**NATURAL LANGUAGE PROCESSING**

**CS491**

**FINAL PROJECT REPORT**

**TOPIC SEGMENTATION WITH A STRUCTURED TOPIC MODEL**

**GROUP NUMBER- G5**

**TEAM MEMBERS DETAILS**

**1) KUMARI RENUKA (U101115FCS111)**

**2) RISHABH KUMAR KANDOI (U101115FCS283)**

**3) SOUMI PAL (U101115FCS158)**

TABLE OF CONTENTS

Objective 3

[Introduction 4](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000066)

Related Paper Review [6](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000073)

Algorithm [9](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000073)

Screenshot of the output [11](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000069)

[Individual Contribution 15](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000073)

[Conclusion and Future Work 16](file:///C:\Users\renuka_pg-2_2601\Desktop\Semester%206\Big%20Data%20Concepts\ProjectReport1.docx#_Toc256000073)

OBJECTIVE

The conversation platforms have now become prevalent for both personal and professional usage. For instance, in a large enterprise scenario, project managers can utilize these platforms for various tasks such as decision auditing and dynamic responsibility allocation. Logs of such conversations offer potentially valuable information for various other applications such as automatic assessment of possible collaborative work among people etc. It is thus vital to invent effective segmentation methods that can separate discussions into small granules of independent conversational snippets. Independent means that a segment should be self-contained and discussing the same topic. Topic modelling is one such method which can be utilized to separate discussions into small granules and provide a topic name to each segment. In the paper named “Topic Segmentation with a Structured Topic Model” presents a new hierarchical Bayesian unsupervised topic segmentation model, integrating a point-wise boundary sampling algorithm with a structured topic model. This new model takes advantage of the high modelling accuracy of structured topic models to produce a topic segmentation based on the distribution of latent topics. This model provides high quality segmentation performance on Choi’s dataset, as well as two sets of meeting transcripts and written texts.

After reading the paper we have implemented the topic segmentation in python. We in our project have used a slack dataset which is further divided into training and test dataset. Initially we have tokenized each message by removing the stop words and punctuations signs. Then each word has been reduced to its root form. After this each word have is converted into n-dimensional vector so that algebraic operations can be performed which carry semantic meaning. Several kinds of constraints are kept into mind such as if there are groups of 1 or 2 messages that are surrounded by messages belonging to the same topic, the bigger topic absorbs the tiny one. This happens when one or two messages are short and so grammatically incoherent that it's not possible to related them to any existing topic. Secondly, to identify reply objects. There a considerable number of messages that are simply reply messages which don't contribute to the topic but are part of the same topic. Semantically, these messages by themselves carry no meaning unless they belong to a bigger topic. We call these message reply objects, and we treat them as a particular case. Thirdly, we have introduced a dynamic window size feature to make the machine remember the last topic for better topic segmentation. Fourthly, we introduced a concept of Similarity Distance. Since each conversation is already converted to vector, so we can apply POS tagging to identify semantic meaning for each conversation and provide a score to be compared to other conversation. This helps to segment the texts more efficiently. The described algorithm has been tested upon various training and test dataset in order to show the efficiency of the proposed algorithm.

Also, we have implemented some necessary change which will help to increase the accuracy of the topic assignment. However, further improvement is necessary related to topic assignments and creating topic detection boundary.

INTRODUCTION

The discourse Topic Modelling provides a convenient way to analyse big unclassified text. A topic contains a cluster of words that frequently occurs together. A topic modelling can connect words with similar meanings and distinguish between uses of words with multiple meanings. In our project we have used the topic modelling and incorporated with text segmentation. The main importance of topic modelling is to discover patterns of word-use and how to connect documents that share similar patterns. So, the idea of topic models is that term which can be working with documents and these documents are mixtures of topics, where a topic is a probability distribution over words. In other word, topic model is a generative model for documents. It specifies a simple probabilistic procedure by which documents can be generated.

Latent Semantic Analysis (LSA) is a method or a technique in the area of Natural Language Processing (NLP). The main goal of Latent Semantic Analysis (LSA) is to create vector-based representation for texts to make semantic content. By vector representation (LSA) computes the similarity between texts to pick the heist efficient related words. In the past LSA was named as Latent Semantic Indexing (LSI) but improved for information retrieval tasking. So, finding few documents that are close to the query has been selected from many documents. LSA should have many aspects to give approach such as key words matching, Wight keywords matching and vector representation depends on occurrences of words in documents. Also, Latent Semantic Analysis (LSA) uses Singular Value Decomposition (SVD) to rearrange the data. SVD is a method that uses a matrix to reconfigure and calculate all the diminutions of vector space. In addition, the diminutions in vector space will be computed and organized from most to the least Important. In our project we have changed the sentences into the vector space and have found the similarity between two sentences on the basis of the Euclidean distance. Topic has been divided on the basis of the similarity distance score.

Clustering is a useful technique that organizes a large quantity of unordered text documents into a small number of meaningful and coherent clusters, thereby providing a basis for intuitive and informative navigation and browsing mechanisms. Partitional clustering algorithms have been recognized to be more suitable as opposed to the hierarchical clustering schemes for processing large datasets. A wide variety of distance functions and similarity measures  
have been used for clustering, such as squared Euclidean distance, cosine similarity, and relative entropy. As mentioned earlier we have used Euclidean distance.

Terms are basically words. But we applied several standard transformations on the basic term vector representation. First, we removed stop words. There are words that are non-descriptive for the topic of a document, such as a, and, are and do. Second, words were stemmed using so that words with different endings will be mapped into a single word. For example, production, produce, produces and product will be mapped to the stem produced. The underlying assumption is that different morphological variations of words with the same root/stem are thematically similar and should be treated as a single word. Third, we considered the effect of including infrequent terms in the document representation on the overall clustering performance and decided to discard words that appear with less than a given threshold frequency. The rationale by discarding infrequent terms is that in many cases they are not very descriptive about the document’s subject and make little contribution to the similarity between two documents. Meanwhile, including rare terms can also introduce noise into the clustering process and make similarity computation more expensive.

**FIG1: WORKING OF A TOPIC SEGMENTER**

RELATED PAPER REVIEWS

**PAPER REFERENCE 1**

*Hearst, M.A., 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics, 23(1), pp.33-64.*

Marti A. Hearst in his paper “TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics”* make use of a technique called TextTiling.TextTiling is a technique for subdividing texts into multi-paragraph units that represent passages, or subtopics. The discourse cues for identifying major subtopic shifts are patterns of lexical co-occurrence and distribution. The algorithm described in this paper has three parts: tokenization into terms and sentence-sized units, determination of a score for each sentence-sized unit, and detection of the subtopic boundaries, which are assumed to occur at the largest valleys in the graph that results from plotting sentence-units against scores. In this paper three methods for score assignment have been explored: blocks, vocabulary introductions, and chains. This article focuses only on the discovery of the segment boundaries, but there is extensive ongoing research on automated topic classification. The algorithm is fully implemented and is shown to produce segmentation that corresponds well to human judgments of the subtopic boundaries of 12 texts. Multi-paragraph subtopic segmentation is useful for many text analysis tasks, including information retrieval and summarization.

