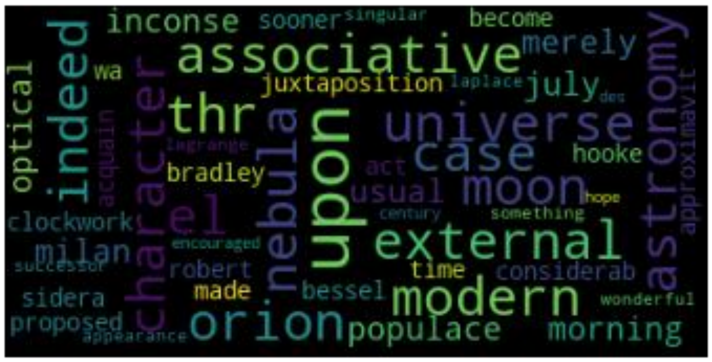
Chaldea book

A Book About Lawyers

The Vicomte de Bragelonne

EBook of Darwinism

A Popular History of Astronomy During the Nineteenth Century



2.3 Perform Feature Engineering using 4 methods:

1- **Bag Of Words (BOW) Transformation**: A bag of words is a representation of text that describes the occurrence of words within a document.

	aaron	abandon	abandoned	abandoning	abandonment	abated	abb	abbe	abbey	abbott	...	zodiacal	zonal	zone	zool	zoologique	zoologist	zoology	zur	zwischen	zygomatic
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows × 16048 columns

2- **TF-IDF Transformation**: Term frequency (TF) vectors show how important words are to documents. They are computed by using:

$$tf(term, document) = \frac{number\ of\ times\ the\ term\ occurs\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

	aaron	abandon	abandoned	abandoning	abandonment	abated	abb	abbe	abbey	abbott	...	zodiacal	zonal	zone	zool	zoologique	zoologist	zoology	zur	zwischen	zygomatic
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

000 rows × 16048 columns

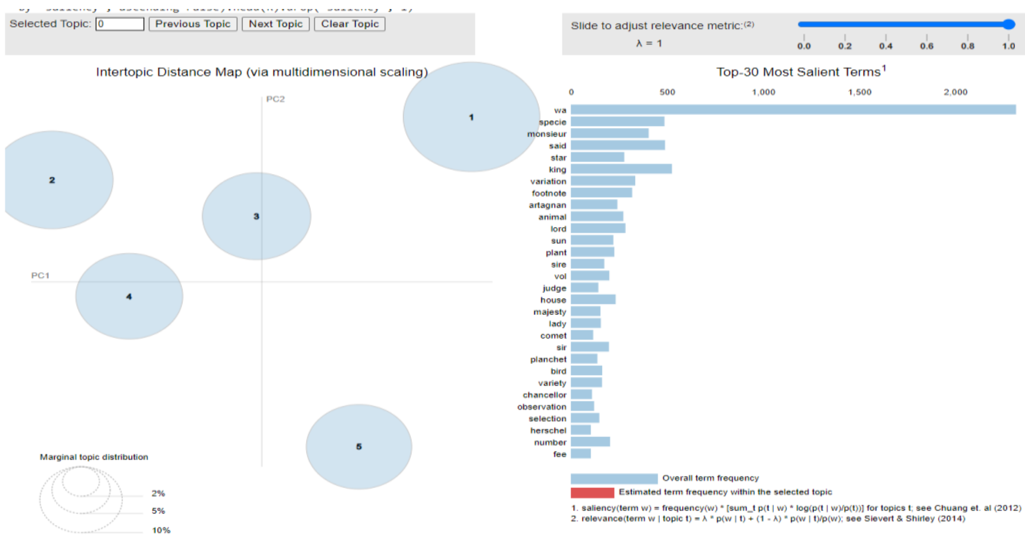
3- LDA Transformation: LDA is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words.

	1	2	3	4	5	res
0	40.195923	0.179589	86.287994	17.819510	6.548666	3
1	6.840732	0.179266	0.174638	9.724366	134.112686	5
2	42.947739	86.382599	3.532179	5.912141	12.257037	2
3	13.155953	1.853795	0.175965	131.157608	4.688347	4
4	11.478949	5.613846	0.174887	15.622274	118.141701	5

The predicted words of LDA Transformation:

```
(array([[ 40.195923  ,   0.17958926,  86.287994  ,  17.81951   ,
          6.548666  ],
        [  6.840732  ,   0.1792659 ,   0.17463806,   9.724366  ,
        134.11269   ],
        [ 42.94774   ,  86.3826    ,   3.5321789 ,   5.9121413 ,
         12.257037  ],
        ...,
        [ 35.19193   , 115.3203    ,   0.17444904,   0.20968111,
          0.13532138],
        [ 20.348946  , 113.14746   ,   7.5316253 ,   0.20784229,
          9.795807  ],
        [  0.37172854,   0.17994398,   4.8293304 ,  26.066813  ,
        119.58385   ]], dtype=float32), None)
```

LDA is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.



4- **Word Embedding (Word2Vec) Transformation:**

Word2Vec consists of models for generating word embedding.

```
[ 6.39032647e-02 -1.00076301e-02 -2.47444082e-02  2.49532200e-02
 -1.64452597e-01 -4.05765250e-02  2.44001463e-01  5.67615628e-02
  2.46330827e-01  1.70743361e-01 -2.93254405e-02  1.92961097e-02
  1.25935256e-01  1.31534727e-03  1.90286748e-02  2.25926861e-02
 -9.74692628e-02  1.18261680e-01  6.58749230e-03  6.97476864e-02
  3.08375317e-03  7.08437189e-02  3.83944809e-02 -5.04423641e-02
 -4.23085783e-03 -7.40122944e-02  1.63575495e-03  2.11675793e-01
 -2.21240018e-02 -2.99621429e-02  1.15621099e-02 -1.41864195e-01
  2.80737784e-02 -5.88704869e-02  3.37919854e-02 -1.00738637e-01
  7.35347271e-02 -6.23905435e-02 -9.10108723e-03  6.61205128e-02
 -1.37935922e-01  2.97885109e-02 -7.19079301e-02  2.71166041e-02
 -5.13751842e-02 -5.43224849e-02 -9.41491723e-02 -1.49340108e-01
 -3.66348475e-02  1.78339824e-01 -6.72631431e-03  3.38511840e-02
  9.75831002e-02  1.75848529e-01  1.65832154e-02  2.20664948e-01
 -1.99374110e-01 -1.83159238e-04  9.93847549e-02  1.08866366e-02
  7.73861483e-02 -2.10308209e-01 -6.40131012e-02  2.25411534e-01
  2.77508423e-02 -1.01526216e-01 -1.68809928e-02  2.53384203e-01
 -1.07851893e-01 -8.49062856e-03 -2.08907742e-02 -2.18841985e-01
  6.51334375e-02 -4.35428321e-02  9.99771878e-02  1.06946655e-01
 -8.23654160e-02 -3.00993957e-02  2.81349123e-01  5.98165877e-02
  1.28292799e-01 -1.07078373e-01  1.92346856e-01  3.58668827e-02
  1.78771645e-01 -4.08566408e-02 -1.65989958e-02  2.05088761e-02
 -1.60516784e-01 -9.39027220e-02  1.24719597e-01 -6.23902828e-02
  1.01312868e-01  1.01647533e-01 -1.16523758e-01  1.58404782e-02
  2.49551699e-01 -2.11534634e-01 -7.85620362e-02 -2.11563744e-02
  1.24335773e-01  1.13423526e-01  8.49671811e-02 -2.49626804e-02
  2.84476101e-01 -2.10208058e-01 -1.88032840e-03  9.00942460e-02
  8.39605778e-02  2.24193856e-01 -4.03151363e-02 -1.30077943e-01
  4.24622511e-03 -2.51844257e-01 -3.11834086e-02 -3.11199576e-02
  2.07286865e-01  2.36877650e-01  1.44839182e-01 -1.97042711e-02
 -1.57025203e-01 -1.11813262e-01 -5.71856424e-02 -1.29472002e-01
  2.70347148e-01  1.84511244e-01  6.89389929e-02 -1.04417324e-01
  1.01622865e-01  2.80982628e-02  1.89747680e-02 -2.16896329e-02
  1.43130109e-01 -9.49415490e-02 -2.38705799e-01  9.39103402e-03
 -3.66527140e-02 -1.06075712e-01 -7.29509890e-02  1.47223040e-01
  5.28361695e-03 -1.24949686e-01  1.47070944e-01 -1.62038818e-01
 -1.51137924e-02  1.67974338e-01 -1.37995034e-01 -1.45792872e-01
  6.50867296e-04  1.07391723e-01]
```

