NATURAL LANGUAGE PROCESSING
CS491

PROJECT REPORT 2

Group Chat Text Segmentation using Topic Modelling

GROUP NUMBER- G5

TEAM MEMBERS DETAILS

1) KUMARI RENUKA (U101115FCS111)

2) RISHABH KUMAR KANDOI (U101115FCS283)

3) SOUMI PAL (U101115FCS158)

TABLE OF CONTENTS

PROJECT DETAILS

GROUP MEMBERS-

S.NO	NAME	EMAIL	E .NUMBER
1	*Kumari Renuka*	*kumari.renuka@st.niituniversity.in*	*U101115FCS111*
2	*Rishabh Kandoi*	*rishabhk.kandoi@st.niituniversity.in*	*U101115FCS283*
3	*Soumi Pal*	*Soumi.pal@st.niituniversity.in*	*U101115FCS158*

ASSIGNED PAPER

Topic Segmentation using bottleneck method

CURRENT PAPER

. Topic segmentation with a structured topic model

PAPER SUMMARY

DESCRIPTION OF THE PAPER ASSIGNED AND THE PROBLEM FACED

Information Bottleneck (IB) method has been successfully applied to clustering in the NLP domain. Specifically, IB attempts to balance the trade-off between accuracy and compression (or complexity) while clustering the target variable, given a joint probability distribution between the target variable and an observed relevant variable. Similar to clustering, this paper interprets the task of text segmentation as a compression task with a constraint that allows only contiguous text snippets to be in a group. In the proposed technique the information bottleneck method augmented with **sequential continuity constraints**. Furthermore, they have utilized critical **non-textual clues** such as time between two consecutive posts and people mentions within the posts.

However, **information bottleneck** is recently proved to a theoretical foundation for deep learning and very complex in its own to be integrated with the natural language processing. Also, we are not be able to find sufficient resources and research papers related to chat text segmentation using information bottleneck.

Reference- Vishal, S., Yadav, M., Vig, L. and Shroff, G., 2017. Information Bottleneck Inspired Method For Chat Text Segmentation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Vol. 1, pp. 194-203).

SUMMARY: GROUP CHAT TEXT SEGMENTATION USING TOPIC MODELLING

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.

Reference- Du, L., Buntine, W. and Johnson, M., 2013. Topic segmentation with a structured topic model. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 190-200).

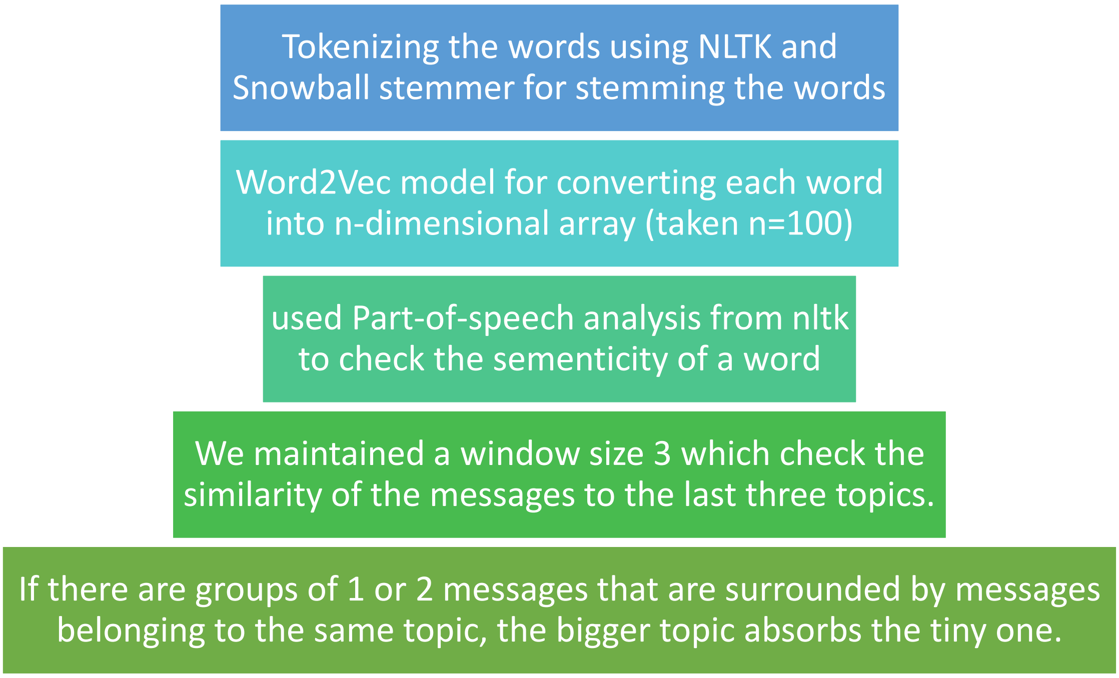
TOPIC SEGMENTER

This tool segment group chat into several conversations and each conversation represents a different subject. We have used the knowledge of Natural Language Processing and Neural Networks.