**PAPER REFERENCE 2**

*Alemi, A.A. and Ginsparg, P., 2015. Text segmentation based on semantic word embeddings. arXiv preprint arXiv:1503.05543.*

Alemi, A.A. and Ginsparg, P. in their paper “Text segmentation based on semantic word embeddings” explored the use of semantic word embeddings in text segmentation algorithms, including the C99 segmentation algorithm and new algorithms inspired by the distributed word vector representation. They have presented a general framework for describing and developing segmentation algorithms, and compared some existing and new strategies for representation, scoring and splitting. By developing a general framework for discussing a class of segmentation objectives, they study the effectiveness of greedy versus exact optimization approaches and suggest a new iterative refinement technique for improving the performance of greedy strategies. They have compared their results to known benchmarks, using known metrics. They have demonstrated the state-of-the-art performance for an untrained method with our Content Vector Segmentation (CVS) on the Choi test set. The Choi dataset is used to test whether a segmentation algorithm can distinguish natural topic boundaries. It concatenates the first n sentences from ten different documents. Finally, they have applied the segmentation procedure to an in the wild dataset consisting of text extracted from scholarly articles in the arXiv.org database. The arXiv is a repository of scientific articles which for practical reasons extracts text from PDF documents.

**PAPER REFERENCE 3**

*Beeferman, D., Berger, A. and Lafferty, J., 1999. Statistical models for text segmentation. Machine learning, 34(1-3), pp.177-210.*

Beeferman, D., Berger, A. and Lafferty in their paper “Statistical models for text segmentation. *Machine learning”* introduces a new statistical approach to automatically partitioning text into coherent segments. The segmentation problem is based on the statistical framework of feature selection for random fields and exponential models. The idea is to construct a model that assigns a probability to the end of every sentence—the probability that there exists a boundary between that sentence and the next. This probability distribution is chosen by incrementally building a log-linear model that weighs different “features” of the surrounding context. For simplicity, they assume that the features are binary questions. The approach is based on a technique that incrementally builds an exponential model to extract features that are correlated with the presence of boundaries in labelled training text. The models use two classes of features: topicality features that use adaptive language models in a novel way to detect broad changes of topic, and cue-word features that detect occurrences of specific words, which may be domain-specific, that tend to be used near segment boundaries. Assessment of their approach on quantitative and qualitative grounds demonstrates its effectiveness in two very different domains, Wall Street Journal news articles and television broadcast news story transcripts. Quantitative results on these domains are presented using a new probabilistically motivated error metric, which combines precision and recall in a natural and flexible way. This metric is used to make a quantitative assessment of the relative contributions of the different feature types, as well as a comparison with decision trees and previously proposed text segmentation algorithms.

**PAPER REFERENCE 4**

*Dias, G., Alves, E. and Lopes, J.G.P., 2007, July.* *Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In AAAI (Vol. 7, pp. 1334-1340).*

Dias, Alves and Lopes in the “Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation” aim of the paper is to divide a document into coherent topical sections. They have used to vector space model algorithm to change the sentence in the document into the vector. Vector of a particular sentence has been formed on the basis of the occurrence of words in that sentence. In order to segment the document into different topics they have consider tf.idf measure to consider the distribution of a particular word in the document. The basic idea of tf.idf measure in to give the score to a particular word not only on the basis of the frequency but also on the basis of the distribution over the document. Also, in order to divide the document into subtopics they have adapted a new technology which the similarity of a sentence has been compared with previous block of k sentence and next block of k sentence. However, after the calculating the similarity topic boundary had been detected. The model is a language independent Topic Segmentation system based on word co-occurrence. Final model has been compared with the c99 and TextTiling model on the basis of three matrices namely F-measures, the Pk estimate (Beeferman, Berger and Lafferty, 1997) and WindowDiff measure. The system showed a 33% improvement over the c99 algorithm and 24% over the TextTiling algorithm in terms of the F-score.

**PAPER REFERENCE 5**

*John, A.K., Di Caro, L. and Boella, G., 2016, November. Text Segmentation with Topic Modeling and Entity Coherence. In International Conference on Hybrid Intelligent Systems (pp. 175-185). Springer, Cham.*

Adebayo Kolawole John and Luigi Di Caro in their paper “*Text Segmentation with Topic Modelling and Entity Coherence*” proposed a system with entity and coherence for improved Text Segmentation (TS) accuracy. They have used the entity mapping techniques across the window to check the transition of the entities within the sentences. Their main aim in the paper was to detect the boundary of the topic shift in the document. In their approach firstly, they have divided the document into small text units. They have employed LDA topic modelling algorithm to obtain a topic for each word. However, while calculating the Topic-based Sentence Similarity they have taken into accounts the most frequent topic assigned and distribution of topics for each word in a particular sentence. They have used the Pk and WindDiff metrics for their system evaluation. Their presented TS approached has overperformed the state-of-the-art systems on the Choi’s TS dataset. s. Observingly, this bond tends to be higher among units with common topics. This notion is what is termed cohesion or coherence within a document. Cohesion is a function of grammatical factors, e.g., co-reference and sentential connectives as well as lexical factors like collocation. Coherence is higher within units that share several topics. The goal of TS is to identify points of weak or no coherence in a text. This idea, otherwise known as lexical cohesion could be tricky as it suffers from lexical ambiguity. This is because there are usually more than one words available to express an idea, i.e., synonyms while some words have multiple meanings, i.e., polysemy. The use of topics has recently been proposed, inspired by distributional semantics-based approaches such as Latent Semantic Analysis (LSA) and LDA topic models. Previous works on Text Segmentation basically adopt two approaches, e.g., lexical cohesion and discourse-based techniques.

**PAPER REFERENCE 6**

*Glavaš, G., Nanni, F. and Ponzetto, S.P., 2016. Unsupervised text segmentation using semantic relatedness graphs. Association for Computational Linguistics.*

Goran Glavas, Federico Nanni, Simone Paolo Ponzetto in their paper built a semantic relatedness graph in which nodes denote sentences and edges are created for pairs of semantically related sentences. They determined the coherent segments by finding maximal cliques of the relatedness graph. They first introduce the two evaluation datasets that we use one being the commonly used synthetic dataset and the other a realistic dataset of political manifestos. Following, they presented the experimental setting and finally describe and discuss the results achieved by GRAPHSEG algorithm and how it compares to other TS models. GRAPHSEG displays competitive performance compared to best-performing LDA-based methods on a synthetic dataset. However, they identify and discuss evaluation issues pertaining to LDA-based TS on this dataset. In their work they presented GRAPHSEG, a novel graph-based algorithm for unsupervised text segmentation. GRAPHSEG employs word embeddings and extends a measure of semantic relatedness to construct a relatedness graph with edges established between semantically related sentences. The segmentation is then determined by the maximal cliques of the relatedness graph and improved by semantic comparison of adjacent segments.