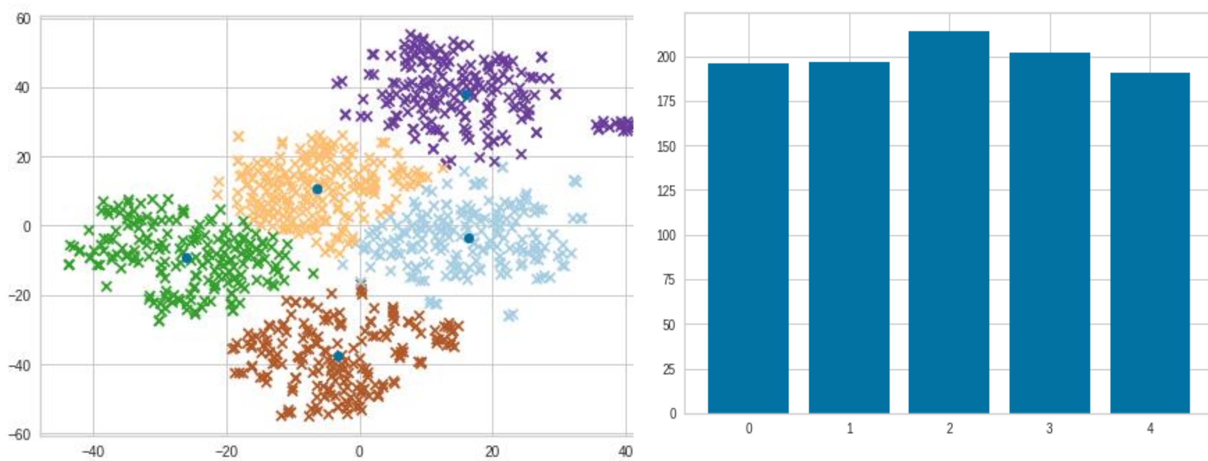
Dimensionality reduction: is a good way to deal with the data that have many features, the TSNE is a good choice here.

TSNE used to reduce the number of dimensions to a reasonable amount if the number of features is very high. It is good for the computation time and good visualization.

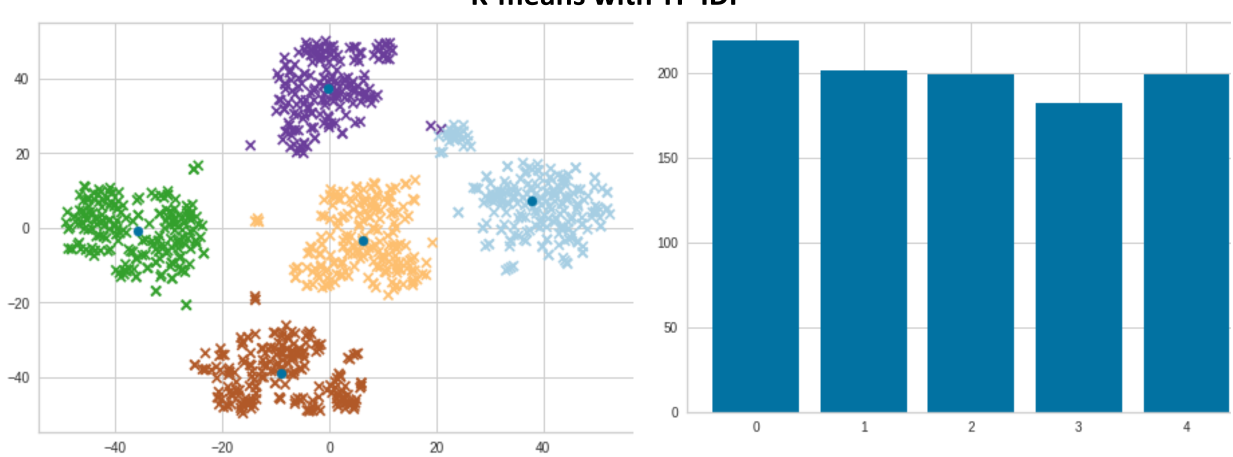
2.4 perform clustering algorithms:

1- K-Means clustering algorithm: K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

