# Mathematical modelling of the statistics of communication in social networks

Gibson Okechukwu Ikoro

October 11, 2017

Submitted for the degree of Doctor of Philosophy

Queen Mary University of London

**DECLARATION**

I hereby declare that the research work reported in this thesis is my own original work carried out under normal terms of supervision.

**ACKNOWLEDGEMENTS**

I would first like to express my profound gratitude to God almighty for the strength and grace he granted during the course of this arduous project.

I am particularly indebted to my supervisor Graham White for helping to conceptualise and shape the project, and for his general advice, counsel, support and encouragement. Thank you very much.

Special thanks go to my enthusiastic co-supervisor Raul Mondragon for his enormous contributions and insights, which have led to the successful completion of this work. Thanks also to Lawrence Pettit for his contributions and insights during the course of this project.

Many thanks go to SAS for awarding me a Graduate Research Fellowship and providing all the relevant SAS software products, which were very useful in obtaining the results of this work.

I specially thank my lovely wife Victoria Ikoro for her continual love and support throughout my study. Many thanks also to my special children- David, Alice and Emmanuel. Thank you so much for always being there.

**ABSTRACT**

Chat rooms are of enormous interest to social network researchers as they are one of the most interactive internet areas. To understand the behaviour of users in a chat room, there have been studies on the analysis of the Response Waiting Time (RWT) based on traditional approaches of aggregating the network contacts. However, real social networks are dynamic and properties such as RWT change over time. Unfortunately, the traditional approach focuses only on static network and neglecting the temporal variation in RWT which may have lead to misrepresentation of the true nature of RWT.

In order to determine the true nature of RWT, we analyse and compare the RWT of three online chat room logs (Walford, IRC and T-REX) putting into consideration the dynamic nature of RWT. Our research shows that the distribution of the RWT exhibits multi-scaling behaviour, which significantly affects the current views on the nature of RWT. This is a shift from simple power-law distribution to a more complex pattern. The previous study on users RWT between pairs of people claims that the RWT has a power-law distribution with an exponent of 1. However, our research shows that multi-scaling behaviour and the exponent has a wider range of values which depend on the environment and time of day. The different exponents observed on different time scales suggest that the time context or environment has a significant influence on users RWT. Furthermore, using the chat characterise, we predicted the factors which could minimize response waiting time and improving the friendship connection during online chat sessions.

We apply our findings to design an algorithm for chat thread detection. Here, we proposed two variations of cluster algorithm. The first algorithm involves the traditional approach while in the second one, the temporal variations in RWT was taken into consideration to capture the dynamic nature of a text stream.

An advantage of our proposed method over the previous models is that previous models have involved highly computationally intensive methods and often lead to deterioration in the accuracy of the result whereas our proposed approach uses a simple and effective sequential thread detection method, which is less computationally intensive.

**CONTENTS**

## LIST OF FIGURES

## LIST OF TABLES

# List of Abbreviations

**ER** ..... Erdos-Rnyi

**WS** ..... Watts-Strogatz

**BA** ..... Barabasi and Albert

**Gg** ..... category of graphs

**IRC** ..... Internet Relay Chat

**MUD** ..... Multi-User Dungeon

**RWT** ..... Response Waiting Time

**LSA** ..... Latent Semantic Analysis

**Avg.CC** ..... Average Cluster Coefficient

**Gp** ..... Generalised Pareto Distribution

**Gev** ..... Generalised Extreme Value

**Zm** ..... Zipf-Mandelbrot

**InvGaus** ..... Inverse Gaussian

**Exp** ..... Exponential

**Weib** ..... Weibull distributions.

**CCDF** ..... Complementary Cumulative Distribution Function

**TDPL** ..... Time at which the graph deviates from Power Law

**SRWTC** ..... Short Response Waiting Time Class

**LRWTC** ..... Long Response Waiting Time Class

**BIC** ..... Bayesian Information Criterion.

**AIC** ..... Akaike Information Criterion

**Allfitdist** ..... MatLab function

**VADER** ..... Valence Aware Dictionary and sEntiment Reasoner

**LIWC** ..... Linguistic Inquiry and Word Count

**ANEW** ..... Affective Norms for English Words

**GL** ..... General Language

**NN** ..... Neural Network

**NNT** ..... Neural Network with Topic

**SVM** ..... Support Visual Machine

**SVMT** ..... Support Visual Machine with Topic

**RBF** ..... Radial Basis Function

**MCR** ..... Misclassification Rate

**ASE** ..... Average Squared Error

**ROC** ..... Receiver Operating Characteristic

**CSS** ..... Detecting a Coherent Short Segment

**CF** ..... Content features

**PF** ..... Participant Features

**PFA** ..... Participant Features Adjustment

**SAS** ..... Statistical Analysis System

**TN** ..... True Negative

**FN** ..... False Negative

**FP** ..... False Positive

**TP** ..... True Positive

# List of Symbols

**N** ..... Vertices

**m** ..... Number of node

**r or L or E** ..... Number of links in a network

**p** ..... Probability

$n_i$ ..... ith of vertice

$\beta$ ..... Probability

$i, j, k, l$ ..... indices of nodes or links

$\alpha$ ..... Alpha

$\Pi(k)$ ..... probability

$k_i$ ..... degree of node $i$

**R** ..... a group of parameters

**Z** ..... a group of values.

$\deg^+(u)$ ..... out-degree

$\deg^-(u)$ ..... in-degree

$T_i$ ..... number of triangles (3-cycles)

**y(G)** ..... clustering coefficient

$\sigma_{st}$ ..... number of shortest paths

$\sigma_{st}(\upsilon)$ ..... number of shortest paths from $s$ to $t$

$\delta$ ..... a global statistic

$x_min$ ..... x minimum

# 1. INTRODUCTION

This thesis studies users' behaviours and their temporal dynamics in chat rooms and then proposes a less computationally intensive algorithm for chat disentanglement. We verify the method proposed by applying it to different real-world data sets and evaluating the results obtained. Chapter 1 gives a brief history of some early chat rooms and discusses the motivation for investigating the dynamics of user behaviour.

Chapter 2 shows related works that highlight the gap in previous research and the originality of this work. In chapter 3 we discussed the basics of social networks and defined some network terms. In chapter 4, we show that our chat room represents human interaction. We time-slice the network and study its evolution across years and quarters of years. Furthermore, we explore the temporal behaviours and the differences that occur in users behaviour as the network evolves. Chapter 5 investigates the temporal variation and dynamics of Response Waiting Time (RWT) of pairs of people in conversation. In chapter 6, we investigate which properties of a chat can be used to predict if a user has a fast response time. Chapter 7 focuses on the dynamics of a group of people in conversation. Unlike most models that involve highly computational intensive methods, such as clustering techniques, our proposed approach uses a simple and effective sequential thread detection method that is less computationally intensive. We presented two experiments to evaluate our system. The thesis closes with a summary and conclusion in chapter 8.

## 1.1 What is a Chat Room?

Chat rooms are where people can meet each other to converse on the internet. There are both public and private chat rooms. In public chat rooms, anyone can meet and chat with strangers while in private chat rooms people can arrange a time to meet people they already know, allowing only the speakers they want into the space. Today, chat rooms are used widely on many levels and by diverse organisations. In addition to its initial use for gaming and engaging in simple conversations it is now being used in education, for example, when receiving feedback on a project or assignment. Many groups in business, medicine and customer service now hold regular meetings using private chat rooms. Chat rooms are ideal for most group discussions because they are interactive and allow users to engage in live-real time conversations with people either far away or near and this can happen with multiple people at the same time. Chat rooms currently exist in a variety of formats. There

are “open” chat rooms where participants can discuss any topic they wish. There are also topic-focused chat rooms where users can discuss a specific topic like a TV show or a game. There are also moderated chat rooms where there is a speaker who can lead the discussions at any time. Those who provide internet services also create majority of the online chat rooms like AOL or through Internet Relay Chat (IRC). The majority of the chat room is text-based and when a user post a message, it will appear immediately on the screen. Online chats provide a way for people to have a conversation with each other by typing on their keyboards and what they posted will immediately appear on the other participants monitors [70].

### 1.1.1 The Early Years and Chat Today

Chat is one of the social networks that has grown over the years. In the 1960s, when the internet had not been developed, there was a system where only users connected to the same computer could chat with each other [22]. The system operation required that users connect to the same system, and only two users were allowed by the chat program to converse at a time. At that time the “talk” function on the Unix operating system was one of the popular early chat utilities and supported chatting on a single multi- user computer [70]. Over time, “talk” grew such that chatting was then allowed across multiple computers. Expansion of the talk system gave room for more than two users to chat in a limited-broadcast way. It required the sender to add all of the recipients addresses. However, since the advent of fast computers, chat has improved significantly. In addition, chat rooms have undergone a series of developments as the internet evolves. An example of online chat rooms include Multi-User Dungeon (MUD Servers), Internet Relay Chat (IRC), Web Chat, Instant Messaging (IM) and Voice Chat.

#### MUD Servers

In 1978, Roy Trubshaw, who studied at the University of Essex, UK, developed MUD Servers [70]. He developed a computer algorithm that permitted individuals to be connected to a fantasy-based game from their home computer[70]. The name of the game was MUD (Multi-User Dungeon) and this was in tribute to the dice-based game Dungeons and Dragons that he enjoyed playing. As time went on Trubshaw's first MUD was no longer confined to his network of friends and acquaintances. Many external users were inspired to develop their version of the program. By 1994 MUDs on the internet had increased to number about 400 and their topics had extended from gaming to general conversations organised by groups and associations where people talked about any general interests. In the past, MUDs were used mostly by those who were very comfortable with computers and served as the first online chat rooms. As the MUDs developed, this gave rise to Internet Relay Chat (IRC).

### Internet Relay Chat (IRC)

IRC was developed in 1988 by Jarkko WiZ Oikarinen at the University of Oulu, Finland. The birth of IRC initiated the widespread use of real-time chat with large groups of people all over the world [70]. However, in its early years, the IRC only occasionally had more than 10 users but gradually gained great popularity around the world in 1991, when a lot of users logged on to access the latest news on Iraqs invasion of Kuwait. This was done by connecting to a particular country that remains operational via IRC functions when radio and television broadcasts were cut off [70].

The documentation of the Internet Relay Chat Protocol was first carried out in May 1993, in RFC 1459 [IRC]. In a group discussion, it was considered as a set of rules for communication sometimes called channels, but also allows one-to-one communication privately, such as chat and data transfer and file sharing [70].

As the IRC network grew and evolved, users began to split off to form their own networks. New networks aimed at adding new features such as proxy detection, additional commands, encryption devices, etc. The first early major split occurred in 1992 and was known as the Undernet network. Today, many other little networks have been developed mostly through modification of versions of DALnet, EFnet, IRCnet and Undernets IRCd. The current largest networks include: Freenode, IRCNet, Quakenet, Efnet, Undernet and Rizon.

### Instant Messaging Chat Rooms

Instant Messaging (IM), as the name suggests, allows real-time conversations between two or more people who send quick text-based messages using their clients software from their computers or other devices [70]. Recent IM also allows the sending of video, audio and picture messages [70] but, for this work, our focus was on text-based messaging. Unlike earlier communication technologies such as MUDs, other public chat rooms in the 1990s were mainly used for communication between unfamiliar people. IM and other social networking sites like Yahoo, Facebook, WhatsApp etc. encouraged people to communicate more with existing friends [56].

## 1.1.2 Role of Chat

The internet provides a major means of human communication and as it grows, chats with friends, family, colleagues and even complete strangers have increased rapidly. There was a time when a family who lived a long distance apart could only communicate with their loved ones through letter writing but with the arrival of the phone human communication improved; however, large phone bills for both parties became the big challenge. Since the advent of chat rooms which only require that the user has a network connection, staying in touch with long distance families, friends and loved ones is much easier and more reliable than before. Advanced chat rooms are now supporting video or voice

conferencing, thus permitting people or families to meet online at the same time with no or less cost to either party, except for maybe a microphone or webcam. In addition to strengthening relationships and friendships, the dynamic changes in social group membership and communication patterns have attracted great attention due to its significant impacts on industry and society [70]. In the business sector, chat is one of the best tools that can foster company-customer relationships. Many companies have installed company-wide messaging service using computers. To stop using up too much time on chatting, companies have developed security systems that deactivate chat outside of a company network. As a result, employees will spend less time chatting with non-employees during work hours. Operators are today in a key position as they can deliver services as well as maintain relationships with customers. To reap rewards, operators must maintain their current privileged position and provoke innovation in the kinds of products and services that will create value for customers and thus lead to continued revenue growth [56]. All of this can be achieved by understanding the variations and fluctuations in the behaviour of their customers conversation patterns over time. Similarly, this can enhance strategic thinking and innovation. Some of the areas we have seen huge impacts in are the marketing industry, banking and telecommunication companies. Marketing managers are often keen to ascertain their customers views or opinions regarding their services. Through assessing timely conversation patterns, they can detect when a customer is switching to their competitors. On the other hand, bank managers want to identify when a customer is on the edge of leaving for another bank while telecommunication companies are doing everything possible to locate customers who are switching from one product community to another within their product categories. Capturing the real fluctuation and variation using customers time varying communication patterns as well as knowing the nature of their communication change will offer a better solution to varying industry sectors [56].

Chat room communication was originally designed with good intentions such as information exchanges and fast information spreading, which aimed to build positive and informed societies. Unfortunately, sometimes chat rooms are misused for illegitimate information exchange. It has been converted into a tool for committing crimes because of its anonymity and totally unrestrained chatting environment [56]. Crimes such as sexual solicitation, online bullying, sensitive and confidential information stealing or leaking, especially for children and youngsters, terrorist contacts and discussion (which pose a great danger to the safety of society) may be reduced by understanding and monitoring how chat room users behave online. Discovering the characteristics of users during online chats may help to improve the safety of chat rooms as well as create better policies to reduce the number of chat room abusers [37]. Lastly, in turn this will build the confidence of users, chat room providers and keep our societies safer at large [64].

### 1.1.3 Motivation

Chat rooms are of enormous interest to social network researchers due to its many uses and applications such as in on-line education, customer services, academic collaborators, General Practitioners and patient consultants. As one of the most interactive internet areas, users behaviour has an effect on the whole structure of a social network. This is important because it provides information about the dynamic process taking place in a real network. In the real world, a person plays a role each time there is an interaction and this role strongly shapes their behaviour, which in turn affect the network structure. The structure of a network is the result of user activities or behaviours.

To understand the behaviour of users in a chat room, there have been studies on the analysis of the Response Waiting Time (RWT) based on traditional approaches of aggregating the network contacts. However, real social networks are dynamic and properties such as RWT change over time. Unfortunately, the traditional approach focuses only on static network while neglecting the temporal variation in RWT which may lead to misrepresentation of the true nature of RWT. Hence our motivation.

### 1.1.4 Research questions

The following questions have been raised as a result of notable gaps in the literature relating to analysing of Chat room logs:

***What is the structure of our chat rooms?***

A Chat room consist of users who send messages to each other. We can model this with a network in which the users are represented by nodes, while links or edges represent the messages. The degree of a node in a network is the number of connections it has to other nodes.

The network represents the interaction between all the users in the whole period of our study. Beside the direct link between users activities and network structure, understanding the social structure hidden behind chat rooms can facilitate the building of a robust model of the flow of information, which could be applied in developing a computer learning tool to support and evaluate the transfer of knowledge between users. It could also be useful for effective message filtering i.e. separating users chatting into different groups. Moreover, understanding of the network structure may enhance the techniques use for interfering of any abnormal or unexpected behaviour in a network.

***How can we best describe the distribution of response waiting time in a chat room?***

Previous studies on response waiting time by Barabasi and others claim that the distribution follows power law [8]. We want to see if it is in fact a power law and if it is the same for everyone. We focus our study on three different chat rooms: Walford chat logs, IRC chat logs and T-REX chat logs.

***What are the dynamic main characteristics of RWT?***

We give a detailed study of the RWT considering the following:

- The impact of number of message exchange on RWT.
- Dynamism of a single user's RWT with other users at different chat sessions.

***Is there temporal variation in RWT?***

Users behave differently at weekdays and weekends due to many factors, for example, working on the weekdays and having more leisure time on the weekends. Is this reflected in the response waiting time of a chat room?

***Is the temporal variation due to the chat room technology?***

Are the differences due to various technologies (different chat rooms) in existence?

**How can we disentangle chat using a computationally less intensive method?** Disentanglement is a task that extracts the different interposed utterances in a chat log and separates them into distinct conversations. Most disentanglement models involve highly computationally intensive methods such as clustering techniques, fuzzy algorithm, etc. These methods often lead to deterioration in the results accuracy.

**How can we dynamically model real social network?** Social interactions are dynamic which often experience tile decay and formation. Also in social networks, nodes enter and exit, interactions between pairs of people or groups are bursty as a result of long dormant periods separated by strong bursts of activity. One of the challenges in chat disentanglement is the temporal variation and dynamics of the network system.

### Aim and Objectives

The aim of this project is to understand the on-line chat room characteristics and study the disentanglement of conversations in real time. In order to achieve this aim we propose to:

- Analyse three chat logs: Walford, IRC and T-REX logs.
- Extract the message contents and other on-line chat room characteristics such RWT, Time stamp etc.
- Use the power law technique to explore chat room characteristics.
- Utilise the on-line chat room characteristics to predict the RWT during on-line chat
- Use the on-line chat log properties to design an algorithm for chat disentanglement.

### 1.1.5 Methodology

This section outlines and summarises the approach I used to carried out my research. Starting with the dataset extraction, the analysis and the findings. We utillsed three chat room logs, which had

already been collected in our research: Walford, IRC and T-REX. The dataset were made available for us from different sources.

Walford was a text-based online social community that had roughly 2446 regular users and the number of communications or edges was 37,981 for four years [38]. It was a corpus that contained 24,040 hours (2001-2004) of chat. In the corpus of chat logs, the following data was recorded: unique numeric IDs for the time, the originator, the originators location, the recipient(s) and their location.

The IRC chat transcript dataset was described in [26] and provided on web(http://www.ling.ohio-state.edu/melsner/). The chat format include: speaker name, recipient name, comment or action and the times which are given in seconds.

The T-REX chat log was generated as part of the T-REX field project and was managed by the Earth Observing Laboratory (EOL) at the National Center for Atmospheric Research (NCAR). These chat logs were reviewed by EOL and edited to maintain the most accurate records for the project.

### Material Analysed

The dataset were in the form of a transcript of the conversations. For each person to person communication or person to many communication, the following data is extracted: the basic command type; unique numeric IDs for: the time, the originator, the originators location, the recipient(s) and their location. This extracted record formed the dataset with the following column heads: chat participants (Sender and Recipient), Timestamp (Day-Month-Year, hour:minute:seconds), chat messages (The content of the chat messages)

One of the chat room characteristics that we calculated from the above dataset is message Response Waiting Time (RWT). The waiting time in a chat room communication is the time difference between successive messages between two people.

```
1
2  40:29 A→(B):grins I think it's the proxy
3                Kevin and Perry that need kicking!
4  40:55 B→(A):what happened last night..the
5                lot of it got or needed a kicking!
6  41:13 C→(D): lsaysl cH kissing bandit...l
7  41:45 H→(I):Kissing bandits are  predators
8                 should not be tolerated
9  41:46 A→(B): it was a Janet router that went,
10               second tie in a week one has died
11 42:08 D→(C):lsaysl cYou're just jealous he
12               took your job
13 42:16 B→(A):grins janet is the of the network
14               the universities and schools are on.
15               router is something that forwards on
16 42:21 I→(H): And I haven't gotten any action since
```

*Fig. 1.1:* Sample of a conversation from our corpus.

For example, in Figure 6.1, three pairs of conversation are going on: $A \rightarrow B$ and $B \rightarrow A$, $C \rightarrow D$

and $D \rightarrow C$, $H \rightarrow I$ and $I \rightarrow H$. The response waiting time distribution between the pair of people A and B is 29 seconds $(40:55-40:29)$, 51 seconds $(41:46-40:55)$ and 30 seconds $(42:16-41:46)$. The response waiting time between the pair of people C and D is 55 seconds $(42:08-41:13)$. The response waiting time between the pair of people H and I is 36 seconds $(42:21-41:45)$. Then, we applied the maximum likelihood method to estimate the power-law scaling parameters for Walford chat room logs and T-REX chat room logs.

### Analysis

My aim in this project is to understand the on-line chat room characteristics. We first explored the dataset to show that chat room represents human interactions. We investigated the dynamics of pairs of people in conversation with respect to their Response Waiting Time (RWT) and this was done by fitting a power law distribution to determine the true nature of the RWT . We also investigated which properties of a chat room can be used to predict the response waiting time.

Focusing on the dynamics of group conversation, we proposed a simple and effective technique that utilised simple statistical information, such as utterance similarities, RWT, turn-taking and the participant-based feature for thread detection in chat logs. This supplemented the traditional qualitative method, which has been proven to be difficult due to the contextual nature of meaning.

### Scope

One of our limitation was that the datasets from the three chat room were not the same size. Walford was the largest while T-REX was the smallest. This may have had an effect when we compared the results from the three datasets. Also since the structure of the three chat log varies from one chat room to another, the model performance tend to favour one over the other.

Another limitation we encountered was the presence of schism. Schism occurs when a conversation splits into two conversations; the new conversation is formed due to certain participants branching of from a specific message and refocusing their attention upon each other. Schisms impose serious difficulty in identifying conversation threads. There are two ways in which new conversations can start: one is through a schism and the other is through a conversation initiating statement. Disentangling chat logs in the absent of a schism may improve model performance and yield better results.

# 2. LITERATURE REVIEW

With the emergence of the internet, communication within and between communities has improved tremendously [56]. In the last decade, there has also been a significant rise in the number of social networks and an increasing number of users. In turn, the impact of these changes has affected the effectiveness and efficiency of both theoretical and practical communication in communities [56]. These changes include: providing an adequate platform for connecting and reconnecting people, an opportunity to produce and share content with others, and extracting and processing group knowledge with feedback. Facebook, Twitter, MySpace, LinkedIn, Flickr and Foursquare are the most common networks and have enhanced communication and the diffusion of information into communities, while sites such as Yahoo and MSN Web Communities provide a directory of chat sites called chat rooms [70].

Many authors have presented studies on patterns of communication and information spread in social networks by examining data collected from them. Most of the studies on chat room networks have focused on time-independent variables and thereby lose the natural properties that time induces as the network evolves. However, the existence of natural and technological applications have made it possible to capture a huge amount of data that is time-dependent from communication systems [35]. The time-ordered sequences of networks are characterised by time-stamped information so that network parameters such as: when and who sent the message, when and who received the message, the delay in response to the message, content structure and duration in group conversation arises naturally in the evolving communication network. With the help of advanced communication systems and cutting-edge technologies, we can analyse social networks to obtain full information on the duration and time occurrence of each link [35].

## 2.1 Waiting Time

As a result of the rapid increase in the use of social networks, computer scientists and sociologists are beginning to investigate their properties. The three major aspects that researchers explore on a social network are structure, content and user behaviour [49]. Barabasi [8], in his research (The origin of bursts and heavy tails in human dynamics), argues that the waiting time distribution for email communication between many users exhibits a power-law. This contrasts with the first-come-first-serve tasks, which are random and may result in uniform dynamics that are Poisson distribution in

nature. In an effort to back up his argument, he conducted a series of studies on email communication between many users. The dataset captured the sender, recipient, size and time details of each e-mail. To determine the waiting time, he noted the time an e-mail arrived to a user and the time the user responded; the time difference between these two occurrences gives the waiting time. His results showed that the time taken by the participant to respond to an email received is best described using a power-law model in a form of $P(x) = x^{-\alpha}$ where $\alpha = 1$, which signifies that the e-mail pattern for an individual has a bursty non-Poisson distribution behaviour(see section 3.1.4). According to the author, since most of the human activity patterns conform to non-Poisson statistics, this indicates that the bursty properties show some significant and common features, which characterise the continuous change of human activities. He suggests that the bursty character of human dynamics occurs as a result of a queuing process, which has human activities as its driving force. Finally, the author stated that the timing of most activities that humans undertake such as communication can be described as bursts of events that occur in quick succession with long periods of inactivity existing in between [61]. It appears that Barabasi jumped to conclusions as there are a lot of distributions that are non-Poisson but are not exponential either. Our work is important because it contribute significantly to the understanding of the true nature of the RWT. Even Ahn et al. [4] in their research have also shown that these distributions are not Poisson but not power-law either.