ALGORITHM

**ALGORITHM 1: TO CONVERT MESSAGES INTO APPROPRIATE TOPICS**

1. **Input:** JsonParsedMessages, WindowSize

2. **Output:** TopicSet

3.  **Initialization:** TopicSet=0,

4. **for** each message i **do**

5. topic[i]=None

6. **end for**

7. **for** each message index i **do**

8. **if** i>0 and author of message[i] is same as before **then**

9. Merge {topic[i], topic[i-1]}

10. **else**

11. **if** message[i] is a replyObject **then**

12. Merge {topic[i], topic[i-1]}

13. **else**

14. Start a new topic

15. **end if**

16. Add the new topic to the window of size WindowSize

17. **end if**

18. **end for**

19. **for** each topic in TopicSet **do**

20. **if** number of messages in topic is <= 2 **then**

21. Absorb the smaller topic in larger one

22. **end if**

23. **end for**

**ALGORITHM 2: TO PREDICT IF CURRENT MESSAGE IS A REPLY OBJECT OR NOT**

1. **Input:** message, thresholdForSimilarity

2. **Output:** Topic to which the current message is a reply, along with appropriate reason**.**

3. **Initialization:** replyStarters = ['ok', 'ok.', 'k', 'k.' 'mine', 'his', 'hers', 'theirs', 'ours']

3. **if** any topic is present in window **then**

4. Calculate SimilarityDistance of current topic with other topics in the window

5. Tokenize the message

6. Find POS tag of the tokens

7. **if** length of tokens <= 1

8.  **or** POS tag is starting with a universal reply tag

9. **or** message starts with a replyStarter

10. **or** message has @UserID in it

11. **or** size of last topic == 1

12. **or** similarityDistance > thresholdForSimilarity **then**

13. Classify it as a reply object with appropriate topic and reasoning

14. **else**

15. Classify it as not a reply object

16. **end if**

17. **end if**

**ALGORITHM 3: TO PREDICT IF CURRENT MESSAGE IS A REPLY OBJECT OR NOT**

1. **Input:** message, windowTopics

2. **Output:** Similarity Score

3. **Initialization:** slidingWindowSize = 10, scoresArray = []

4. Define decay function as decay = 0.993\*(difference of IDs of messages)

5.  **for** each message in each topic of windowTopics **do**

6. Tokenize the message

7. Convert tokens to vector

7. **for** each vector of tokens according to slidingWindowSize **do**

8. Calculate sum of absolute values to give Euclidean distance

9. Calculate cosine score

10. Store the best pair of both values

11. **end for**

12. Add the pair of values to scoresArray

13. **end for**

14. Sort on the basis of scoresArray[‘eucideanDistance’]

15. Take out only top 5% of the scores from the array

16. Sort on the basis of scoresArray[‘cosineScore’]

17. Classify similarity score as the scores from first index of the sorted array

**ALGORITHM 4: PROBABILISTIC GENERATIVE PROCESS FOR TOPIC SEGMENTATION MODEL**

1. **Input:** topics, documents

2. **Output:** probabilistic weightage of each topic segment

3. **for** each topic k ∈ {1, ..., K}:

4. Draw a word distribution φk ∼DirichletW (γ).

5. **end for**

6. **for** each document d ∈ {1, ..., D}:

7. Draw topic shift probability πd ∼Beta (λ0, λ1)

8. Draw µd ∼ DirichletK (α)

9. **for** each text passage (except last) u ∈ {1, ..., Ud −1}:

10. Draw ρd,u ∼Bernoulli (πd)

11. **end for**

12. Compute Sd the number of segments as 1 + ρd,u

13. **for** each segment s ∈ {1, ..., Sd}:

14. Draw νd,s ∼ PYP(a, b, µd)

15. **end for**

16. **for** each text passage u ∈{1,...,Ud}:

17. Set segment sd,u = 1 + ρd,v

18. **for** each word index n ∈ {1, ..., Nd,u}:

19. Draw topic zd,u,n ∼ DiscreteK (νd, sd,u)

20. Draw word wd,u,n ∼ DiscreteK (φzd,u,n).

21. **end for**

22. **end for**

23. **end for**

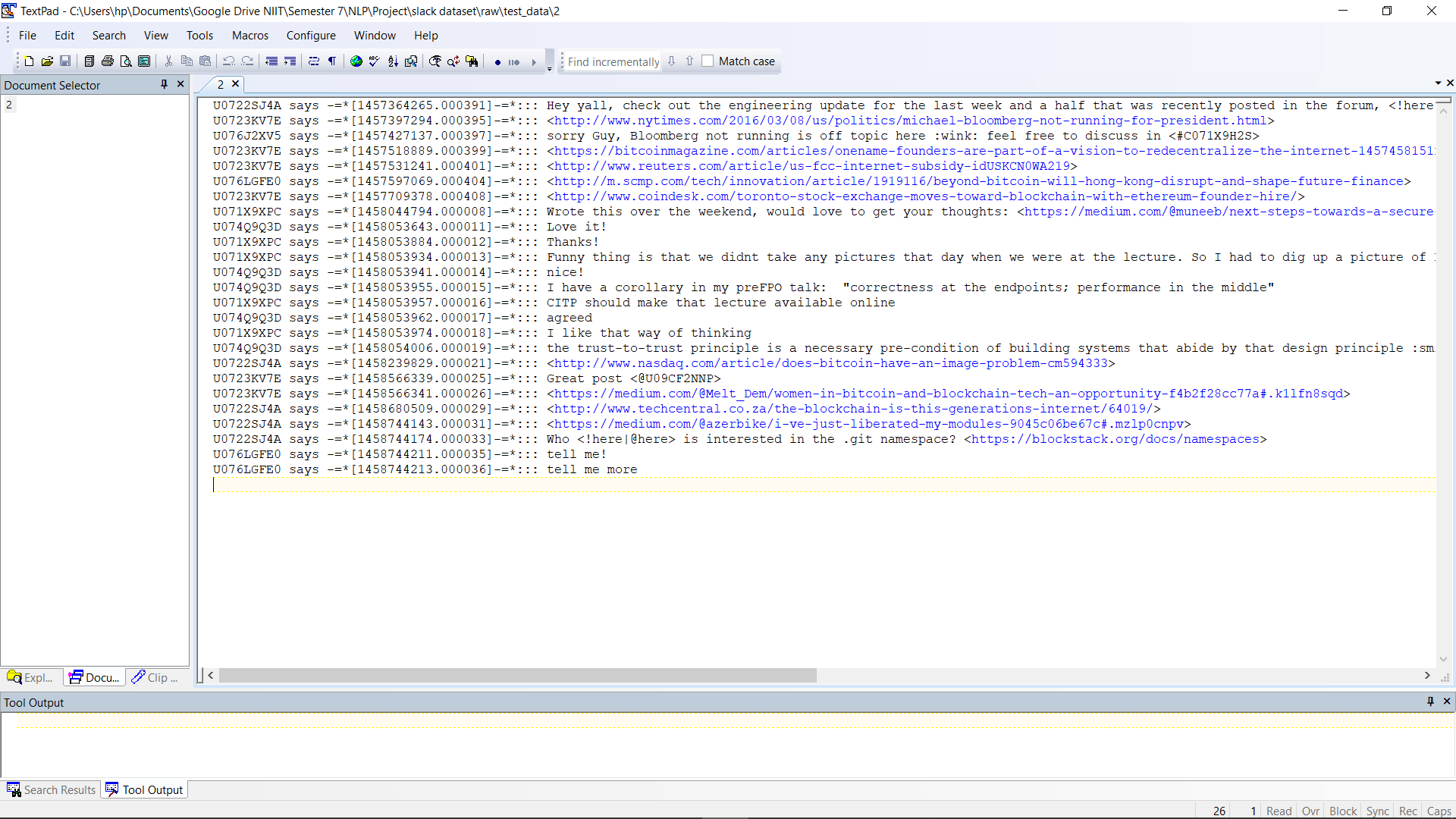
IMPLEMENTATION OF TOPIC MODELING

***Topic Modelling Techniques used for comparative analysis:***

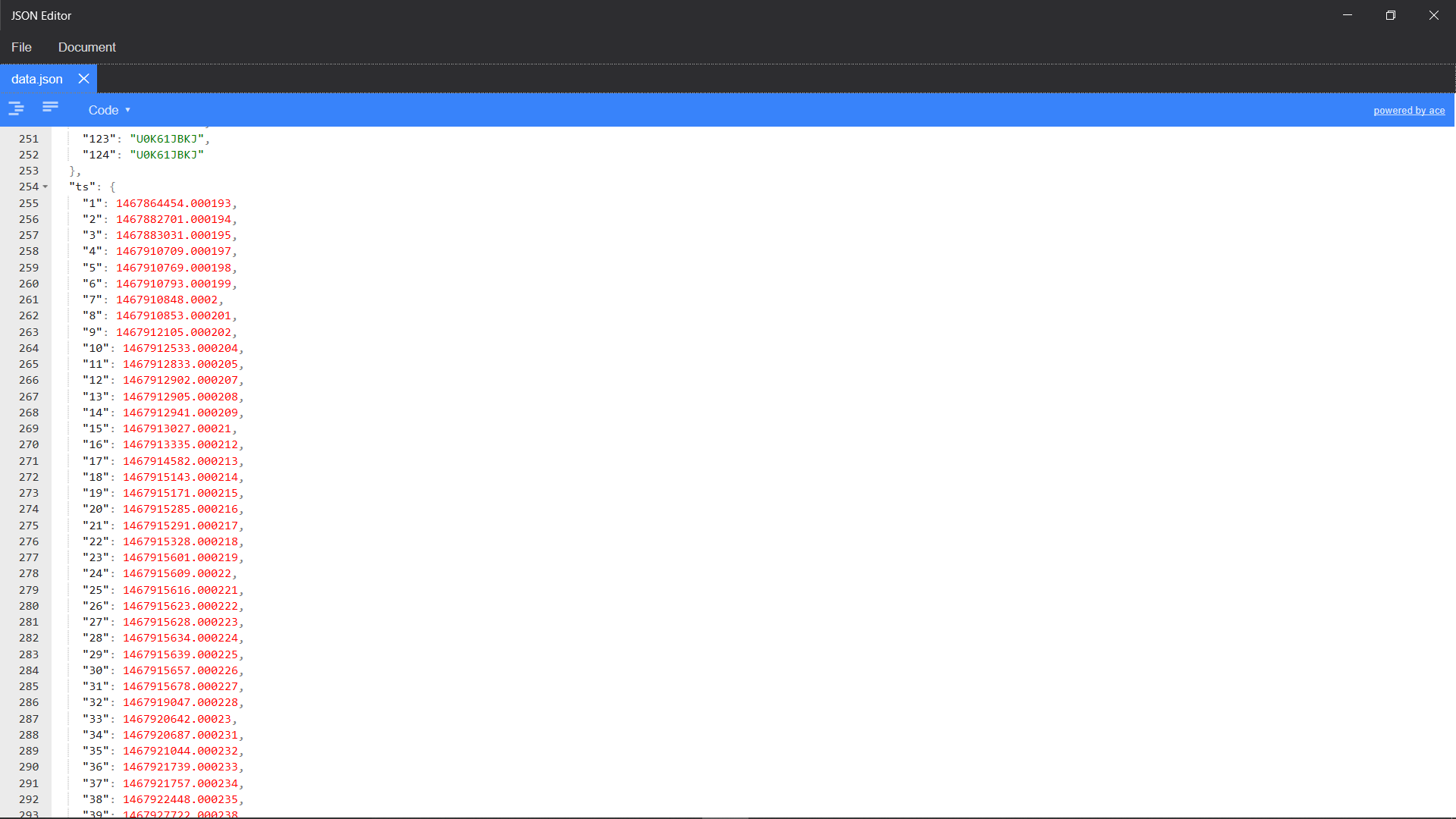
1. **Latent semantic analysis (LSA)** is a technique in natural language processing, in particular distributional semantics, of analysing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text (the distributional hypothesis). A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Words are then compared by taking the cosine of the angle between the two vectors (or the dot product between the normalizations of the two vectors) formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words.
2. **Hierarchical Dirichlet process (HDP)** is a nonparametric Bayesian approach to clustering grouped data. It uses a Dirichlet process for each group of data, with the Dirichlet processes for all groups sharing a base distribution which is itself drawn from a Dirichlet process. This method allows groups to share statistical strength via sharing of clusters across groups. The base distribution being drawn from a Dirichlet process is important, because draws from a Dirichlet process are atomic probability measures, and the atoms will appear in all group-level Dirichlet processes. Since each atom corresponds to a cluster, clusters are shared across all groups.
3. **Latent Dirichlet allocation (LDA)** is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is an example of a topic model.