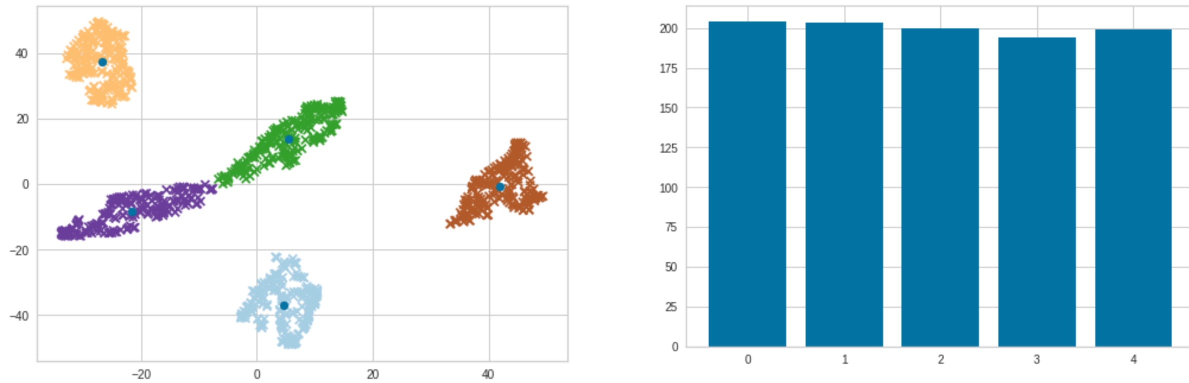
K-means with BOW



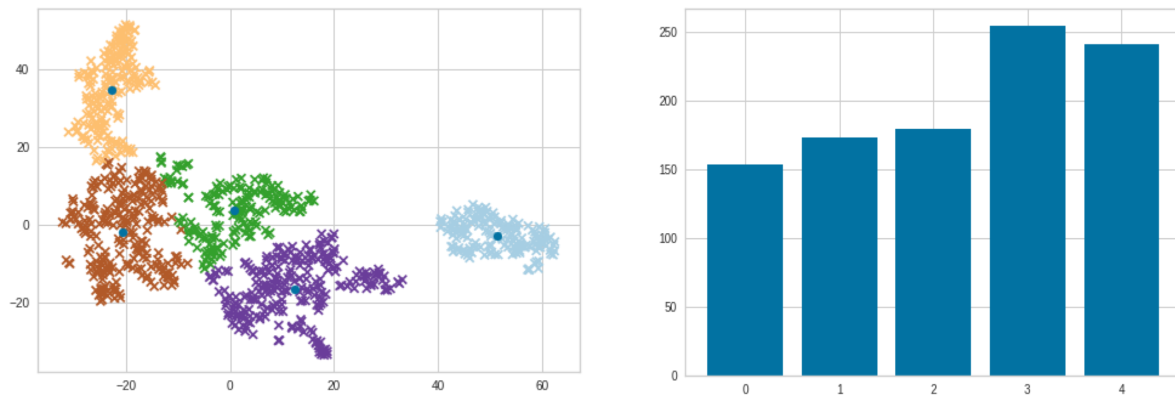
K-means with TF-IDF



K-means with LDA

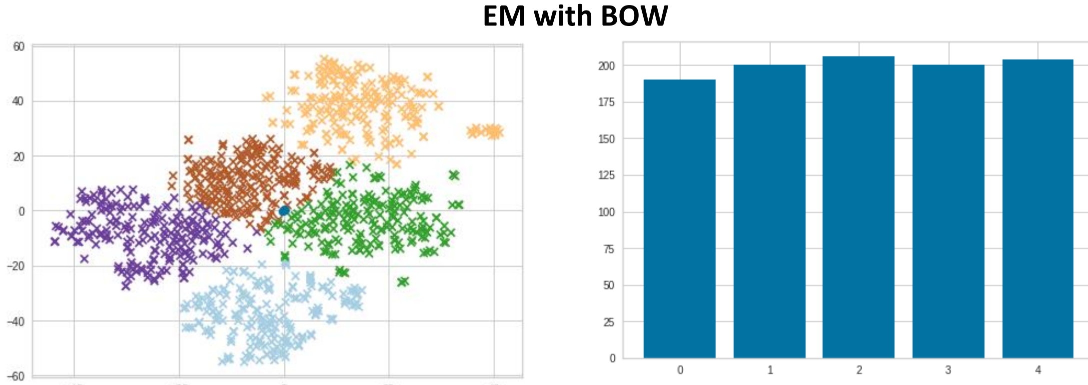


K-means with Word2Vec

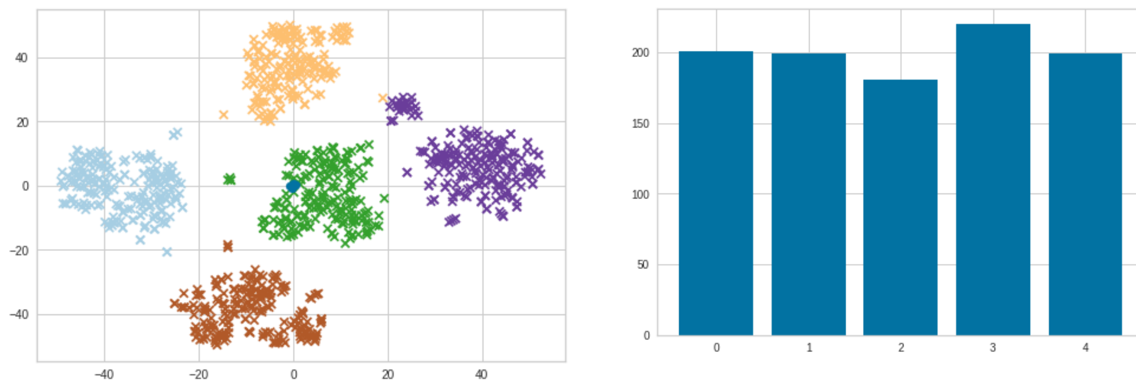


2- **Expectation Maximization(EM) algorithm:** ***Expectation-Maximization algorithm*** can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of the other observed variables) too in order to predict their values with the condition that the general form of probability distribution governing those latent variables is known to us. This algorithm is actually at the base of many unsupervised clustering algorithms in the field of machine learning.