Ahn et al. [4] examined the structural properties of social networks by analysing and comparing the structure of MySpace, South Koreas Cyworld and Orkut (since closed), which are social networking services. They conducted a snowball sampling of the network (this means to randomly select one seed node and perform a breadth-first search; this is done until the total nodes selected equal the required sampling ratio. Only the edges that exist between the selected nodes are taken into account during the final stage of the network sampling). This sampling approach has been found to preserve the power-law nature in the degree distribution (the degree of a node in a network is the number of connections it has to other nodes). They started the analysis of the network characteristics by examining their power-law degree distribution, which can be described using:

> $P(x) = x^{-\alpha}$ where $x$ is the node degree and $\alpha < 3$, authenticates the existence of a fairly small number of nodes that have a large number of links.

They also examined the clustering coefficient, which represents the ratio of subsisting links over the total possible or potential links between its neighbours. In addition, the authors examined the degree of correlation $K_{nn}$, which they defined as.

> "a mapping between a node degree $k$ and the average degree of the closest neighbours of those nodes of degree $k$ [4]."

Analysing the Cyworld network shows that their degree distribution has multi-scaling behaviour:

a region which is heavy-tailed with $\alpha = 2$ and a region which decayed exponentially. The observed multi- scaling behaviour suggests that Cyworld has two distinct types of users. Different scaling regions were also observed in the cluster coefficient and degree correlation, indicating a mixture of different types of users. However, Orkut and MySpace exhibited simple scaling behaviours that had different exponents. In another large-scale study, Ahn et al. examined data collected from four widely used online social networks: YouTube, Flickr, Orkut and LiveJournal. The author confirmed the existence of a power-law in Flickr, YouTube and LiveJournal datasets while the Orkut data deviated significantly [56].

They reported a significant diameter and short path for the social networks mentioned above (i.e. small-world). Analysing the joint degree distribution, which is an approximation of the degree correlation function, they show that vertexes with a high degree are more likely to relate to other high-degree vertexes in all networks except for YouTube. $K_{nn}(k)$ is the average neighbour degree for the nodes with degree $k$, and is used to measure correlations in power-law networks. It is observed that, if $K_{nn}$ increases, the vertexes with a higher-degree are more likely to relate to other high-degree vertexes, while on the other hand if $K_{nn}$ decreases it features an opposite trend. Furthermore, they observed significant scale-free behaviour in all the datasets except YouTube. Scale-free metrics are computed from the joint degree distribution of the graph and are concerned with the degree to which the network possesses a hub-like core. The metrics value ranges from 0 to 1, where the value 1 suggests that nodes with a high-degree are more likely to relate to other high-degree nodes, and 0 indicates that nodes with low-degree are more likely to be associated with other low-degree nodes. Lastly, they observed a similarity between the in-degree of user nodes and that of out-degree.

Researchers have also investigated the content generated by users over time, which may aid in discovering knowledge. For instance Althoff et al. [5] carried out a study on how topics evolved across three online media streams. They started by crawling the top threads everyday, from the major online media channels across multiple channels, with the purpose of capturing peoples communication needs and searching for patterns. Retrieving threads from Twitter, Google and Wikipedia, they cluster similar threads across time and channels. Their results suggested that, when getting information about current events, Twitter is not the only social media to turn to, there are other sources one can get information about the current events, though this depends on individual requirements. Lastly, the author proposed a novel forecasting approach that put together a time series from many semantically identical topics, exploiting the notion that semantically identical topics will show similar behaviours.

Another area of study that has attracted many researchers attention is the users' behaviour in online social network. This is of interest because users' behaviour has impact on the structure of a social network and can be utilised to describe the users themselves at best [49]. For instance, users' activities on Twitter have been studied by Lerman et. al. [44] and Comarela et. al. [17]. The

authors carried out an empirical analysis to examine the effect of a users' activities on spreading patterns. They show that the variation that occurs in the exponent of a power-law indicates our aptitude to maximise the transfer of information and the cost of communication enforced by the handicap of the human brain [33]. In summary, they represented a communication system in the form of mapping signals to stimuli, assuming that signals can be roughly compared to words and stimuli are the fundamental tools for determining the meaning of a word. They assumed that the signal connects to stimuli, as the words are associated with the activation of different brain areas. For instance, nouns have the tendency to activate visual areas, while verbs which are "doing" words, might activate motor areas. The region activated is associated with the diverse kinds of stimuli experienced with the word.

A recent paper by Lim et al. [49] explored user behaviour on several social networks such as Google+, Flickr, Instagram, Twitter, Tumblr and YouTube with the goal of uncovering behavioural trends in multiple network use. They presented the analysis of each users profile, which explored how descriptions differ across multiple Online Social Networks (OSNs). Furthermore, they explored the message post-time using time of day and day of week analysis in order to understand how users sharing-behaviours vary with time. Lastly, they conducted cross-network interaction analysis, mapping how users post from one source network to a sink network. The authors show that users exhibited diverse behaviour on the different OSNs.

The study by Giovanni et al. [17] shows a detailed characterisation of response waiting time on Twitter. They investigated the time difference between when a user receives a tweet and when it is replied to or retweeted. Results suggest that the time a tweet waits before it is replied to or retweeted seems to have several types of scaling and shows a sharp drop off near 108 seconds.

This indicates that the RWT for Twitter replies or retweets do not possess pure power-law tails. The authors show that nearly all messages will wait more than 100 seconds before they are replied to or retweeted. In addition about 90% wait for up to 1000 seconds. Also, the authors claim that the main factors that affect participants response rate or the probability of retweeting include the tweeters sending rate, earlier replies to the same tweeter, the age and some basic wording of the tweet [17]. Their method is a combination of two types of machine learning: a Support Vector Machine classifier and a Naive Bayes predictor, which was used for tweets ranking, showing the possibility of reordering tweets with the aim of increasing the fraction of replied to or retweeted messages.

Moreover, Acar et al. [1] studied the modelling and multi-way analysis of chat rooms. The authors believe that chat room communication data provide valuable information to study the evolution of social groups in cyberspace as well as the changes of social group membership over time. They also argued that chat room communication data, although gotten from streaming real-time communica-

tion, still has information that is full of noise and multidimensional. They further reasoned that the conventional methods such as Singular Value Decomposition (SVD), which depend on a linear relationship, may not be applied to multidimensional data. Hence, they proposed multiple dimensions techniques that capture the different facets of chat room communication. However, it is obvious that there is no clearly defined approach that universalises SVD to a higher dimension [1], therefore using this method to model multidimensional data like chat room data may return unsatisfactory results and may not be reliable. As a solution to the limitation of SVD, Evrim et al. [1] proposed a multiple dimension technique to capture many aspects of chat room communication by constructing a multi-way data array, which is called high order tensors.

Next we will review some previous work on power-law because of its valuable mathematical properties and its applications to a broad range of both natural events such as floods, volcanic eruptions, earthquakes, tsunamis and man-made events such as cyber attacks. According to Newman [59],

> "When the probability of measuring a particular value of some quantity varies inversely as a power of that value, the quantity is said to follow a power-law, also known as Zipfs law or the Pareto distribution."

Researchers have developed several models to identify the occurrence of a power-law in communication traffic. Michalis et al. [32] did a study of power-law relationships of internet topology. The authors suggested a formulas which links power-laws exponents with the number of edges, the number of nodes and the average neighbourhood size. They also proposed the power-law exponents as an efficient way to describe the highly skewed graph metrics instead of averages.

Clauset et al.[16] in their study of the power-law distribution in empirical data claim that the standard practice of detecting and evaluating power-law distribution by fitting a straight line to a log-log plot in order to estimate the exponent is not a reliable method. In the authors words "it is not straightforward to say with certainty whether a particular data set has a power-law distribution". The authors have a procedure for estimating the parameters of a power-law description, which we will follow, and we will describe it in detail in Chapter 3.

Also, Barabasi et al. [10] studied the competition and multi-scaling in evolving networks. They claim that the rate of increase in node connectivity within a network is based on their ability of the node to contest for links. The authors believed that the different fitness translates into multi-scaling in the dynamic evolution and that the time dependence of the nodes connectivity depends on the fitness of the node. It is clear, for the last decade that power law has been applied in modelling, such as in the topology of the internet. However, little or no effort has been made in identifying and characterising communication chat room data using power-law, and this remains a challenge that this proposed research seeks to solve. Kwak et al. [43] presented an extensive study on characterising Twitter and its properties as a new means of transferring information. The author claims that the

followers distribution is a non-power-law. They also reported a low reciprocity and a short diameter. These characteristics show many anomalies when compared with the already known characteristics of human social networks. In their methodology, they used a "queuing model with a priority discipline" which presumes that each person prioritizes different activities and select an event with the highest priority for execution.

In their paper titled "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations" [45], these authors studied how networks evolve over time by examining a small portion of a network at regular time interval. Using seven data sources which contain a column that describes the time each node was added, they were able to create at any point in time a snapshot of the network. The authors found that in contrast to previous standard studies the mean out-degrees in a network is not constant over time rather they grow in a natural pattern. They also found that across a range of networks, the network diameter tends to decrease slowly as the network grows. To further understand what might cause a network to densify and also possess a shrinking diameter the authors propose two models which they call the Community Guided Attachment Model and the Fire Forest model [45]. In the Community Guided Attachment model, they argue that graph densification is based on nodes decomposing into nested sets of communities such that creating connections becomes more challenging between communities as the network grows. The Fire Forest model is more complex in the sense that it shows the densification as well as a decreasing effective diameter as the network enlarges. The process involves adding new nodes through the existing edges to the network in a widespread manner.

In addition to their findings in [45], the authors latest research in [46] found that the Forest Fire Model shows a sharp change between scanty networks and dense graphs. Graphs that have a shrinking diameter are observed within the sharp changing point. Finally, they noticed that the time dependent evolution of the graphs power-law degree distribution and the densification power law exponent have a basic relationship.

## 2.2 Techniques for Multi-participant Chat Analysis

There are various methods that can be used to analyse multiparty chat room logs. Figure 2.1 shows a topology of these techniques and we can split these methods into two levels: Low-Level Analysis and High-Level Analysis [71]. Low-Level includes Chat Pre-processing, Chat Room Feature Processing and Disentanglement of Chat Threads. High-Level consists of User Profiling, Message Attribute Identification and Automatic Summarisation, Topic Detection and Social Phenomenon Detection.

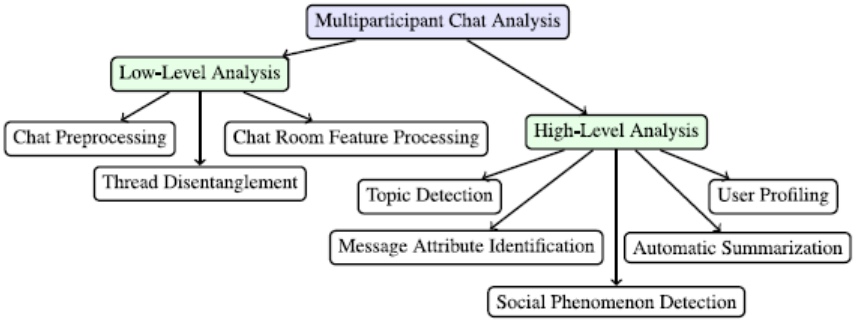



*Fig. 2.1:* Techniques for Multi-participant Chat Analysis. Source:[71]

### 2.2.1 Chat room feature processing

As stated earlier, chat room feature processing is often used to discover associations that exist among posted text. These features could be time stamps [3, 27, 28, 29, 58, 14], nickname augmentation [3, 27], name mentions [27, 28, 69], and speaker identity [28]. The applications of feature processing can be seen in thread disentanglement, where word similarities aid in determining distinct chat conversations and detecting the topic of discussion. In addition, the cohesion of messages can assist in deciding on their general topic. For instance, in order to determine if two utterances belong to the same conversation in thread detection, Elsner [28] suggested in their paper that time stamps and identities of the speakers are better cues than the contents of the messages and stated the following for utterance disentanglement:

- Penalising the time gap distance is vital in finding the relationship among utterances: The farther apart in time two utterances are, the less likely it is that they belong to the same temporal conversation and, the closer they are, the more likely it is that the two utterances are related.

- Generally, name-mention are often seen at the beginning of chat messages and can serve as a connection between utterances. Mentioning names is useful in finding threads of conversation and some participants make use of them more than others.

- In some conversations, only a few users speak while the majority are recipients. In such environments, the speakers identity can help in disentangling of the chat thread since only a few core speakers who are speaking frequently dominate the conversation and these speakers may be in the same conversation.

These features above are very important in providing extra knowledge for analysis beyond the message composition.

### 2.2.2 Thread disentanglement

Multi-party conversation is different from the groups spoken communication in the sense that it lacks some characteristics we see in spoken dialogue such as turn-taking [71]. The absence of these features results in threads becoming entangled such that interpreting individual threads may be misleading. Moreover, initiating and responding to messages further entangles threads; for instance, many users can respond to a speaker at the same time, and one user can respond to many users at the same time. It has also been mentioned that users can start a new thread that is unplanned and this rarely happens in spoken group conversation [71]. Disentangling conversation chat threads is important because it makes future analysis techniques much easier. After the disentanglement process, the focus will be on single conversations. It also makes it easy to apply popular Artificial Intelligence methods such as topic detection and automatic summarisation to a single conversation. One of the common methods used when studying thread disentanglement is the clustering method [71, 26]. In a recent study, Elsner [28] employed coherence models to investigate chat disentanglement. They validated their models using recorded telephone conversations for thread disentanglement. In their method they utilised the tabu search method to arrive at a solution to this problem; this involves conducting two stages of experiments with different chat corpus. The first experiment involved disentangling single messages and the second involved disentangling the entire chat log.

Another interesting method of chat disentanglement is described in [29] and [27]. Their work utilised correlation clustering for thread detection. The method involved searching for a group of clusters that maximise the degree of similarities between pairs in a cluster and the level of dissimilarity among pairs across clusters. The two models employ a maximum-entropy classifier to decide whether an utterance belongs to a given conversation [71]. In another recent work, Mayfield et al. [53] utilised a two-pass method for thread detection. In the first pass, the method labels sentences using a negotiation framework. After the labelling process, a single-pass clustering algorithm is used to detect sequences.

The approach used by Shen et al. [68], Wang and Oard [74] and Wang et al. [72] was a Single-Pass Clustering Technique (SPCT). The SPCT procedure involves holding the first utterance as the first cluster which allows each subsequent utterance to be allocated to an existing one if the utterance is similar to the cluster and surpasses a certain threshold or else it allocates them to a new cluster. To assign an utterance to a given conversation, the authors first used a vector space model with term frequency-inverse document frequency (TF-IDF) to depict the utterances and employed the similarity of the vectors for each utterance as well as the sentence types and personal pronouns. In another research, Wang and Oard [74] appear to employ a similar approach using a clustering method but utilising a slightly different approach. The authors approach is in two stages: in the first stage, their method involves single-pass clustering, as described above. The second phase involves renovating

the links between utterances based on the new utterance by going through all the conversations. This permits utterances to be re-classified assuming that the cluster which holds the new utterance is strongly linked with a given utterance.

Another approach used by Elsner and Schudy involved greedy techniques and local searches for the NP-hard problem of discovering a better solution for cluster correlation. However unlike the previous single-pass, their method makes multiple passes with random permutations and the authors report suggests that using the greedy method followed by local searches resulted in the highest performance. The voting schema technique was employed by Elsner and Charniak [27] as well as correlation clustering. In their approach, messages are processed incrementally as they were received. Wang et al. used parent- child associations that exist among utterances and represented it with connectivity matrices. Their techniques are in two stages. In the first stage, they computed the similarity matrix of the messages using standard TF-IDF term vector representation while, in the second phase, the author builds a direct graph of the utterances and relations can only exist if utterance similarity surpasses a threshold value. The author developed four approaches - one is a baseline, and the other three are just an extension of the baseline, penalising the time-distance between messages with different techniques. Camtepe et al. [11] presented an approach which claims to have found every subgroup conversation that exists in that particular network. Their algorithm depends mainly on the statistical information contained in the sequence of posts. In their model, they assume that:

- The number of subtopics is predetermined at the beginning
- The overall number of users in the channel is unchanged
- Each user has a life span within the channel.
- Each user randomly picks a subtopic at arrival within the channel and stick to it throughout the life span
- Only one user is permitted to send a message within a subtopic at a time
- A user who is picked to send a message will be allowed for message inter-arrival time before submitting the message.
- The size of the content of the messages is not fixed and message inter-arrival times are not fixed as well rather these are at random, at a given distribution and mean
- All the messages posted are collected from each subtopic, combined at any time and intermix randomly

Firstly, because it is the speakers that are clustered, not the utterances, the algorithm fails when speakers shift from one conversation to another. Secondly, they used a chat feature such as a time gap between utterances and turn-taking behaviour, however, speakers are not allowed long gaps or to cause an interruption. However, in reality, speakers can interrupt or leave long gaps, group conversations can be interrupted by other conversations, and two or more group conversations can occur at the same time. These chat room behaviours were not taken into consideration by the authors. Moreover, they did not use standard corpora.

Camtepe et al. [11] conducted research on data collection and analysis of a chat room using SVD technique. Their primary aim was to test the challenges of the n-way data analysis method and see if there is a relationship between the way data is collected (i.e. tensor construction) and how these techniques work. Their model is based on five parameters:

> "(i) distribution for the inter-arrival time, (ii) distribution for size of the message (number of words), (iii) distribution for the number of messages per user, (iv) noise ratio (NR), and (v) time period."

As a part of an effort to provide solutions to these challenges, Acar et al. [1] clustered speakers by using a fuzzy algorithm approach. However, the algorithm focuses on the speaker and does not include the utterances made in decision-making. As a result, it may be hard to know or classify the utterances to different conversations. In addition, several threads of research attempted to develop programs to solve the disentanglement problem. PieSpy [58]is an IRC program that gathers chat room messages and visualises social networks. It is based on simple heuristic rules that decide the origin and destination of the messages. These rules involve direct addressing, i.e. writing the nickname at the start of the message; however, it is not uncommon for the destination nickname to be used in a post. There is another method called Temporal Proximity. After a long interval of inactivity, if a participant posts a message which is at once followed up by a post from a different participant, then we can assume that the second message was a reply to the first one. Furthermore, a method called Temporal Density is often used in cases where Temporal Proximity cannot be applied. In this case, if within a stipulated period of time many messages have been sent and these messages have come from only two users then we can assume that there is a concrete conversation going on between these two participants. However, this method cannot be used when there are multiple conversations occurring simultaneously.

### 2.2.3 Topic detection

Topic detection may be difficult due to the dynamics of chat as multiple topics often occur at the same time and change over time. Topic detection can be used to know if users are staying on topic. It can also assist chat participants to discover the particular chat room they are interested in. Users can

match their interests by randomly sampling the open chat rooms, word construction vectors of the rooms, and then make a selection based on their preference. Lastly, it can support in creating user and chat room profiles. Elnahrawy [25] applied techniques such as Nave Bayes, k- Nearest Neighbour and Support Vector Machines (SVM) to newsgroup postings and chat logs, with an aim of discovering pre-determined topics in a log. The authors claim that the Nave Bayes classifier outperformed SVM with highest accuracies and shorter training times. Also, the classification time was longer in k-Nearest Neighbour than Nave Bayes. Similarly, zyurt and Kse [63] focused on a Turkish language chat log, applying the same techniques as Elnahrawy [25] for topic detection. However, their classification was based on individual utterances, unlike that of Elnahrawy [25] approach. The authors claim that SVMs yielded the highest accuracy compared to the other two classifiers. Anjewierden et al. [6] focused on an educational chat room made up of students collaboratively learning and classified their utterances into groups using the Nave Bayes classifier. Before they trained and evaluated their classifier, the authors combined manual and automated methods to limit noise, such as misspellings, in their chat data. Durham [23] looked at binary classification for chat messages using the Latent Dirichlet Allocation model as a featured vector for SVM. The authors argued that their method performs best if all messages sent by users are considered as a single document.

# 3. FUNDAMENTALS

We divided this section into two parts. Part one deals with the network structure while part two is concerned with the chat disentanglement.

## 3.1 Social networks

Social networks denote relations among social entities [70], for instance, communications among members of a group or economic relations between corporations. Social network analysis emphasises the structural analysis of networks in order to explain social behaviours [37]. The methods are widely used in the social and behavioural sciences but have also gained application in economics, marketing, banking, telecommunication and other areas as well [13]. A social network is made up of nodes and edges. Nodes stand for participants while edges denote the link between the participants. A link between two nodes signifies the relation between them [57]. In a network, an interaction between nodes represents their relations [13].

### 3.1.1 Models of Networks

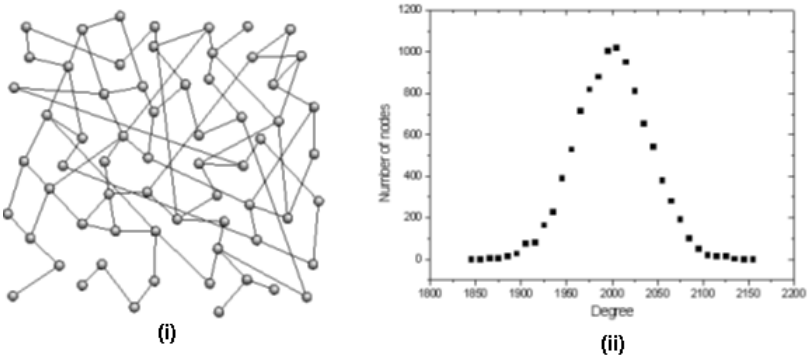
Several models for investigating the topological properties of social networks have been proposed. Among others models, the prominent ones include the Random graph model, Small-world model and Model of a scale-free network.

#### The Random Graph of Erdős and Rényi

Erdős and Rényi constructed a random graph that could be seen as the basic model of a complex network [31, 18]. This graph (see Figure 3.1) is made up of $N$ vertices and $E$ edges while multiple edges and self-loops were excluded. In building the network, $L$ edges were added at random to already defined $N$ disconnected vertices. In another related approach, the models were constructed by first defining the N vertices and then, for each pair of vertices, calculate the probability ($p$) of connecting them. The latter method is called Erdős - Rényi (ER) model [31, 30]

#### The Small-World Model of Watts and Strogatz

Networks show a small world property when most of its vertices are reachable from others via a few edges [18]. A good example is a social network, where you can connect to everybody in the globe




(a) The Random Graph of Erdős and Rényi

*Fig. 3.1:* (i)is an example of a random graph and (ii) the average degree distribution over 10 random networks formed by 10,000 vertices using a probability $p = 0.2$. Source: [18].

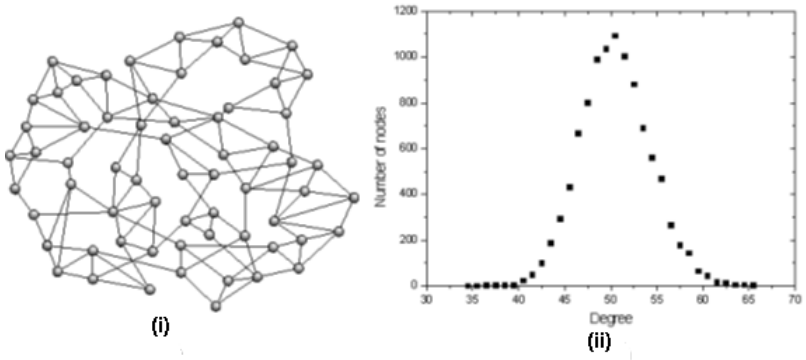
via a few linked social acquaintances [75, 76]. Watts and Strogatz constructed a random network (see Figure 3.2) which has a small-world feature with many short loops and is known as the Watts-Strogatz (WS) Small-World model [77]. The Watts-Strogatz model was first introduced by Duncan J. Watts and Steven Strogatz in their joint work published in Nature (1998). The author used the following steps to create the Watts and Strogatz model:

- Step 1: Build a ring lattice with $N$ vertices and each vertice has a link to $2R$ neighbours ($R$ on each side).
- Step 2: In each of vertice $n_i = n_1, ..., n_N$ take each edge $(n_i, n_j)$ with $i < j$, and with probability $\beta$ wire it again.
- Step 3: To wire again, substitute $(n_i, n_j)$ with $(n_i, n_k)$ where $k$ is selected with uniform probability from every plausible values that avoid self-loops ($k \neq i$) and edge that has duplicate .

**Scale-free Networks of Barabasi and Albert**

another research from Barabasi and Albert reveals that many natural world systems have a degree distribution that exhibits a power-law [9]. Unlike ER and WS models, which have random pattern connected vertices, some vertices in BA have a high connection while others have a few connections [9, 18]. This model degree distribution exhibits power-law when the $k$ value is large. The BA model is called a scale-free network. A network is scale-free when its degree distribution exhibits a power-law, at least asymptotically [9] which can be described by:

$$P(k) \sim k^{-\alpha}$$




(a) The Small-World model of Watts and Strogatz

*Fig. 3.2:* (i) an example of a network with $N = 64$ vertices, $k = 2$, p = 0.1, and (ii) average degree distribution over 10 WS networks with 10 000 vertices, $k = 25$ and $p = 0.3$. Source: [18].