SCREENSHOTS OF INPUT/OUTPUT

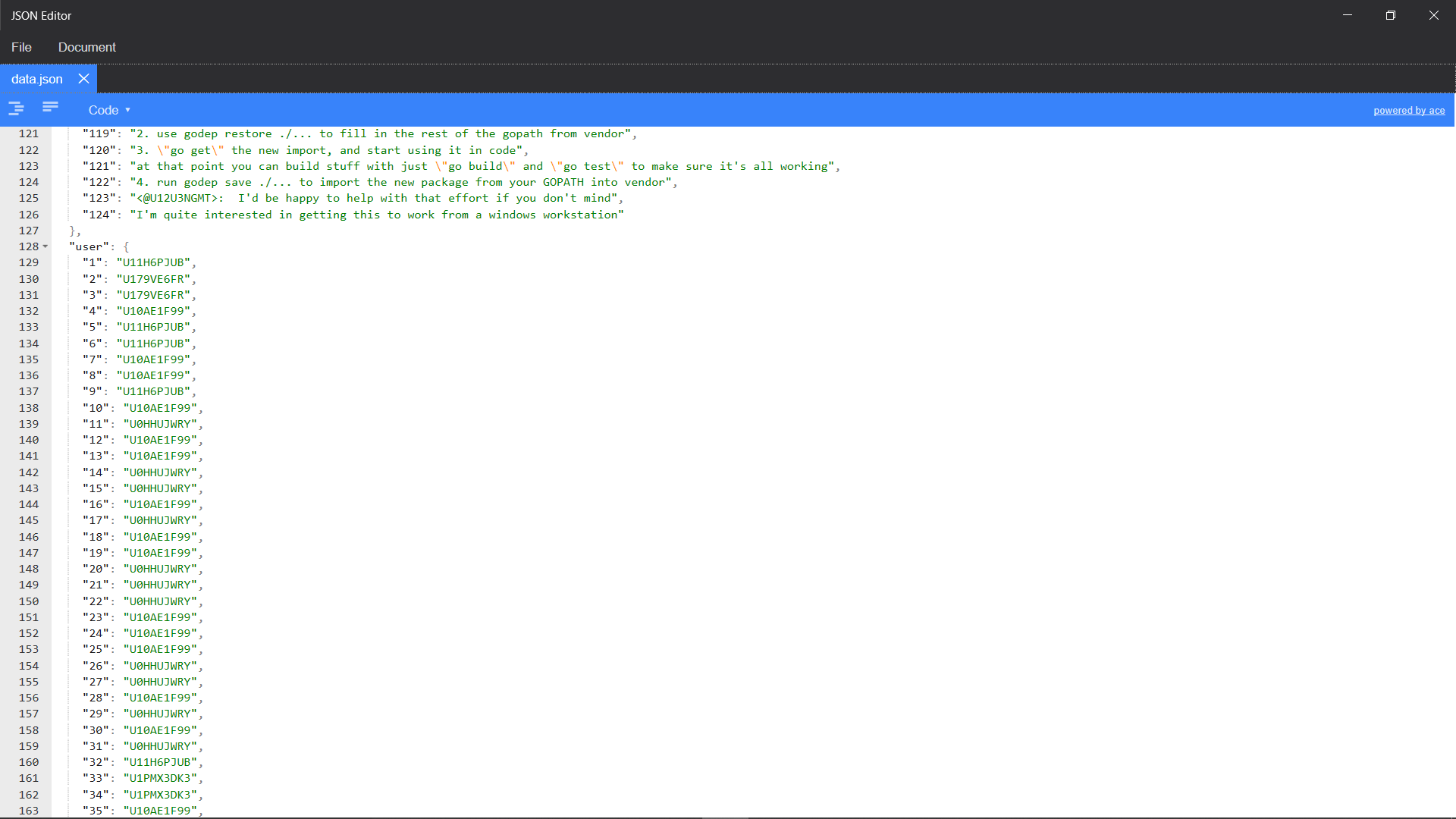
**FIG1: RAW DATASET**



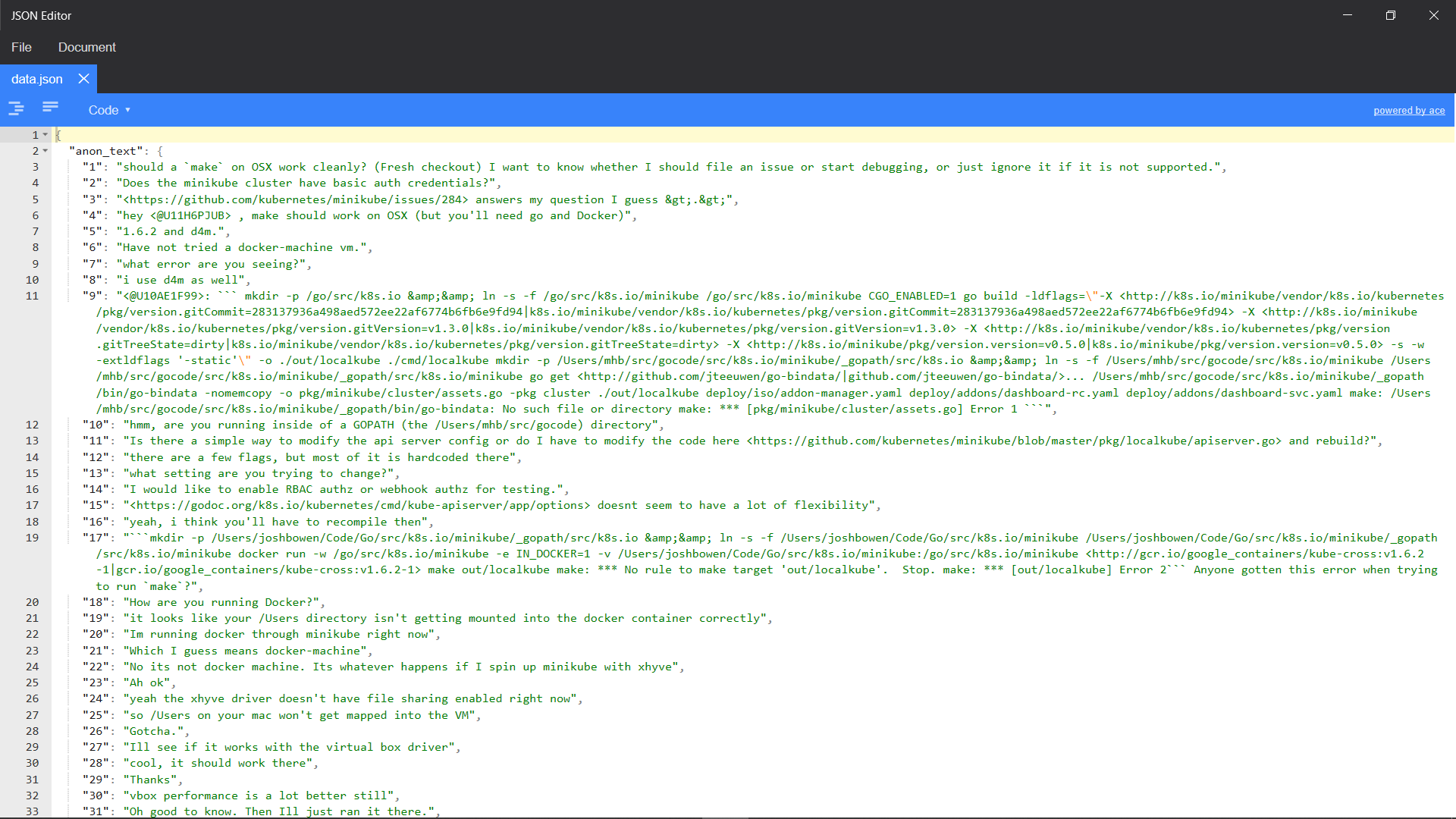
**FIG2: JSON DATA 1**



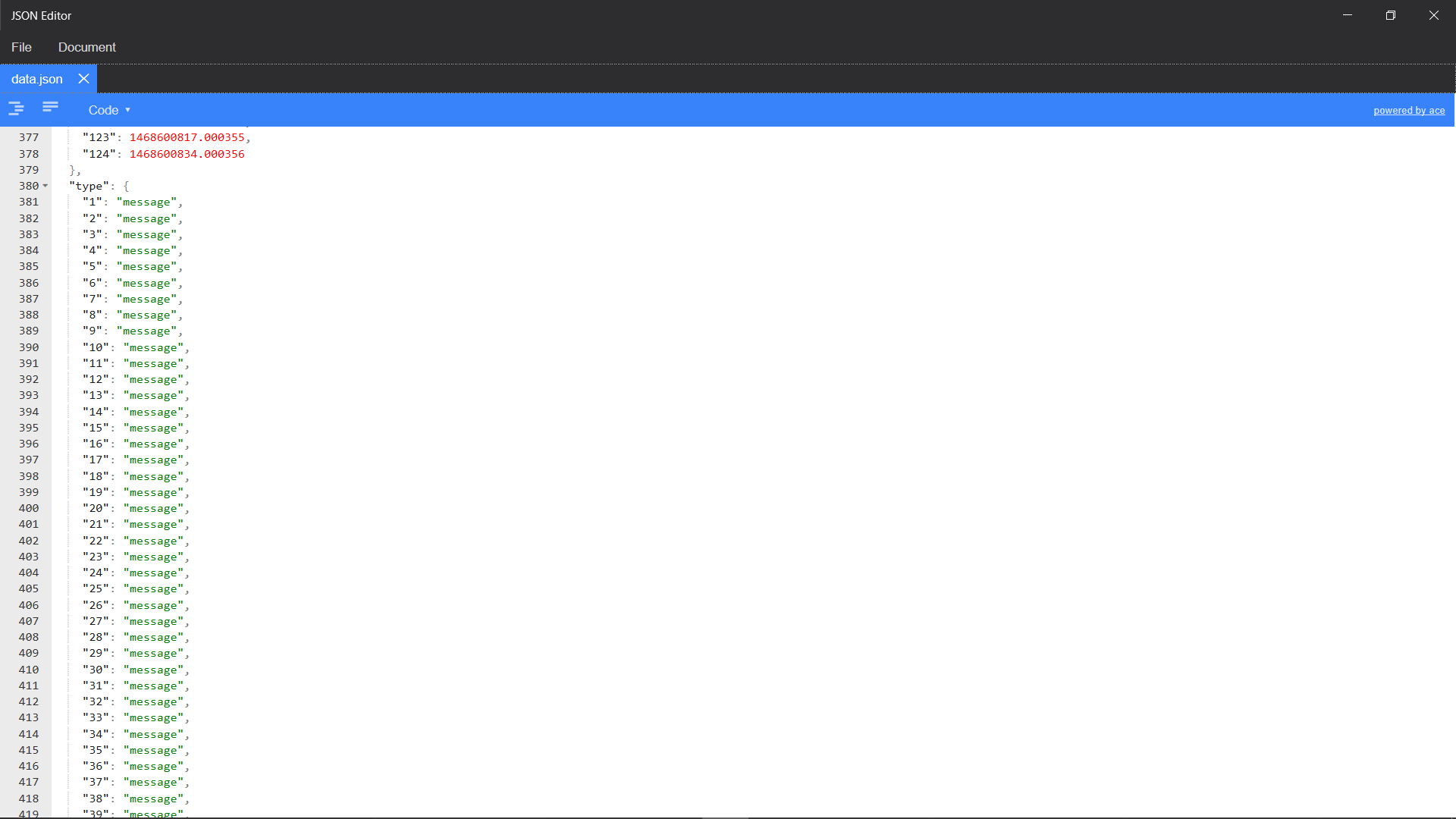
