IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Mining the Hierarchical Dimensions of Twitter Data for OLAP

[1]Chitlaa Harshithha, [2]N.Naveen Kumar
[1]Student, [2]Associate professor
Software Engineering,
School of Information Technology, Hyderabad, India.

Abstract: Social media platforms like, Twitter, Facebook disclose lot of information regarding the tastes of the people. In order to help in promotion of products and investigation of sentiments, there are many researches that focus on the content analysis. Also, OLAP (online analytical processing) has been demonstrated to be exceptionally powerful to examine multidimensional structured data. Text OLAP, the sole target behind applying OLAP to the text messages is to mine and develop the dimension which is hierarchical depending upon the data content which is unstructured. Usually text OLAP handles plain texts, but in the contrary the social platforms' information/content incorporates an abundance of information regarding social relationships which can be utilized to dig out an effective dimensional hierarchy. So here we come up with a topic model which can be called THLDA, Twitter Hierarchical Latent Dirichlet allocation. The target of THLDA model is to naturally mine the hierarchical dimension of tweets' topics, which can be additionally utilized for text OLAP on the tweets. Moreover, THLDA utilizes a procedure word2vec to break down the semantic connections of words in tweets to get an increasingly viable measurement. We lead broad examinations on tremendous amounts of Twitter information and assess the viability of THLDA. The results of the experiment show that the method beats other current subject models in mining and building the various leveled measurement of tweeters' points.

***Index Terms* - Topic Modeling, Text Modeling, Twitter data,Latent Dirchlet Allocation.**

I. Introduction

During the previous scarcely any years, Twitter has gotten progressively famous as a developing social stage for informing and correspondence among people. The enormous amounts of Twitter information aggregated so far is making it conceivable in order to find conveyance, float of mass taste and suppositions, which incredibly aids target marketing, product recommendation, etc. Then again, Online Analytical processing, empowers investigation to intelligently see information from various perspectives in layered granularities, which has just been demonstrated particularly helpful for the business insight. Lamentably, OLAP strategies are effective in managing shape information which are organized and formalized, however face troubles in handling literary substance, for example, Twitter information. To effectively apply the OLAP methods to Twitter, it is essential to mine the concealed delegate measurements from its broad substance.

Latent Dirichlet Allocation(LDA) model, which is typical unsupervised model, is proficient at factually breaking down literary information for the hidden themes. We have put forward MS-LDA, which is an LDA-based model to recognize the concealed layered interests from the Twitter information. MS-LDA, as the expansion of LDA, incorporated tweets and social connections between people who use twitter. All things considered, the crude LDA model can just mine single layer/monolayer themes, as opposed to the various leveled ones which OLAP requires. Then again being unsupervised hierarchical topic model,hLDA can acquire the sibling-sibling relationships connections among themes and also can arrange the points into a progressive tree consequently. Truth be told, Twitter information contain plentiful social conduct data about tweeters, for example, following, mentioning, retweeting. Moreover, some semantic connections also exist among the words in the tweets, which can influence the viability of the demonstrating procedure. At the end of the day, to viably find the concealed layers of subjects from the Twitter information in order to develop the progressive measurement for OLAP, we have to put forward another topic model, thatwhich can use the qualities of Twitter in the demonstrating procedure.

Existing System

During the previous scarcely any years, Twitter has gotten progressively famous as a developing social stage for informing and correspondence among people. The enormous amounts of Twitter information aggregated so far is making it conceivable to find the conveyance and float of mass tastes and suppositions, which incredibly aids target marketing, product recommendation etc. Then again, online Analytical processing or OLAP, empowers investigation to intelligently see information from all perspectives in layered granularities, which has just been demonstrated particularly helpful for business insight. Lamentably, OLAP strategies are effective in managing shape information which are organized and formalized, however face troubles in handling literary substance, for example, Twitter information. To effectively apply OLAP methods to Twitter, it is basic to mine the concealed delegate measurements from its broad substance.

Disadvantages

- Facing difficulties in processing data/textual content.
- It is basic to mine the concealed agent measurements from its broad substance

PROPOSED SYSTEM

We thought of a model, which is based on LDA, MS-LDA ,called Multilayered Semantic LDA, to discover the concealed layered interests from the information got from Twitter, a social networking platform. As the expansion of LDA, MS-LDA incorporated the social connections among people who tweet and tweets. By and by, the crude LDA model can just mine single layer subjects, instead of the hierarchical ones which OLAP actually wants. Then again, as an unaided various leveled topic model, hLDA can get the kin connections among points and can sort out the subjects into a hierarchical tree consequently. Truth be told, Twitter information contain bountiful social conduct data about tweeters, for example, retweeting, mentioning and following. Some semantic connections among the words in tweets also exist, which may influence the viability of the displaying procedure. As such, to adequately find the concealed layers of subjects from Twitter information for building the progressive measurement for OLAP, we need to put forward another topic model which use the attributes of Twitter in its demonstrating procedure.

Advantages

- Detection of the layered interests which are hidden, from the data obtained i.e., Twitter data.
- HLDA can attain the sibling-sibling relationships between topics and can put forth the topics into a hierarchy based tree consequently.

I. SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- Hard Disk : 20GB
- RAM : 256 MB
- Processor : Core – i5

SOFTWARE REQUIREMENTS

- Operating System : Windows 8
- Coding Language : Java
- IDE : Eclipse

II. RELATED WORK

Mining Hidden Interests from Twitter Based on Word Similarity and Social Relationship for OLAP [1]

Online Analytical Processing, is a way to deal with answering multidimensional analytical questions in a natural way. Be that as it may, the conventional OLAP approaches can just do arrangement with organized information, however not with unstructured textual information like tweets. To work on this issue, we put forward LDA (Latent Dirichlet Allocation)- based model, which can be identified as, MS-LDA , Multilayered Semantic LDA ,which can distinguish the concealed layered premiums from information obtained from Twitter dependent on Latent Dirichlet Allocation. The layered component of interests can also be used additionally to apply Online Analytical Processing ,OLAP procedures to information obtained from Twitter. Besides, Multilayered Semantic LDA utilizes the semantic comparability between tweets' expressions dependent on word2vec, and furthermore the social relationship among people who tweet ,so that adequacy can be improved. The broad examinations exhibit that Multilayered Semantic LDA can viably remove the measurement pecking order of tweeters' inclinations towards Online Analytical Processing.
Now, we put forward an improvised subject model, for example MS-LDA, Multilayered Semantic LDA ,which can be utilized to extricate the measurement progressive systems of tweeters' inclinations, regularly covered up in the huge measure of unstructured Twitter information. We directed broad investigations on Twitter information to assess the viability of MS-LDA. The outcomes show that Multilayered Semantic LDA in fact has the great acknowledgment impact. The word2vec model utilized in this project is prepared utilizing news given by Google. In any case, the introduction of news when all is said in done is to some degree thorough, while tweets are increasingly casual.

