**Summary: Information Bottleneck Inspired Method for Chat Text Segmentation**

The paper highlights the increasing importance of efficient methods to process chat text, in particular for **text segmentation**. It is useful in various fields like Decision auditing and dynamic responsibility allocation by project managers. Logs of such conversations can help to make automatic assessment of possible collaborative work among people.

Applications

Segmentation Methods

There were many different algorithms and methods to implement it, but due to following **challenges** in chat segmentation, none of the methods worked properly:

1. Informal Nature of text
2. Frequently short length of post
3. Significant proportion of irrelevant interspersed text

This paper proposes to use **Information Bottleneck (IB)** method. This is because it balances 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. Additionally, it helps to incorporate **non-textual clues** that arise in chat scenario, i.e., time between two consecutive posts & people mentions within the posts. The only **constraint** used in the paper is to allow only contiguous text snippets in a group, i.e., text should maintain **sequential continuity.**

The **Problem Statement** is:

“Let C be an input chat text sequence C = {c1, ..., ci, ..., c|t|} of length |C|, where ci is a text snippet such as a sentence or a post from chat text. In a chat scenario, text post ci will have a corresponding time-stamp cti . A segment or a subsequence can be represented as Ca:b = {ca, ..., cb}. A segmentation of C is defined as a segment sequence S = {s1, ..., sp}, where sj = Caj :bj and bj + 1 = aj+1. Given an input text sequence C, the segmentation is defined as the task of finding the most probable segment sequence S.”

The paper proposes an IB inspired agglomerative (Bottom-up) Text segmentation **algorithm,** which aims at maximising the balance trade-off between most informative segmentation of a variable & most compact representation of input chat text sequence. It involves agglomeratively merging an adjacent pair of posts that correspond to least value of ‘d’, where ‘d’ represents textual dissimilarity between a pair of posts to achieve optimal segment sequence. The same algorithm is being modified to incorporate non-textual clues.

The **dataset** can be described in following ways:

1. Chat text datasets – Slack & Fresco.
2. Collected raw data was in the form of threads, which was later divided in segments.
3. Multiple documents created, with each document containing ‘N’ (range in 5 to 15) continuous segments from original thread.
4. Take 60% of data as training and rest as test dataset.
5. Insert manual annotation to apply constraint.

The **output** helps to interpret our models, along with the comparison with other algorithms, and study the efficiency of the result. The results demonstrate that the proposed IB method yields an absolute improvement of as high as 3.23%. Also, a significant boost (3.79%-7.32%) in performance is observed on incorporation of non-textual clues indicating their criticality.

The **future work** involves incorporating semantic word embeddings in the proposed IB method.