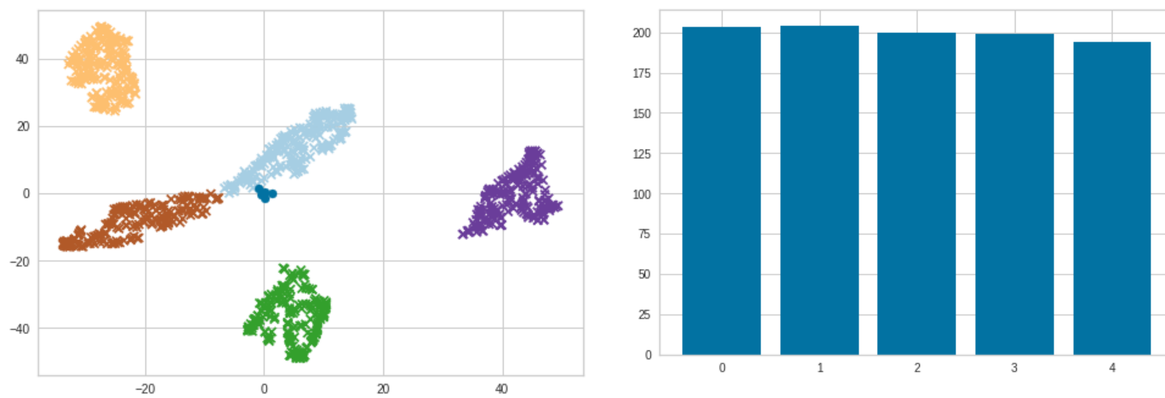
EM with BOW



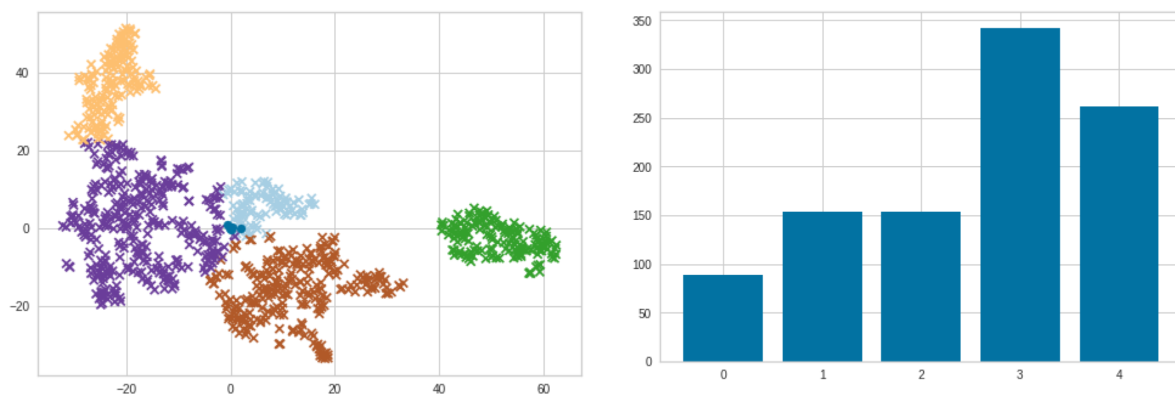
EM with TF-IDF



EM with LDA

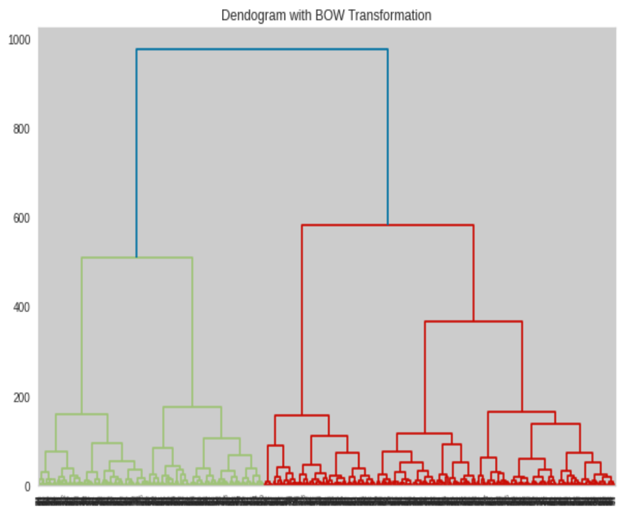


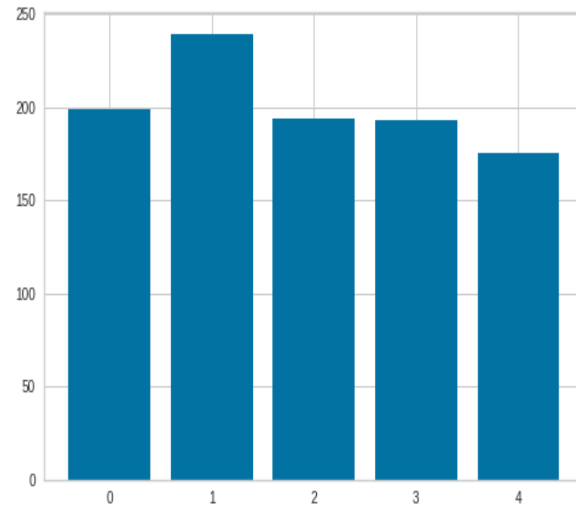
EM with Word2Vec



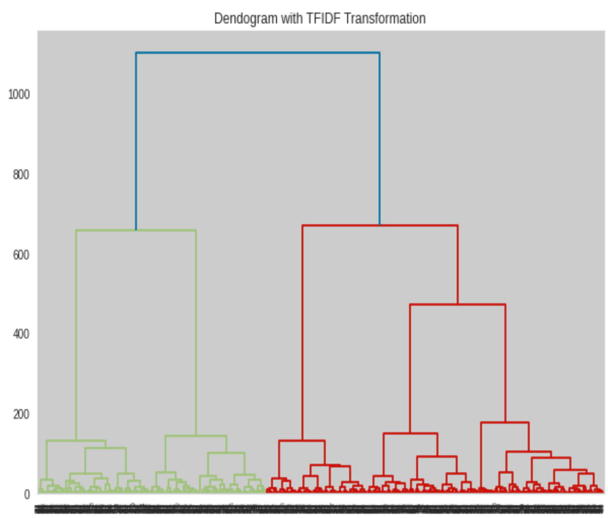
3- Hierarchal clustering algorithm: Hierarchical clustering, also known as hierarchical cluster analysis, is **an algorithm that groups similar objects into groups called clusters**. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