A Method for Online Analytical Processing of Text Data [2]

There are progressively noticeable requests for organized/unstructured data reconciliation and progressed investigation. Be that as it may, ordinary database innovation has not had the option to introduce a vigorous and down to earth execution of a really coordinated engineering for such purposes. Subsequent to taking a shot at a few mechanical applications (specifically, in the life sciences and medicinal services region), we have recognized essential problems and specialized ways to deal with the issues.Here, we come up with information portrayals and logarithmic activities for coordinating semantic data (Ex: ontologies) into OLAP frameworks, which will permit to break down a tremendous arrangement of literary archives with their fundamental semantic data. The presentation of the model execution has been assessed utilizing genuine world datasets, and the high adaptability and adaptability of our methodology have been affirmed regarding the calculation time.

In this paper, we put forward an information portrayal and its polynomial math activities to coordinate ontologies with OLAP frameworks to break down an immense arrangement of literary reports. By using our strategy, two kinds of data (structured and unstructured data) can commonly improve data disclosure and investigation extent. Utilizing preorder and post order in a hierarchy, the proposed strategy was executed with a tireless store. The proficiency of our methodology has been affirmed as for the calculation time. Our strategy is so effective and strong that it empowers an expert to intelligently.

III. PRODUCT DESIGN

A. UML DIAGRAMS:

Class Diagram:

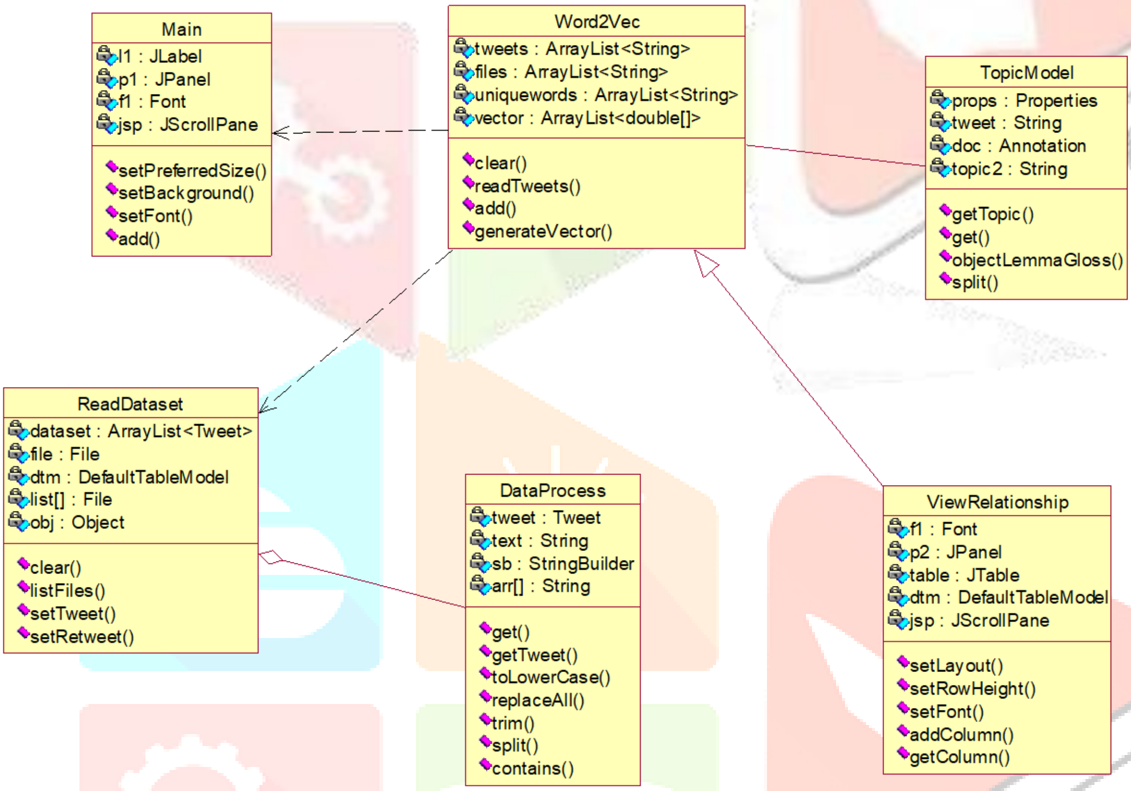


Fig -1: Class diagram

Use Case Diagram:

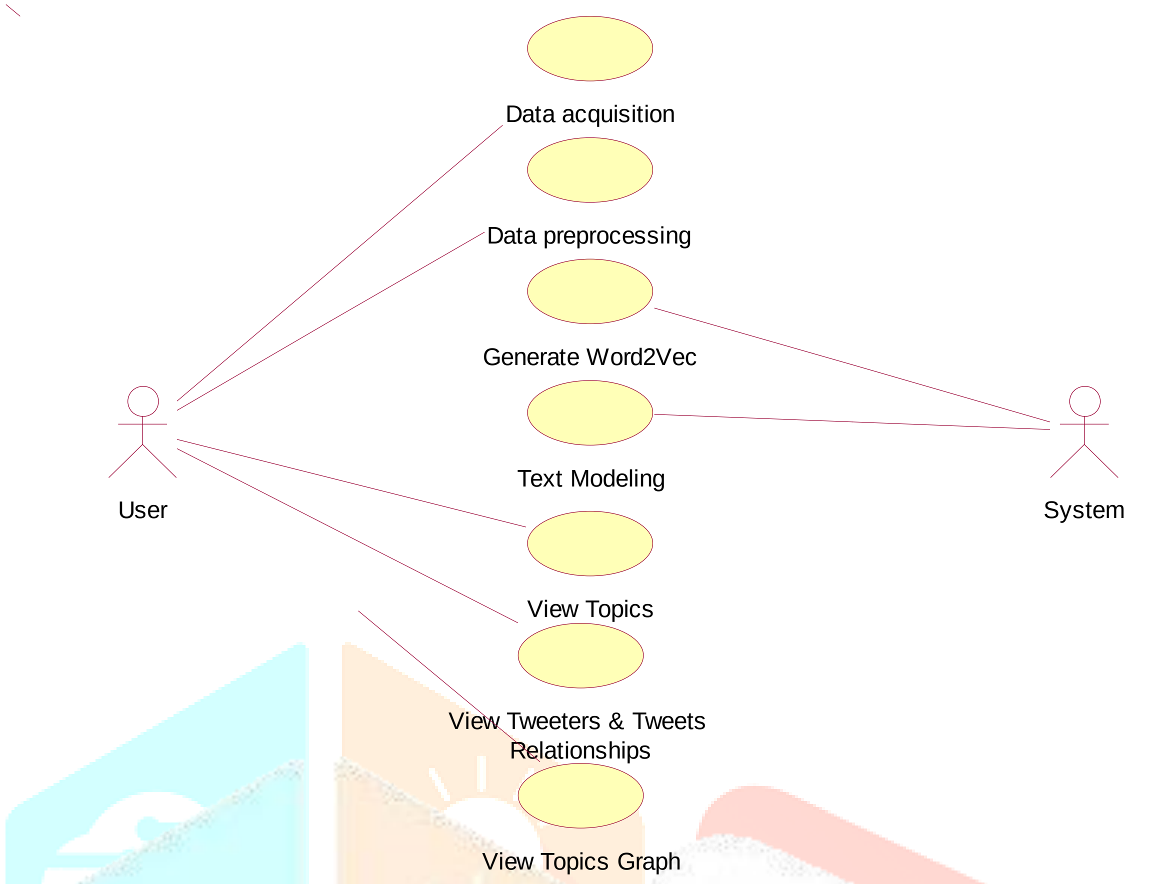


Fig -2: Use Case diagram

B. SYSTEM ARCHITECTURE

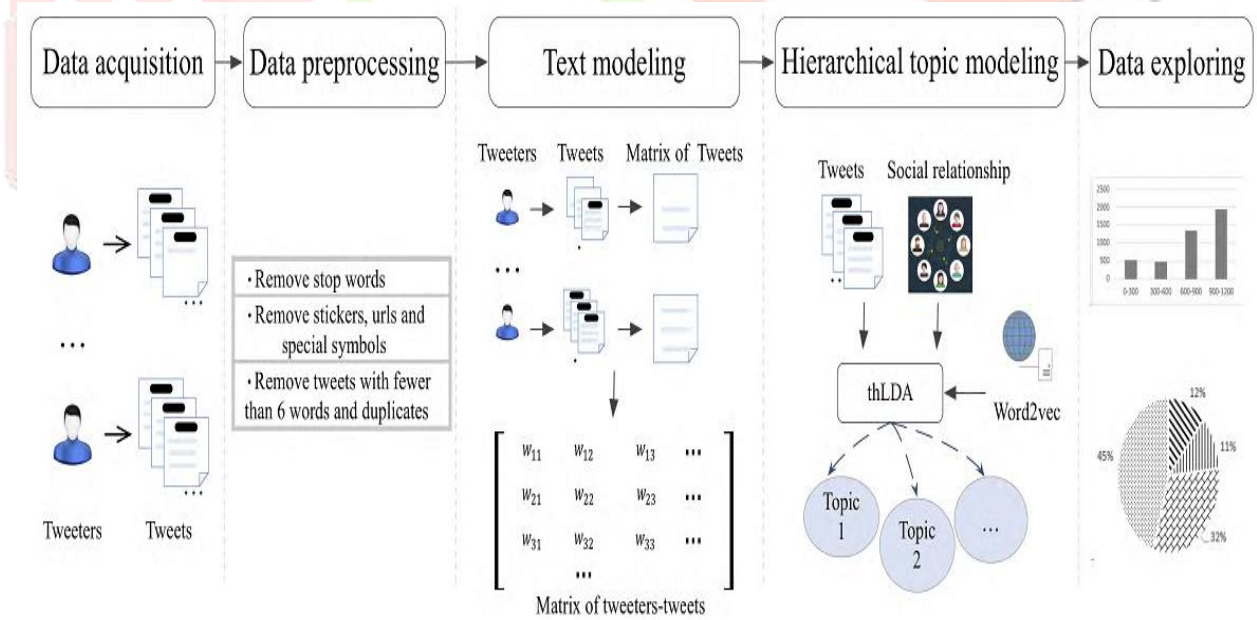


Fig -9: System Architecture

IV. IMPLEMENTATION

The general procedure of investigating Twitter information dependent on the OLAP method can be portrayed as follows:

• **Data acquisition:** Using the REST APIs which are provided by the twitter, tweets, social relationship and tweeters' profiles, can be obtained.

• **Data preprocessing:** Short words i.e., the most common words like the, is, at etc., should be removed along with the web links. Then parts of speech analysis should be done to remain with verbs and nouns in the unstructured tweets.

• **Text modeling:** The Relationship between tweeters' and tweets can be recognized based on the text modeling.

• **Hierarchical topic modeling:** Interests(or topics) from the data obtained from Twitter can be extracted, and the hierarchical topic dimension can be constructed based on the probability distribution of various topics and subtopics.

• **Data exploring:** Using OLAP, analysis of tweeters from multiple dimensions can be done.

Methodologies

In this paper the concept to extract topics from tweets and can extract relationships between tweeters and tweets. Relationship can be extracted by analyzing two person's tweets and look for semantic similarity between their tweets and if both are talking on same matter then similarity will be higher and both tweets users will have similarity and relationship can be bond between them.

In this paper to extract relationship and topic author is asking to generate WORD2VEC (word to vector) and also called as BAG of WORDS (BOG). After forming vector we can easily extract relationship between two vectors and can also extract topics.

WORD2VEC conversion means converting tweets into vector array and each array will be consider as one tweet and in each tweet each occurrence or count of each word will put inside that array. If two tweets have common words then that array column will have value > 0 and similarity will be found.

IV. TESTING AND RESULTS

The procedure or strategy for discovering errors or defects in an application or software program with the goal that the application functions as indicated by the end client's prerequisite is called testing. The test case is defined as a set of conditions or factors under which a tester will decide if a system or application under test satisfies requirements and works properly.

Following are the testing strategies followed during the test period:

TEST CASES:

Table: Test Cases and Results

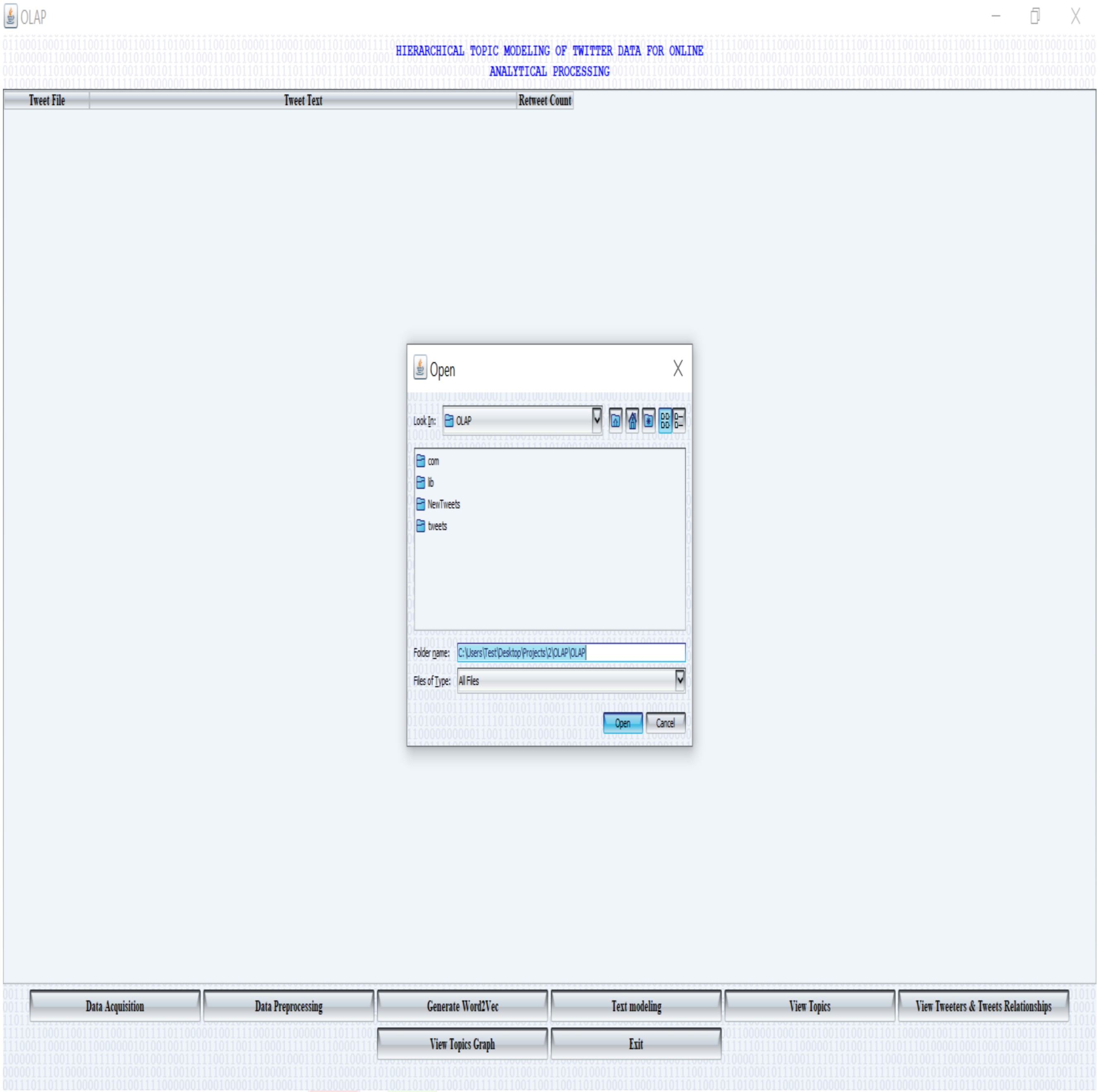
Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Data Acquisition	Verify dataset is available or not	If it is available	We can load the dataset	Tweets dataset loaded	High	High
02	Data preprocessing	Verify dataset loaded or not	If it's loaded	We can clean tweets	All special symbols and URL's remove from tweet	low	High