In this case, $\alpha$ is a parameter with values ranging from $2 < \alpha < 3$ and sometime it may occur outside these ranges [9, 18].

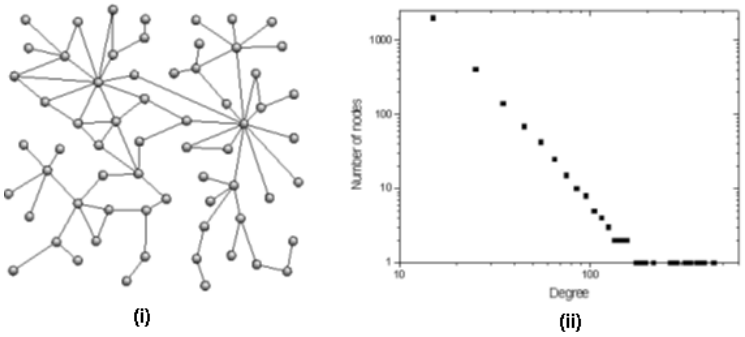
Another feature in a scale-free network is the existence of hubs [9]. Nodes with the highest degree in a network are sometimes known as hubs and are believed to have a definite purpose, even though they are domain dependant. The two core principles on which the Barabasi-Albert network model are based are growth and preferential attachment [9, 18]. According to the author [9] Barabsi-Albert model can be generated using following procedure: First, beginning with $m_0$ nodes, the network develops by following the two steps below:

Growth: A new node with $r(\leq r_0)$ link is added at every timestep and this link connects the new node to r which is in the already existing network

Preferential attachment: The probability $\Pi(k)$ of each edge of the newly formed node links to node $i$ which depend on the degree $k_i$ of node $i$ is $\Pi(k_i) = \frac{k_i}{\sum_j k_j}$

Preferential attachment appear to be a probabilistic rule: a newly formed node is allowed to link to any already existing node in the network.

Because of the importance and usefulness of online social networks, researchers have been investigating their properties. As mentioned in the introduction, discussing the principles and statistics of a social network will be relevant when exploring the dataset in the next chapter. We will examine some of the properties and statistics of the social network that are most useful in exploring our data set. These include degree distribution, cluster coefficient, centrality measure, density, diameter and power-law.




(a) The scale-free network of Barabasi and Albert

*Fig. 3.3:* (i) an example of a scale-free network and (ii) average degree distribution over 10 Barabasi-Albert networks formed by 10,000 vertices using m = 5. The degree distribution follows a power-law, in contrast to that presented in Figure 3.3. Source: [18].

### 3.1.2 Local statistics

The local statistics focuses on a property within a graph (ie node as propety within the whole graph). Assume $Gg$ is a category of graphs, $R$ is a group of parameters and $Z$ a group of values. In addition, let us consider $KG$ to be a group of graph properties in $G$, which could include vertices, edges, subgraphs, paths, etc. Then, a local statistic $\rho G$ is the one that allocates a single value $\rho G(k) \in Z$ to a certain graph property k of a given graph $G \in Gg$. [70] Examples are $in - degree$ and $out - degree$, edge weight, distance, clustering coefficient of a vertex and centrality measures.

#### Degree

The degree of a vertex $k$ represents the number of edges that have $k$ as its endpoints, for example, in figure 3.4, node A has 5 degrees and node B has 3 degrees. The average node degree $K$ can be defined as $K = \frac{2E}{N}$ [70]. The node degree distribution $p(k)$ is the probability that a randomly selected node has a specified degree $k$. The node degree distribution can be represented as $p(k) = \frac{n(k)}{N}$. In this case, n(k) is the number of nodes that have the degree of $k$. We look at the power-law degree distributions, $P(x) \sim x^{-\alpha}$, where $x$ is the node degree [4].

In a directed graph $G = (K; E)$, the out-degree of u $\in K$, denoted by $\deg^+(u)$ is the number of edges that have their origin in u. The in-degree of u $\in K$, denoted by $\deg^-(u)$, is the number of edges that have their destination in u [70].

#### Clustering coefficient

Another important property of the network is the clustering coefficient and centrality measure. The clustering coefficient measures the degree to which friends of friends know each other [4]. The clustering coefficient of a graph is an aggregation of the local clustering coefficient for each node.

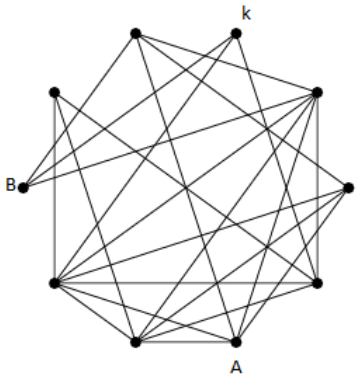



*Fig. 3.4:* The degree of a vertex

According to [70]: Given a graph $G = (V, E)$ and a vertex $vV$, the clustering coefficient of $v$ is $CC(v)$ = number of pairs of neighbours which are connected by edges and halved by the number of pairs of neighbours. The clustering coefficient for the whole graph is the average of the local values CCi.

The average clustering coefficient of the whole nodes in a network is called the clustering coefficient of the network [4]. According to Hamed et.al [36], given a node i with $k_i$ links, at maximum, these links may be involved in triangles

$$\frac{k_i(k_i - 1)}{2}$$

. Increase in the number of triangles indicates a rise in the clustering of the node as well. The clustering coefficient, y(G) can be defined as the average number of triangles(3-cycles) subdivided by the total number of possible triangles.

$$CC = 1/N \sum_i \frac{T_i}{k_i(k_i - 1)/2}, k_i \geq 2$$

Where $T_i$ is the number of triangles (3-cycles) for node i and $k_i$ is the degree of the node i. It also measures the degree at which neighbours are connected in a network. The clustering coefficient of a node in a perfectly-connected mesh network is 1 [47, 45].

### Centrality measure

Furthermore, we considered the centrality properties (betweenness centrality and closeness centrality). The betweenness centrality measures the degree of a node that influences or controls information flows between other nodes [34]. Betweenness can also be defined in terms of shortest path: the number of shortest paths through a node within a network [70].

The betweenness centrality $C_b(n)$ of a node n is calculated by [36]:

$$C_b(n) = \sum_{s \neq v \neq t \epsilon N} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where $\sigma_{st}$ is the number of shortest paths from s to t and $\sigma_{st}(v)$ is the number of shortest paths from $s$ to $t$ that passes through node $v$. The average value of the node betweenness over all nodes is called average node betweenness.

On the other hand, closeness centrality is concerned with how fast information flows within a network, starting with a given node to other reachable nodes [4].

Closeness centrality is calculated as the mean geodesic distance between a vertex and all other vertices reachable from it [34]. The closeness centrality, Cc(n) of a node n is defined as the reciprocal of the sum of the shortest path from this node to other reachable nodes in a graph.

### 3.1.3 Global statistics

Assume $Gg$ is a category of graphs, $R$ is a group of parameters and $Z$ a group of values. In addition, let consider $KG$ to be a group of graph properties in $G$, which could include vertices, edges, subgraphs, paths, etc. Then, a global statistic $\delta$ is the one that allocates a single value $\delta G$ $\in$ to each graph $\in \Gamma$ [70]. Examples are number of vertices or edges, diameter, density, and mean geodesic distance of the graph.

#### Diameter

This can be described as the maximal distance in-between two vertices. For instance, If the link from node A to B is the shortest path in a network, the number of edges on that short path is described as the diameter of a graph [34]. For example, in figure 3.5, central point here is one of the nodes where three edges meet. The first graph has a diameter 4 and the second graph has a diameter of 3.

Mathematically, it can be described as:

$$\text{diam(G)} := max(d(u, \nu)|u, \nu \in V)$$

Where $d(u, \nu)$ represents the distance in-between two vertices and it can be defined as

$$d(u, \nu) = min|P|$$

Where P represents the path starting from u and end in $\nu$

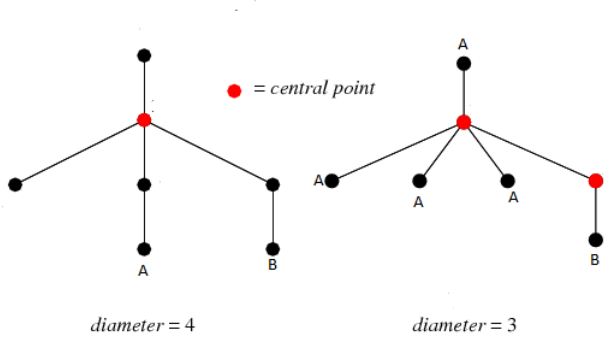



*Fig. 3.5:* The diameter of a graph

### Density

We can define the density of a graph $GE$ as a fraction of edges available in $GE$ to the maximum possible edges [34]. According to [70] the density of a graph $GE$ is given by

$$(GE) = \frac{m}{n(n-1)/2}$$

Where $n$ is the number of vertices available, $m$ the number of edges available and $n(n-1)/2$ is the maximum number of edges that can be present in an undirected graph. Having talked about the network properties, in the next section, we will introduce the power-law because of the relevance of its mathematical properties to our study.

### 3.1.4 Power Law

Power-law has received intense attention for the last decade as a result of its valuable mathematical properties and occurrence in a range of natural and man-made phenomena [16]. As defined in section 2.1: Mathematically, a quantity $x$ obeys a power-law if it is drawn from a probability distribution:

$$P(x) \sim x^{-\alpha}$$

Where $x$ is the node degree and $\alpha < 3$ attests to the existence of a relatively small number of nodes with a very large number of links.

Quantities such as the sizes of cities, the strength of earthquakes etc., which are not clearly described by their average values, have been found to follow power-law distributions.

While most studies in the past relied on least squares fit of a straight line to detect and characterise power-law behaviours [32], a later study [59] criticised the approach and argued that it introduces biases into the value of the exponent. They proposed a maximum likelihood and a goodness of fit test in their approach as a statistical framework for detecting and measuring true power-law behaviour empirically. Supporting this,

### 3.1.5 Three-steps approach to test the power-laws hypothesis

Clauset et.al [16] suggested a maximum likelihood as a reliable approach for calculating power-law parameters as we stated in section 2.1. The authors proposed a three-step process to test if a distribution actually exhibits power-law. The first step is to estimate the power-law parameters $x_{min}$ and $\alpha$, the second step is to calculate the goodness-of-fit and the third step is to conduct a likelihood test ratio. We will briefly discuss each step.

#### Estimating the power-law parameters

Most of the empirical data in nature has a power law degree distribution, in the form $P(x) \sim x^{-\alpha}$, for (x> $x_{min}$) where $x$ is the node degree and $\alpha$ is the exponent. The exponents of the power-laws can be used to characterise graphs. According to [16], when a lower value of $x$ or $x$ below the lower limit bound (x< $x_{min}$) is fitted, it will means mean fitting a power-law model to a non-power-law data (data below the limit bound i.e $x_{min}$ ). This will produce bias during the estimation of the scaling parameter. On the other hand, too a very high value of $x$ or $x$ above the lower limit ($x > x_{min}$) will produce an increase of both statistical errors on the scaling parameter and the bias from finite size effects because it will be effectively excluding data points that are valid. Maximum likelihood is a reliable technique for calculating the exponent of power-law [16]. In general, the following approach was proposed by [16] for the analysis of power-laws:

In the rest of this work, we use the method proposed by

#### Goodness-of-fit

According to Clauset et al, the hypothesis can be tested using a goodness-of-fit test, through a bootstrapping procedure. This approach calculates the goodness-of-fit between the data and the power-law, if the resulting p-value is greater than 0.1, then power-law is a plausible hypothesis for the data, otherwise it is rejected.

#### Direct comparison (likelihood test ratio)

Lastly, Clauset et al . for detecting and quantifying true proposes that a comparison of the power-law behaviour in chat room data . We will now discuss part two, which deals with the contents in conversations. with an alternative hypotheses via a likelihood ratio test be made. This involves a direct comparison of two models (in our case with the log-normal distribution). A standard technique is to use Vuongs test, which is a likelihood ratio test for model selection using the Kullback-Leibler criteria. The test statistic, R, is the ratio of the log-likelihoods of the data between the two competing models. The sign of R indicates which model is better. The p-value from these tests will be used to quantify the plausibility of the hypothesis as follows:

- If the p-value is large (close to 1), then any difference between the empirical data and the model can be explained with statistical fluctuations.
- If p-value is small (close to 0), then the model does not provide a plausible fit to the data and another distribution may be more appropriate in this scenario,

## 3.2 Conversation Analysis

Conversations involve two or more people and consist of a series of utterances [15]. Conversations are governed by norms and traditionally have been analysed using qualitative protocols [66]. However, this analysis is difficult and labour-intensive due to the contextual nature of the meaning of words [62].

According to Clark [15], conversations consist of different levels of parts: section, adjacency pairs and turns. Conversations are usually purposive and unplanned [15]. More often than not two people entering into conversation may have certain purposes, but no concrete plans of how it will be achieved. According to [15], conversations may appear prearranged and objective-oriented only in retrospect, but in practice they are generated cleverly bit by bit as the participants negotiate collaboratively and what emerges are adjacency pairs, sections and, ultimately, the entire conversation itself [15].

### 3.2.1 Adjacency pairs

An adjacency pair is defined as a sequence of two utterances generated by two different speakers that are, subsequent to one another [22]. According to [22], adjacency pairs consist of two sequences of successive actions - first pair segment and second pair segment. The two segments are performed by different agents X and Y.

- X: Hi ($1^{st}$ pair part)
- Y: Hi ($2^{nd}$ pair part)
- X related kind of greeting adjacency pair is the English closing greeting.
- X: (great) bye! ($1^{st}$ pair part)
- Y: (great) bye! ($2^{nd}$ pair part)

### 3.2.2 Sections in conversations

In the turn-taking rule allocation, Sacks et al. [66] identified two adjacent turns: current turn and the next turn, and the transition between them. According to Sacks et al. [66], conversations are handled locally and controlled by joint action which means that conversations are managed contribution by contribution. The shape of each turn is dependent on all participants because they manage the section of the current speaker as well as influence the course and the length of each turn [22]. When

people are entering a conversation, they may have general goals but will not always make specific plans on how to achieve these aims before the conversation [22]. This means that conversations are managed contribution by contribution. Conversation periods can be divided into three [22]:

- Opening conversation session
- Conversation body
- Closing conversation session

According to [22], what takes place in the "opening conversation" period can be called an opening section. Opening sections are joint projects for the participants and emerge from the usual entry-body-exit structure: entry (orienting to the possibility of conversing), body (establishing a joint commitment to converse) and exit (opening of first topic). The exit from the conversation is called the closing section. Two participants in a conversation cannot exist merely by stopping. They first exist from the last topic and jointly agree to close. The close section also has an entry-body-exit structure: entry (terminating the last topic), body (taking leave) exit (terminating contact).

### 3.2.3 Turn taking in conversations

In a good group conversation, people take turns talking. Only one person is permitted to speak at a time. In reality, it is not usually so, more than one person speaks to either an individual or group at the same time. Sacks et al. proposed rules to govern the common observations about everyday conversation, in turn-taking which is called the turn-taking allocation rule [66] and the rules are as follows:

- There is often change in the speakers at least once.
- There is only one party that talks at a time.
- Often more than one person are allowed to speak at a time briefly.
- Non-overlap one gap is popular when moving from one turn to the next.
- There is a variation in the order of the turn.
- The size of the turn also varies.
- The duration of the conversation is not predetermined .
- There is no limitation in what parties say or specify in advance.
- Turn distribution is not predetermined.

Using the system of Sacks et al., the current speaker may "select" the next speaker, for example, by asking a question that obliges the addressee to take the next turn. This means that the next speaker must be one of the audiences of the current speaker (i.e. the next sender must be one of the receivers of the current sender). Sacks et al. proposed the following rules:

(A) If the turn is designed in such a way that the use of a current speaker selects the next technique, then the selected party retains the right to have the next turn and at this point that transfers occur.

(B) If the turn is designed in such a way that the use of a current speaker selects the next technique, then self-selection may be initiated for the next speaker and the right to take a turn is given to the first starter.

(C) If the turn is designed in such a way that the use of a current speaker selects the next technique, then the current speaker may carry on if there is no another self-selection.

If at the starting of the transitional-relevance place of the first turn-built unit, neither number 1 nor number 2 has acted and with the prearrangement of number 3, the current speaker has carried on, then the rule-set A-C will apply again in the next the transitional-relevance place and repeatedly at every of the next transitional-relevance place till the transfer comes into effect.

### 3.2.4 Beginning and end in a multi-party conversation

In a chat conversation, detecting certain word patterns used in fixing the starting and ending of a conversation is very important. Duranti [21], in his book Key Terms in Language and Culture, states that turn-taking in everyday conversation may be identified by words or phrases; for example, a conversation often begins with: hello, welcome, welcome on board etc. while the end of a conversation often includes: bye, it was nice meeting you, it was nice chatting with you, see you next time, see you tomorrow etc. Also, questions and normal sentences can be used to initiate a conversation as well as select the next speaker, e.g. (Hello, how are you?), (Joe, did you watch the football?), etc. To determine when a conversation begins and when it ends, we need to know the common conversation starting and closing pattern techniques of that particular culture [15]. For instance, when opening a conversation, people say something like hello, while in closing conversation you will hear something like see you later, have a nice day, goodbye and okay.

In the real world, closing a conversation is done in a smooth and acceptable way so that the other participant will not be offended. The participant who wants to initiate the closing of the conversation must prepare the other participant before final closure [22]. The participants must jointly agree to close the conversation. For instance, in preparation for closure, one of the participants involved may

initiate the closure by saying goodbye while the other participant responds with either goodbye or another form of closing salutation.

However, in some cases, after saying goodbye the participants may remember a very important thing to discuss and they will continue talking. This often introduces a lot of closing salutation in the middle of the conversation before it finally ends. Having a lot of starting and closing salutations in the middle of a conversation makes it difficult to use conversation starting and finishing patterns to determine the begin and end of a conversation. Wittgenstein [22] suggests that words should, at all times, be considered as being enclosed in whole activities since the same word can mean different things at any point where it occurs. The author stated that we should put into consideration the large sequences in which the words occur before putting a meaning to it. They suggested that seeing it as part of a bigger unit will provide insight into how the words work.

| | | | | |
|---|---|---|---|---|
| | (19) | Dorinne: | Uh-you know, it's just like bringin the- blood up. | |
| | | Theresa: | Yeah well. THINGS UH ALWAYS WORK OUT FOR THE BEST | |
| | | | [ | |
| | | Dorinne: | Oh certainly. | |
| → | | | Alright Tess. | **(pre-closing: first pair part)** |
| | | | [ | |
| | | Theresa: | Uh huh, | |
| → | | Theresa: | Okay. | **(pre-closing: second pair part)** |
| | | Dorinne: | G'bye. | **(closing: first pair part)** |
| | | Theresa: | Goodnight, | **(closing: second pair part)** |
| | (20) | Johnson: | ... and uh, uh we're gonna see if we can't uh tie in our plans a little better. | |
| | | Baldwin: | Okay fine | |
| | | | [ | |
| | | Johnson: | ALRIGHT? | |
| | | Baldwin: | RIGHT. | |
| → | | Johnson: | Okay boy. | **(pre-closing: first pair part)** |
| → | | Baldwin: | Okay. | **(pre-closing: second pair part)** |
| | | Johnson: | Bye bye. | **(closing: first pair part)** |
| | | | [ | |
| | | Baldwin: | G'night. | **(closing: second pair part)** |

*Fig. 3.6:* Conversation activity. Source:[21]

For example, in conversation (20) of Figure 3.6, the first okay by the speaker named Baldwin is different from the okay subsequently produced by the speaker named Johnson. The first okay is part of an agreement (okay fine) to a proposal and hence closes a topic. The second okay (okay boy) is the first pair part of an adjacency pair that sets the tone for the forthcoming closing salutations. Supporting, Searle and Vanderveken [21] argued that hello could be used to indicate recognition in a courteous fashion and does not capture the use of hello or other kinds of greetings in conversations.

This can occur when we do not take into consideration the larger contexts in which greetings and closing salutations appear in conversation. This is a significant challenge in determining when a conversation started and closed. Traditionally, conversational analysis has used qualitative protocols to establish the beginning and end of a conversation [24]. However, it has proved to be difficult, due to the contextual nature of meaning that we have just discussed. We propose to supplement these protocols with simple statistical data, such as response waiting time and turn-taking.

### 3.2.5 Distributions

Here, we introduce some of the statistical distributions that were used later in this research.

#### Generalised Pareto

The Pareto distribution is used for quantities that are distributed with very long right tails. It is a power law probability distribution that is used in describing many types of observable phenomena. It is named after the Italian economist Vilfredo Pareto.

#### Inverse Gaussian Distribution

The Inverse Gaussian Distribution, also called the Wald or normal-inverse Gaussian, is an exponential distribution with a single mode and long tail. The distribution is used to model non-negative, positively skewed data

#### Log-normal Distribution

The log normal as a probability distribution has logarithm that are normally distributed. This distribution is appropriate if a positive quantity outcome is of a great interest because we can only have log(x) if x is positive. Mathematically, the log-normal distribution is given by:

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\Pi}}\epsilon^{-\frac{1}{2}(\frac{lnx-\mu}{\sigma})^2}$$

#### Burr Distribution

Burr distribution was first discussed by Burr (1942) as a two-parameter family. An additional scale parameter was introduced by Tadikamalla (1980). It is a very flexible distribution family that can express a wide range of distribution shapes. The cumulative distribution function (cdf) of the Burr distribution is

$$f(x|\alpha,c,k) = 1 - \frac{1}{(1+(\frac{x}{\alpha})^c)^k}, x > 0, \alpha > 0, c > 0, k > 0.$$

#### Gamma Distribution

In the read world, the gamma distribution is often used to define a range of processes where the data are positively skew. It is a distribution with two parameters: shape and rate.

#### Generalized Extreme Value Distribution

The generalized extreme value distribution is often used to model the smallest or largest value among a large set of independent, identically distributed random values representing measurements or observations.

### 3.2.6 Algorithm

We would like to discuss two algorithm that will be applied later in this study.

#### Neural Network(NN)

The development of a Neural Network is inspired by human brain activities. As such, this type of network is a computational model that mimics the pattern of the human mind. The neural network is constructed with an interconnected group of nodes, which involves the input, connected weights, processing element, and output. Neural networks can be applied to many areas, such as classification, clustering, and prediction.

The neural network is a network made up of artificial neurons (or nodes). There are three types of neurons within the network: input neurons, hidden neurons, and output neurons. In the network, neurons are connected; the connection strength between neurons is called weights. If the weight is greater than zero, it is in an excitation status. Otherwise, it is in an inhibition status. Input neurons receive the input information; the higher the input value, the greater the activation. Then, the activation value is passed through the network in regard to weights and transfer functions in the graph. The hidden neurons (or output neurons) then sum up the activation values and modify the summed values with the transfer function. The activation value then flows through hidden neurons and stops when it reaches the output nodes. As a result, one can use the output value from the output neurons to classify the data.

Neural Network Model: Organic neural networks are composed of billions of interconnected neurons that send and receive signals to and from one another. Artificial Neural Networks are a class of flexible nonlinear models used for supervised prediction problems. The most widely used type of Neural Network in data analysis is the multilayer perceptron (MLP). MLP models were originally inspired by neurophysiology and the interconnections between neurons, and they are often represented by a network diagram instead of an equation. The basic building blocks of multilayer perceptrons are called hidden units. Hidden units are modeled after the neuron. Each hidden unit

receives a linear combination of input variables. The coefficients are called the (synaptic) weights. An activation function transforms the linear combinations and then outputs them to another unit that can then use them as inputs.

#### Support Vector Machine (SVM)

SVM is a binary classification model that constructs a hyperplane to separate observations into two classes. In constructing a hyperplane, we define a margin around the hyperplane, the point that lies on the two margins are called support vectors and we define an optimisation problem that will maximise the margin of the defined hyperplane. The size of the margin governs the tradeoff between correctly classifying the training dataset and generalising the future dataset:

Wide Margin means more training point misclassified but better generalised for future dataset while Narrow Margin means fits the training point better but might be overfit to the training dataset

Because of challenges in getting a perfect separation of the two classes, SVM introduces a concept of penalty C for observation that falls on the wrong side of the margin. The penalty is based on distance from misclassified points to their right side of the margin. This C is called the tuning parameter or regularising parameter. It helps to avoid over fitting as well.

Tuning parameters of 0.1 to 1.0 were optimised, this means that C ranged from 0.1, 0.2, 0.31 and a model was trained for value of C, then the best model was selected as the champion model

In some cases, the two classes are inseparable e.g. A circle blue points in a square red points. It is difficult to separate them linearly. Here a concept of kernel function is used to map to a higher-dimension space which makes the separation much more possible. There are 3 kernel functions in SVM for transforming complex data spaces into a form that can be more easily separated: Sigmoid function, Radial basis function and polynomial (2 to 4 degrees).