IMPORTANT POINTS TO CONSIDERED

1. Messages from a same group and related to a particular topic many not be sequential.
2. We have segmented a single large conversation related to the same topics into smaller subtopics.
3. The reply texts such as ‘OK’, ‘nice ‘, ’I agree’ are treated as the category of the same topic.

FIG1: WORKING OF A TOPIC SEGMENTER



Tokenizing the words using NLTK and Snowball stemmer for stemming the words

Word2Vec model for converting each word into n-dimensional array (taken n=100)

used Part-of-speech analysis from nltk to check the sementicity of a word

We maintained a window size 3 which check the similarity of the messages to the last three topics.

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.

WORK DONE SO FAR

After going through the reference paper provided and analysing the feasibility of the implementation of the paper, we found that the method involved in doing chat text segmentation is beyond our scope to implement it in the same way as described in the paper. The unavailability of in-depth information about implementing Information Bottleneck method for Text Segmentation made us search for some other possible methods to do so.

We found there are other numerous ways that can be adopted to get the same results with rather less complexity like Naïve Bayes, SciPy, etc. One such method is using nltk library which is built for NLP in Python. We chose this library because it can work simply over unsupervised data as well. This means we don't have to spend more time in training the dataset rigorously to get the valid output and thus more time can be spent in analysing and refining the output as per the requirements.

Till now, our method is designed to work as follows:

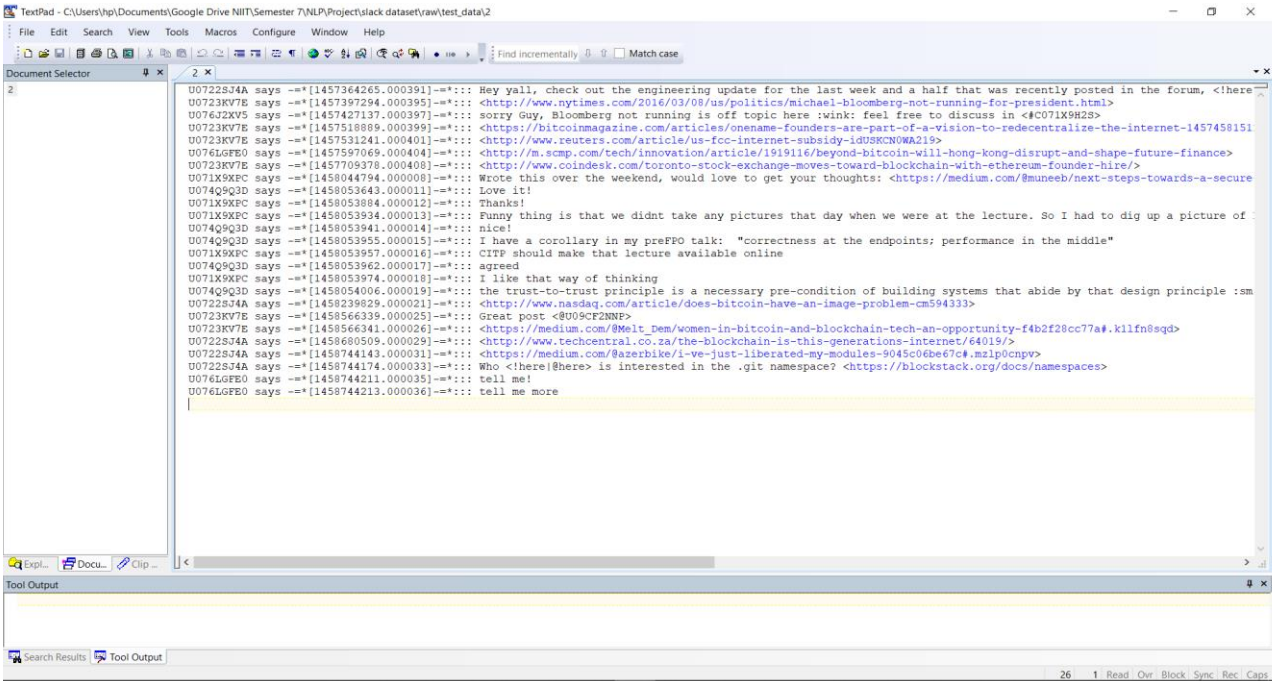
1. Built the dataset as described below.
2. Use "nltk" to tokenize each message removing stop words (e.g. 'to', 'and', 'a') and punctuation signs, except in cases like "It's", where the apostrophe is necessary.
3. Then we use a Snowball stemmer to reduce each word to its root form (e.g. 'agrees' and 'agreed' to 'agree').
4. Now we have a simplified vector text for each message.
5. Then we use a Word2Vec model to convert each word into an n-dimensional vector (by default n is 100). The advantage of this model is that algebraic operations on these vectors carry semantic meaning (e.g. vector('king') - vector('man') ~= vector('queen') - vector('woman')). We use the whole set of messages as the training data of the model.
6. As a post-step, 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.

Contribution of each team member:

1. Rishabh: Dataset conversion and tokenization.
2. Renuka: Stemming and vectorizing.
3. Soumi: Integrating all to convert raw data to topics. Researched and came out with the idea adopted and explained it briefly to the group members.