03	Generate Word2Vec	Verify the tweets are cleaned or not	If it is cleaned	We can generate vector	We get the average occurrence of words in tweets	Medium	High
04	Text modeling	Verify vector generated or not	If it's generated	We can apply the OLAP & LDA	We will get processing of each tweet to extract topic	High	High
05	View topics	Verify topics extracted or not	If it's extracted	We can view the topics	We can view the all topics from tweets	High	High
06	View tweeters & tweets relationship s	Verify topics are viewed or not	If it's viewed	We can view the relationship s	We get the cosine similarity between tweeters & tweets	High	High
07	View topics graph	Verify all topics viewed or not	If it's viewed	We can get the graph	We get the topics graph	High	High

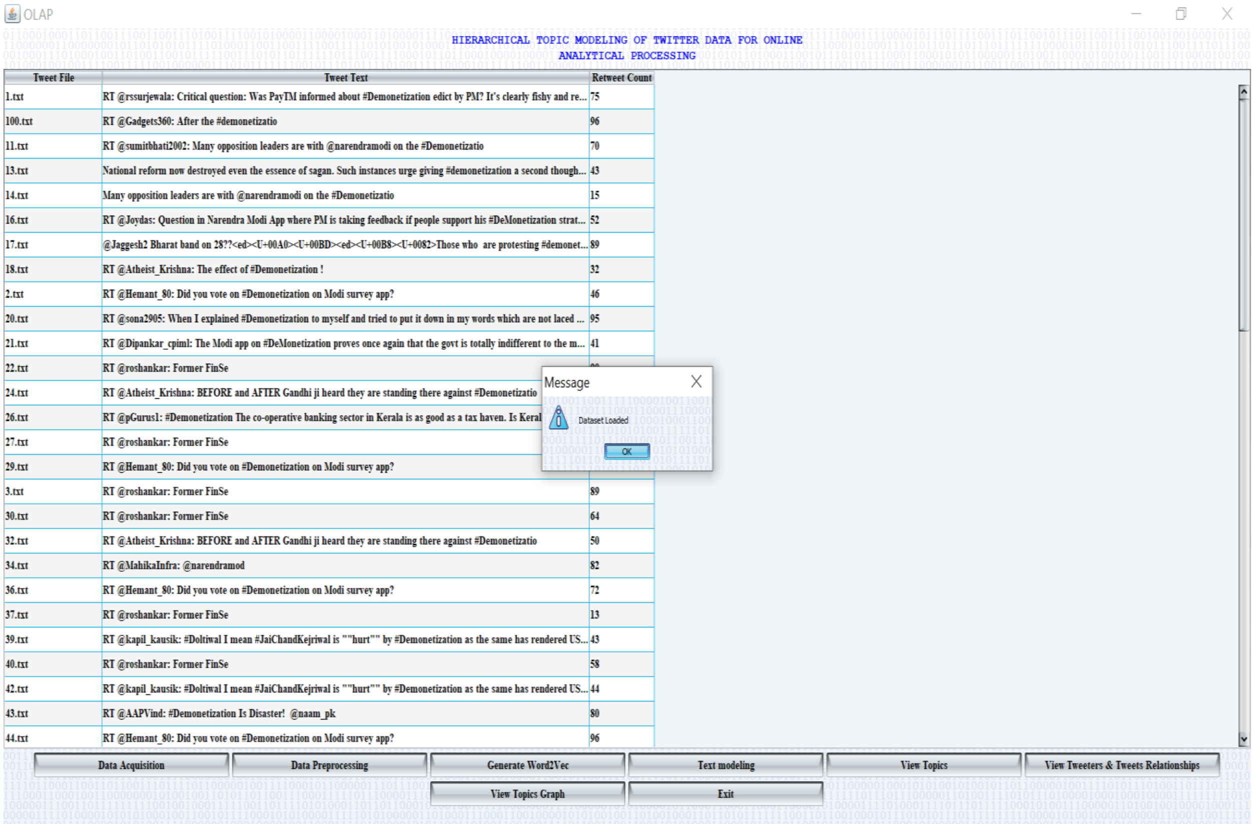
RESULTS:

The dataset which I have used is downloaded from Kaggle website. Then I have made each tweet into a separate file.