### 3.2.7 Evaluation metric

Model evaluation metrics are used to assess goodness of fit between model and data.

#### Receiver operating characteristic

ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve plots the sensitivity against one minus the specificity for a series of cutoffs for the fitted probability. The ROC plot is a unit square plot, and the higher the curve rises above the 45-degree line, the more desirable it is. The 45-degree line corresponds to an area under the curve (AUC) of 0.5 and represents where the fractions of true positives and false negatives are equal.

Sensitivity or Recall measures the ability of a model to correctly predict, or "rule in" the event of interest among those observations under analysis in which the event occurred. So, a model sensitivity of 90% means that if 100 observations in your data set had the event of interest, the model correctly predicted that 90 of them would have the event. Specificity, on the other hand, measures the ability of a model to "rule out" the event in those observations under analysis in which the event DID NOT occur. If the model specificity is 90%, that means that out of 100 observations that did not have the event of interest, the model correctly predicted that 90 of them would not have the even. Also the Positive Predictive Value or Precision measures the proportion of positive cases that were correctly identified and the Recall measures the proportion of actual positive cases which are correctly identified. The F-measure or F-score combines precision and recall using the harmonic mean. The harmonic mean is used rather than the more common arithmetic mean since both precision and recall are expressed as proportions between zero and one.

- True Negative (TN) a negative class data point was identified as negative.
- False Negative (FN) a positive class data point was identified as negative;
- False Positive (FP) a negative class data point was identified as positive;
- True Positive (TP) a positive class data point was identified as positive;

*Tab. 3.1:* Confusion matrix

| 3* | Predicted class | | |
|---|---|---|---|
| | | Yes | No |
| Actual class | Yes | True Positive | False Negative |
| | No | False Positive | True Negative |

**Latent Semantic Analysis**

Latent Semantic Analysis is a new automatic mathematical or statistical method for extracting and inferring relations of expected contextual usage of words in passages of discourse [19]. it uses knowledge bases, semantic networks, grammars, syntactic parsers or morphologies [19]. It takes as an input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs [7]. The latent semantic structure analysis starts with a matrix of terms by documents. Each row stands for a unique word and each column stands for a text passage or other context. Each cell indicates the frequency with which each term occurs in each document. The cell entries are subjected to a preliminary transformation, in which each cell frequency is weighted by a function that expresses both the word's importance in

the particular passage and the degree to which the word type carries information in the domain of discourse in general [19]. This matrix is then analyzed by singular value decomposition (SVD) to derive our particular latent semantic structure model which was then used for indexing [7].

### Cosine Similarity Measure

Cosine Similarity Measure is one of the popular approaches in finding similarities between two document using the cosine function, which is the measure of similarity between two vectors derived from the cosine of the angle between them. For example, here are two very short texts to compare:

- Julie loves me more than Linda loves me.
- Jane likes me more than Julie loves me.

We want to know how similar these texts are, purely in terms of word counts (and ignoring word order). We begin by making a list of the words from both texts:

- me Julie loves Linda than more likes Jane

Now we count the number of times each of these words appears in each text:

*Tab. 3.2:* Convert the documents from text to vector.

| Terms | Document 1 | Document 2 |
|---|---|---|
| me | 2 | 2 |
| Jane | 0 | 1 |
| Julie | 1 | 1 |
| Linda | 1 | 0 |
| likes | 0 | 1 |
| loves | 2 | 1 |
| more | 1 | 1 |
| than | 1 | 1 |

We are not interested in the words themselves though. We are interested only in those two vertical vectors of counts. For instance, there are two instances of 'me' in each text. We are going to decide how close these two texts are to each other by calculating the cosine of the angle between the two vectors.

The two vectors are, again:

- a: [2, 1, 0, 2, 0, 1, 1, 1]
- b: [2, 1, 1, 1, 1, 0, 1, 1]

The cosine of the angle between them is about 0.822.

# 4. CHARACTERISATION OF THE CHAT ROOM NETWORK

Several platforms such as email, Twitter, Facebook and other chat rooms used for online communications appear to have different characteristics. Even within the same platform, various type seem to have varying characteristics due to different technology bases; for example, chat rooms such as Multi-User Dungeon (MUD Servers), Internet Relay Chat (IRC), Web Chat, IM and Voice Chat, and Walford may have different features. In this chapter, we uncover some hidden characteristics of various types of chat rooms, but first, we will describe the datasets and I want note that we did not collect the dataset ourself.

## 4.1 Datasets

We have three datasets from different chat rooms: Walford, IRC, and T-Rex. The summary of the datasets is displayed in Table 4.1

*Tab. 4.1:* Summary of the three datasets.

| Parameters | IRC | TEX | Walford |
|---|---|---|---|
| No. nodes | 473 | 97 | 2446 |
| No. edges | 1431 | 138 | 37982 |
| No. hours | 72 | 72 | 26297 |
| No. days | 3 | 3 | 1095 |

### 4.1.1 IRC chat logs

The IRC chat transcript is Elsners dataset from the Linux channel at http://freenode.net. The dataset can be downloaded from http://www.ling.ohio-state.edu/melsner/. The chat format include: speaker name, recipient name, comment or action and the times which are given in seconds.

### 4.1.2 T-REX chat log

This dataset came from the National Center for Atmospheric Research (NCAR). This chat log was generated as part of the T-REX field project and was managed by the Earth Observing Laboratory (EOL) at the National Center for Atmospheric Research (NCAR). These chat logs were reviewed by EOL and edited to maintain the most accurate records for the project.

### 4.1.3 Walford chat log

Walford was a text-based online social community that had roughly 2446 regular users and the number of communications or edges was 37,981 for four years [38]. It was a corpus that contained 24,040 hours (2001-2004) of chat. In the corpus of chat logs, the following data was recorded: unique numeric IDs for the time, the originator, the originators location, the recipient(s) and their location.

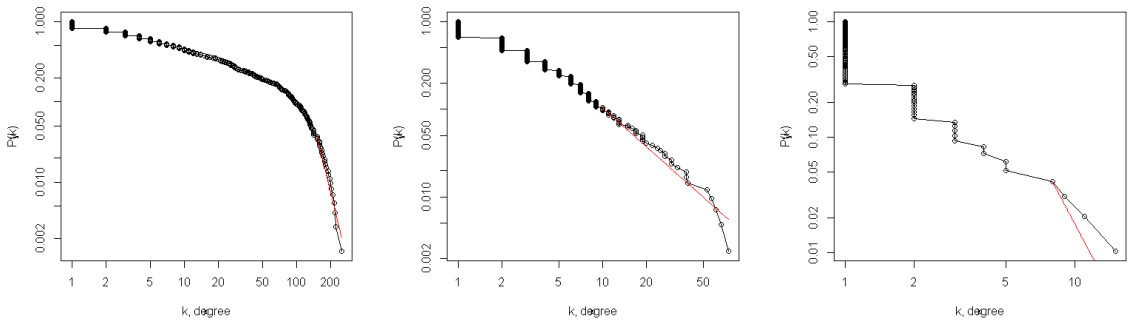
### 4.1.4 Node degree for chat logs

A chat room is made up of users who send messages to each other. We can model this with a network in which the users are represented by nodes while the messages are represented by links or edges. The number of links a node has to other nodes in a network is called node degree. As explained in chapter three, many natural and man-made occurrences follows power law distribution and to this effect, we explored our datasets to confirm if power-law exists in their node degree distribution. This was done by applying the maximum likelihood method [59] to compute the exponent and X-min.

After filtering with $X >= x - min$, we have the following remaining number of users in the dataset: 1430 for Walford, 203 for IRC and 50 for T-REX and then, plotted the degree distributions in Figure 4.1.

*Tab. 4.2:* Number of users before and after filtering.

| Parameters | IRC | TEX | Walford |
|---|---|---|---|
| No. users before filtering | 473 | 97 | 2446 |
| x-min | 10 | 8 | 25 |
| No.users after filtering($X >= x - min$) | 203 | 50 | 1430 |




(a) Walford (b) IRC (c) T-REX

*Fig. 4.1:* Chat logs degree distribution

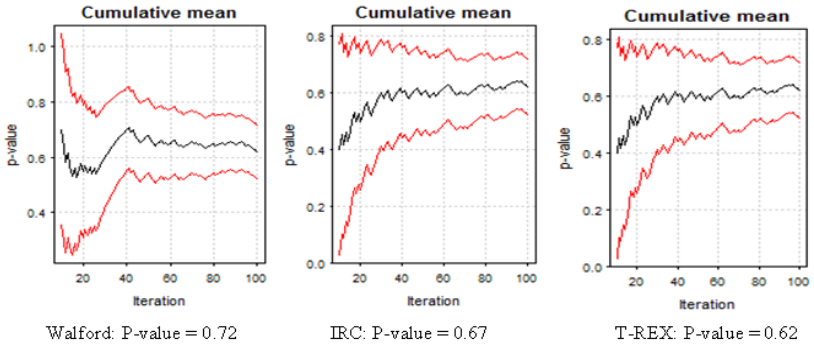
The degree distributions across the chat logs appear to follow power- law, which attests to the existence of a relatively small number of nodes with a very large number of links. Even more, the tail

of the degree distribution is well estimated by a power-law in Walford and IRC. On the other hand, T-REX tends to deviate at the tail; this may be because our dataset for T-REX is small compared to Walford and IRC.

### Testing the power law hypothesis

Following Clauset et al three-step principles in section 3.1.5, the first step involves using maximum likelihood to compute the $x_{min}$) value, which is the minimum number of node degree in the distribution used to fit the power law and $\alpha$ value which is the exponent of the power law. The $x_{min}$) and $\alpha$ values are 25 and 3.18 for Walford, 10 and 2.03 for IRC and 8 and 1.78 for T-REX respectively.

In the second step, we tested the hypothesis using goodness-of-fit test which we implemented through a bootstrapping procedure. The results of the p-values show 0.72 for Walford, 0.67 for IRC and 0.62 for T-REX respectively. The result is displayed in Figure 4.2




(a) Chat logs

*Fig. 4.2:* The graph gives the cumulative estimate of the p-value; the final value of the black line corresponds to P-value. Also the red-lines give approximate 95% confidence intervals

The third step is ratio test. This entails a direct comparison of two distribution using Vuongs test and in our case the alternative distribution is log-normal. The ratio test p-value is as follows: 0.74 for Walford, 0.90 for IRC and 0.38 for T-REX.

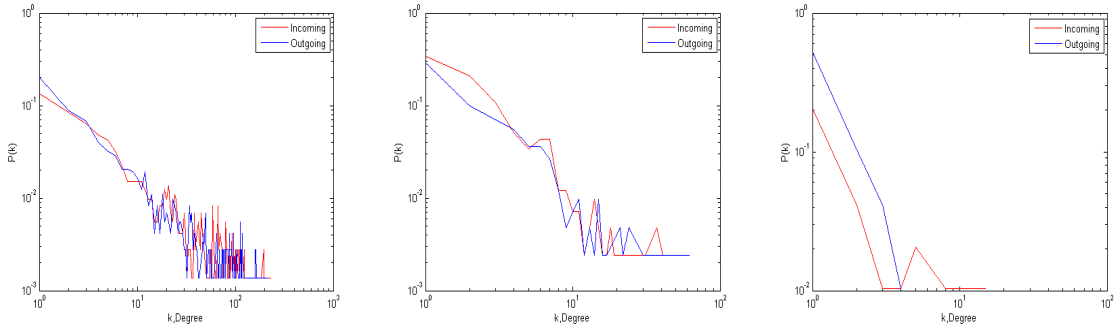
*Tab. 4.3:* Goodness of fit of the tail of the degree distribution to the power-law exponent $P(x) \sim x^{-\alpha}$. Goodness of fit done using the maximum likelihood method [59]. The p-value is the goodness-of-fit metric.

| Parameters | Walford | IRC | T-REX |
|---|---|---|---|
| Alpha | 3.18 | 2.03 | 1.78 |
| x-min | 25 | 10 | 8 |
| GoF (p-value) | 0.72 | 0.64 | 0.62 |
| Ratio (p-value) | 0.74 | 0.90 | 0.38 |

The summary of the result of the three steps is displayed in Table 4.3. Based on the goodness-

of-fit and the ration test values, there is a sufficient evidence to suggest that the model is a plausible fit to the data (i.e. it follows power-law distribution).

To check the reciprocity of the chats between the users, we evaluated the directionality of the messages. The out-degree of a node reflects the messages sent by a user and the in-degree the incoming messages to the user.




(a) Walford (b) IRC (c) T-REX

*Fig. 4.3:* In and Out Degree distribution

The similarity on the in- and out-degree in Walford and IRC (see Figure 4.3) reflects that the users are involved in conversations, instead of just a monologue or a hailing system where the out-degree would be the dominant feature. Again the difference we see in the in- and out-degree of T-REX may be as a result of dataset limitations (we have only a small dataset for T-REX). Having explored the structure of each chat room and confirmed the existence of power-law, we further investigated the temporal differences that exist in the chat rooms.

## 4.2 Temporal Differences

This network is a time-varying graph, which means it changes over time and the edges are not continuously active; as a result, both nodes and edges fluctuate [48, 55]. Temporal difference evaluation focuses on the network structure on weekdays and weekends, across years and across the quarter of years. Also we confirmed that the in-degree and out-degree have very similar distribution, hence, for the temporal difference study, we consider the undirected degree.

### 4.2.1 Behaviour on Weekdays and Weekends

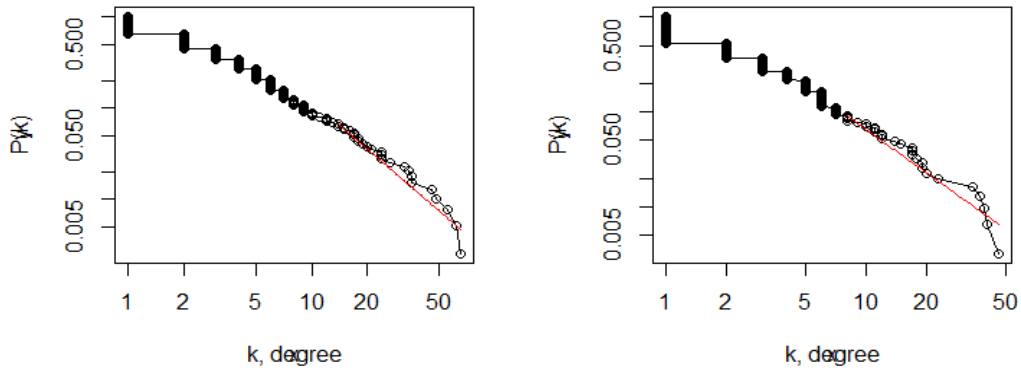
In this section, we investigate the behaviour of users on weekends and weekdays. Weekdays start from 00:00 on Monday to 24:00 on Friday while weekends start from 00:00 on Saturday to 24:00 on Sunday. The summary is of the weekdays and weekends distribution and cluster coefficients and, secondly, we explore the users behaviours through response waiting time (RWT). The dataset summary is displayed in Table 4.4.

*Tab. 4.4:* weekdays and weekends dataset summary

| Logs | Weeks | no.node | no.link |
|---|---|---|---|
| IRC | Wkday | 393 | 1212 |
| | Wkend | 309 | 709 |
| T-REX | Wkday | 68 | 80 |
| | Wkend | 45 | 59 |
| Walford | Wkday | 476 | 9373 |
| | Wkend | 643 | 14113 |

### Degree distribution

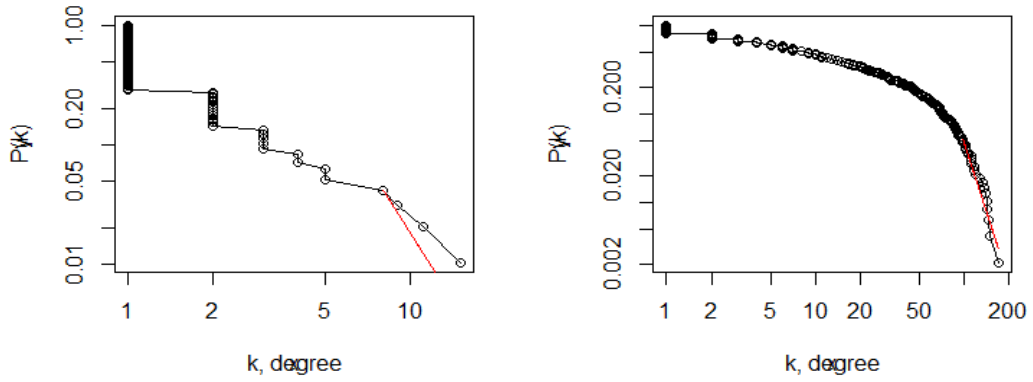
The weekdays and weekends degree distribution for IRC, the T-REX and Walford chat rooms are displayed in Figure 4.4, Figure 4.6 and Figure 4.5 respectively.




(a) Weekdays (b) Weekends

*Fig. 4.4:* IRC:Degree distribution and fitted power law (red line)

Unlike the case of aggregating the whole data, putting the users environment into consideration, the degree distribution appears to differ depend on the chat room (Walfor, IRC and T-REX). On Weekdays, IRC degree distribution tends to decay as a power-law while that of Walford appears to deviate at the tail, suggesting that it cannot be described by a simple power-law. Looking at the degree distribution on Weekends, Walford is well approximated by a simple power-law while IRC seems to have a deviation at the tail. The differences we see on the degree distribution on weekends reflect the fact that IRC are platforms predominately used during working hours compared to the Walford platform, which appears to be used in non-working hours. Another difference is that the weekends have a relatively lower exponent compared to weekdays in the IRC and T-REX chat rooms (see Table 4.5). However, the weekend exponent for Walford is very similar to the weekday exponent,




(a) Weekdays (b) Weekends

*Fig. 4.5:* Walford:Degree distribution and fitted power law (red line)

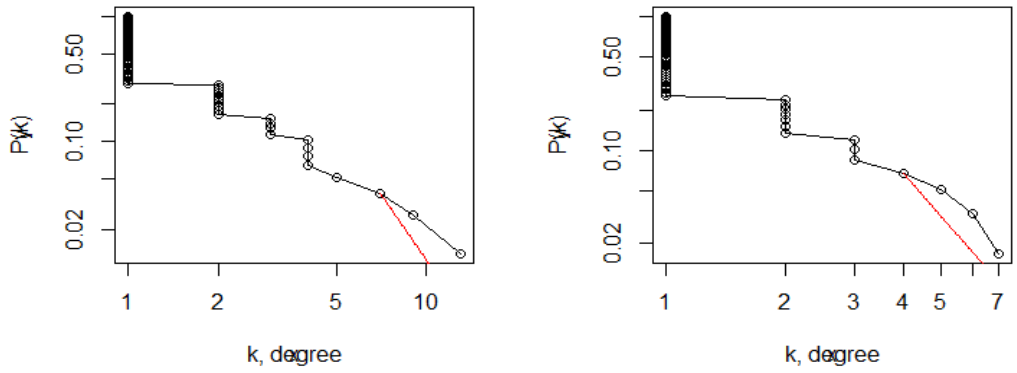
*Tab. 4.5:* Results compared for degree distribution.

| Logs | Week | $\alpha$ | Gof | Ratio | |
|---|---|---|---|---|---|
| Walford | day | 2.35 | 0.94 | 0.51 | 0.74 |
| | end | 2.12 | 0.87 | 0.41 | 0.96 |
| IRC | day | 2.24 | 0.79 | 0.33 | 0.56 |
| | end | 1.67 | 0.27 | 0.74 | 0.69 |
| T-REX | day | 2.13 | 0.59 | 0.39 | 0.79 |
| | end | 1.84 | 0.24 | 0.38 | 0.189 |

perhaps reflecting that Walford chat room is not operated around working topics so is used more for leisure. Hence, environment factor has great influence on users behaviour. Regarding degree distribution in T-REX, we observed a deviation on both weekdays and weekends and this could be as a result of limitation in T-REX dataset

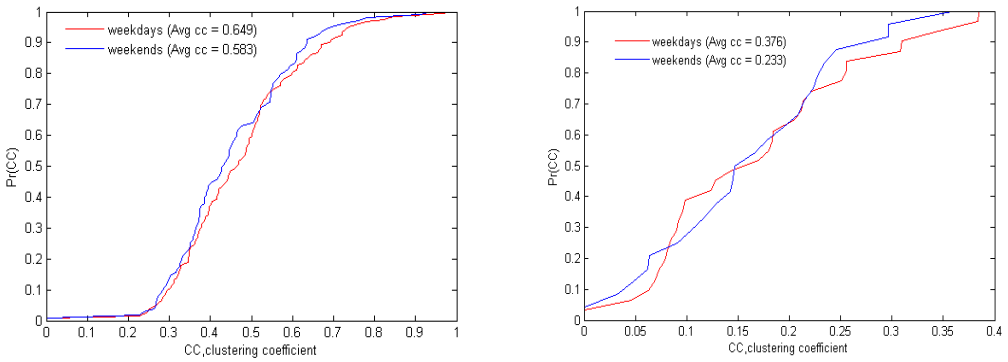
To study the cohesion between the users, we evaluated the local and global clustering coefficient. This coefficient measures the triadic relationship between the users [4]. This study was done for IRC and Walford chat logs only since we have large datasets for them, however due to dataset limitations we excluded T-REX chat logs. The CDF of the local cluster coefficient is plotted in Figure 4.7 (for each node, we calculated the clustering coefficient). The global cluster coefficient (average clustering coefficient of all nodes, having their degree greater than 1) on weekdays and weekends are 0.6 and 0.5 for Walford logs and 0.2 and 0.3 for IRC logs, respectively. For both chat rooms, the clustering coefficient for the weekdays and weekends is very similar. In comparison with IRC, Walford, the clustering coefficient is relatively high, suggesting that Walford has larger clusters of users chatting together.

Next, for each node degree, we calculated the average clustering coefficient of all nodes that had




(a) Weekdays (b) Weekends

*Fig. 4.6:* T-REX:Degree distribution and fitted power law (red line)




(a) Walford (b) IRC

*Fig. 4.7:* Cluster coeff. for weekdays and weekends.

this degree and plotted the distribution of the clustering coefficient as a function of the degrees in Figure 4.8.

The average cluster coefficient decreases as the node degree increases. This suggests that the mean cluster coefficient tends to depend on the degree; the node with a low degree tends to associate with a high average cluster coefficient while the node with a high degree tends to correlate with a low average cluster coefficient. One of the possible reasons for this may be that if one has a few friends; it is highly likely to know friends of your friends, thereby forming a strong tie. On the other hand, if one has too many friends, the probability of knowing the friends of your friends may be very low.

Lastly, we employed Latent Semantic Analysis (LSA) for the automatic indexing of terms in weekdays and weekends of Walford log only since is the largest dataset we have. According to Landauer et al. (1997)

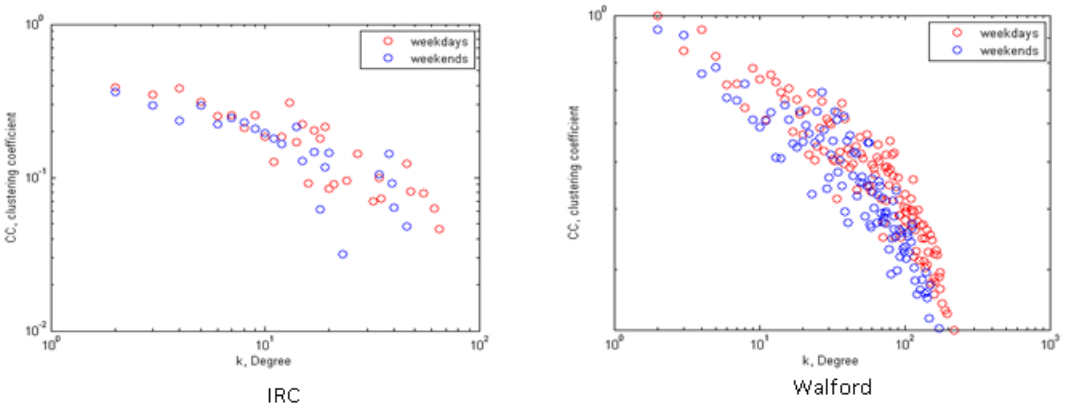



*Fig. 4.8:* Cluster coeff. as a function of Degrees distr.

> "Latent Semantic Analysis (LSA) is a method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. The underlying idea is that the aggregation of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other".

This automatic indexing resulted in 150,074 terms and two documents. Some additional characteristics of the dataset are given below.