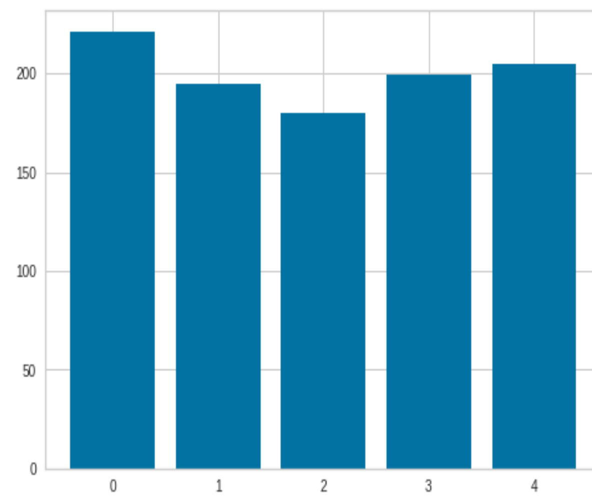
Hierarchal clustering with BOW



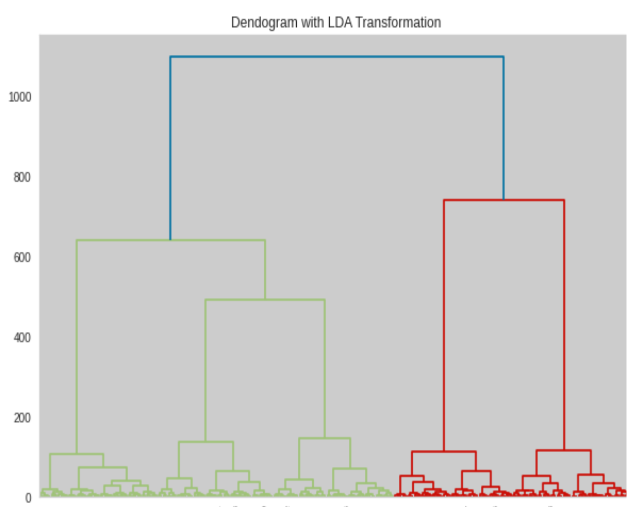


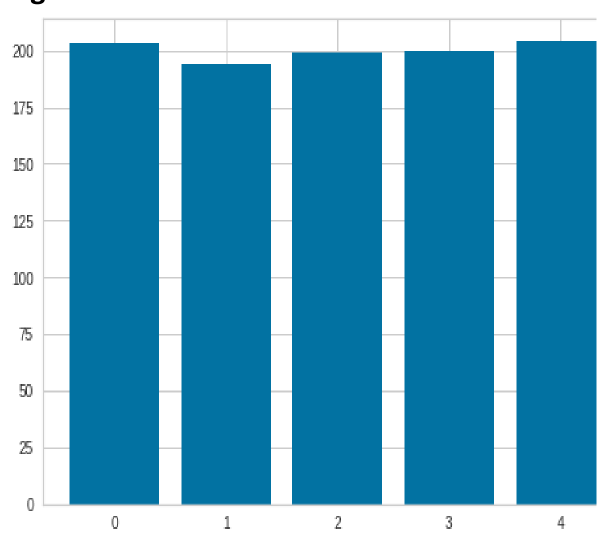
Hierarchal clustering with TF-IDF



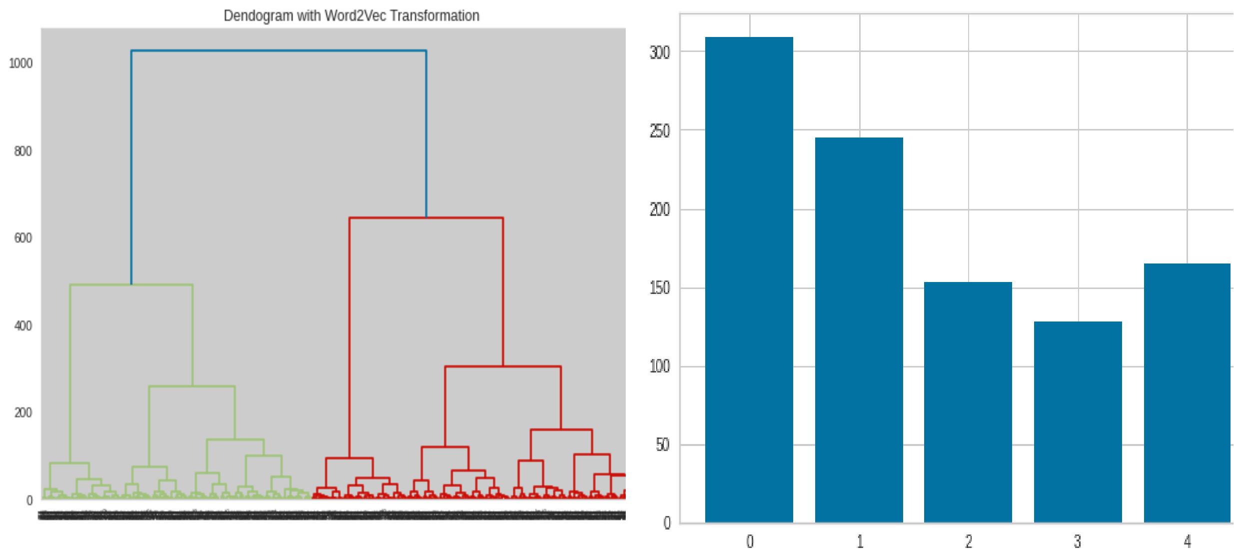


Hierarchal clustering with LDA





Hierarchal clustering with Word2Vec



2.5 Perform Evaluation using 4 methods:

1- Model Evaluation using Kappa: Cohen's kappa: a statistic that measures inter-annotator agreement.

Kappa's Score with K-Means clustering algorithm			
K-means with BOW	**K-means with TF-IDF**	**K-means with LDA**	**K-means with Word2Vec**
0.92125	0.96375	0.98625	0.73375
Kappa Score of LDA as Topic Modeling		0.27795	
Kappa's Score with Expectation Maximization(EM) algorithm clustering algorithm			
EM with BOW	**EM with TF-IDF**	**EM with LDA**	**EM with Word2Vec**
0.92750	0.96250	0.98625	0.70125
Kappa's Score with Hierarchal clustering algorithm			
Hierarchal clustering with BOW	**Hierarchal clustering with TF-IDF**	**Hierarchal clustering with LDA**	**Hierarchal clustering with Word2Vec**
0.91000	0.96125	0.98625	0.72875

The predicted labels were mapped to the true label before measuring kappa score to ensure we get a correct kappa score {for ex: most predicted cluster named 1 are mapping to cluster named 0 in the true labels , then change each 1 to zero to get correct kappa} merely changing cluster names only not predictions.

2- Model Evaluation using consistency with V-Score: V-measure cluster labeling given a ground truth.

V-Score with K-Means clustering algorithm			
K-means with BOW	**K-means with TF-IDF**	**K-means with LDA**	**K-means with Word2Vec**
0.8466708764209049	0.9227255819276063	0.9604186793621567	0.604108703376123
V-Score with Expectation Maximization(EM) algorithm clustering algorithm			
EM with BOW	**EM with TF-IDF**	**EM with LDA**	**EM with Word2Vec**
0.8523648741765133	0.9212490869283759	0.9604186793621567	0.6168610207344851
V-Score with Hierarchal clustering algorithm			
Hierarchal clustering with BOW	**Hierarchal clustering with TF-IDF**	**Hierarchal clustering with LDA**	**Hierarchal clustering with Word2Vec**
0.8460145486690573	0.9197807272127881	0.9604186793621567	0.627040876468714

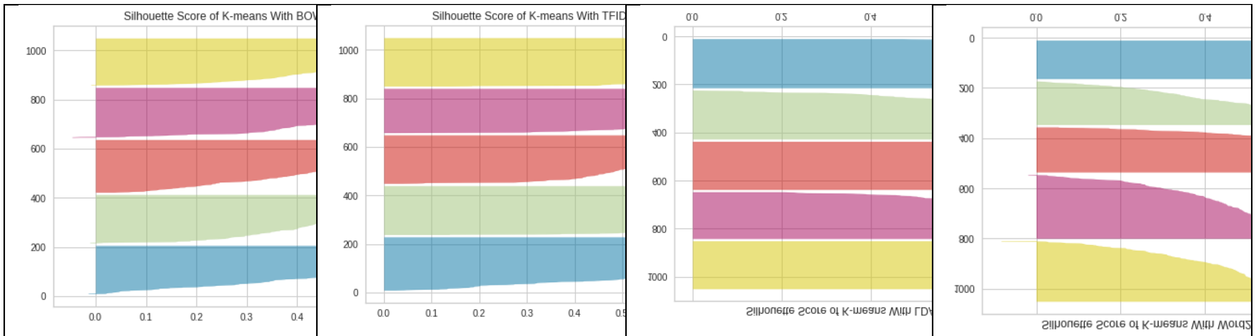
So the champion model is K-Means with TF-IDF, because it's closed to human labels.

3- Model Evaluation using Coherence: used with LDA
coherence is used to measure how well the topics are extracted.

Coherence score With LDA using c_v: **0.4363**
Higher value is better.
Coherence score With LDA using u_mass: **-8.0567** **Lower** value is better.

4- **Model Evaluation using Silhouette Score:** Silhouette score is used to evaluate the quality of clusters created using clustering algorithms such as K-Means in terms of how well samples are clustered with other samples that are similar to each other. The Silhouette score is calculated for each sample of different clusters. Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.