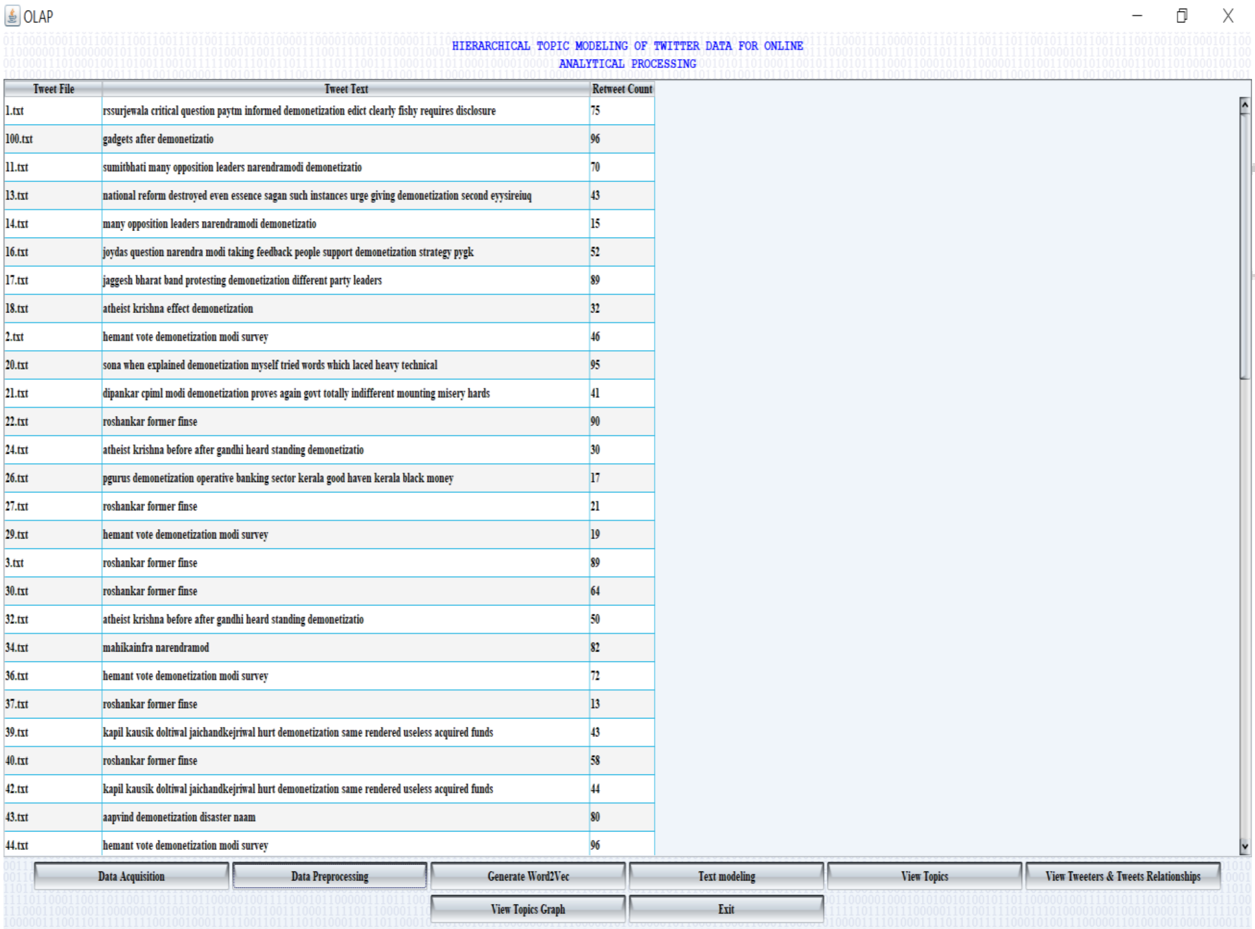
On running the batch file, we will get this screen. Here we can see the buttons that are created and in the empty field the headers are properly arranged. On clicking the buttons the functional changes can be seen in th e empty area of the screen.



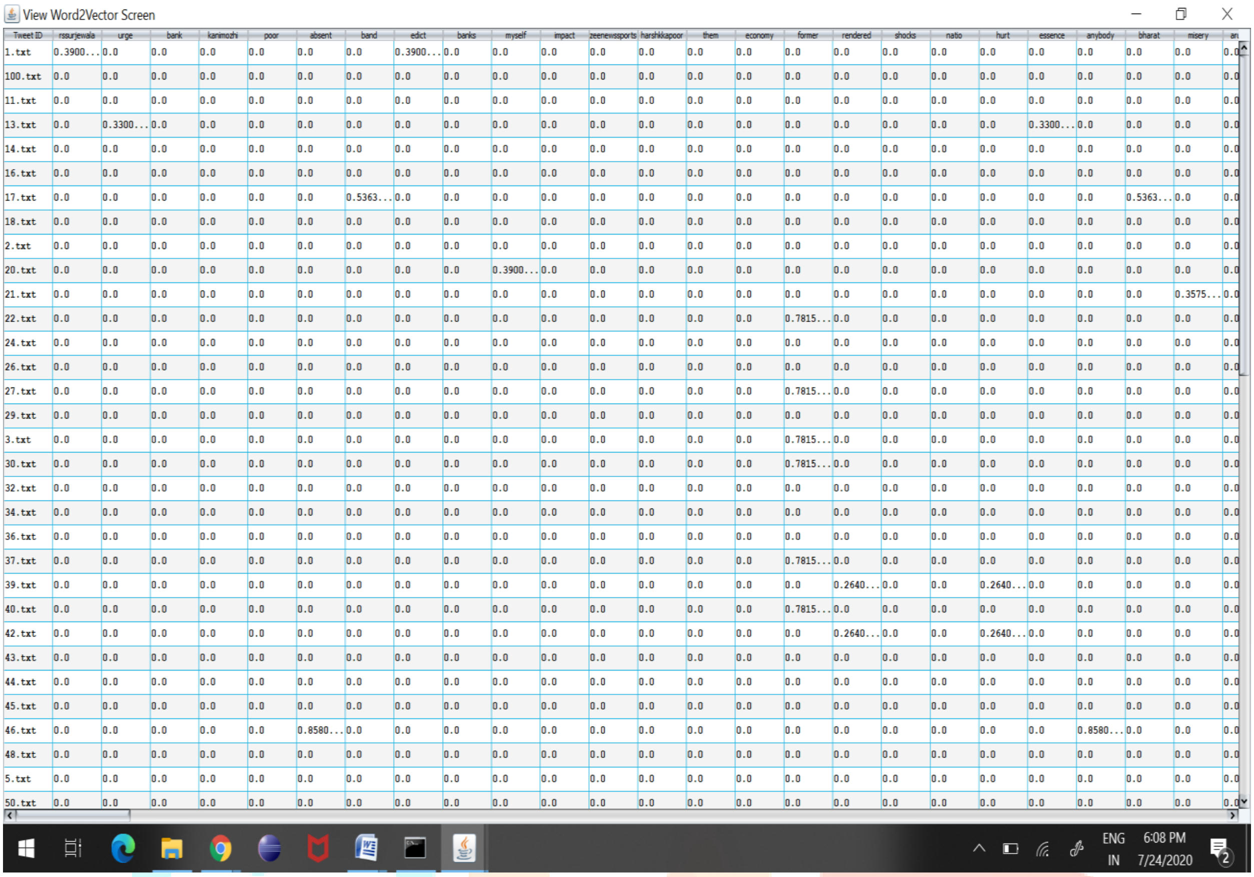
On clicking Data acquisition button on the screen we can selects the folder in which the tweet individual documents are stored as input to the application. This is uploading the data obtained from Twitter. After choosing the required folder we can click on open. Then the changes can be seen on the screen.



After the data gets loaded the alert is displayed.