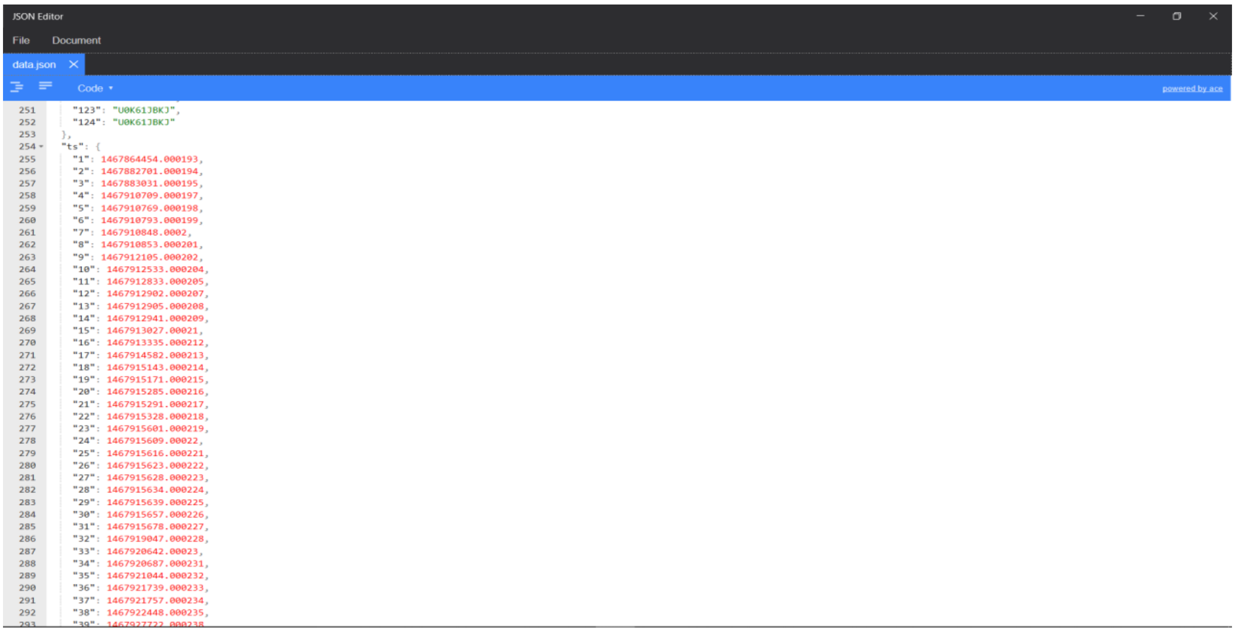
SCREENSHOTS OF OUTPUT

FIG2: RAW DATASET



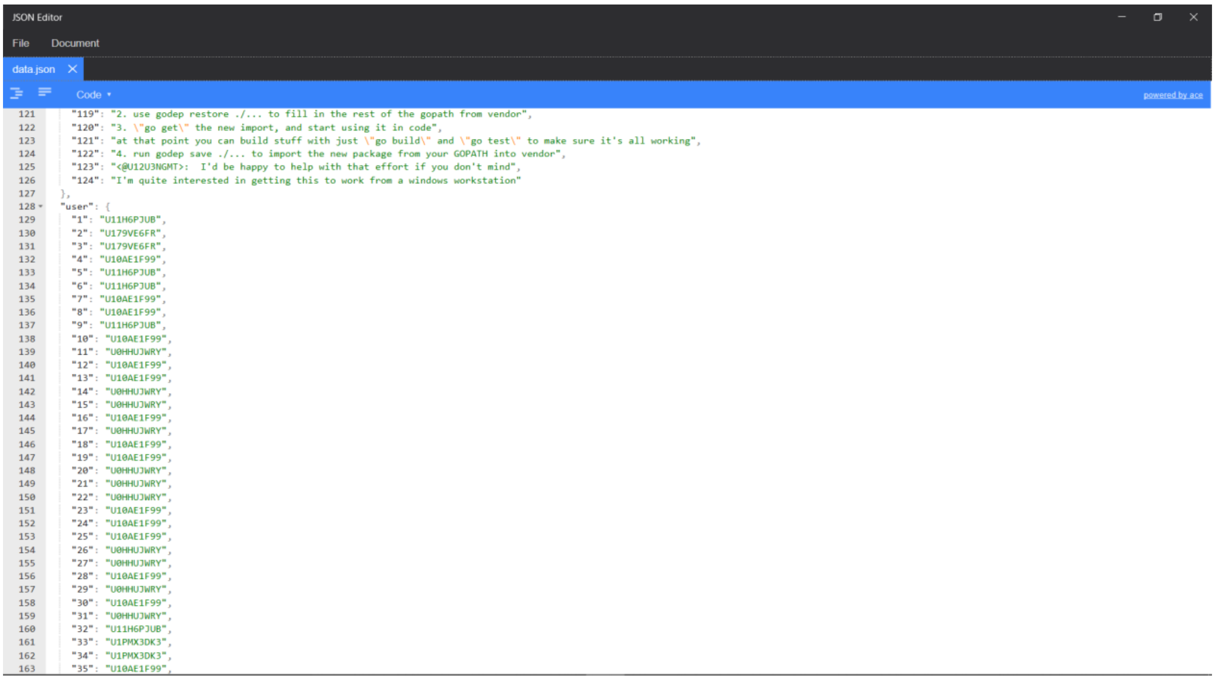
```
U0722SJ4A says -=*[1457364265.000391]-=*::: Hey yall, check out the engineering update for the last week and a half that was recently posted in the forum, <!here
U0723KV7E says -=*[1457397294.000395]-=*::: <http://www.nytimes.com/2016/03/08/us/politics/michael-bloomberg-not-running-for-president.html>
U076J2XV5 says -=*[1457427137.000397]-=*::: sorry Guy, Bloomberg not running is off topic here :wink: feel free to discuss in <#C071X9H2S>
U0723KV7E says -=*[1457518889.000399]-=*::: <https://bitcoinmagazine.com/articles/onename-founders-are-part-of-a-vision-to-redecentralize-the-internet-1457458151
U0723KV7E says -=*[1457531241.000401]-=*::: <http://www.reuters.com/article/us-fcc-internet-subsidy-idUSKCN0WA219>
U076LGFE0 says -=*[1457597069.000404]-=*::: <http://m.scmp.com/tech/innovation/article/1919116/beyond-bitcoin-will-hong-kong-disrupt-and-shape-future-finance>
U0723KV7E says -=*[1457709378.000408]-=*::: <http://www.coindesk.com/toronto-stock-exchange-moves-toward-blockchain-with-ethereum-founder-hire/>
U071X9XPC says -=*[1458044794.000008]-=*::: Wrote this over the weekend, would love to get your thoughts: <https://medium.com/@muneeb/next-steps-towards-a-secure
U074Q9Q3D says -=*[1458053643.000011]-=*::: Love it!
U071X9XPC says -=*[1458053884.000012]-=*::: Thanks!