Silhouette Score with K-Means clustering algorithm			
K-means with BOW	**K-means with TF-IDF**	**K-means with LDA**	**K-means with Word2Vec**
0.47652	0.62407	0.72321	0.51574



Silhouette Score with Expectation Maximization(EM) algorithm clustering algorithm			
EM with BOW	**EM with TF-IDF**	**EM with LDA**	**EM with Word2Vec**
0.47449	0.62412	0.72321	0.48069
Silhouette Score with Hierarchal clustering algorithm			
Hierarchal clustering with BOW	**Hierarchal clustering with TF-IDF**	**Hierarchal clustering with LDA**	**Hierarchal clustering with Word2Vec**
0.46485	0.62326	0.72321	0.46807

2.6 Perform Error Analysis:

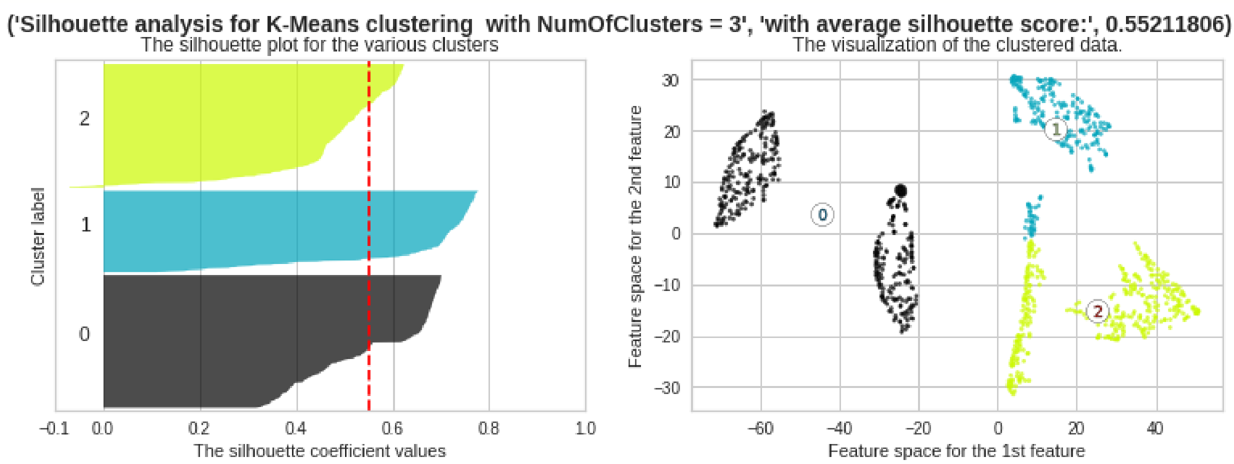
1- compare between silhouette scores using different K numbers:

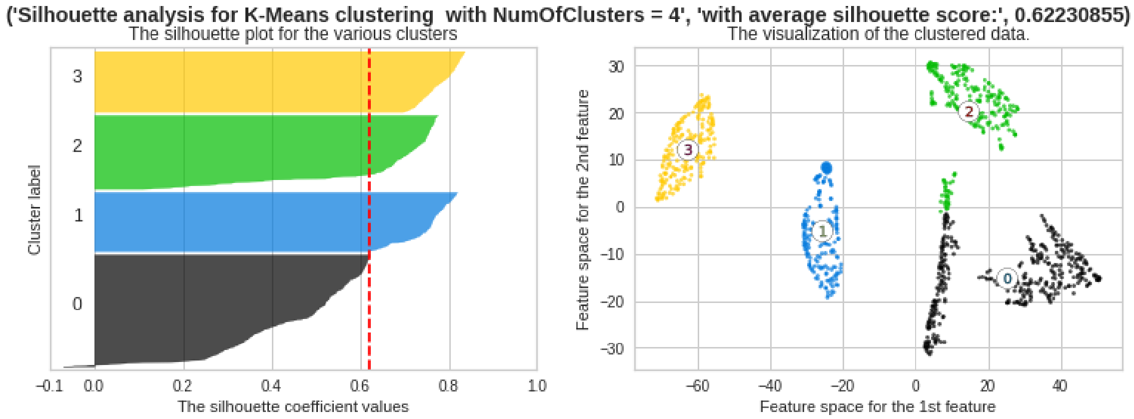
When K = 3 and K = 4:

```
For NumOfClusters = 3 The average of the silhouette score is : 0.55211806

For NumOfClusters = 4 The average of the silhouette score is : 0.62230855
```

The clusters are very bad.

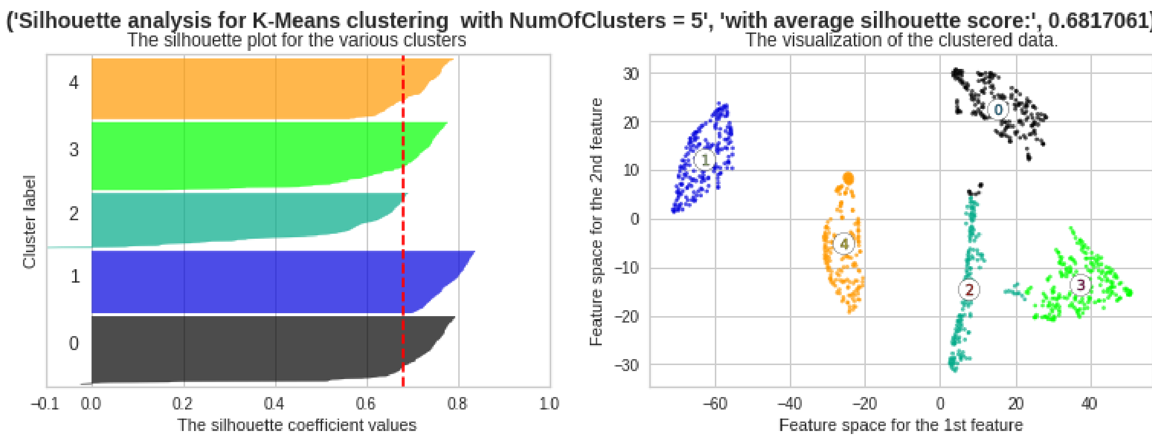




When K = 5:

`For NumOfClusters = 5 The average of the silhouette score is : 0.6817061`

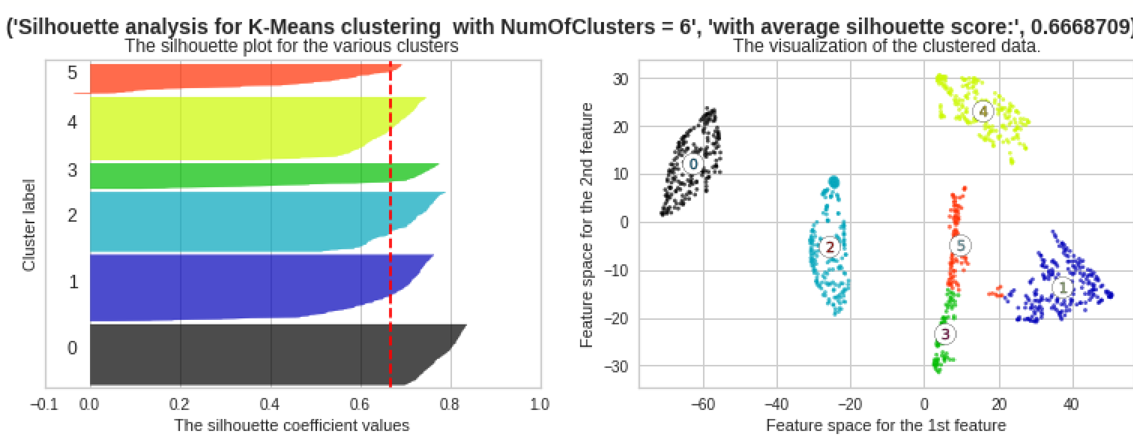
All clusters have the same size and clear clusters, the silhouette score is closest to 0.68, so this indicates the clusters and separated the clusters well.



When K = 6:

`For NumOfClusters = 6 The average of the silhouette score is : 0.6668709`

It is not a good K choice, because we have 2 clusters conflicted together.