**FIG3: JSON DATA 2**



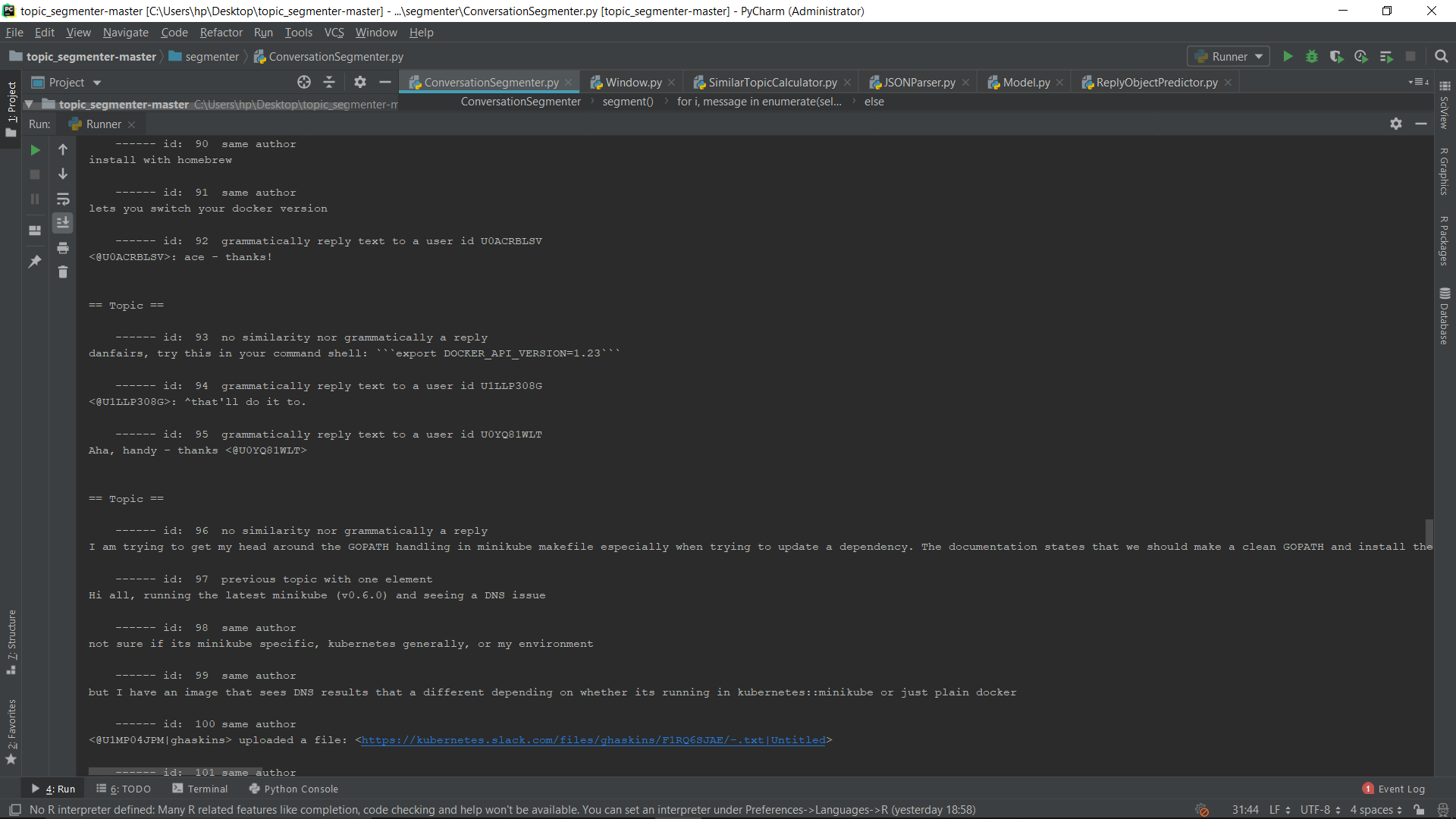
**FIG4: JSON DATA 3**



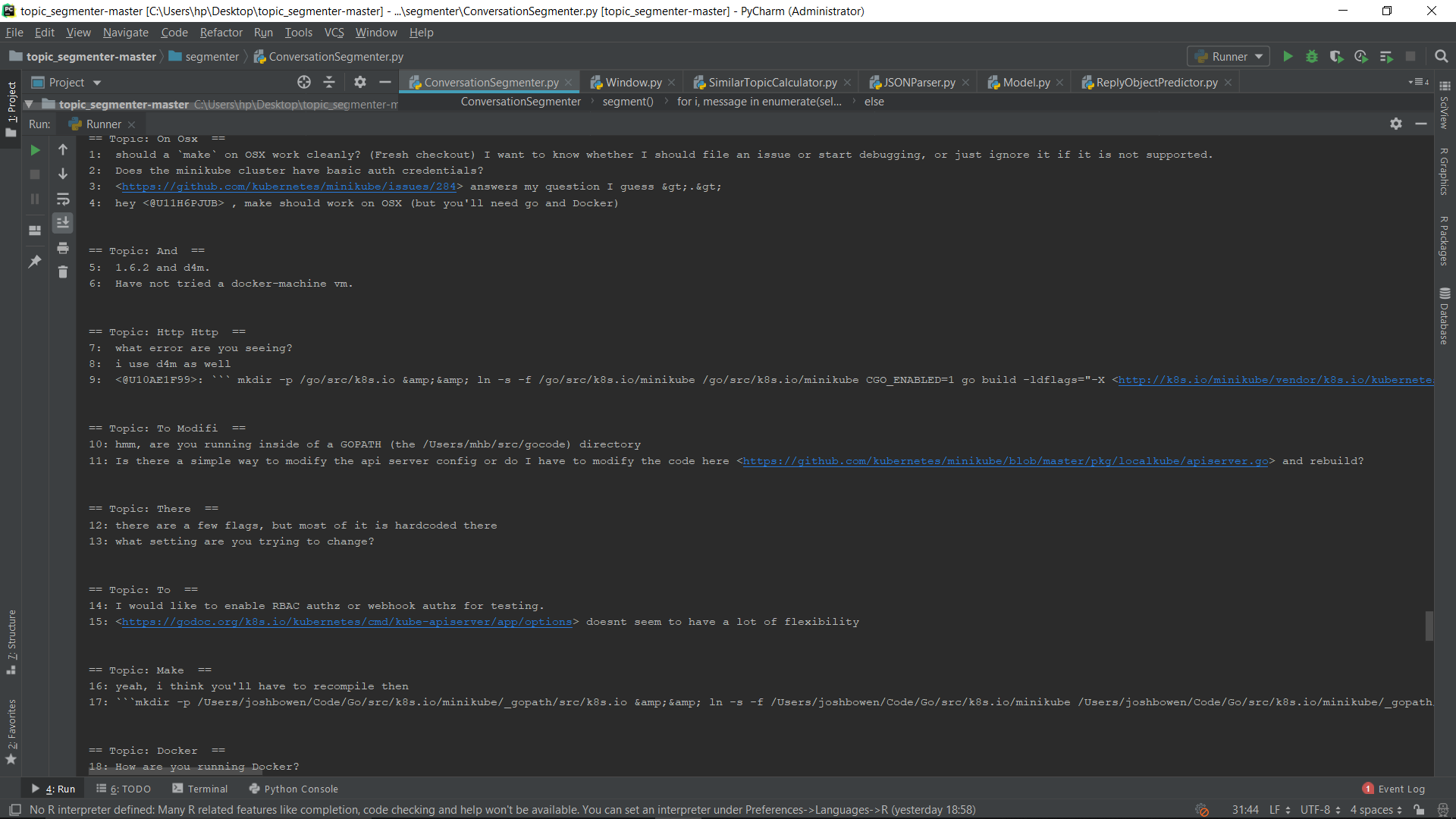
**FIG5: JSON DATA 4**



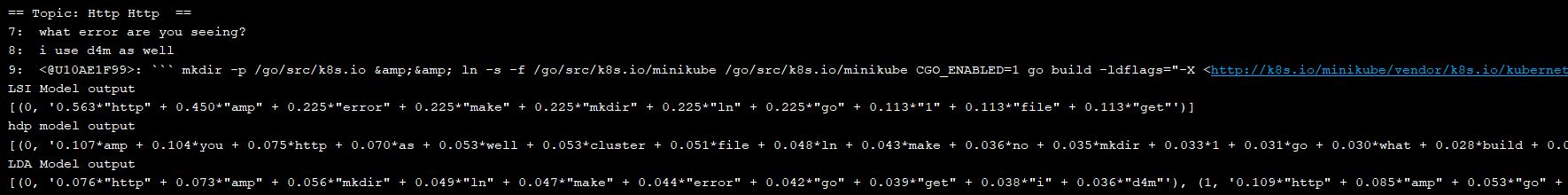