U071X9XPC says -=*[1458053934.000013]-=*::: Funny thing is that we didnt take any pictures that day when we were at the lecture. So I had to dig up a picture of
U074Q9Q3D says -=*[1458053941.000014]-=*::: nice!
U074Q9Q3D says -=*[1458053955.000015]-=*::: I have a corollary in my preFPO talk:  "correctness at the endpoints; performance in the middle"
U071X9XPC says -=*[1458053957.000016]-=*::: CITP should make that lecture available online
U074Q9Q3D says -=*[1458053962.000017]-=*::: agreed
U071X9XPC says -=*[1458053974.000018]-=*::: I like that way of thinking
U074Q9Q3D says -=*[1458054006.000019]-=*::: the trust-to-trust principle is a necessary pre-condition of building systems that abide by that design principle :sm
U0722SJ4A says -=*[1458239829.000021]-=*::: <http://www.nasdaq.com/article/does-bitcoin-have-an-image-problem-cm594333>
U0723KV7E says -=*[1458566339.000025]-=*::: Great post <@U09CF2NNP>
U0723KV7E says -=*[1458566341.000026]-=*::: <https://medium.com/@Melt_Dem/women-in-bitcoin-and-blockchain-tech-an-opportunity-f4b2f28cc77a#.k1lfn8sqd>
U0722SJ4A says -=*[1458680509.000029]-=*::: <http://www.techcentral.co.za/the-blockchain-is-this-generations-internet/64019/>
U0722SJ4A says -=*[1458744143.000031]-=*::: <https://medium.com/@azerbike/i-ve-just-liberated-my-modules-9045c06be67c#.mzlp0cnpv>
U0722SJ4A says -=*[1458744174.000033]-=*::: Who <!here|@here> is interested in the .git namespace? <https://blockstack.org/docs/namespaces>
U076LGFE0 says -=*[1458744211.000035]-=*::: tell me!
U076LGFE0 says -=*[1458744213.000036]-=*::: tell me more
```