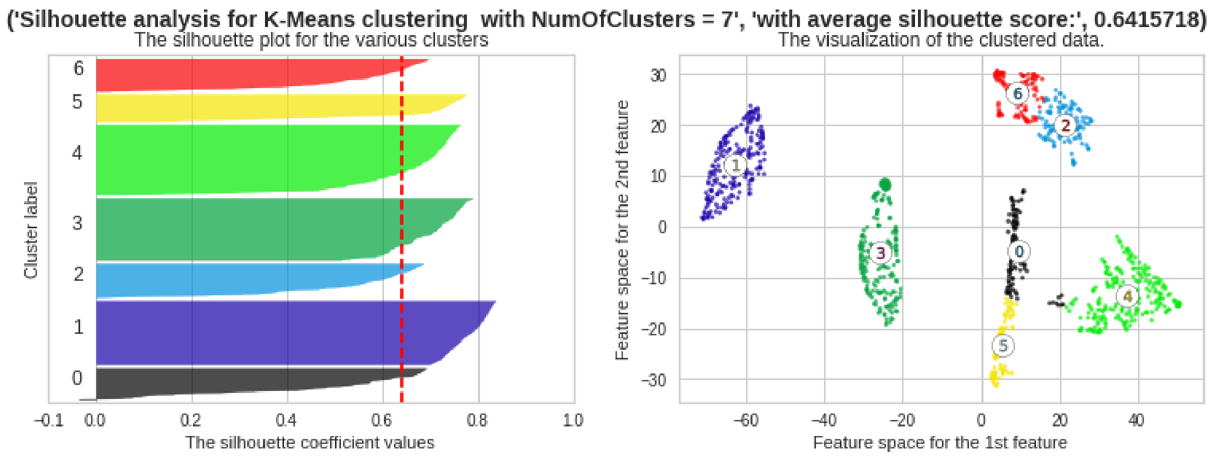


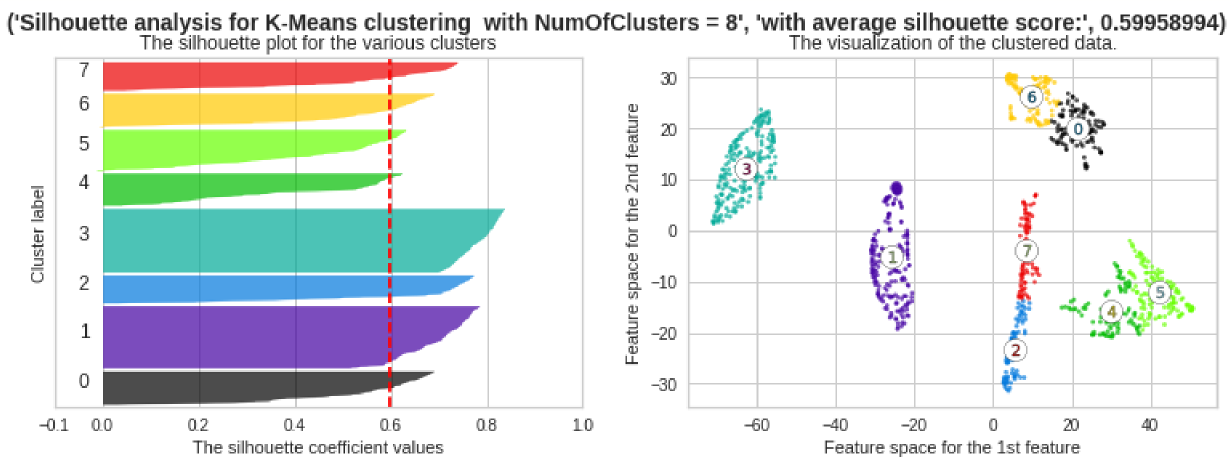
When K = 7 and K = 8

```
For NumOfClusters = 7 The average of the silhouette score is : 0.6415718

For NumOfClusters = 8 The average of the silhouette score is : 0.59958994
```

The silhouette score is not good as k = 5 and the classes have conflict together





The model Misclassified 64 rows:

	PartitionsList	Label_of_Book	index	clustersOutput
177	vol rosenberger calculated though lived lynn o...	e	4	0
3	may aware project gutenberg ha involved writin...	d	3	0
189	dim perception already arrived perhaps observa...	a	0	4
71	body much exception part agree large marked in...	c	2	4
81	considerable number spread varying distance si...	c	2	4
...	...	...	...	...
174	beneficial le beneficial le size body would be...	c	2	4
25	wa thus recognised domain far reaching specula...	e	4	1
50	trans vol footnote ibid vol cvii footnote bull...	e	4	0
175	dissipation extinction footnote footnote allge...	e	4	0
115	undoubtedly wa accads clear authentic insight ...	a	0	2

64 rows × 4 columns

The Most Frequent words with its occurancies through the actual class:

```
most frequent words in label : 0
 [('wa', 529), ('said', 401), ('monsieur', 380), ('king', 369), ('artagnan', 235), ('sire', 208), ('one', 197), ('well', 176), ('upon', 161), ('majesty', 146), ('man', 143), ('two', 143), ('would', 136), ('ha', 133), ('planchet', 129), ('know', 122), ('good', 119), ('yes', 116), ('louis', 115), ('say', 113), ('without', 112), ('hand', 111), ('time', 110), ('shall', 107), ('day', 106)]

most frequent words in label : 1
 [('wa', 728), ('footnote', 310), ('star', 267), ('sun', 183), ('comet', 152), ('first', 147), ('one', 143), ('year', 139), ('vol', 129), ('herschel', 118), ('time', 118), ('body', 107), ('observation', 106), ('system', 104), ('astronomy', 101), ('two', 101), ('solar', 99), ('light', 96), ('may', 90), ('great', 88), ('ha', 86), ('discovery', 84), ('object', 83), ('however', 82), ('result', 81)]

most frequent words in label : 2
 [('specie', 517), ('variation', 352), ('plant', 256), ('animal', 249), ('ha', 223), ('one', 222), ('case', 173), ('bird', 162), ('wa', 154), ('may', 149), ('form', 145), ('number', 144), ('selection', 137), ('many', 137), ('variety', 134), ('would', 122), ('darwin', 121), ('fact', 119), ('character', 119), ('part', 118), ('great', 117), ('two', 116), ('change', 110), ('among', 109), ('nature', 108)]

most frequent words in label : 3
 [('wa', 429), ('one', 253), ('great', 166), ('name', 137), ('time', 134), ('ha', 132), ('king', 118), ('race', 111), ('first', 109), ('many', 104), ('even', 103), ('could', 100), ('city', 98), ('year', 97), ('land',
```

```
91), ('must', 90), ('god', 88), ('called', 86), ('work', 85), ('also',
84), ('ancient', 83), ('found', 83), ('may', 82), ('would', 78),
('country', 77)]
```

most frequent words in label : 4

```
  [('wa', 726), ('lord', 225), ('house', 197), ('year', 188), ('judge',
184), ('time', 175), ('made', 138), ('one', 138), ('would', 136), ('sir',
136), ('fee', 128), ('lady', 125), ('court', 125), ('inn', 120),
('lawyer', 118), ('law', 118), ('chancellor', 105), ('upon', 102), ('day',
101), ('justice', 95), ('chief', 91), ('wife', 90), ('present', 89),
('great', 87), ('street', 84)]
```