*Tab. 4.6:* characteristics of the dataset for weekday and weekend

| Parameters | Value |
|---|---|
| No of documents | 2 |
| No of unique terms | 150074 |
| Avg. terms per doc | 1.39723e+007 |
| Avg. indexing terms per doc | 7.07427e+006 |
| Cosine measure | 0.0005887655 |

Evaluating the similarity between the content of weekday and weekend chats, we apply a cosine similarity measure. The results in Table 4.6 show that the cosine measure is 0.0005887655, indicating a broad range of dissimilarity between the word content in weekday and weekend conversations.

Having looked at the weekdays and weekends, we will now explore the chat rooms considering years, quarters and times of day.

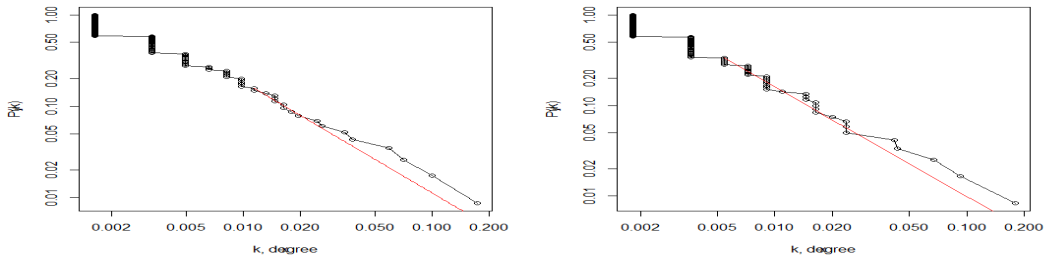
### 4.2.2 Behaviour across the years

In order to study the historical growth of this chat room network in depth, we time-sliced the network and for each year examined the degree distribution and cluster coefficient.

*Tab. 4.7:* Power-law fits and the corresponding p-value across the year

| Parameters | 2001 | 2002 | 2003 | 2004 |
|---|---|---|---|---|
| No.nodes | 611 | 550 | 1575 | 1075 |
| No.links | 8055 | 7695 | 21968 | 16685 |
| Avg.C.C. | 0.489 | 0.490 | 0.622 | 0.641 |
| Alpha | 2.21 | 2.20 | 1.846 | 1.846898 |
| x-min | 0.011 | 0.0054 | 0.0038 | 0.1507 |
| Gof p-value | 0.75 | 0.01 | 0.22 | 0.22 |
| Ratio p-value | 0.77 | 0.32 | 0.73 | 0.73 |

As of 2001, the numbers of participants (nodes) in chat room communication were 611 with 8055 edges and this reduced to 550 with 7695 edges by 2002. In 2003, there was a sudden jump to 1575 participants with 21,968 edges. Finally, the number of participants decreased again to 1075 with 16,685 edges in 2004. These fundamental properties are shown in Table 4.7. In the last row of Table 4.7 we present the p-values for the power-law model, which represents an estimate of how possible it is for the power-law to fit the data [16]. The power-law exponent values, as well as the p-values, show that the degree distributions for 2003 and 2004 are consistent with the power-law. The power-law exponent is between 2 and 3, which indicates that there are relatively few people who have large numbers of friends in the network [4].

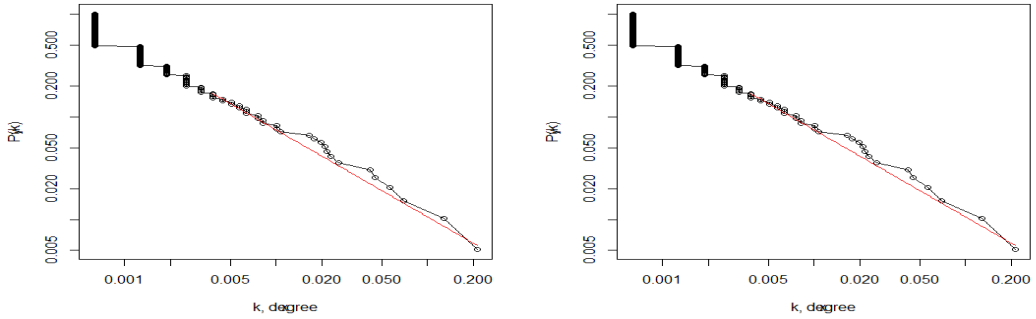



(a) 2001 (b) 2002

*Fig. 4.9:* Power-law degree distribution across the year

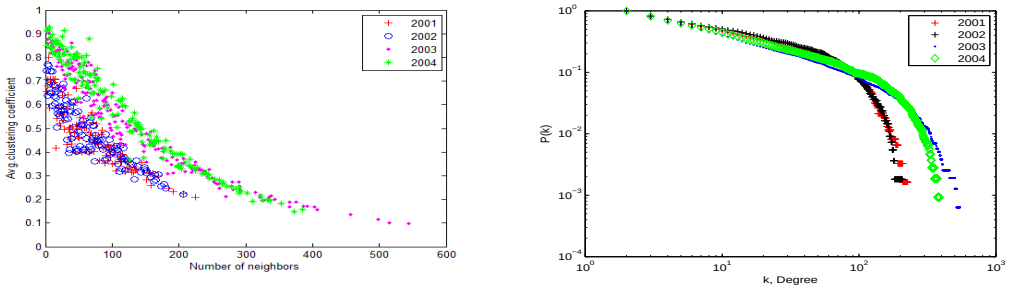
However, the power-law exponents and p-values of the degree distribution for 2001 and 2002 are large; the power-law exponent values are well above 3, which suggest that the power-law model may be ruled out[16]. Moreover, a plot of the power-law degree distribution across the year in Figure 4.9 reveals that 2003 and 2004 are more consistent with the power-law model than 2001 and 2002.

Next, we examine the network clustering coefficient that measures the number of triadic relationships between the users. Table 4.7 shows a gradual increase in the average cluster coefficient value from 2001 to 2004. The highest average clustering coefficient is 0.641 in 2004, indicating a high




(a) 2003 (b) 2004

*Fig. 4.10:* Power-law degree distribution across the year



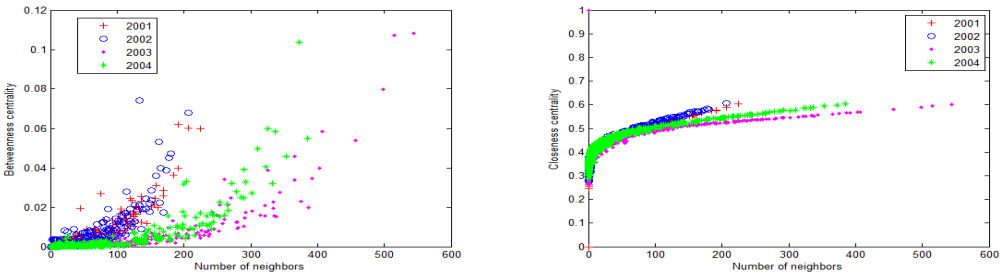

(a) Avg.clustering coef. (b) Node degree distrib

*Fig. 4.11:* Avg.cluster coefficient and centrality distribution across the year

number of triangles (3-cycle, see section 3.1.2), while the least average clustering network is 0.489 in 2001, suggesting a small number of triangles. To support our claim, we plot the distributions of the clustering coefficient in Figure 4.11. The graphs from 2001 and 2002 are the shortest with a low clustering coefficient, reaching only up to k = 300. The graphs of 2003 and 2004 extend longer than that of 2001 and 2003 with a higher clustering coefficient value.

Also, Figure 4.12 reveals that 2003 has the highest value of betweenness centrality(the red point) followed by 2004 (the green point), and that the least betweenness centrality value is found in 2001. This observation may account for the fast information spread and low speed of information flow in 2003 and 2001, respectively, as displayed in the closeness centrality graph.

In summary, for this particular network, the results suggest that 2003 and 2004 are more consistent with the power-law model than 2001 and 2002 and there was a gradual increase in the average cluster coefficient value from 2001 to 2004. So 2004 has a higher number of triangles (3-cycle) than the other years. Also, it is revealed in Figure 4.12 that 2003 has the highest value of betweenness centrality and the least betweenness centrality value was found in 2001.




(a) Betweenness centrality (b) Closeness centrality

*Fig. 4.12:* Avg.cluster coefficient and centrality distribution across the year

### Behaviour across quarters of year

Similarly, we investigate how the network evolved across quarters of all years together. For each quarter, we explored the degree distribution and the cluster coefficients. The basic parameters are displayed in

Table 4.8.

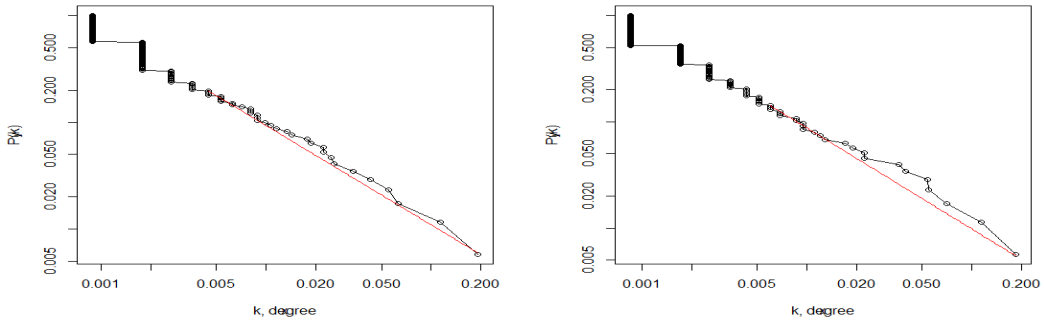
*Tab. 4.8:* Power-law fits and the corresponding p-value across the quarters

| Parameters | Qrt1 | Qrt2 | Qrt3 | Qrt4 |
|---|---|---|---|---|
| No.nodes | 1126 | 1165 | 914 | 1038 |
| No.links | 17060 | 17971 | 12831 | 15355 |
| Avg.C.C. | 0.578 | 0.630 | 0.575 | 0.586 |
| Alpha | 1.929 | 1.951 | 2.126 | 2.599 |
| x-min | 0.0044 | 0.006 | 0.010 | 0.022 |
| Gof test(p-value) | 0.3412 | 0.5014 | 0.83 | 0.641 |
| Ratio test (p-value) | 0.748 | 0.729 | 0.741 | 0.874 |

The second quarter has the highest number of chat room participants of about 1165 nodes with 17,971 edges while the third quarter has the lowest number of chat room participants with 914 nodes and 12,831 edges. Examining the power-law exponent and the p-value across all of the quarters revealed an appearance of the power-law in their degree distribution. Also, the power-law exponent for the quarter lies between 2 and 3, which shows that there is a relatively limited number of nodes with a big number of links.

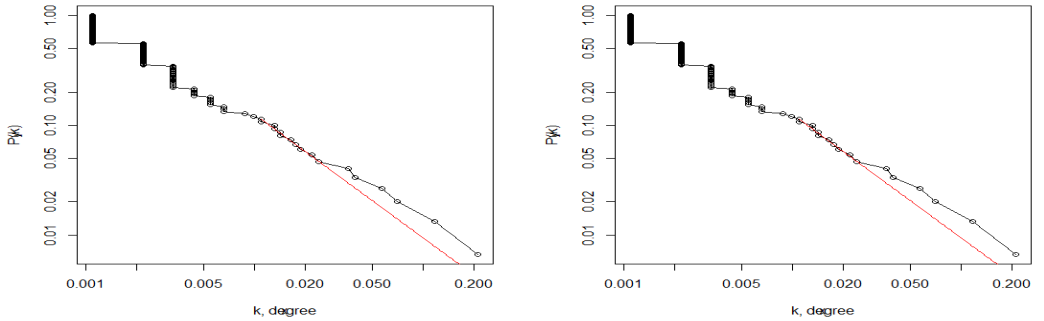
These also attest to the existence of hubs or a few people with a very large number of friends in the network. Furthermore, a plot of the cumulative distribution functions $P(x)$ across the quarter in Figure 4.13 clearly reveals that the degree distribution for all quarters is consistent with the power-law.

Figure 4.15(b) compares the node degree distribution across the quarter on a log-log scale plot.




(a) First quarter (b) Second quarter

*Fig. 4.13:* Network evolution: Power-law degree distribution across quarter




(a) Third quarter (b) Fourth quarter

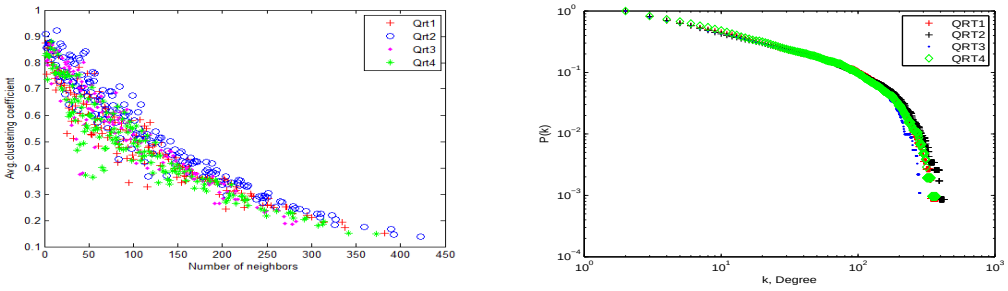
*Fig. 4.14:* Network evolution: Power-law degree distribution across quarter

We notice that the graphs across all the quarters are almost the same. In Table 4.8 we presented the average clustering coefficients across the quarters. The highest average clustering coefficient value (0.630) is in quarter 2, which suggests that the participants in the second quarter have more triangles than the rest of the quarters.

We plot the distributions of the clustering coefficients across quarters in Figure 4.15(a), which suggests that the chat room participants in the second quarter have a high number of triangles.

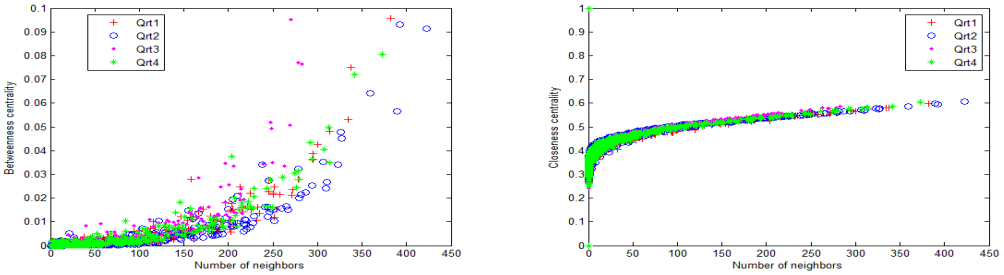
Moreover, the third quarter has the shortest graph with a low clustering coefficient, reaching only up to $k = 255$ and the graph of the fourth quarter extends longer than all other quarters reaching up to $k = 455$ neighbours. It is apparent that the fourth quarter has the least betweenness centrality value while the rest of the quarters seem to have the same betweenness centrality value. This may be the reason we have an overlap in the closeness centrality graph of the first, second and third quarters. This suggests that the three-quarters have the same speed of information spread.

In summary, though, there is an appearance of power-law degree distribution in all the quarters; however, we observed slight difference across the quarters. Quarter 2 has the highest number of




(a) Avg.clustering coeff. (b) Node degree distribution

*Fig. 4.15:* Avg.cluster, centrality and node degree distribution across quarter




(a) Betweenness centrality (b) Closeness centrality

*Fig. 4.16:* Avg.cluster, centrality and node degree distribution across quarter

triangles and the fourth quarter has the least betweenness centrality value while the rest of the quarters appear to have the same betweenness centrality value.

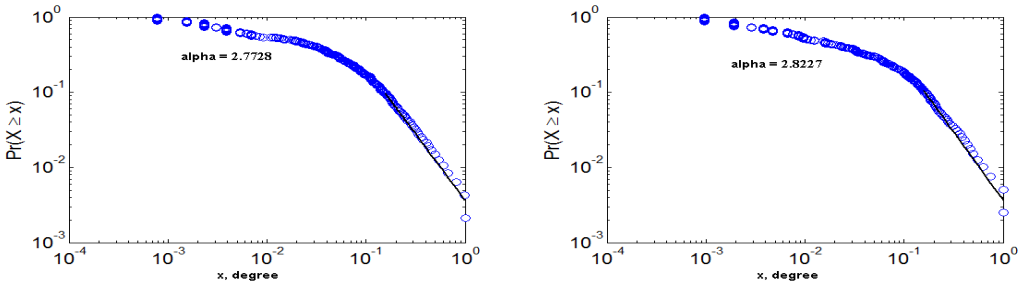
### 4.2.3 User's behaviour across the time of day

In a time-varying network, links exist for only short periods of time, then disappear and reappear again. To track and capture this rapid change over time we slice the network according to the time of day by dividing the hours of a day into 4 intervals of 6 hours each. Table 4.9 presents a summary of the chat room network across the time of day.

*Tab. 4.9:* power-law fits and p-value across time of day

| Parameters | 00:00AM - 6:00AM | 6:00AM -12:00PM | 12:00PM -6:00PM | 6:00PM -11:59PM |
|---|---|---|---|---|
| No.nodes | 1296 | 1049 | 1152 | 1456 |
| No.links | 16963 | 11570 | 16299 | 23285 |
| Avg.C.C | 0.591 | 0.514 | 0.588 | 0.635 |
| Alpha | 2.7728 | 2.8227 | 2.9432 | 2.8522 |
| x-min | 0.1590 | 0.1602 | 0.1962 | 0.1724 |
| $\rho - value$ | **0.3822** | **0.4065** | **0.4022** | **0.4387** |

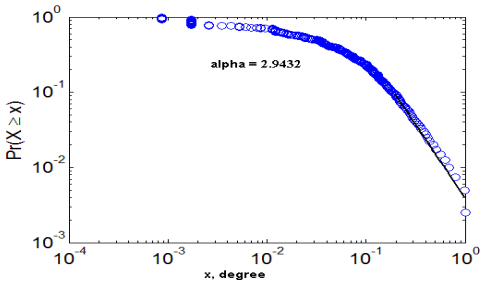
As we can see 6pm to 11:59pm has the highest number of participants or nodes (1456) with 23,285 edges while the least number of participants or nodes (1049) with 11,570 edges occurred between 6am to 12pm which indicate an existence of diurnal pattern in users behaviour. This suggests that more people tend to chat between 6pm and 11.59pm and less people chat between 6am and 12pm in this particular chat room. This is expected as the platform is often used on weekends and during non-working hours. In Table 4.9, all power-law exponents lie between 2 and 3, which implies the existence of a relatively small number of nodes with a very large number of links or attests to an appearance of hubs or people with a very large number of friends in the network.



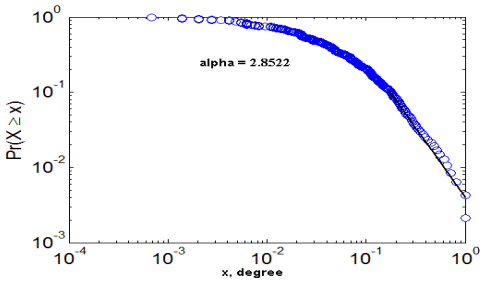

(a) 12am - 6am




(b) 6am - 12PM




(c) 12PM - 18PM

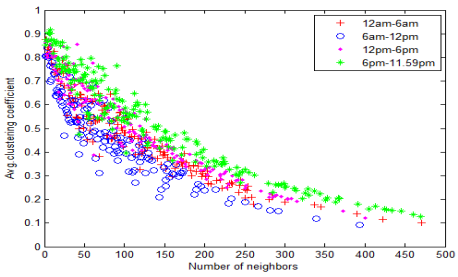



(d) 18PM - 23.59PM

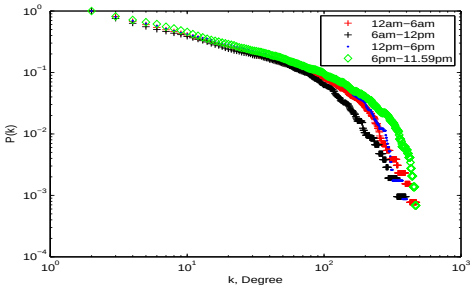
*Fig. 4.17:* Network evolution: Power-law degree distribution across period

The $\rho - value$ indicates an existence of power-law across time of day. Moreover, a plot of the cumulative distribution functions P(x) clearly reveals that and the existence of power-law (see Figure 4.17). A closer look at Table 4.9 shows that 6pm to 11:59pm has the highest average clustering coefficient value (0.635), suggesting a high number of triangles. In addition, Figure 4.18(b) compares the node degree distribution across the time of day on a log-log scale plot. We notice that the graph of 6pm to 11:59pm is higher and extended longer than the rest. The least average clustering coefficient value (0.514) falls between 6am to 12pm, suggesting a low number of triangles.

Also, a plot of the clustering coefficient distributions across time in Figure 4.18(a) clearly suggests that the chat room participants from 6pm to 11:59pm have the highest cluster coefficient value as well as the largest population while the chat room participants from 6am to 12pm have the least cluster coefficient value and less population.

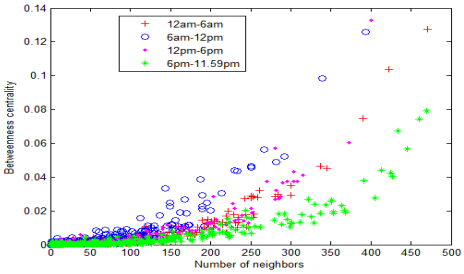



(a) Avg.clustering coeff.

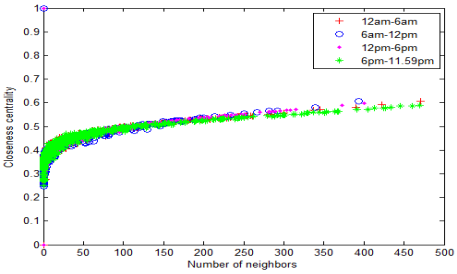



(b) Node degree distribution

*Fig. 4.18:* Avg.cluster coefficient, centrality and node degree distribution




(a) Betweenness centrality




(b) Closeness centrality

*Fig. 4.19:* Avg.cluster coefficient, centrality and node degree distribution

Moreover, a plot of the betweenness centrality against the number of neighbours in Figure 4.19(a) indicated that people who chat from 6pm to 11:59pm have a large number of neighbours but less betweenness centrality value while at other times of day with a small number of neighbours has a high betweenness centrality value. We observed that there was an existence of power-law degree distribution across the chatting period. Also, the results suggest that more people chat between 6pm and 11:59pm while fewer people chat between 6am and 12pm. Interestingly, people chatting from 6pm to 11.59pm are well connected during the rest of the other periods.

## 4.3 Summary

In summary, we show that our chat logs represent a human interaction by examining the characteristics of the chat rooms. Furthermore, we explored the temporal differences in Walfords chat log as it is our largest dataset.

We investigated the degree distributions, cluster coefficient and information flow across the year, the quarter of the year, time of day, weekdays and weekend. The degree distributions show that a large proportion of the users only chat with one other. At the other extreme, there are few users that chat with a very large group of users. Even more, the tail of the degree distribution is well approximated by a power-law, reflecting that there is no average user in the chat rooms. In addition,

most of the temporal difference has a high value of clustering coefficient, suggesting a presence of a high number of triangles in the network, which is evidence of people chatting with one another. Lastly, the high values of betweenness across the temporal differences confirm an existence of few people with a very large number of friends in the network.

## 5. USER BEHAVIOUR DYNAMICS FOR PAIR CONVERSATION

Having explored the network structure of our chat room in chapter 4, our goal in this chapter is to investigate users behaviour through Response Waiting Time (RWT). The statistics and dynamics of online social networks generated by the way users behave online are of enormous importance to social networking service providers, sociologists, linguists and those interested in online commerce. Investigating distribution could provide information about the dynamic process that takes place in a real network. For now, our study will focus on the dynamics of RWT for pairs of people in a chat room. The waiting time in chat room communication can be defined as the time difference between successive messages between two people. Mathematically, the waiting time is given by:

$$dt = t_{i+1} - t_i,$$

where $t_i$ is the time at i and $t_{i+1}$ is the time at i+1.

```
1
2   40:29 A→(B):grins I think it's the proxy
3                  Kevin and Perry that need kicking!
4   40:55 B→(A):what happened last night..the
5                  lot of it got or needed a kicking!
6   41:13 C→(D): lsaysl cH kissing bandit...l
7   41:45 H→(I):Kissing bandits are  predators
8                   should not be tolerated
9   41:46 A→(B): it was a Janet router that went,
10                 second tie in a week one has died
11  42:08 D→(C):lsaysl cYou're just jealous he
12                 took your job
13  42:16 B→(A):grins janet is the of the network
14                 the universities and schools are on.
15                 router is something that forwards on
16  42:21 I→(H): And I haven't gotten any action since
```

*Fig. 5.1:* Sample of a conversation from our corpus.