FIG3: JSON DATA 1



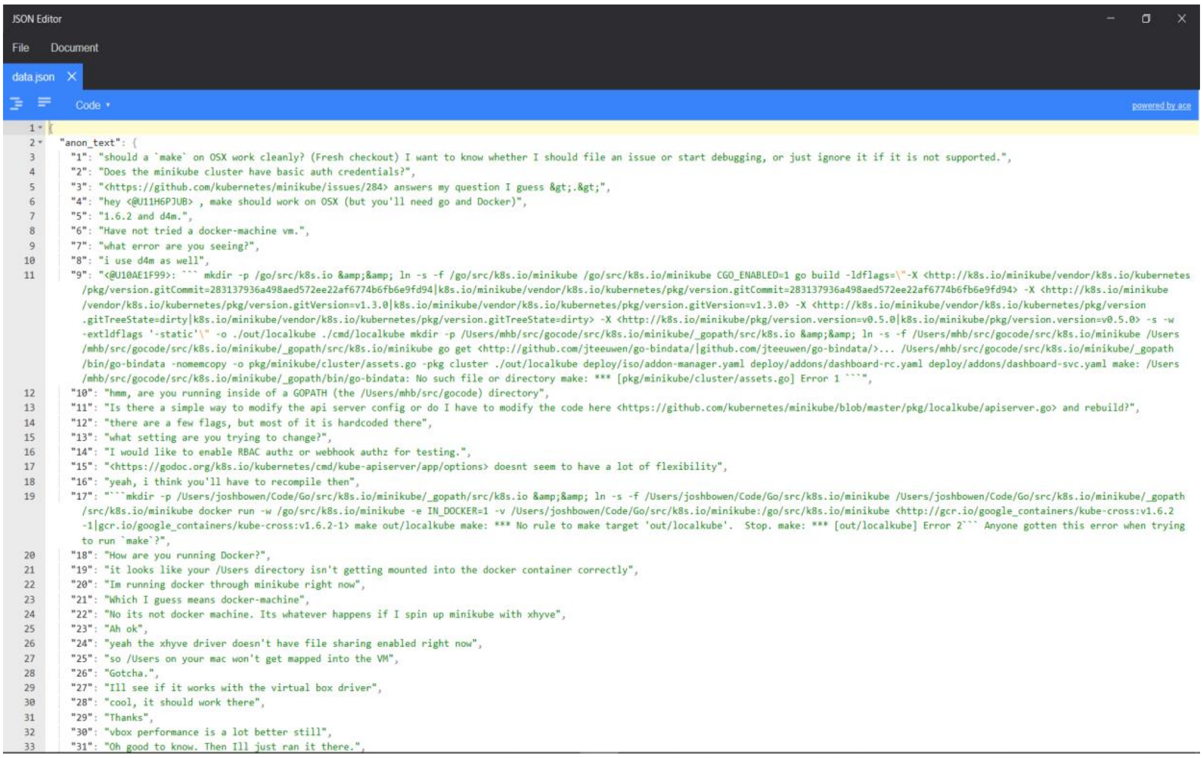
```
    "123": "U0K61JBKJ",
    "124": "U0K61JBKJ"
  },
  "ts": {
    "1": 1467864454.000193,
    "2": 1467882701.000194,
    "3": 1467883031.000195,
    "4": 1467910709.000197,
    "5": 1467910769.000198,
    "6": 1467910793.000199,
    "7": 1467910848.0002,
    "8": 1467910853.000201,
    "9": 1467912105.000202,
    "10": 1467912533.000204,
    "11": 1467912833.000205,
    "12": 1467912902.000207,
    "13": 1467912905.000208,
    "14": 1467912941.000209,
    "15": 1467913027.00021,
    "16": 1467913335.000212,
    "17": 1467914582.000213,
    "18": 1467915143.000214,
    "19": 1467915171.000215,
    "20": 1467915285.000216,
    "21": 1467915291.000217,
    "22": 1467915328.000218,
    "23": 1467915601.000219,
    "24": 1467915609.00022,
    "25": 1467915616.000221,
    "26": 1467915623.000222,
    "27": 1467915628.000223,
    "28": 1467915634.000224,
    "29": 1467915639.000225,
    "30": 1467915657.000226,
    "31": 1467915678.000227,
    "32": 1467919047.000228,
    "33": 1467920642.00023,
    "34": 1467920687.000231,
    "35": 1467921044.000232,
    "36": 1467921739.000233,
    "37": 1467921757.000234,
    "38": 1467922448.000235,
    "39": 1467927722.000238
```