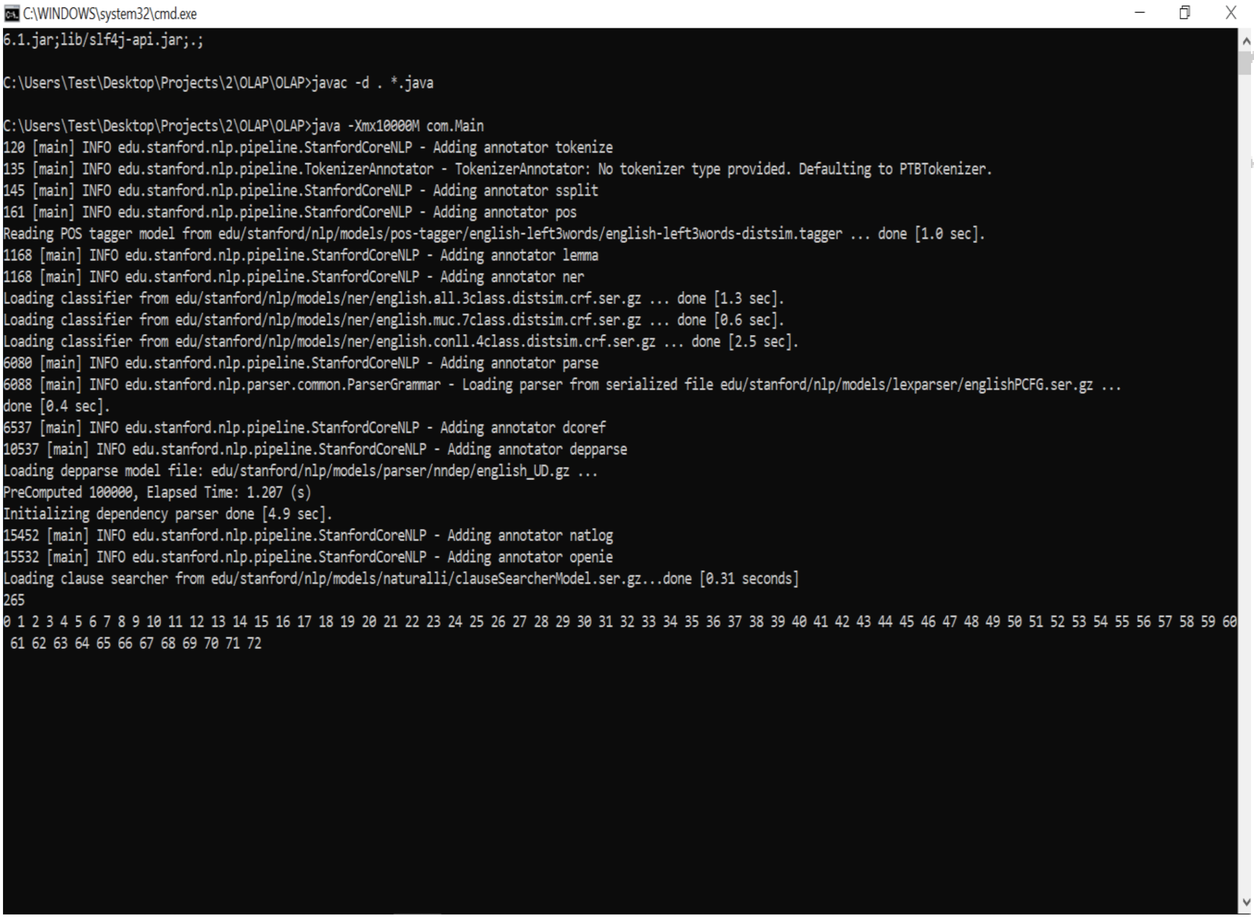


Data after pre processing.



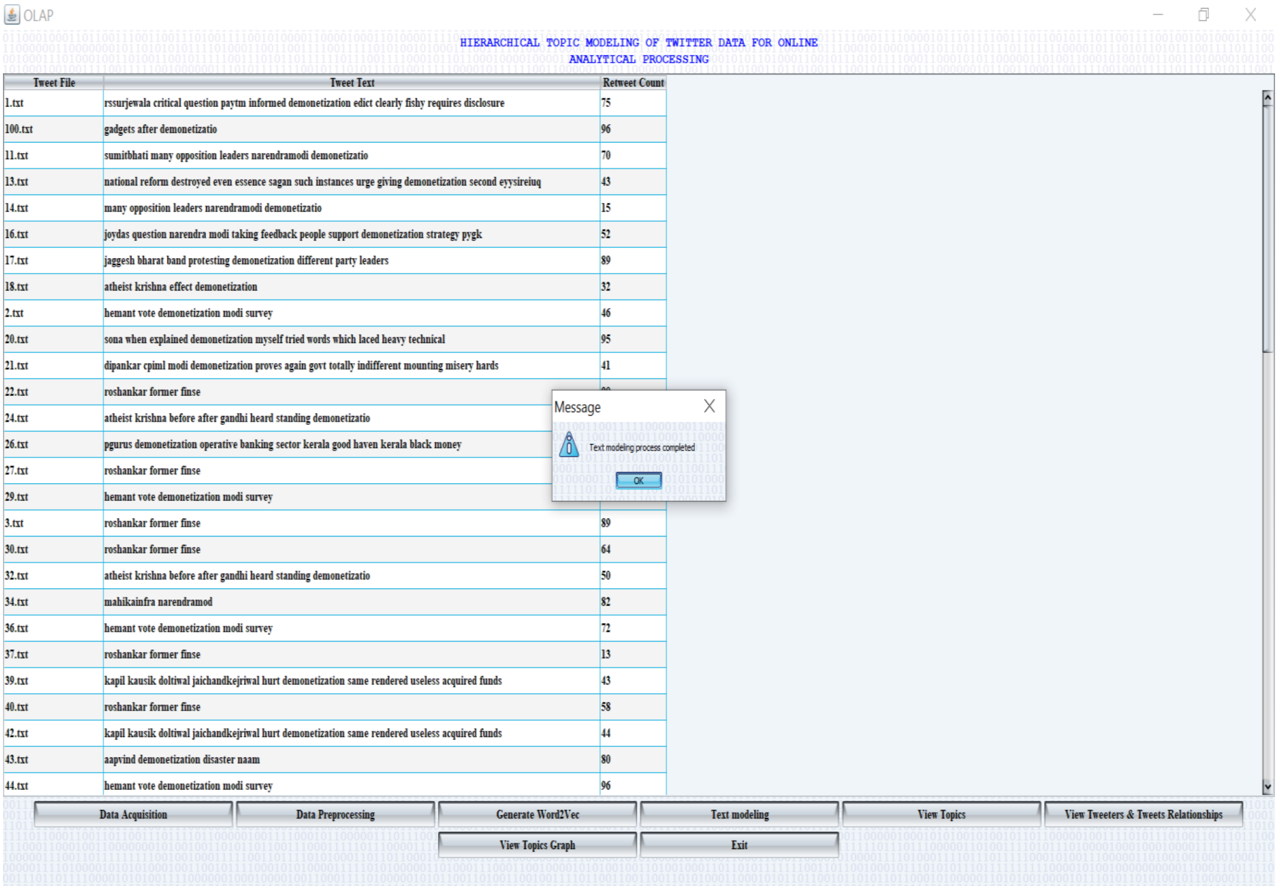
Tweet ID	rssurjewala	urge	bank	kanimozhi	poor	absent	band	edict	banks	myself	impact	zeenewssports	harshkkapoor	them	economy	former	rendered	shocks	natio	hurt	essence	anybody	bharat	misery	an
1.txt	0.3900...	0.0	0.0	0.0	0.0	0.0	0.0	0.3900...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100.txt	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	0.0	0.0	0.0	0.0	0.0
11.txt	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	0.0	0.0	0.0	0.0	0.0
13.txt	0.0	0.3300...	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	0.3300...	0.0	0.0	0.0	0.0
14.txt	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	0.0	0.0	0.0	0.0	0.0
16.txt	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	0.0	0.0	0.0	0.0	0.0
17.txt	0.0	0.0	0.0	0.0	0.0	0.0	0.5363...	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	0.5363...	0.0	0.0
18.txt	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	0.0	0.0	0.0	0.0	0.0
2.txt	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	0.0	0.0	0.0	0.0	0.0
20.txt	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3900...	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
21.txt	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	0.0	0.0	0.0	0.3575...	0.0
22.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24.txt	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	0.0	0.0	0.0	0.0	0.0
26.txt	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	0.0	0.0	0.0	0.0	0.0
27.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
29.txt	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	0.0	0.0	0.0	0.0	0.0
3.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
32.txt	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	0.0	0.0	0.0	0.0	0.0
34.txt	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	0.0	0.0	0.0	0.0	0.0
36.txt	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	0.0	0.0	0.0	0.0	0.0
37.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
39.txt	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	0.2640...	0.0	0.0	0.2640...	0.0	0.0	0.0	0.0	0.0
40.txt	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	0.7815...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42.txt	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	0.2640...	0.0	0.0	0.2640...	0.0	0.0	0.0	0.0	0.0
43.txt	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	0.0	0.0	0.0	0.0	0.0
44.txt	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	0.0	0.0	0.0	0.0	0.0
45.txt	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	0.0	0.0	0.0	0.0	0.0
46.txt	0.0	0.0	0.0	0.0	0.0	0.8580...	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	0.8580...	0.0	0.0	0.0
48.txt	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	0.0	0.0	0.0	0.0	0.0
5.txt	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	0.0	0.0	0.0	0.0	0.0
50.txt	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	0.0	0.0	0.0	0.0	0.0