For example, in Figure 5, let us assume that three pairs of conversation are going on: $A \to B$ and $B \to A$, $C \to D$ and $D \to C$, $H \to I$ and $I \to H$. The response waiting time distribution between the pair of people A and B is 29 seconds $(40:55 - 40:29)$, 51 seconds $(41:46 - 40:55)$ and 30 seconds $(42:16 - 41:46)$. The response waiting time between the pair of people C and D is 55 seconds $(42:08 - 41:13)$. The response waiting time between the pair of people H and I is 36 seconds $(42:21 - 41:45)$. Then, we applied the maximum likelihood method to estimate the power-law scaling parameters for Walford chat room logs and T-REX chat room logs.

## 5.1 What are the statistics of RWT in our network

There have been a significant number of studies which rely on the statistics of Response Waiting Time (RWT) to understand users behaviour. Although, previous research have shown that the degree distribution of users RWT in email, tweeter etc. exhibit power law, i.e., simple scaling, However, not much work has been done on the temporal variation in RWT. So for each chat room, we unveil the hidden statistics for RWT.

### 5.1.1 Modelling Walford chat log

As Walford logs is the largest data set we have, we will resume our analysis with it. We first examined the distribution of the RWT by plotting the Complementary Cumulative Distribution Function (CCDF) for the waiting time shown as in Figure 5.2. Unlike many empirical data in nature the RWT distribution is not a pure power law rather the graph reveals several distinct regions.

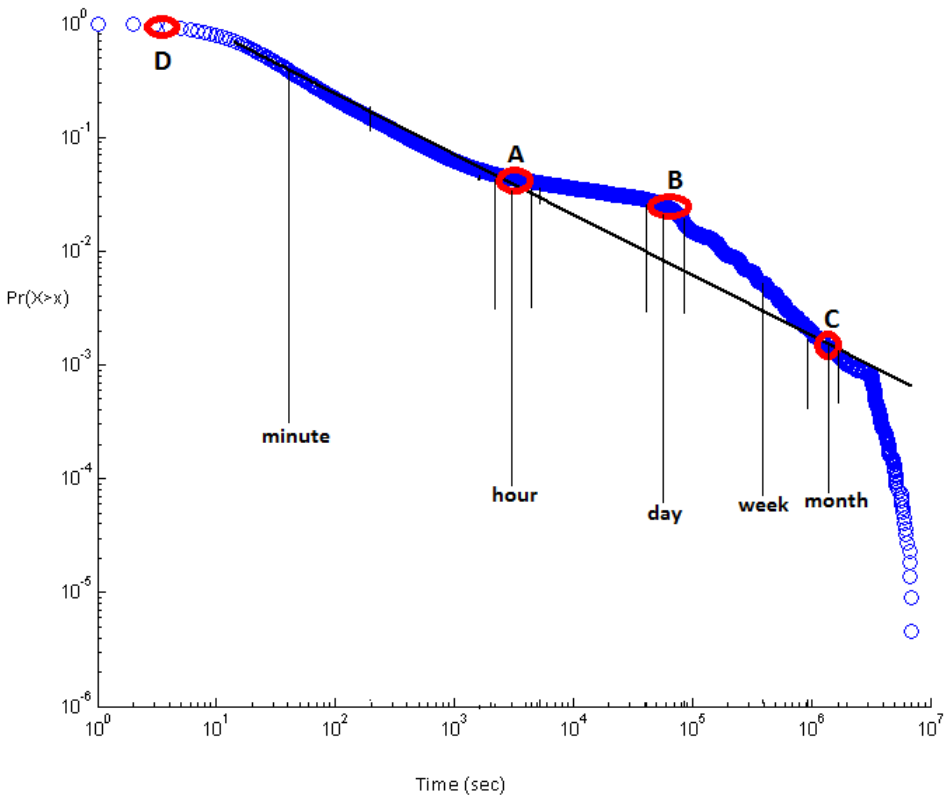



*Fig. 5.2:* Waiting time

We note that after one hour, the behaviour of waiting times for message responses exhibits a different pattern. Up until one hour, the distribution is in the form of a power-law with an exponent of 1.53, and then beyond one hour, the graph suddenly deviates. This indicates that they are different factors that influence the RWT at various time scales. Finally, the curve drops off sharply near $10^7$

seconds.

To test these hypothesis, we use the bootstrap procedure. Before applying this procedure, we first selected where the dataset is greater than X-minimum and the remaining number of rows in our dataset reduced from 78916 to 1822. The result of the bootstrap procedure is displayed in Figure 5.3 with 0.110 goodness-of-fit and 0.018 ratio test which suggest that the model does not provide a plausible fit to the data and another distribution may be more appropriate.

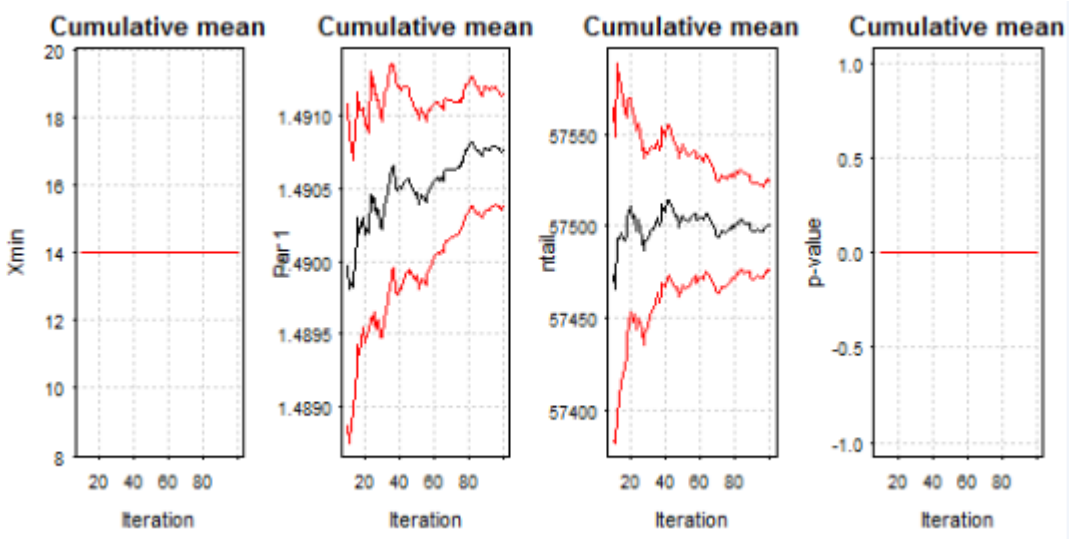



*Fig. 5.3:* The alpha value and Xmin are 1.49 and 14 respectively. Goodness of fit done using the maximum likelihood method [59]. The p-value which is the goodness-of-fit and ratio test metric is 0.110 and 0.018 respectively

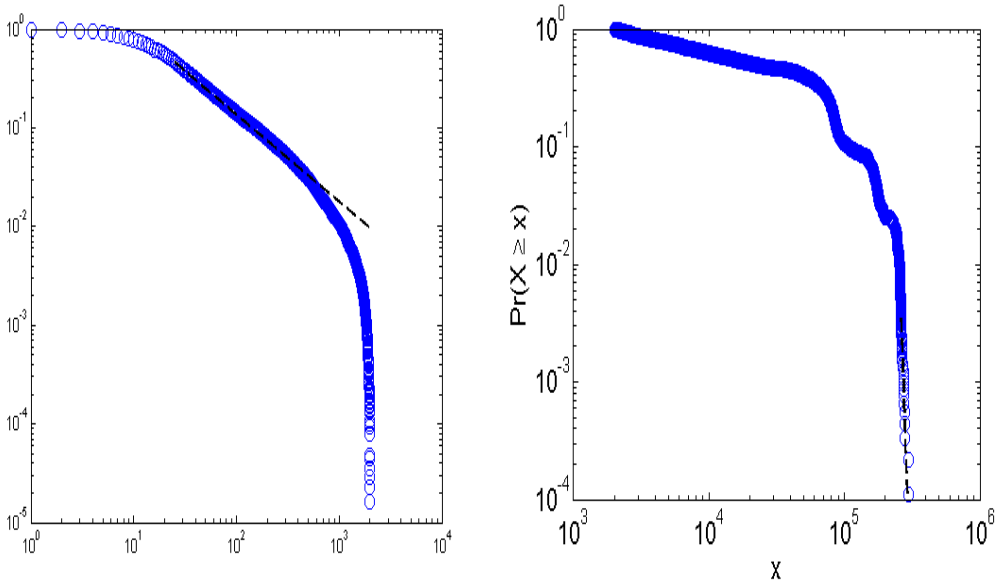
.

For simplicity, we divided the RWT distribution into two major regions. As we noted above that the behaviour of waiting times for message responses exhibits a different pattern after one hour. Hence, region 1 represents where the waiting time(seconds) is less or equal to one hour while region 2 represent areas where the waiting time is between one hour and a month.

| Message | Time(Seconds) |
|---|---|
| loves being IGNORED! | 66 |
| lwonders why monthspod hasnt updated since the 15th.... | 65 |
| saysL gLBut I have directory forwarding on, when I type http | 64 |
| ooeeerrr we are top end heavy today.... lol | 64 |
| http | 63 |
| yawns...bored with article writing now...need some inspiration.... | 62 |
| llgasps in astonishment! | 60 |
| thinks l . o O (I Positive feedback | 60 |
| saysL gLWell, at least they aren't animated. | 60 |
| dsays lMy sheep isnt happy.d to lCherubd. | 58 |
| saysL gLNice, but slow compared to 350-400 from Microsoft ;p.L | 56 |
| lsaysl cYou're just jealous he took your job | 55 |
| glwonders if Dafty would let us have access to modify the code ;) d To lxalond] | 55 |
| hey hey | 54 |
| saysL gLDamn. I was going to sneak a button in there ;)L | 53 |
| saysl wBeats sitting around ****ing all day. And he gets paid 65k a year to boot. | 51 |

*Fig. 5.4:* Sample

For example, considering the sample in Figure 5.4, the messages where time is less or equal to 60 seconds will be in region 1 while messages where the time is greater than 60 seconds and less than a month will belong to region 2.

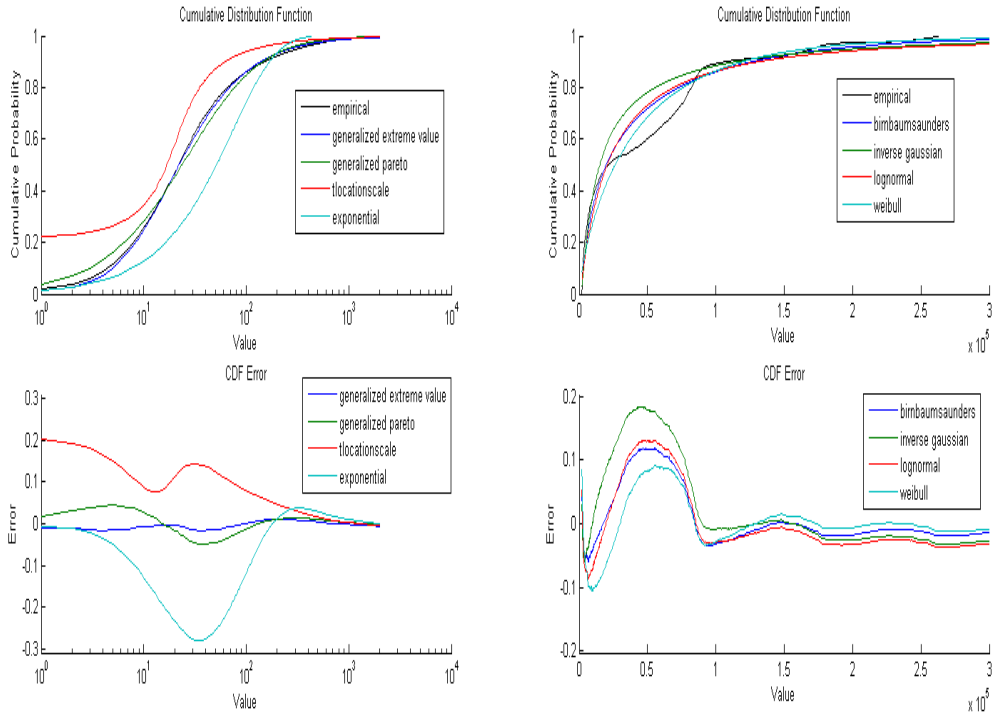



(a) Pair: D - A (b) Pair: A - C

*Fig. 5.5:* Response waiting time for a Pair of people

We extracted the waiting time distribution in the two regions and examined them separately as shown in Figure 5.5 and 5.6. It shows that short waiting time can be described with power law distribution while long waiting time does not follow power law distribution which suggests more than one user pattern in the network. Furthermore, we explored MatLab function called allfitdist to select the distribution that best described the two regions. Allfitdist function fits all valid parametric distributions to the data and sorts them using a metric such as. BIC or AIC then, finds the best distribution that describes the data. The model with the lowest BIC is preferred. The analysis in Figure 5.6 shows that short waiting time is best described by generalized extreme value distribution which is one of the power law family. On the other hand, long waiting time is described using Birnbaum-Saunders distribution. These graphs suggest that response waiting time do not exhibit a simple power law rather it is a multi-scaling network.

To further understand the underlying factors that may be responsible for the RWT behaviour in figure 5.2, we compared the two regions based on their word content and the way the users interact with each other to exchange information. We start by extracting the word content of the two regions and then, used cosine measure to compare the word similarity between them. The result shows that the word content in the two regions is less similar with a cosine measure of 0.130. From this result, we see the correlation between the response waiting time and the type of words people use when




(a) Pair: D - A

(b) Pair: A - C

*Fig. 5.6:* CDF for the curve region

chatting. The dominant words in the two regions are displayed in Table 5.1. Term and Term2 are the dominant words that appear within a region, Freq is the number of times a word occurred in the region and NumUtt is the number of utterance in which a particular word occurs in the region.

Next, we compared the statistics of the network structure in the two regions since user activities or behaviours give rise to network structure formation, suggesting a possible link between RWT and network structure. A network is made up of users (nodes) and the links between them. The way these links (edges) are organised has a big effect on who gets what information. The first network property to be examined is the degree distribution; the degree distribution of a node (user) k is the number of edges that have k as a vertex.

First we extracted every pair of communication which have a response waiting time less or equal to an hour (see Figure 5.2). This represents a network of users who took less or equal to an hour to respond to messages and it forms region 1. Secondly, region 2 is formed by extracting every pair of communication which have a response waiting time greater than an hour but less or equal to a month (see Figure 5.2). Region 2 represents a network of users who took greater than an hour but less or equal to a month to response to messages. We then, examined the degree distribution of each region.

The analysis of the Cumulative Distribution graph suggests that the degree distribution is higher

*Tab. 5.1:* A sample of the Predominant words in each region

| Terms in Regoin 1 | Freq | No. of Utt | Terms2 in Region 2 | Freq | No. of Utt |
|---|---|---|---|---|---|
| heeheeing | 290 | 290 | jason1 | 858 | 858 |
| seconds | 90 | 90 | vock | 807 | 807 |
| spin | 37 | 37 | Imeg | 622 | 622 |
| dhour | 27 | 37 | jupiter | 571 | 571 |
| largliquidlikeess | 26 | 26 | knight | 514 | 514 |
| ixalon | 19 | 19 | anne | 478 | 478 |
| triplewordscore | 19 | 19 | lorass | 448 | 448 |
| mike | 14 | 14 | dstaks | 445 | 432 |
| llilith | 13 | 13 | jazz | 411 | 409 |
| glerror | 11 | 11 | dave | 358 | 345 |
| ginge | 9 | 9 | wild | 355 | 354 |
| leerald | 8 | 8 | child | 319 | 317 |
| ljillian | 8 | 8 | toyboy | 171 | 167 |
| leeraldserpent | 7 | 7 | mistress | 167 | 165 |
| allnightertonight | 7 | 7 | mars | 169 | 166 |
| allnighter | 7 | 7 | sir | 160 | 157 |
| ginus | 7 | 7 | light | 143 | 143 |
| Iiris | 6 | 5 | connected | 131 | 128 |

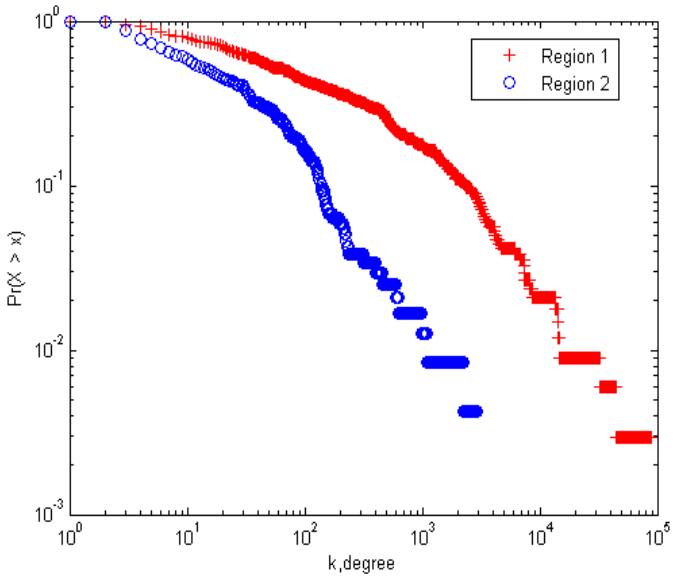



*Fig. 5.7:* Degree distribution

and longer in region 1 (Region with shorter waiting time) compare to region 2 (Region with a longer waiting time). This indicates that Region 1 is denser than region 2. Secondly, the degree distribution in Region 2 clearly exhibits multi-scaling behaviour i.e. a simple power law can not describe it. The second network property we examined is the clustering coefficient that is a measure of the extent to which one's friends are also the friends of each other. The clustering coefficient of region 1 and region 2 are 0.54 and 0.30 respectively.

*Tab. 5.2:* Topological characteristics

| Region | 1 | 2 |
|---|---|---|
| No. of Node | 338 | 186 |
| No.of Edges | 1074 | 519 |
| Clustering coef | 0.54 | 0.30 |
| Network density | 0.030 | 0.019 |

This indicates that region 1 (with short waiting time) has more triangles compare to region 2 (with long waiting time). Table 5.2 displays the additional network properties such as node, edges, clustering coefficient and network density for each region. So, we see a clear difference in their network properties, Region 1 is a dense network with more triangles, hence short waiting time while Region 2 is the less dense network with long waiting time.

### 5.1.2 Modelling IRC chat log

The second dataset we analysed is IRC chat log. A plot of the response waiting time distribution for all the pairs of people in Figure 5.8 shows that it is also not a pure power law distribution; rather it possesses a more complex pattern, which is evident of a multi scaling behaviour.

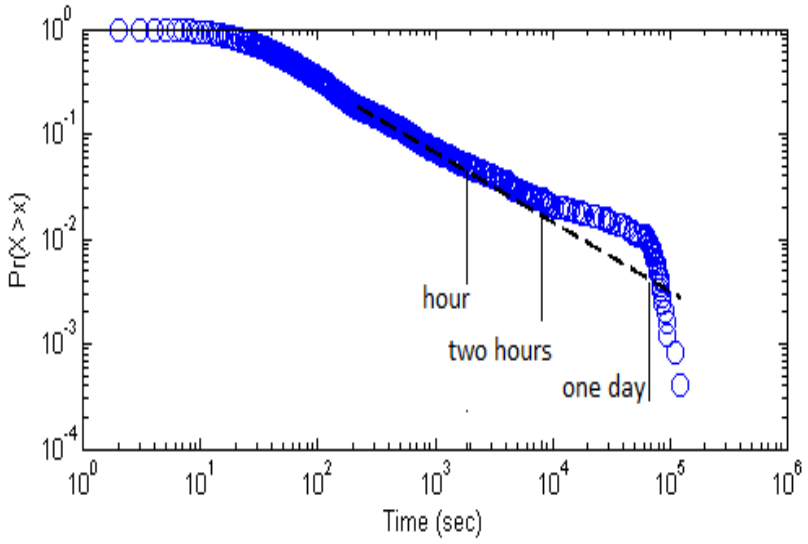



*Fig. 5.8:* Waiting time

Up until two hours, the behaviour exhibits a power-law distribution, and then beyond two hours, the behaviour of the response waiting times becomes different (the graph suddenly deviates with a sharp curve).

Also, we tested the hypothesis using the bootstrap procedure and Figure 5.9 suggest that the model does not provide a plausible fit to the data and another distribution may be more appropriate.

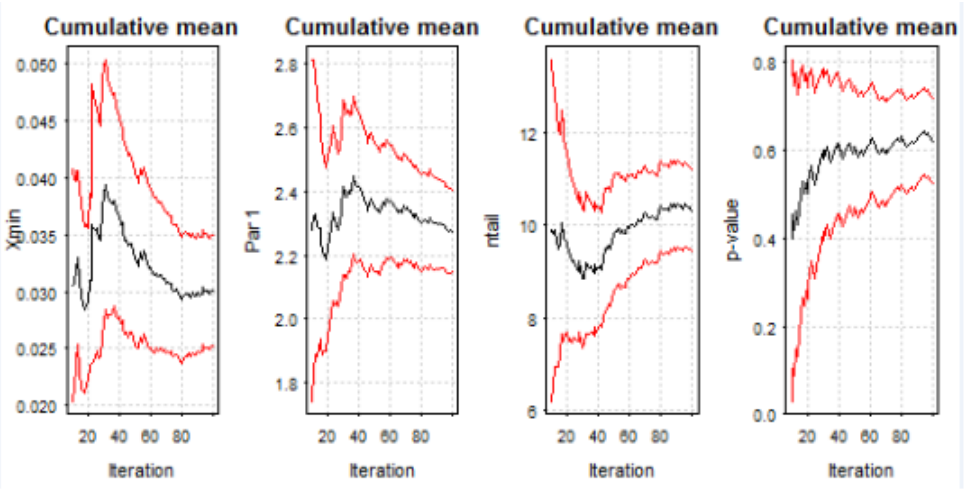



*Fig. 5.9:* The alpha value and Xmin are 1.59 and 192 respectively. Goodness of fit done using the maximum likelihood method [59]. The p-value which is the goodness-of-fit and ratio test metric is 0.85 and 0.48 respectively

.

We then divided the distribution into two major regions. Region one consist of users with waiting time less or equal to two hours and can be described using power law while region two are users whose waiting time is between two hours and a day. To evaluate the correlation between RWT and network structure, we compute and compare the degree distributions of the two regions in Figure5.10. The degree distribution of region 1 is higher and extends longer compare to region 2; indicating that region 1 is a dense network with shorter waiting time while region 2 is less dense with longer waiting time.

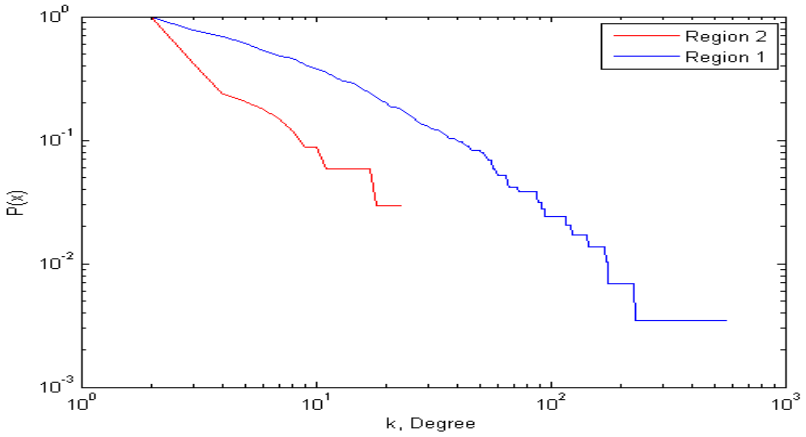



*Fig. 5.10:* Degree distribution

### 5.1.3 Modelling T-REX chat log

Lastly, we will consider T-REX dataset. The RWT is displayed in Figure 5.11. The graph clearly suggests that the RWT for all the pairs of people exhibits more complex pattern, just as the response

waiting time in Figure 5.8; indicating an existence of multi scaling. The behaviour of the response waiting times becomes different after one and half hour. Up until one and half hour, the distribution is in the form of a power-law with an exponent of 1.53, then, beyond one hour, the graph suddenly deviates with a sharp curve and finally drops off sharply. This suggests that waiting time consists is influence by different factors at different time scales.