FIG4: JSON DATA 2



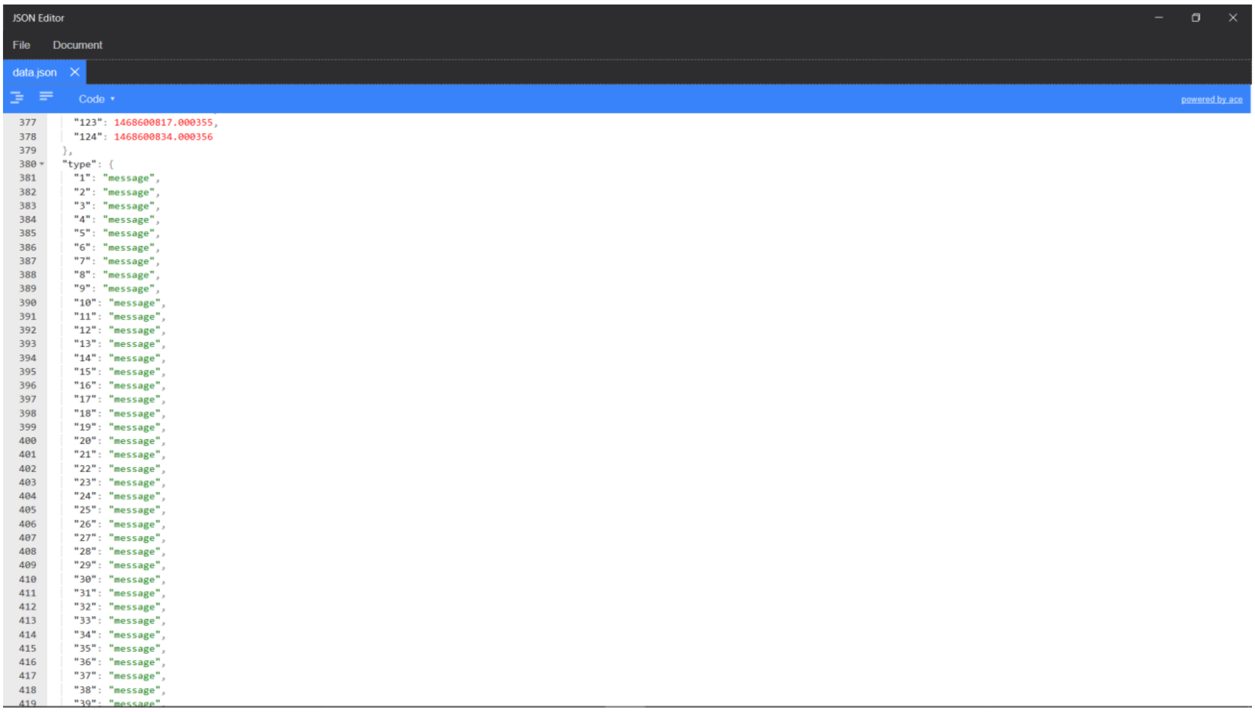
```
    "119": "2. use godep restore ./... to fill in the rest of the gopath from vendor",
    "120": "3. \"go get\" the new import, and start using it in code",
    "121": "at that point you can build stuff with just \"go build\" and \"go test\" to make sure it's all working",
    "122": "4. run godep save ./... to import the new package from your GOPATH into vendor",
    "123": "<@U12U3NGMT>:  I'd be happy to help with that effort if you don't mind",
    "124": "I'm quite interested in getting this to work from a windows workstation"
  },
  "user": {
    "1": "U11H6PJUB",
    "2": "U179VE6FR",
    "3": "U179VE6FR",
    "4": "U10AE1F99",
    "5": "U11H6PJUB",
    "6": "U11H6PJUB",
    "7": "U10AE1F99",
    "8": "U10AE1F99",
    "9": "U11H6PJUB",
    "10": "U10AE1F99",
    "11": "U0HHUJWRY",
    "12": "U10AE1F99",
    "13": "U10AE1F99",
    "14": "U0HHUJWRY",
    "15": "U0HHUJWRY",
    "16": "U10AE1F99",
    "17": "U0HHUJWRY",
    "18": "U10AE1F99",
    "19": "U10AE1F99",
    "20": "U0HHUJWRY",
    "21": "U0HHUJWRY",
    "22": "U0HHUJWRY",
    "23": "U10AE1F99",
    "24": "U10AE1F99",
    "25": "U10AE1F99",
    "26": "U0HHUJWRY",
    "27": "U0HHUJWRY",
    "28": "U10AE1F99",
    "29": "U0HHUJWRY",
    "30": "U10AE1F99",
    "31": "U0HHUJWRY",
    "32": "U11H6PJUB",
    "33": "U1PMX3DK3",
    "34": "U1PMX3DK3",
    "35": "U10AE1F99",
```