**FIG6: OUTPUT SCREEN WITH REASON FOR SEGMENTATION**



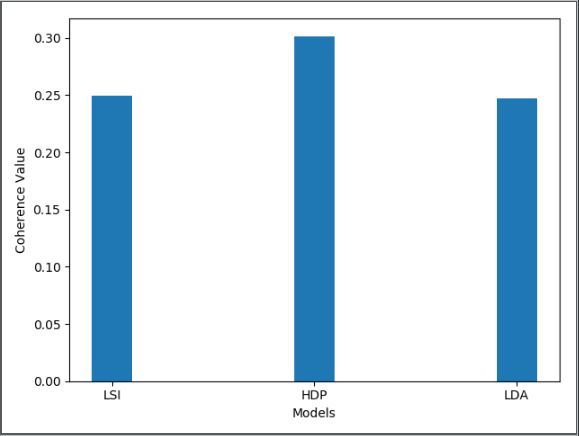
**FIG7: OUTPUT SCREENSHOT WITH TOPIC NAME**

****

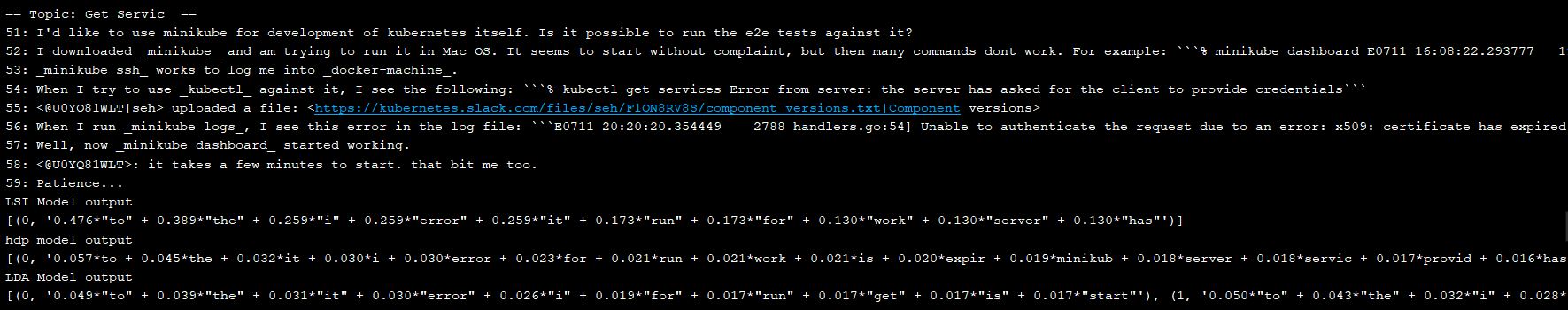
**FIG8: OUTPUT SCREENSHOT 1 WITH TOPIC MODELING**