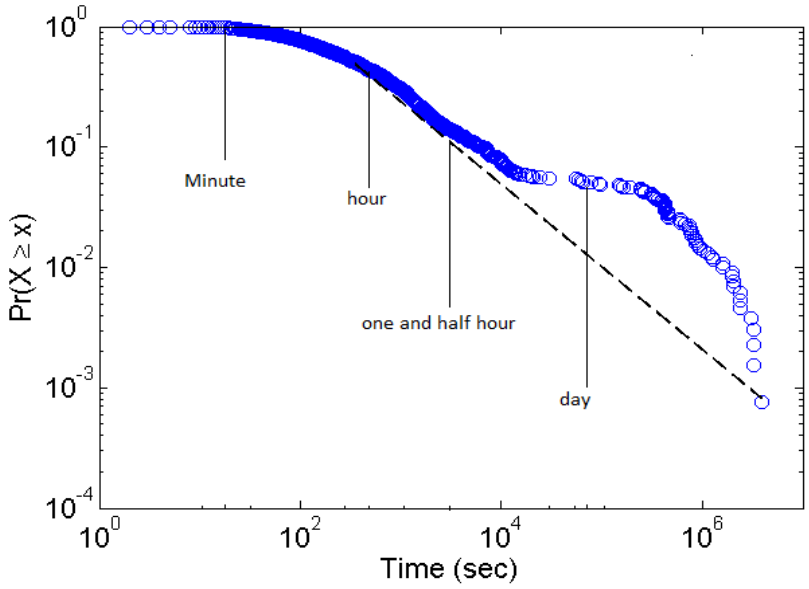



*Fig. 5.11:* Waiting time

In testing for the hypothesis, we first created a subset of the data set for X greater than the X-minimum and then use the bootstrap procedure to estimate the goodness-of-fit and the ratio test. The result which is shown in Figure 5.12, suggest that the model does not provide a plausible fit to the data and another distribution may be more appropriate.

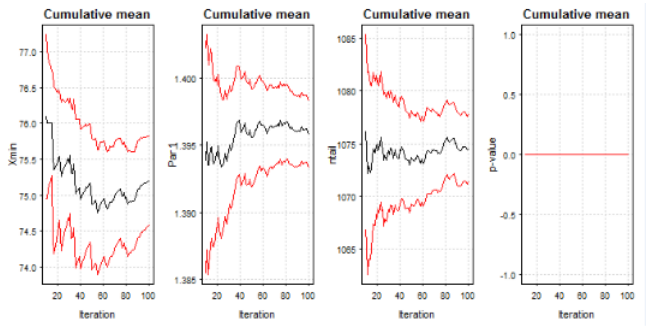
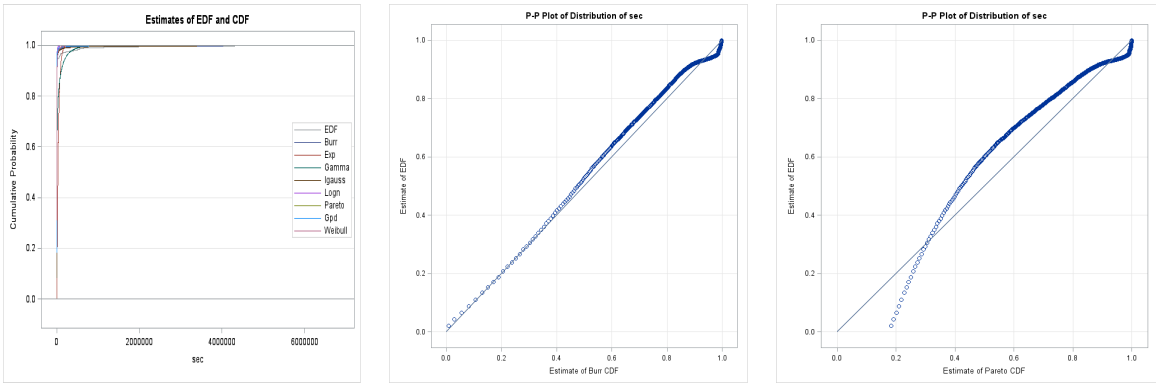



*Fig. 5.12:* The alpha value and X-min are 1.39 and 76 respectively. Goodness of fit done using the maximum likelihood method [59]. The p-value which is the goodness-of-fit and ratio test metric is 0.102 and 0.006 respectively.

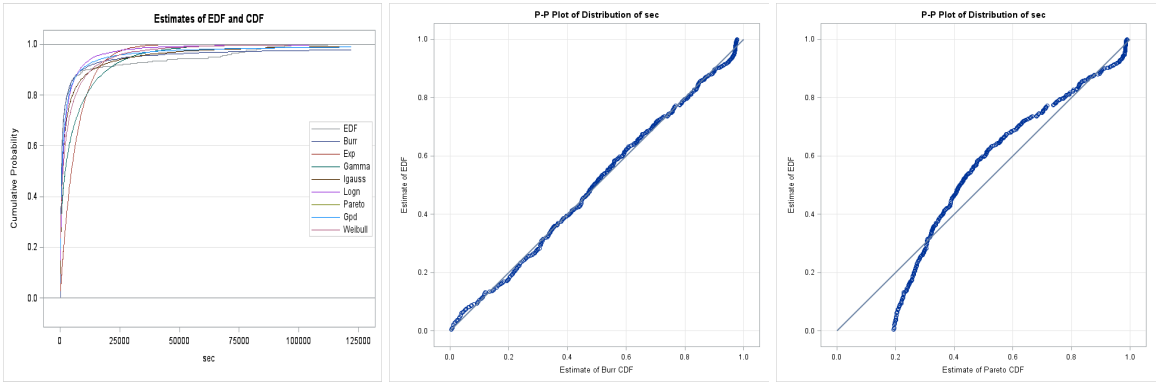
### 5.1.4 Comparing results from the fitted Power law with other distributions

Our results suggest that we cannot best describe RWT in a chat room using power law distribution. So to determine the distribution that best describe the RWT, we will compare two or more distributions. To do this, each distribution must have the same lower threshold. So we first subset the distribution to have the same x-min as the distribution we used in fitting power law. We fitted the power law above with the following X-min: 14 for Walford, 192 for IRC and 76 for T-REX chat log. Figure 5.13 - 5.15 show that the distribution of response waiting time is quite closer to burr than pareto(power law) which suggest that Burr as the best distribution to describe response waiting time in an on-line chat room. In the next section, we will discuss temporal variation that occurs in RWT.




(a) Waldord CDF (b) Walford: EDF vers CDF for Burr (c) Walford: EDF vers CDF for Pareto

*Fig. 5.13:* Comparison

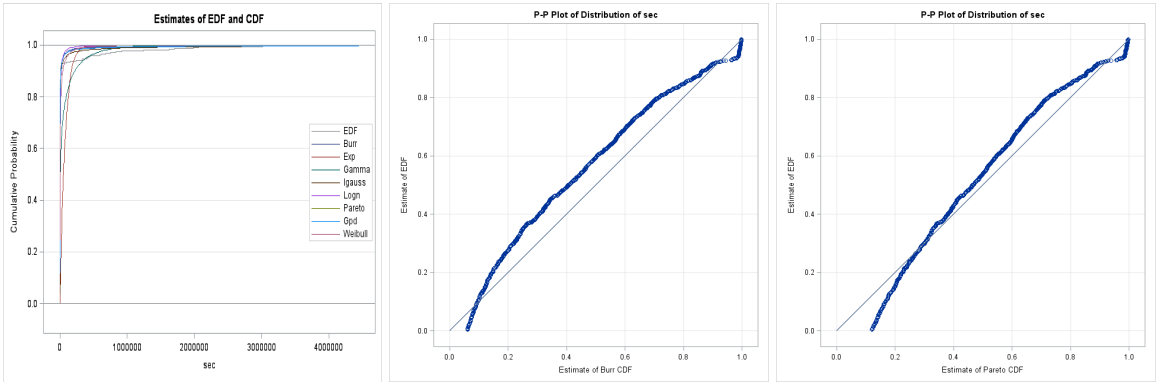



(a) IRC CDF (b) IRC: EDF vers CDF for Burr (c) IRC: EDF vers CDF for Pareto

*Fig. 5.14:* Comparison

## 5.2 Temporal variation in RWT

Some previous studies have relied on the analysis of the Response Waiting Time (RWT) to characterise users behaviour. So, in this section, we tend to use the temporal variation in RWT to




(a) T-REX CDF (b) T-REX: EDF vers CDF for Burr (c) T-REX: EDF vers CDF for Pareto

*Fig. 5.15:* Comparison

understand users behaviours dynamics in a chat room.

### 5.2.1 Effect of communication count on response waiting time

Using Walford chat logs, we investigated the impact of communication count (number of messages exchanged between pairs of people) on response waiting time (see Figure 5.16).

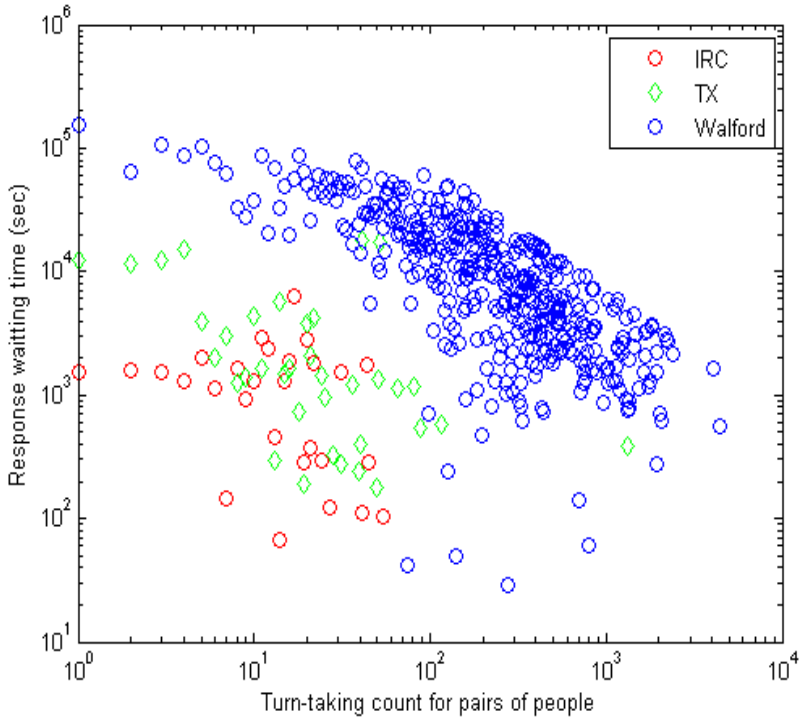



*Fig. 5.16:* RWT vs Turn-taking

For each pair, we counted the number of times they have sent messages to each other and the average of their response waiting times. A plot of the averaged RWT and the communication count

for pairs of people reveals that the RWT decreases as the communication count increases. Secondly, analysis of IRC chat room, which has a different technology (mode of operation), indicates an apparent effect of communication count on RWT (see Figure 5.16). RWT tend to be shorter as the communication count increases. Lastly, the result of T-REX chat logs analysis in Figure 5.16, inversely associated communication count with RWT, suggesting a relationship between communication count and the RWT. This suggests that pairs of participants with a low number of turns are associated with prolonged response waiting time while pairs of participants with a high number of turns are related to short response waiting time. If two people communicate to each other more often, they get to know each other and have a high tendency of responding to each other messages quickly.

### 5.2.2 The RWT considering one user with other participants

In another strand of analysis, we investigated the RWT considering one user with other participants in three different chat rooms: Walford, IRC and T-REX. Table 5.3 and Table 5.4 show two users and their RWT pattern with others in Walford chat room. In Table 5.3, we show a real-time modelling of waiting time for participant A with others.

*Tab. 5.3:* Waiting time between user A and others

| Num | Users | dist | $\chi$ | df | P-v | alpha |
|---|---|---|---|---|---|---|
| 1 | A $\rightarrow B$ | zm | 0.40 | 3 | 0.93 | 0.59 |
| 2 | A $\rightarrow C$ | gp | 2.86 | 6 | 0.82 | 0.91 |
| 3 | A $\rightarrow D$ | gev | 3.61 | 7 | 0.72 | 1.57 |

*Tab. 5.4:* Waiting time between user B and others

| Para | B$\rightarrow C$ | B $\rightarrow D$ | B $\rightarrow A$ |
|---|---|---|---|
| Distri | InvGaus | Exp | gp |
| $\mu$ | 147544 | 89568.2 | |
| | (106822) | (2982.3) | |
| $\lambda$ | 9.672 | | |
| | (0.801) | | |
| $\alpha$ | | | 2.14 |
| | | | (0.15) |
| Sig | | | 52.30 |
| | | | (2.17) |

First we computed the RWT for different pairs of conversation involving user A (A and B, A and C, A and D) and applied maximum likelihood method to estimate their statistics. The RWT of

number 1 ($A \rightarrow B$) has a Zipf-Mandelbrot(zm) distribution with an alpha value of 0.5995, indicating long waiting period while number 2 ($A \rightarrow C$) and number 3 ($A \rightarrow D$) follow generalised Pareto distribution and generalised extreme value distribution with an alpha value of 0.9176 and 1.5700 respectively. This indicates that a single person chatting with others can have several different responses waiting time patterns.

Similarly, Table 5.4 shows a waiting time for participant B with others (B and C, B and D, B and A with Inverse Gaussian (InvGaus), exponential (Exp) and generalised Pareto (gp) distributions respectively). Hence, one participant, due to some interference factors can exhibit different waiting time pattern. Furthermore, investigating a single user response waiting time in IRC chat logs and T-REX chat logs show a similar pattern as in Walford chat room.

*Tab. 5.5:* Waiting time between user G and others

| Num | Users | dist | $\chi$ | df | P-v | alpha |
|---|---|---|---|---|---|---|
| 1 | G $\rightarrow H$ | zm | 4.20 | 50 | 0.52 | 0.57 |
| 2 | G $\rightarrow K$ | gp | 5.93 | 60 | 0.23 | 0.30 |
| 3 | G $\rightarrow L$ | gev | 4.37 | 60 | 0.62 | 1.34 |

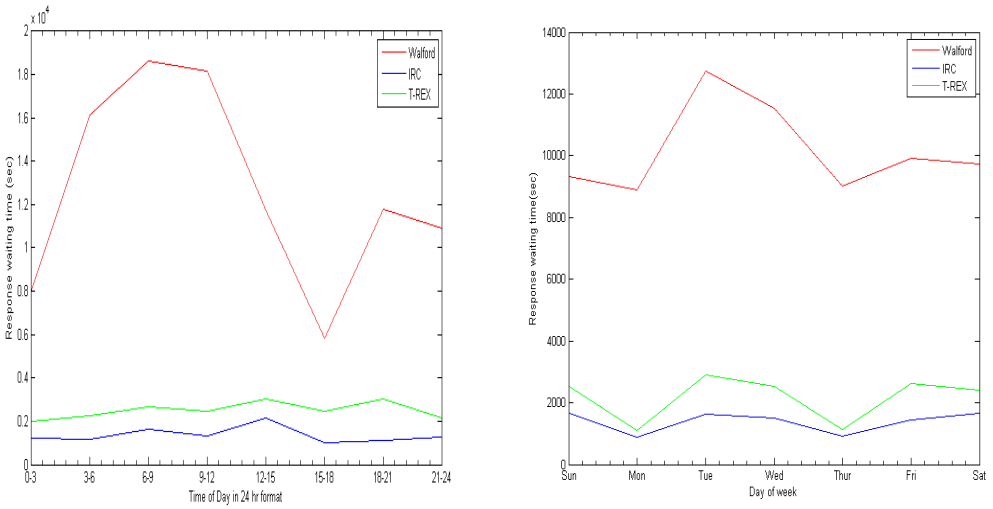
Table 5.5 shows the diverse distribution of one user G response waiting time with others in IRC ($G \rightarrow H$, $G \rightarrow K$ and $G \rightarrow L$) while Table 5.6 displays the changes in the response waiting time distribution of single user with respect to other users in T-REX chat room ($W \rightarrow X$, $W \rightarrow Y$ and $W \rightarrow Z$). The distribution of this response waiting time covers Zipf-Mandelbrot, generalized extreme value, generalized Pareto and Weibull distributions. The response waiting of $A \leftrightarrow B$ in Table 5.3 and $B \leftrightarrow A$ in Table 5.4 appear to have a different distribution. This indicates that pairs of people involve in conversation $A \leftrightarrow B$ may not necessarily have the same response waiting time distribution or behave alike. Our result shows that an individual can have several waiting time depending on the interference factors which may be fatigue, lack of interest, lack of communication, boringness, lack of interest or concentration etc. This suggests that communication dynamics depends on the group or pairs rather than being simply about the individual. For more on the variations in RWT distributions for pairs of people see Table C.1 - C.3 and Figure C.8 - C.16 in the appendix.

*Tab. 5.6:* Waiting time between user W and others

| Num | Users | dist | $\chi$ | df | P-v | alpha |
|---|---|---|---|---|---|---|
| 1 | W $\rightarrow X$ | zm | 0.89 | 103 | 0.83 | 0.75 |
| 2 | W $\rightarrow Y$ | gp | 1.75 | 86 | 0.94 | 2.93 |
| 3 | W $\rightarrow Z$ | gev | 2.01 | 74 | 0.92 | 1.48 |

### 5.2.3 Time of Day, Day of Week and RWT Interaction

We examined the response waiting time in relation to the time of day by slicing the hours of a day into 8 intervals of 3 hours each. For each slice, we aggregate the response waiting time and plot the aggregated distribution as seen in Figure 5.17(a). Response waiting time appears to have the same pattern in IRC and T-REX chat room but slightly different in Walford, suggesting that IRC and T-REX chat room may have been used for the same purpose while Walford is used for different purpose. Also in the three logs, the users RWT behaviours shows a predominately diurnal component of variation. Starting with short waiting time, very early morning, just after midnight gradually increases during working hours and then, the responses waiting time begin to fluctuate. This indicates that users responses to messages are unpredictable within this time.



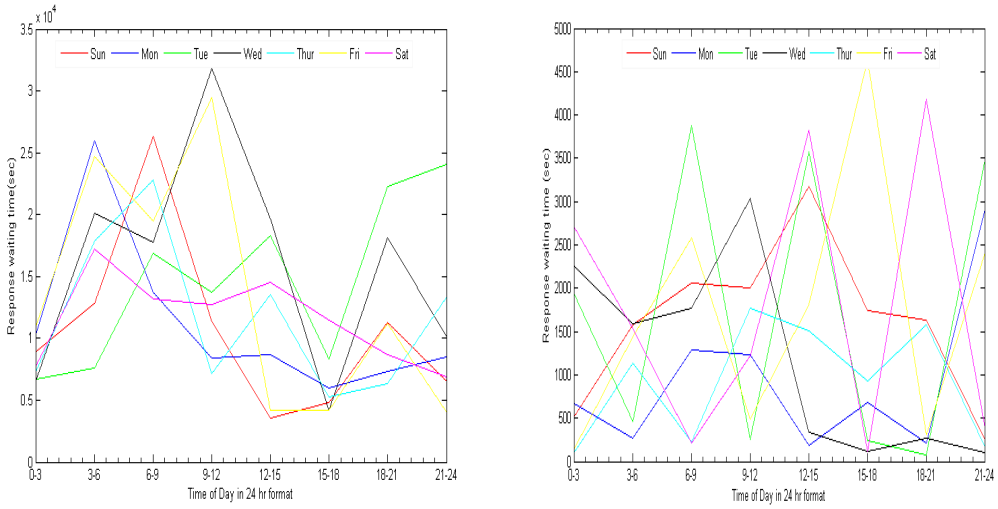

(a) Time of day

(b) Day of Week

*Fig. 5.17:* Time of day

Likewise, we see the similarity between IRC and T-REX chat rooms when we conduct the same RWT analysis over the days of week in Figure 5.17(b). Starting with relatively long waiting time on Sunday, short waiting time on Monday, long waiting time on Tuesday and Wednesday, short waiting time on Thursday and then, short waiting time on Friday. One possible reason for this is that users tend to respond to message quickly on Monday just after the weekend and probably get involve in many task on Tuesday and Wednesday. Then, on Thursday, users start preparing toward weekend by responding to messages which have been waiting since Tuesday. Unlike IRC and T-REX, Walford chat logs appear to have short waiting time on weekends and long waiting time on weekdays. Probably this Walford platform is often used on weekends and non-working hours. Lastly, we look at the interaction among response waiting time, time of day and day of week. For each day in a week, we divide the hours into 8 intervals of 3 hours each, aggregate the response waiting time in

each interval and then, plot the response waiting time against the day of time for each day of week (see Figure 5.18).




(a) Walford (b) IRC

*Fig. 5.18:* Day of week and Time of day interaction

Walford and IRC seem to exhibit different patterns. In Walford, each day tends to start with a short waiting time and gradually increases together. Beyond this point, they started exhibiting different response waiting time pattern. In comparison with other days, on Sunday and Saturday, the response waiting time appears to be low throughout the time of day. The shortest RWT occurred on Sunday between 15:00 PM to 18:00PM while the longest RWT occurred on Wednesday between 12:00 PM to 15:00 PM. Similarly, in IRC, we notice that RWT on Monday and Thursday appears to be low throughout the time of day compare to other days of week.

#### Results compared with previous models

In Table 5.7, we compare our results with the previous results reported by others on the same task [8]. Our analysis which focuses on three different chat rooms yield results that is slightly different from previous RWT models. In all the dataset (Walford chat logs IRC chat logs and T-REX chat logs) our model indicates an existence of multi scaling behaviour between the RWT for pairs of people in a chat room. The Time at which the graph Deviates from Power Law (TDPL) is within 2 hours for chat rooms, one day for twitter and none for email.

### 5.2.4 Temporal Variation on Weekdays and Weekends

Having explored the impact of communication count on RWT and the diversity of one user RWT, we further examined the temporal variation in RWT considering the users environment. Users behave

*Tab. 5.7:* Comparison with other models. Source for E-mail and Twitter: [17] and [8]

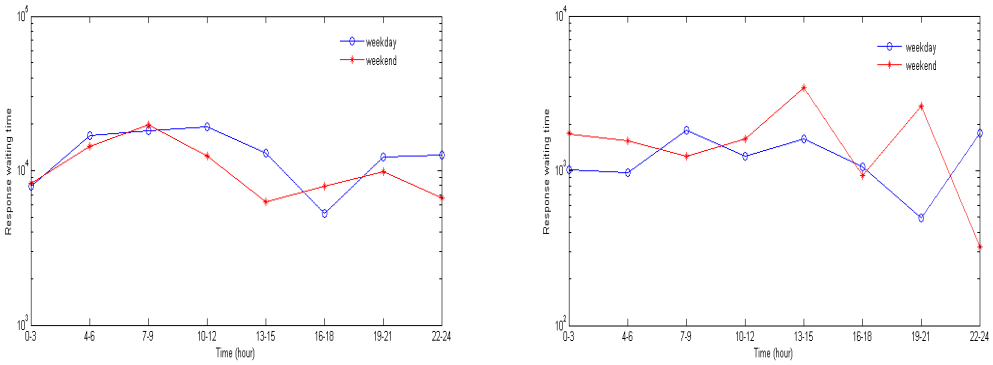
| Dataset | Scaling | TDPL | $\alpha$ |
|---|---|---|---|
| E-mail(A.L. Barabasi model) | Simple | Non | $\pm 1$ |
| Twitter(G.Comarela model) | Multi | after 1 day | |
| Our models | | | |
| IRC | Multi | after 2 hour | 1.6 - 2.8 |
| Walford | Multi | after 1 hour | 0.3 - 3 |
| T-REX | Multi | after $1\frac{1}{2} hour$ | 1.5 - 2.3 |

differently on weekdays and weekends due to factors like working on weekdays and reserve leisure time for weekends. We investigate if this is reflected in the response waiting time of a chat room.

**Time of Day Analysis for weekdays and weekends**

We analyse the user behaviour based on the response waiting time and time of day. After dividing the RWT into weekdays and weekends, we slice the hours of a day into 8 intervals of 3 hours each and study the changes in the RWT across the time of day. Figure 5.19 shows the trend of users RWT across the time of day on weekdays and weekends are slightly dissimilar in both Walford and IRC chat logs. In Walford chat logs, we observed shorter waiting time at the very early morning, just after midnight, longer waiting time during work hours of the day, shorter waiting after work hours in the evening and longer waiting time late in the night. A closer look shows that on weekdays and weekends the RWT have a similar increasing trend at the very early morning, just after midnight, when most people are asleep and activity is at a low level. After 0700-0900 hours which is the peak, the RWT gradually reduces on weekends (because most of the users are at home) and the shortest RWT on weekends occurs between 1300 and 1500 hours. However on weekdays, the RWT started to decline at 1200 hour (probably lunch break) and the shortest RWT occurs just after working hours. Considering IRC chat logs, we notice a short RWT at the very early morning, just after midnight which slightly increased during working hours of the days on weekdays. RWT started declining after working hours and the shortest RWT occurs between 1600 and 1800 on weekdays.