FIG5: JSON DATA 3



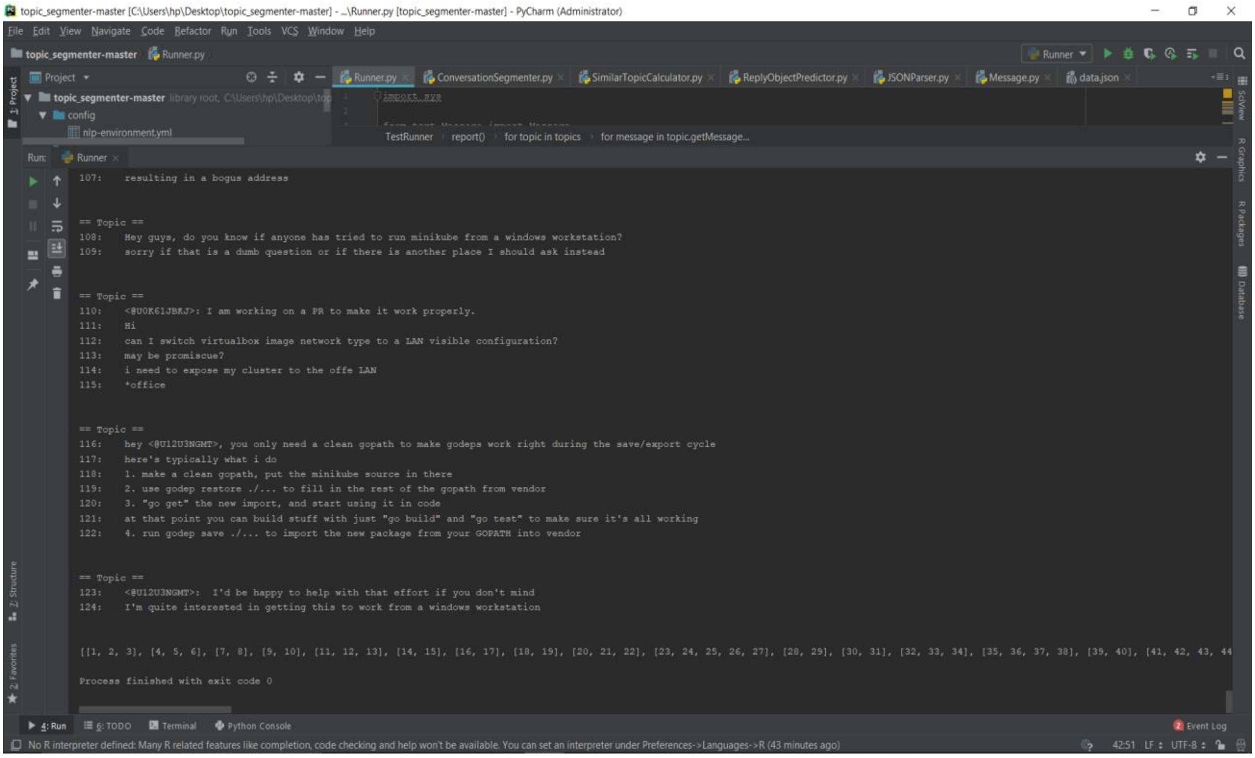
```
{
  "anon_text": {
    "1": "should a `make` on OSX work cleanly? (Fresh checkout) I want to know whether I should file an issue or start debugging, or just ignore it if it is not supported.",
    "2": "Does the minikube cluster have basic auth credentials?",
    "3": "<https://github.com/kubernetes/minikube/issues/284> answers my question I guess &gt;.&gt;",
    "4": "hey <@U11H6PJUB> , make should work on OSX (but you'll need go and Docker)",
    "5": "1.6.2 and d4m.",
    "6": "Have not tried a docker-machine vm.",
    "7": "what error are you seeing?",
    "8": "i use d4m as well",
    "9": "<@U10AE1F99>: ``` mkdir -p /go/src/k8s.io &amp;&amp; ln -s -f /go/src/k8s.io/minikube /go/src/k8s.io/minikube CGO_ENABLED=1 go build -ldflags=\"-X <http://k8s.io/minikube/vendor/k8s.io/kubernetes
      /pkg/version.gitCommit=283137936a498aed572ee22af6774b6fb6e9fd94|k8s.io/minikube/vendor/k8s.io/kubernetes/pkg/version.gitCommit=283137936a498aed572ee22af6774b6fb6e9fd94> -X <http://k8s.io/minikube
      /vendor/k8s.io/kubernetes/pkg/version.gitVersion=v1.3.0|k8s.io/minikube/vendor/k8s.io/kubernetes/pkg/version.gitVersion=v1.3.0> -X <http://k8s.io/minikube/vendor/k8s.io/kubernetes/pkg/version
      .gitTreeState=dirty|k8s.io/minikube/vendor/k8s.io/kubernetes/pkg/version.gitTreeState=dirty> -X <http://k8s.io/minikube/pkg/version.version=v0.5.0|k8s.io/minikube/pkg/version.version=v0.5.0> -s -w
      -extldflags '-static'\" -o ./out/localkube ./cmd/localkube mkdir -p /Users/mhb/src/gocode/src/k8s.io/minikube/_gopath/src/k8s.io &amp;&amp; ln -s -f /Users/mhb/src/gocode/src/k8s.io/minikube /Users
      /mhb/src/gocode/src/k8s.io/minikube/_gopath/src/k8s.io/minikube go get <http://github.com/jteeuwen/go-bindata/|github.com/jteeuwen/go-bindata/>... /Users/mhb/src/gocode/src/k8s.io/minikube/_gopath
      /bin/go-bindata -nomemcopy -o pkg/minikube/cluster/assets.go -pkg cluster ./out/localkube deploy/iso/addon-manager.yaml deploy/addons/dashboard-rc.yaml deploy/addons/dashboard-svc.yaml make: /Users
      /mhb/src/gocode/src/k8s.io/minikube/_gopath/bin/go-bindata: No such file or directory make: *** [pkg/minikube/cluster/assets.go] Error 1 ```",
    "10": "hmm, are you running inside of a GOPATH (the /Users/mhb/src/gocode) directory",
    "11": "Is there a simple way to modify the api server config or do I have to modify the code here <https://github.com/kubernetes/minikube/blob/master/pkg/localkube/apiserver.go> and rebuild?",
    "12": "there are a few flags, but most of it is hardcoded there",
    "13": "what setting are you trying to change?",
    "14": "I would like to enable RBAC authz or webhook authz for testing.",
    "15": "<https://godoc.org/k8s.io/kubernetes/cmd/kube-apiserver/app/options> doesnt seem to have a lot of flexibility",
    "16": "yeah, i think you'll have to recompile then",
    "17": "```mkdir -p /Users/joshbowen/Code/Go/src/k8s.io/minikube/_gopath/src/k8s.io &amp;&amp; ln -s -f /Users/joshbowen/Code/Go/src/k8s.io/minikube /Users/joshbowen/Code/Go/src/k8s.io/minikube/_gopath
      /src/k8s.io/minikube docker run -w /go/src/k8s.io/minikube -e IN_DOCKER=1 -v /Users/joshbowen/Code/Go/src/k8s.io/minikube:/go/src/k8s.io/minikube <http://gcr.io/google_containers/kube-cross:v1.6.2
      -1|gcr.io/google_containers/kube-cross:v1.6.2-1> make out/localkube make: *** No rule to make target 'out/localkube'.  Stop. make: *** [out/localkube] Error 2``` Anyone gotten this error when trying
      to run `make`?",
    "18": "How are you running Docker?",
    "19": "it looks like your /Users directory isn't getting mounted into the docker container correctly",
    "20": "Im running docker through minikube right now",
    "21": "Which I guess means docker-machine",
    "22": "No its not docker machine. Its whatever happens if I spin up minikube with xhyve",
    "23": "Ah ok",
    "24": "yeah the xhyve driver doesn't have file sharing enabled right now",
    "25": "so /Users on your mac won't get mapped into the VM",
    "26": "Gotcha.",
    "27": "Ill see if it works with the virtual box driver",
    "28": "cool, it should work there",
    "29": "Thanks",
    "30": "vbox performance is a lot better still",
    "31": "Oh good to know. Then Ill just ran it there.",
```