The Most Frequent words with its occurancies through the actual cluster:

most frequent words in cluster : 0

```
  [('wa', 526), ('said', 400), ('monsieur', 379), ('king', 364), ('sire',
208), ('artagnan', 208), ('one', 195), ('well', 175), ('upon', 161),
('majesty', 145), ('two', 140), ('would', 136), ('man', 134), ('planchet',
126), ('ha', 126), ('know', 122), ('good', 119), ('yes', 116), ('say',
112), ('without', 111), ('hand', 110), ('louis', 110), ('shall', 107),
('time', 107), ('day', 104)]
```

most frequent words in cluster : 1

```
  [('wa', 430), ('footnote', 300), ('one', 252), ('vol', 169), ('great',
161), ('time', 136), ('name', 136), ('ha', 133), ('king', 123), ('first',
121), ('year', 120), ('race', 109), ('chapter', 103), ('many', 100),
('even', 100), ('city', 97), ('may', 92), ('land', 91), ('also', 89),
('could', 89), ('must', 88), ('work', 87), ('found', 86), ('ancient', 83),
('called', 83)]
```

most frequent words in cluster : 2

```
  [('specie', 466), ('plant', 256), ('animal', 240), ('ha', 206),
('variation', 203), ('one', 189), ('wa', 155), ('case', 143), ('form',
137), ('variety', 131), ('many', 130), ('may', 129), ('bird', 125),
('selection', 122), ('darwin', 109), ('change', 106), ('nature', 105),
('great', 105), ('would', 102), ('number', 99), ('two', 98), ('condition',
96), ('fact', 95), ('distinct', 90), ('natural', 89)]
```

most frequent words in cluster : 3

```
  [('wa', 723), ('star', 252), ('one', 181), ('sun', 176), ('first', 152),
('variation', 146), ('year', 142), ('comet', 138), ('body', 128), ('time',
124), ('two', 120), ('ha', 109), ('great', 105), ('may', 105),
('observation', 102), ('part', 100), ('herschel', 99), ('system', 98),
('light', 96), ('solar', 94), ('astronomy', 92), ('thus', 87), ('even',
83), ('however', 83), ('upon', 82)]
```

most frequent words in cluster : 4

```
  [('wa', 732), ('lord', 225), ('house', 194), ('year', 189), ('judge',
184), ('time', 175), ('made', 138), ('sir', 137), ('one', 136), ('would',
134), ('fee', 128), ('court', 124), ('lady', 123), ('inn', 119), ('law',
```

115), ('lawyer', 113), ('chancellor', 105), ('upon', 104), ('day', 101), ('justice', 95), ('chief', 91), ('wife', 89), ('present', 89), ('great', 87), ('street', 83)]

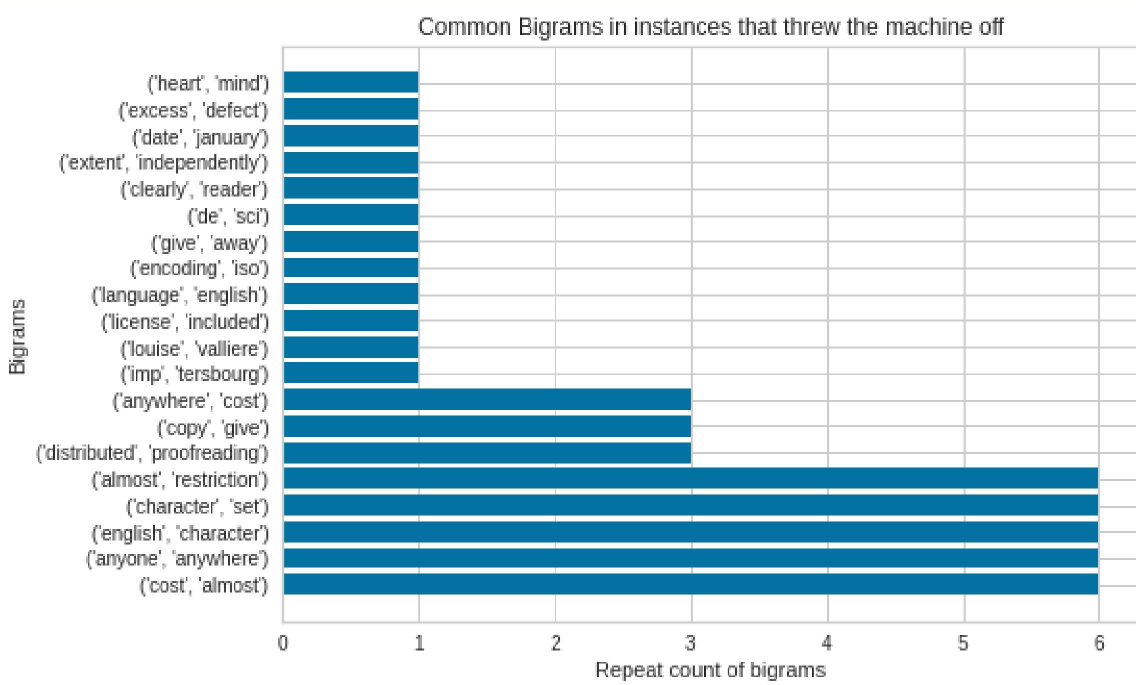
Most common words with their repeat count in all records that were labelled uncorrectly according to the human label:

[('one', 630), ('wa', 496), ('ha', 378), ('part', 378), ('may', 325), ('specie', 325), ('number', 325), ('variation', 300), ('case', 276), ('two', 253), ('see', 210), ('amount', 210), ('large', 190), ('fact', 171), ('even', 171), ('vol', 153), ('specimen', 153), ('size', 153), ('first', 153), ('also', 136), ('three', 136), ('great', 136), ('almost', 136), ('character', 136), ('among', 120)]

Most common collocations with their repeat count in all records that were labelled uncorrectly according to the human label:

[(('cost', 'almost'), 6), (('anyone', 'anywhere'), 6), (('english', 'character'), 6), (('character', 'set'), 6), (('almost', 'restriction'), 6), (('distributed', 'proofreading'), 3), (('copy', 'give'), 3), (('anywhere', 'cost'), 3), (('imp', 'tersbourg'), 1), (('louise', 'valliere'), 1), (('license', 'included'), 1), (('language', 'english'), 1), (('encoding', 'iso'), 1), (('give', 'away'), 1), (('de', 'sci'), 1), (('clearly', 'reader'), 1), (('extent', 'independently'), 1), (('date', 'january'), 1), (('excess', 'defect'), 1), (('heart', 'mind'), 1)]

The common Words that causes the machine threw-off:



Conclusion:

In this assignment, we learned how to apply different clustering algorithms with different transformation techniques. Then we compared between them using some evaluation models, like Kappa, Silhouette, consistency, and coherence with LDA topic modeling, and how to choose the best champion model, and apply error analysis to get the most frequent words that causes the machine off.

References:

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html?highlight=em

https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

https://docs.rapidminer.com/latest/studio/operators/modeling/segmentation/expectation_maximization_clustering.html#:~:text=The%20goal%20of%20EM%20clustering,the%20observed%20data%20(distribution).