Word2Vector is generated.

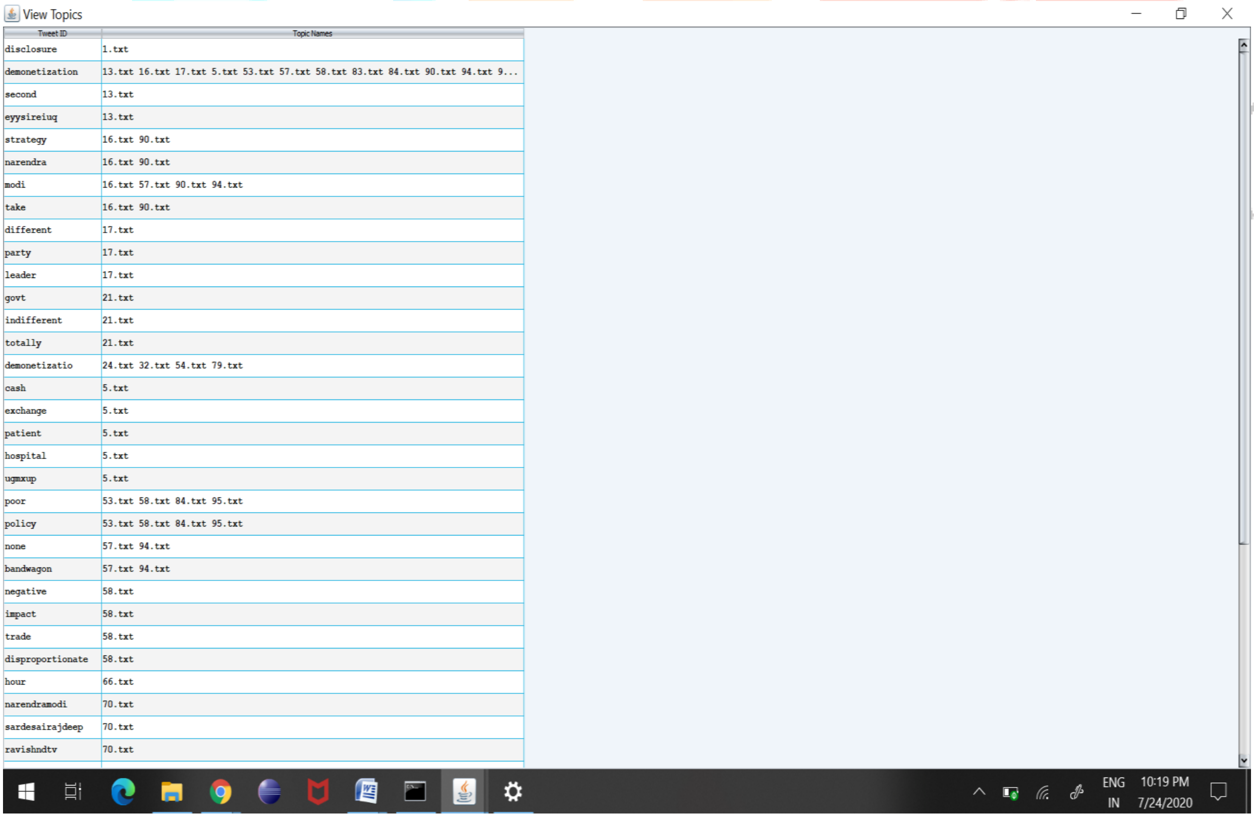


```
C:\WINDOWS\system32\cmd.exe
6.1.jar;lib/slf4j-api.jar;.;

C:\Users\Test\Desktop\Projects\2\OLAP\OLAP>javac -d . *.java

C:\Users\Test\Desktop\Projects\2\OLAP\OLAP>java -Xmx10000M com.Main
120 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator tokenize
135 [main] INFO edu.stanford.nlp.pipeline.TokenizerAnnotator - TokenizerAnnotator: No tokenizer type provided. Defaulting to PTBTokenizer.
145 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator ssplit
161 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator pos
Reading POS tagger model from edu/stanford/nlp/models/pos-tagger/english-left3words/english-left3words-distsim.tagger ... done [1.0 sec].
1168 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator lemma
1168 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator ner
Loading classifier from edu/stanford/nlp/models/ner/english.all.3class.distsim.crf.ser.gz ... done [1.3 sec].
Loading classifier from edu/stanford/nlp/models/ner/english.muc.7class.distsim.crf.ser.gz ... done [0.6 sec].
Loading classifier from edu/stanford/nlp/models/ner/english.conll.4class.distsim.crf.ser.gz ... done [2.5 sec].
6080 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator parse
6088 [main] INFO edu.stanford.nlp.parser.common.ParserGrammar - Loading parser from serialized file edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz ...
done [0.4 sec].
6537 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator dcoref
10537 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator depparse
Loading depparse model file: edu/stanford/nlp/models/parser/nndep/english_UD.gz ...
PreComputed 100000, Elapsed Time: 1.207 (s)
Initializing dependency parser done [4.9 sec].
15452 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator natlog
15532 [main] INFO edu.stanford.nlp.pipeline.StanfordCoreNLP - Adding annotator openie
Loading clause searcher from edu/stanford/nlp/models/naturalli/clauseSearcherModel.ser.gz...done [0.31 seconds]
265
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
 61 62 63 64 65 66 67 68 69 70 71 72
```