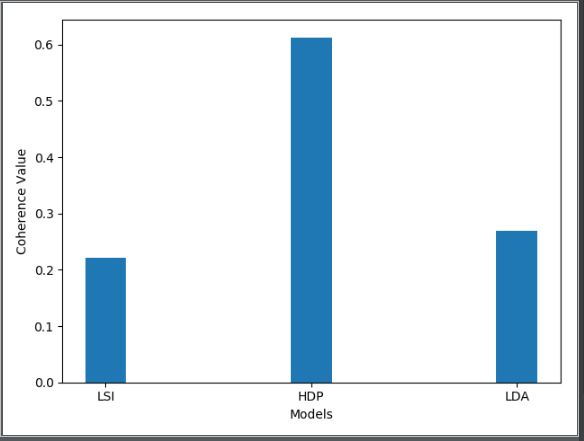
**FIG9: OUTPUT SCREENSHOT 1 WITH TOPIC MODELING COMPARISON**



**FIG10: OUTPUT SCREENSHOT 2 WITH TOPIC MODELING**



**FIG11: OUTPUT SCREENSHOT 2 WITH TOPIC MODELING COMPARISON**



**DATASET DESCRIPTION**

We have used the same dataset that is being provided to us along with the reference paper. It is basically a slack dataset. The format of the dataset is as follows:

1. Each text file contains around 25 chat conversations.
2. Each conversation consists of the text being written by the user along with the User ID and timestamp.
3. Each conversation comes in a single line.

This dataset is then converted to JSON format for further processing. The JSON format thus formed is as follows:

1. There are 4 objects named as “anon\_text”, “user”, “ts” and “type”.
2. Extract all chat text from each conversation and store it to “anon\_text”.
3. Extract all User IDs from each conversation and store it to “user”.
4. Extract all timestamp information from each conversation and store it to “ts”.
5. Since currently each conversation if by default of type “message”, so for each conversation, store the string “message” to the object “type”.

INDIVIDUAL CONTRIBUTION

***Contribution of each team member:***

**1. Rishabh:**

*Dataset conversion :* The raw dataset consisted of various chat files, in which each file’s format has User Id, timestamp and chat text in each line. So, a JSONParser was built that converted the file to the JSON file with User Id, timestamp and chat text as objects for quick processing.

*Tokenization :* It refers to dividing a string into appropriate tokens. By appropriate, we mean removing stop words and punctuations.

I*dentify Reply Objects* : It is very important to identify if a message is a reply or not, in order to merge it to the earlier topic. So, in order to consider a message as a reply, we considered various situations like if we have a singleton message, whether the message starts with words like ok, fine, etc., whether the message contains a user id or not, etc.

*Find Similarity distance :* Similarity Distance calculation involves quantifying the similarity between two objects, which are vectors of tokens in our case. This also involves calculating Euclidean Distance of a message to provide a central score of that message.

**2. Renuka:**

*Stemming :* The use of snowball stemmer as a library in python lead us to achieve stemming. It chops off the ending letters to reduce inflectional forms and sometimes derivationally related forms of a word or token to a common base form.

*Vectorizing :* Converting each token to vector give the ample opportunity to perform mathematical computations like calculating Similarity Distance.

*Apply Topic Modelling Techniques :* Implementation of the LDA, HDP and LSI topic models on each topic segment is done using an NLP framework called as gensim. It results in the probability distribution of the set of words present in that segment for each of the topic modelling techniques and shows graphical comparison.

**3. Soumi:**

*Integrating all to convert raw data to topics :* Dividing the complete objective into modules and classes in order to debug the code precisely lead to simplifying the process of integration.

*Naming the topics :* It involves tokenizing the stemming the messages in a topic and using probabilistic model to get the most likely word(s) as a name of the topic.

*Incorporate window size functionality* : We kept the size of window to be 3, which means the last three topics need to be remembered so that the comparison can be made for analysing whether a message is a reply object or not. Since large window size would lead to keeping messages that have huge difference in timestamp, so deciding a reasonable size is important.

CONCLUSION AND FUTURE WORK

**CONCLUSION**

The goal of Text Segmentation (TS) is to identify boundaries of topic shift in a document. Discourse structure studies have shown that a document is usually a mixture of topics and sub-topics. A shift in topics could be noticed with changes in patterns of vocabulary usage. The process of dividing text into portions of different topical themes is called Text Segmentation. We in our project have used a slack dataset which is further divided into training and test dataset. Initially we have tokenized each message by removing the stop words and punctuations signs. Then each word has been reduced to its root form. After this each word have is converted into n-dimensional vector so that algebraic operations can be performed which carry semantic meaning. Several kinds of constraints are kept into mind such as if there are groups of 1 or 2 messages that are surrounded by messages belonging to the same topic, the bigger topic absorbs the tiny one. This happens when one or two messages are short and so grammatically incoherent that it's not possible to related them to any existing topic. Secondly, to identify reply objects. There a considerable number of messages that are simply reply messages which don't contribute to the topic but are part of the same topic. Semantically, these messages by themselves carry no meaning unless they belong to a bigger topic. We call these message reply objects, and we treat them as a particular case. Thirdly, we have introduced a dynamic window size feature to make the machine remember the last topic for better topic segmentation. Fourthly, we introduced a concept of Similarity Distance. Since each conversation is already converted to vector, so we can apply POS tagging to identify semantic meaning for each conversation and provide a score to be compared to other conversation. This helps to segment the texts more efficiently. After the topics being segmented, they were given names according to Unigram/Bigram probability and then using different Topic Modelling techniques, the efficiency of segmentation was compared. We found that Hierarchical Dirichlet Process, being independent of input for number of topics, gives the best result among them all. The described algorithm has been tested upon various training and test dataset in order to show the efficiency of the proposed algorithm.

**FUTURE WORK**

We have used various in-built libraries so that we don’t have to spend more time in training the dataset rigorously to get the valid output and further more time can be spent in analysing and refining the output as per the requirements. This model can further be extended by using topic modelling with LDA. The model can be trained by using training dataset then it can be tested using test dataset in order to provide more better and accurate results. Training of model can also be used for appropriate naming of the topics. The model can be further extended which will give suggestions for any issue raised by searching the related topics.