FIG6: JSON DATA 4



```
    "123": 1468600817.000355,
    "124": 1468600834.000356
  },
  "type": {
    "1": "message",
    "2": "message",
    "3": "message",
    "4": "message",
    "5": "message",
    "6": "message",
    "7": "message",
    "8": "message",
    "9": "message",
    "10": "message",
    "11": "message",
    "12": "message",
    "13": "message",
    "14": "message",
    "15": "message",
    "16": "message",
    "17": "message",
    "18": "message",
    "19": "message",
    "20": "message",
    "21": "message",
    "22": "message",
    "23": "message",
    "24": "message",
    "25": "message",
    "26": "message",
    "27": "message",
    "28": "message",
    "29": "message",
    "30": "message",
    "31": "message",
    "32": "message",
    "33": "message",
    "34": "message",
    "35": "message",
    "36": "message",
    "37": "message",
    "38": "message",
    "39": "message",
```

FIG5: JSON DATA 3


topic_segmenter-master [C:\Users\hp\Desktop\topic_segmenter-master] - ...\Runner.py [topic_segmenter-master] - PyCharm (Administrator)
File Edit View Navigate Code Refactor Run Tools VCS Window Help
topic_segmenter-master
Runner.py
Runner
Project
Runner.py
ConversationSegmenter.py
SimilarTopicCalculator.py
ReplyObjectPredictor.py
JSONParser.py
Message.py
data.json
topic_segmenter-master library root, C:\Users\hp\Desktop\top
config
nlp-environment.yml
import sys
TestRunner
report()
for topic in topics
for message in topic.getMessage...
Run:
Runner
1: Project
Z: Structure
2: Favorites
SciView
R Graphics
R Packages
Database
4: Run
6: TODO
Terminal
Python Console
2 Event Log
No R interpreter defined: Many R related features like completion, code checking and help won't be available. You can set an interpreter under Preferences->Languages->R (43 minutes ago)
42:51
LF
UTF-8

```
107:    resulting in a bogus address


== Topic ==
108:    Hey guys, do you know if anyone has tried to run minikube from a windows workstation?
109:    sorry if that is a dumb question or if there is another place I should ask instead


== Topic ==
110:    <@U0K61JBKJ>: I am working on a PR to make it work properly.
111:    Hi
112:    can I switch virtualbox image network type to a LAN visible configuration?
113:    may be promiscue?
114:    i need to expose my cluster to the offe LAN
115:    *office


== Topic ==
116:    hey <@U12U3NGMT>, you only need a clean gopath to make godeps work right during the save/export cycle
117:    here's typically what i do
118:    1. make a clean gopath, put the minikube source in there
119:    2. use godep restore ./... to fill in the rest of the gopath from vendor
120:    3. "go get" the new import, and start using it in code
121:    at that point you can build stuff with just "go build" and "go test" to make sure it's all working
122:    4. run godep save ./... to import the new package from your GOPATH into vendor


== Topic ==
123:    <@U12U3NGMT>:  I'd be happy to help with that effort if you don't mind
124:    I'm quite interested in getting this to work from a windows workstation


[[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

Process finished with exit code 0
```

DATASET DESCRIPTION:

We have used the same dataset that is being provided to us along with the reference paper. 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”.

PLAN FOR THE REST OF THE SEMESTER

The work remaining are to refine the output in the best way possible. To do so, the following are planned to be implemented till the next/final submission:

1) To identify reply objects. There a considerable number of messages that are simply reply messages (e.g. 'Ok. let's do it', 'I agree', 'Mine is better', etc.) 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. According to the paper “Topic Detection in Group Chat Based on Implicit Reply” more than 90% of the messages are reply objects in a group conversation.

2) Complete Incorporate a message containing <@user_id> also as a reply message.

3) Introduce a dynamic window size feature to make the machine remember the last topic for better topic segmentation.

4) Understanding the dataset used and the different ways to process it.

5) Introduce 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.

These works are planned to be distributed in following manner:

1. **Rishabh:** Identify Reply Objects.
2. **Renuka:** Find Similarity distance.
3. **Soumi:** Incorporate dynamic window size functionality