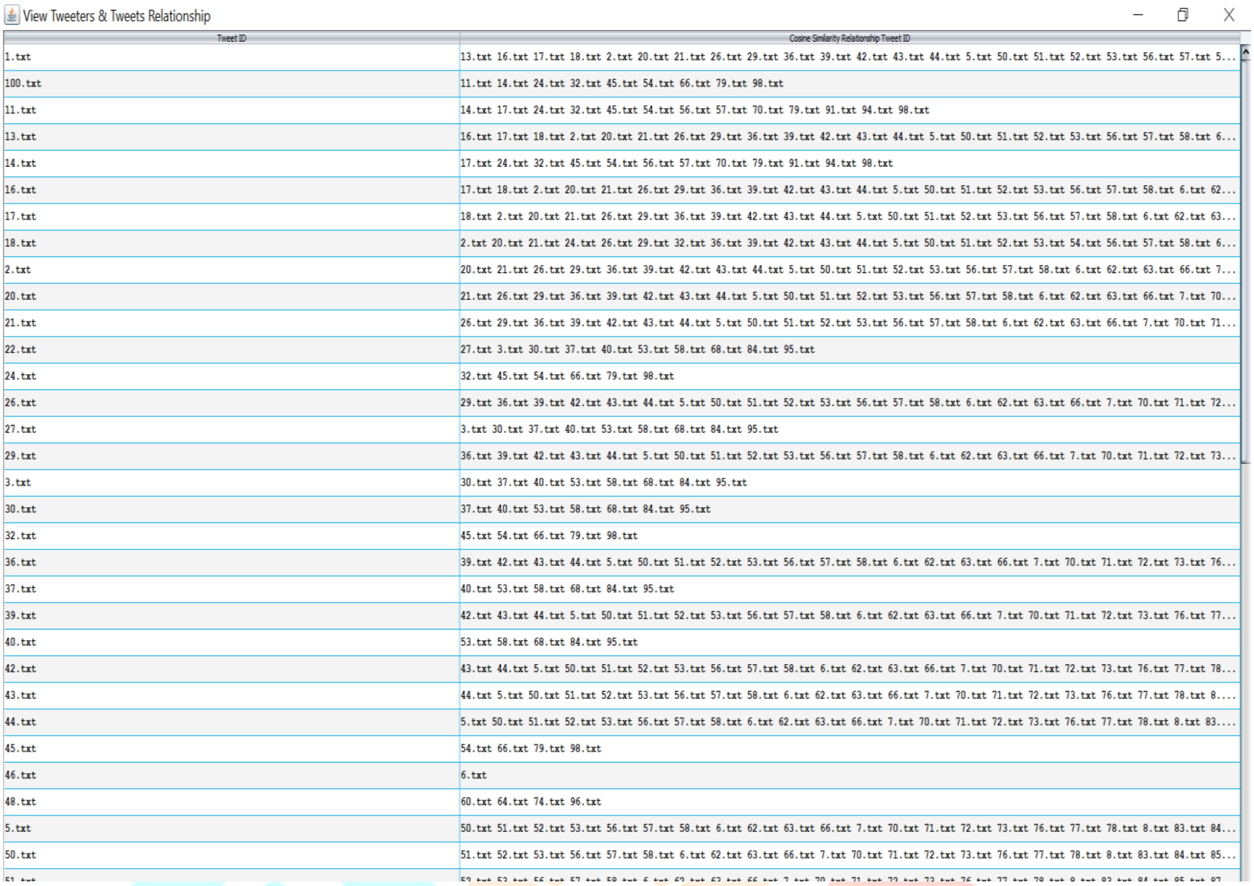
Here we can see that each tweet is getting processed on clicking Text Modeling button.



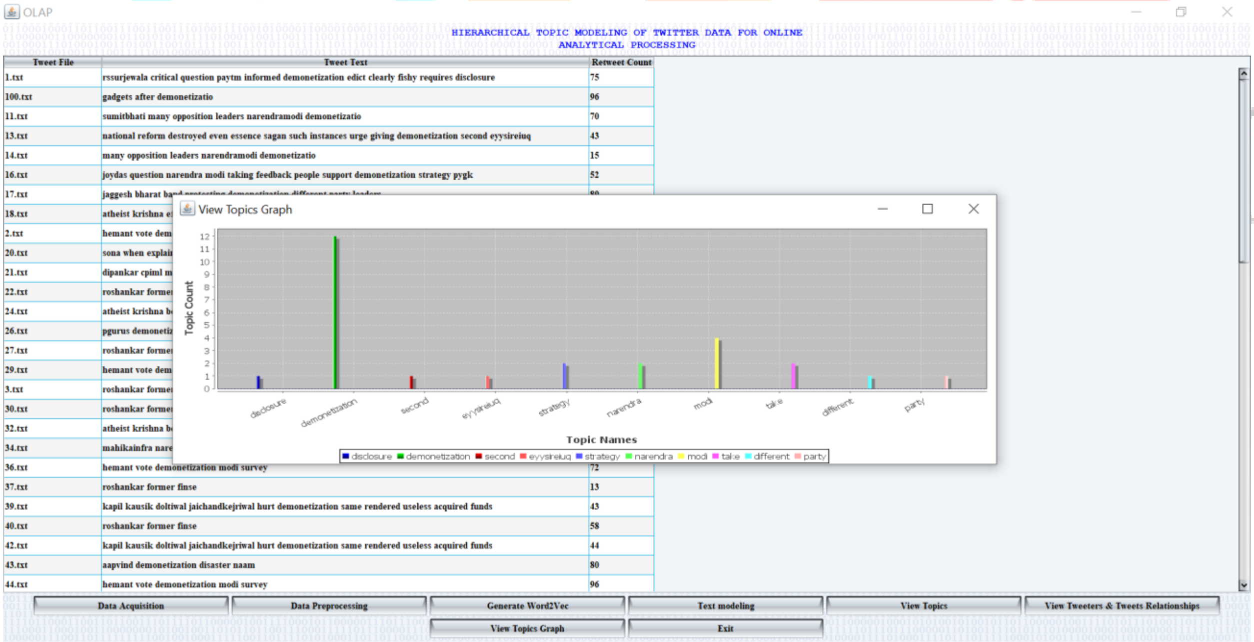
Topic Modeling is done and Topics obtained can be seen on clicking View Topics.



List of topics obtained from given dataset.



Tweeters and Tweets Relationship can also be obtained.



In above graph x-axis showing topic name and y-axis showing number of times that topic appear in all tweets.

VI CONCLUSION

Here, we have proposed Twitter Hierarchical Latent Dirchlet Allocation method i.e.,thLDA which is a novel hierarchical topic model. This method is applied to the large quantity of unstructured Twitter data in order to mine the dimension hierarchy of tweets' topics. The effectiveness of this method is found out by performing extensive experiments on real data from Twitter. Results of the experiment confirm that this method is more effective than any other models.

VII FURUTE SCOPE

Indirect social relationships between the tweeters can be analyzed in order to enhance our current model, in the future.

REFERENCES

[1] D. Yu et al., ''Mining hidden interests from Twitter based on word similarity and social relationship for OLAP,'' Int. J. Softw. Eng. Knowl. Eng., vol. 27, nos. 9–10, pp. 1567–1578, 2017.

[2] A. Inokuchi and K. Takeda, ''A method for online analytical processing of text data,'' in Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage., 2007, pp. 455–464.