Shorter RWT is observed on a weekend while longer RWT is noticed on weekdays in both IRC and Walford logs within these hours. In comparing the two chat room, we observed weekend users have the shortest RWT during working hours (0900 - 1500 hrs) in Walford. This suggests that Walford chat room is a platform more frequently used during weekends and non-working hours. On the other hand, IRC appears to have the shortest RWT which covers a relatively long period of time (0900 - 1600 hrs) during working hours on weekdays. This indicates that IRC is a platform more frequently used during weekdays and working hours.

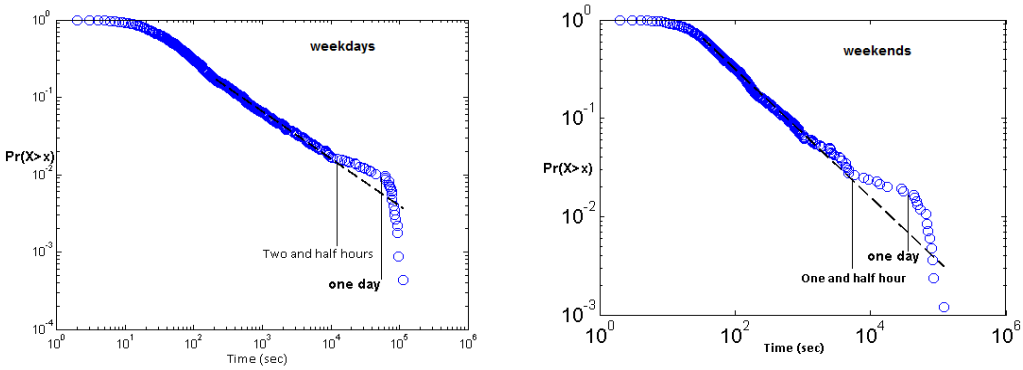



(a) Walford (b) IRC

*Fig. 5.19:* Response waiting time verse Time of day

### Analysis of RWT distribution for weekdays and weekends

We investigate the distribution of the RWT by plotting the Complementary Cumulative Distribution Function (CCDF) for the waiting time. We start our analysis with IRC chat log. A plot of the response waiting time distribution for all the pairs of people in Figure 5.20 show that it is not a pure power law distribution (simple-scaling), rather it possesses a more complex pattern, which is evident of a multi scaling behaviour. Up until two and half hours on weekdays and one and half hour on weekends, the behaviour exhibits a power-law distribution, then beyond these hours, the behaviour of response waiting times becomes different (the graph suddenly deviates with a sharp curve).

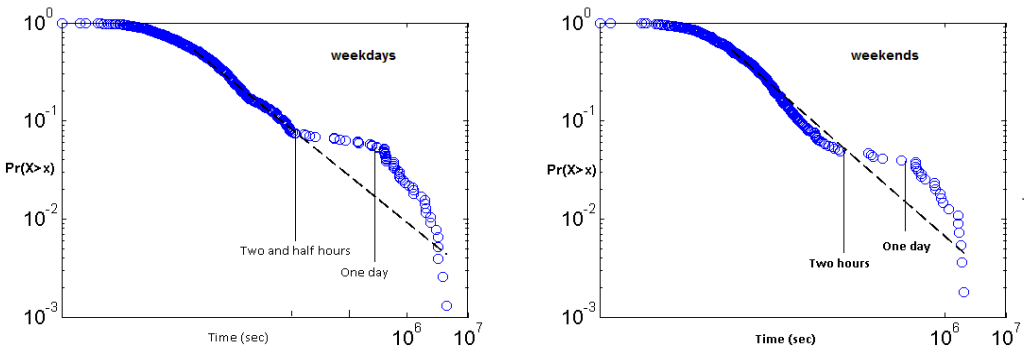



(a) (b)

*Fig. 5.20:* IRC:Degree Distribution of users RWT

Next we consider T-REX dataset. The RWT is displayed in Figure 5.21 and Figure 5 23. The graph clearly suggests that the response waiting time for all the pairs of people exhibits a more complex pattern, just as the response waiting time in Figure 5.20; indicating an existence of multi-scaling.

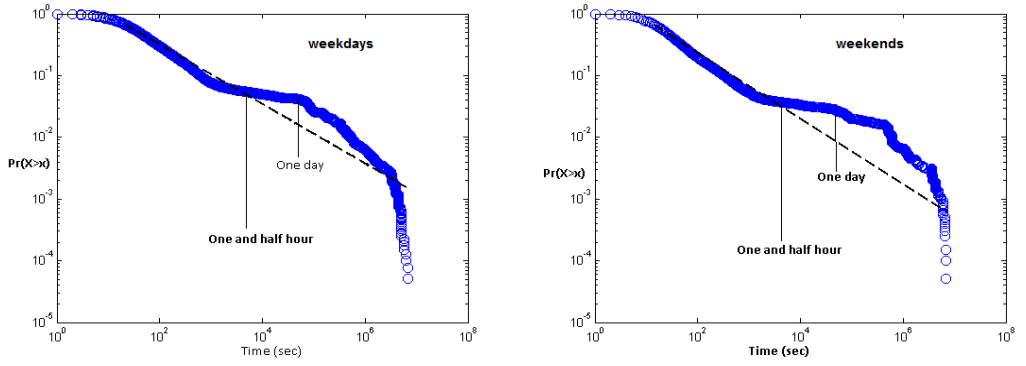
In Figure 5.21(a) and Figure 5.21(b), the behaviour of the response waiting times becomes differ-




(a) (b)

*Fig. 5.21:* T-REX: Degree Distribution of uses RWT

ent after two and half hour on weekdays and two hours on weekends. Up until two and half hour, the distribution is in the form of a power-law with an exponent of 1.48 on weekdays and until two hours, the distribution is in the form of a power-law with an exponent of 1.53 on weekends, then beyond this hours, the graph suddenly deviates and appear flat till one day before dropping off sharply near 107 seconds. The flat shape suggest that if a response is not received after two and half hours on weekdays and two hours on weekends, it is highly likely that the response will still come within a day. This indicates that they are different factors that influence the RWT at different time scales. Lastly, we consider the Walford dataset. The RWT is presented in Figure 5.22(a) andFigure 5.22(a). Also, the graph indicating an existence of multi- scaling and the behaviour of response waiting times becomes different after one and half hour for both weekdays and weekends.




(a) (b)

*Fig. 5.22:* Walford:Degree Distribution of users RWT

Up until one and half hours for both weekdays and weekends, the behaviour exhibits a power-law distribution, then beyond these hours; the behaviour of the response waiting times becomes different and appears flat till one day before dropping off sharply. The flat shape suggest that if a response is

not received after one and half hours on weekdays and weekends, it is highly likely that the response will be received within a day. Table 5.8 compares the RWT on weekdays and weekends across the three chat rooms. For both weekdays and weekends, our model indicates an existence of multi-scaling behaviour in the RWT for pairs of people. Although the exponents vary across the three chat rooms, however, the exponents on weekends tend to be higher than on weekdays. The Time at which the graph deviates from Power Law (TDPL) also varies (one and half to two and half hours). In summary, one behaviour the three chat rooms have in common on weekdays and weekends is that the RWT distributions are multi-scaled, that is they cannot be described using a single power law. Also, the distributions have in common a change of behaviour near the one and a half to two and a half hour response time, and again changes around the one day mark (see Table 5.8). The time at which the graph deviates from Power Law (TDPL)). This suggests that if a message has not been answered in one and a half hour there still a good chance that would be answered within a day.

*Tab. 5.8:* Results compared for RWT.

| Logs | Week | $\alpha$ | x | P | TDPL |
|---|---|---|---|---|---|
| IRC | day | 1.61 | 184 | 0.266 | $2\frac{1}{2}hr$ |
| | end | 1.65 | 30 | 0.16 | $1\frac{1}{2}hr$ |
| T-REX | day | 1.50 | 172 | 0.20 | $2\frac{1}{2}hrs$ |
| | end | 1.54 | 155 | 0.16 | 2hrs |
| Walford | day | 1.48 | 18 | 0.239 | $1\frac{1}{2}hr$ |
| | end | 1.53 | 12 | 0.139 | $1\frac{1}{2}hr$ |

## 5.3 Discussion

As described in section 2, Barabasi claimed that the distribution of the time taken by the user to reply to a received message is best approximated by power-law with an exponent equal to 1". This suggests that an individuals email pattern has a bursty non-Poisson character with an exponent of 1. Our results on the RWT shows multiscaling behaviour and reveals that the value of the exponents has a wide range of variation depending on the user's environment (weekdays or weekends). This is significantly different from Barabasi results and indicates that the distribution of the RWT could be more bursty than Barabasi found.

Further investigation of the behaviour of users RWT reveals that the distribution of the RWT can not be best described using power law as reported in previous literature rather the distribution exhibits multi-scaling behaviour with different exponents for the three chat logs. Even more, considering different context (weekdays and weekends) there is an evidence of multi-scaling behaviour in users RWT for the different time context(weekdays and weekends).

Our results are important because they show that the time context or environment (for example weekdays and weekends) has an impact on the pattern of the users RWT. This could be as a result of the difference in users routine on weekdays and weekends. Secondly, the response waiting time is different on different time scales, which may be because participants generally only spend a maximum of few hours in every chat session before any interruption. In contrast, Barabasi in his RWT analysis did not put into consideration the interfering factors such as environment which may hinder the users from having a smooth chat flow or responding to messages.

So to determine the distribution that best describe the RWT, we will compare two or more distributions for each of the three chat log. Our result show that the distribution of response waiting time is quite closer to Burr than power law such as Pareto. This suggests Burr as the best distribution to describe response waiting time in an on-line chat room.

Again, considering different context of the users' RWT (weekdays and weekends), we compared the network structure of users' RWT on weekdays against users' RWT weekend by examining the degree distribution and cluster coefficients. There is an evidence of multi-scaling behaviour in the degree distributions. The degree distribution reveal a difference in exponents for IRC and T-REX chat rooms. The weekends have a lower exponent than the weekdays. However, the weekend exponent for Walford is very similar to the weekday exponent, perhaps reflecting that Walford chat room is not operated around working topics so is used more for leisure.

To study the cohesion between the users, we evaluate the global clustering coefficient. For both chat rooms, the clustering coefficient for the weekdays and weekends is very similar. For the IRC logs, the clustering coefficient is relative low. In comparison, Walford clustering coefficient is high suggesting that Walford has larger clusters of users chatting together.

We also show that users RWT appear to have different behaviour depending on the time of day. After dividing the RWT into weekdays and weekends, we slice the hours of a day into 8 intervals of 3 hours each and study the changes in chat room communication across the time of day. Our results reveal a shorter waiting time at the very early morning, just after midnight, longer waiting time during work hours of the day, shorter waiting after work hours in the evening and longer waiting time late in the night. This pattern suggests that users also commit their time to other things which may affect their response to messages.

Although, all the users RWT exhibited multi-scaling behaviour, however, there were significant differences between the statistics on weekdays and at weekends, for example, RWT on weekends appears to have higher exponents than weekdays. The different exponents on different time scales suggest that time context or the environment have a significant influence on the users RWT.

Moreover, the distributions have in common a change of behaviour near the one and a half to two and a half hour response time and again changes around the one day mark. This suggests that

if a message has not been answered in one and a half hour there still a good chance that would be answered within a day.

# 6. PREDICTING THE RESPONSE WAITING TIME IN A CHAT ROOM

## 6.1 Introduction

In the previous chapter, we explored the dynamics of Response Waiting Time (RWT) for pairs of people in conversation. In this chapter, we investigate which properties of a chat can be used to predict if a user has a fast response time. In a chat–session, the response waiting time is crucial in the establishment and maintenance of an online–conversation [54]. This online–conversation can be in the academic field where collaborators often meet online to chat for a few minutes [40, 52]. Also, it could occur during on–line education. Educators are looking for a better way of incorporating a web–based chat–room into the academic curriculum [40, 52]. A well–implemented chat–room should make the students feel that they have a direct connection with their instructors as well as their classmates [20].

Furthermore, customer services which is paramount in business growth [41] are using live chats as a means of resolving customers' problems as well as running general customer support services [67]. The chat between a company agent and the customer, which is in text format, has the advantage over traditional phone support as the text logs may be used to identify customers' opinion about products and the overall customer experience.

A major factor that drives customer satisfaction is the time taken for the agent and client to respond to each other during the online chat [65]. The above examples suggest that a short response waiting time is vital to ensure a good rapport between on-line communicators.

The approach we follow to predict if a user is fast responder is to first define what a fast response is (short waiting time), see section 6.2.1. Thereafter we evaluate different properties of the chat. We estimate which is the best statistical model to describe the RWT of a user. The assumption is that the statistical model can be used as a way to discriminate between users response. Also we consider that the "mood" of a chat can be a predictor of a fast response. So we conducted sentiment analysis on the posted messages and categorised the mood of each utterance(post) as either positive, negative or neutral. We also assumed that the different topics under discussion can also influence user's response time, so we carried out topic detection analysis and the top 5 topics were chosen. Lastly we consider the number of messages exchanged between pairs of people in conversation. The assumption here is that the more two people talk to each other the better they understand themselves and the quicker they will respond to each other's messages.

We believe that these quantities influence a user's response time and we used them as the inputs in the NN and SVM to predict which users are fast responders.

## 6.2 Walford chat–room and its properties

We only focused on Walford as it is the largest data set in our possession.

### 6.2.1 Response Waiting Time (RWT)

A typical example of a Walford Direct–chat log is in Figure 6.1, where three pairs of conversation are going on: A→B and B→A, C→D and D→C, H→I and I→H. The second column in the figure is the time when the chat was sent, so from the figure, the response waiting time between A and B is 26 seconds ($40:55 - 40:29$). To characterise the corpus we evaluated the Complementary Cumulative Distribution Function (CCDF) of the RWT, shown in Figure 6.2. The graph suggest that RWT distribution is not a power law rather reveals distinctive patterns which describe users' behaviours. After 15 seconds, the RWT pattern appears to change slightly indicating a change in user's RWT behaviours. Beyond 15 seconds, we notice another sharp change in the RWT pattern graph suggesting another rapid change in user's RWT behaviours.

Given that 15 seconds is a short response time in which an average user would type around 10 english words and we noticed that a third of the response time is less than 15 seconds (Figure 6.3). Based on this observation and for simplicity we partition the behaviour only into two classes: the first 15 seconds were chosen as Short Response Waiting Time Class (SRWTC)) and any RWT duration beyond 15 seconds were classified as Long Response Waiting Time Class (LRWTC).

```
1  40:29 A→(B):grins I think it's the proxy
2                Kevin and Perry that need kicking!
3  40:55 B→(A):what happened last night..the
4                lot of it got or needed a kicking!
5  41:13 C→(D): lsaysl cH kissing bandit...l
6  41:45 H→(I):Kissing bandits are  predators
7                 should not be tolerated
8  41:46 A→(B): it was a Janet router that went,
9                second tie in a week one has died
10 42:08 D→(C):lsaysl cYou're just jealous he
11               took your job
12 42:16 B→(A):grins janet is the of the network
13               the universities and schools are on.
14               router is something that forwards on
15 42:21 I→(H): And I haven't gotten any action since
```

*Fig. 6.1:* Sample chats in Walford.

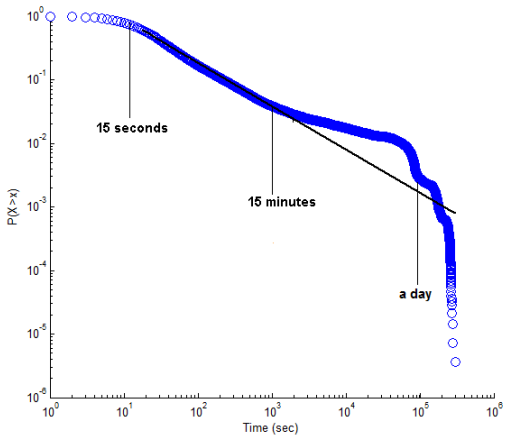



*Fig. 6.2:* Complementary Cumulative Distribution of the RWT

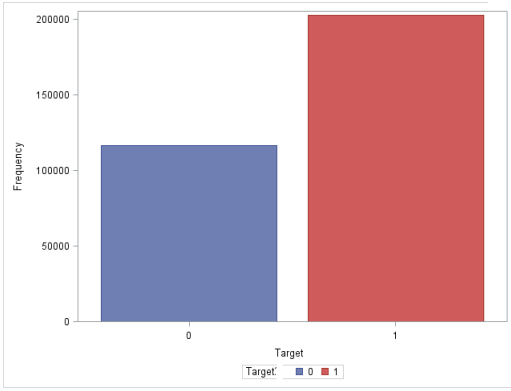



*Fig. 6.3:* Frequency of RWT divided into less than 15 seconds (blue) and larger than 15 seconds (red).

### 6.2.2 Statistical model describing Individual RWT

To characterise the individual differences in the response waiting time for each pair of people in a conversation, we extracted their RWT and, using the allfitdist in Matlab, fitted a statistical model that best describes it. For example, the RWT between user A and user B may be best described by a Pareto distribution while the RWT between user A and user C may be best described with a Weibull distribution. This captures the fact that communication dynamics depends on the pair of individuals rather than being simply about one of the individuals [54] involved in the conversation.

Figure 6.4 shows the number of times that a particular distribution describes the RWT of a conversation. More than 50% of the conversations can be modelled with a generalised extreme value model, the next most predominant model is a generalised Pareto model.

#### Recoding the statistical model of the RWT to a vector

The statistical model of Individual's RWT variable is a categorical variable (also called nominal variable) which contain character values. The values of this variable were recoded with numerical values.

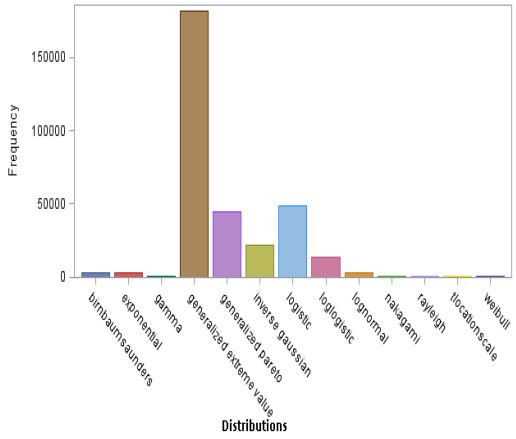



*Fig. 6.4:* Frequency of which statistical model best describes the RWT between a pair of individuals.

The reason is that Neural Network which is one of algorithm we used to develop the model only works with numerical values and not character values. The recoded variable is shown in Table 6.1.

*Tab. 6.1:* Recoding the models with numerical value

| Model | code |
|---|---|
| Burr | 1 |
| Exponential | 2 |
| Birnbaumsaunders | 3 |
| Gamma | 4 |
| Generalized extreme value | 5 |
| Generalized Pareto | 6 |
| Inverse Gaussian | 7 |
| Logistic | 8 |
| Loglogistic | 9 |
| Lognormal | 10 |
| Nakagami | 11 |
| Rayleigh | 12 |
| Tlocationscale | 13 |
| Weibull | 14 |

### 6.2.3 User utterance sentiment

The purpose of using sentiment analysis, is to investigate if the duration of a conversation is related to the "mood" of the utterances. The utterances are categorised into positive, negative or neutral using VADER (Valence Aware Dictionary and sEntiment Reasoner) [54]. VADER is a lexicon and rule-based sentiment tool which incorporates a wide range of human validated sentiment lexicons

such as Linguistic Inquiry and Word Counts (LIWC), Affective Norms for English Words (ANEW) and Generative Lexicon (GL) and performs well even in the case of short utterance.

Two examples of our chat utterances and the sentiment analysis results are shown below. The value in bracket shows the valence (scaled from -1 to 1), a positive score indicates a positive sentiment and a negative score indicates a negative sentiment. Zero scores are considered to be a neutral.

Positive utterance: `that is sooo great!  soooooo what does your ring look like?` (0.784)) and a negative utterance: `she goes crazy.  I hate not working` (-0.7269).

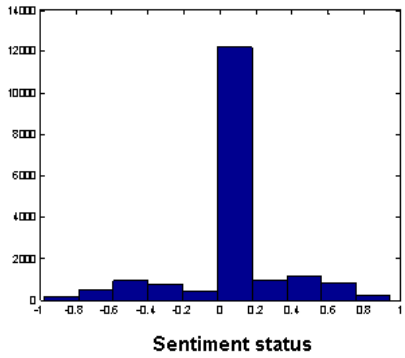



*Fig. 6.5:* Frequency distribution of the utterances sentiment.

The graph in Figure 6.5 shows the distribution of sentiment scores. The algorithms classified the majority of the utterances as neutral, there are also some positive and negative sentiments across the utterances.

### 6.2.4 Number of messages exchanged vs. average RWT

The number of message exchanged (NME) is the total number of messages exchanged by a pair of individuals.

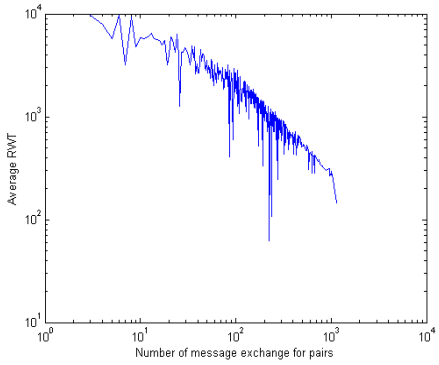



*Fig. 6.6:* Frequency of total number of massages exchange by pair of individuals.

Figure 6.6 shows that there is a power law dependence between the number of messages exchanged between two individuals and their average RWT. This suggests that individuals that exchanged fewer messages take a longer time to respond, while individuals with a high number of message exchange have shorter response waiting time.

### 6.2.5 Topic detection

Another possibility is that the response time is related to the topic discussed in the chat. To do topic analysis, we pre-processed the unstructured data using Statistical Analysis System Text (SAS) mining tool. This tool has three nodes that are relevant in our text analysis: text parsing node, text filtering node and topic node. Starting with text parsing node, we parsed the text to quantify the words or terms that are present in the conversations. Next, the text filtering node was used to eliminate the unwanted terms so that we only focus on the most important and relevant information. One of the benefits of filtering is that it reduces the list of parsed terms. To determine the number of relevant topics, the prepared dataset was passed through text topic node [37].

## 6.3 Model used for prediction

Our main aim is to predict if a user is going to be a fast responder and we are going to do the prediction using a Neural network and a Support Vector Machine. First,we will train the Neural Network and Support Vector Machine with some of the data set, then predict the rest of the data and finally validate the prediction

The input parameters are; the statistical model of individual's RWT for the past 6 months, the number of message exchanged, the sentiment status and the different topics of discussions. The target variable used is a binary number where 0 represent short response waiting time and 1 stands for long response waiting time.

Our data sets consisted of over 35,000 thousand chat records which we then partitioned into two parts called train/test split approach. One part to be trained, the other for fine tuning and evaluating how well the model has been trained. The train/test split approach is ideal when we have a large enough data set as in our case. Another common method of splitting the dataset is called cross validation. It is very similar to train/test split, but involves creating more subsets of the data set. It involves splitting the dataset into k subsets, and training is done on k-1 of these subsets. The last subset is held for testing and this is repeated for each of the subset. Cross validation approach is often used when there is no sufficient data to create a sizeable training set and a validation set that represent the predictive population as well but in our case we had a sizeable data set to use the train/test split method.

### 6.3.1 Experiment

In our experiment we considered two cases and in both cases we trained with Neural Network (NN) and Support Vector Machine (SVM).

- Case 1: All the topic parameters were considered as one of the inputs. The total number of predictors were 8
- Case 2: The topic parameters were not considered as one of the inputs. The total number of predictors were 3

**Case 1**

In this case, we had 8 inputs: the number of message exchanged, the sentiment status, the best fit model which describes the the individual's RWT for the past 6 Months and 5 topics of discussion. We trained the data with two Machine learning algorithms a Neural Network with Topics (NNT) and a Support Vector Machine with Topics (SVMT). The two algorithms, which have been described in chapter 3, will enable us to compare performance.

**SVMT Training**: The reason we are considering SVM is scalability. SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space. The SVMs are trained with our dataset which consists of 35,000 data points: 60% for training and 40% for testing. Each point is located in the n-dimensional space, with each dimension corresponding to a feature of the data point. We used a training set of 11862 data points with 8 features and tried different kernel functions such as linear, sigmoid, radial basis and polynomial. The sigmoid kernel gave the best output.

**SVMT Testing**: We apply SVMs model to the test dataset to predict if a user is going to be a fast responder. The testing set, consisting of 40% data points with 8 features which resulted in AUC of 0.915.

**NNT Training** Multi-layer networks are used. The scaled conjugate gradient descent algorithm is used for training. Our data consists of the same set of 35,000 data points: 60% for training and 40% for testing. The metrics we used for model evaluation are Area Under Curve, recall, precision and F-score as defined in section 3.1. In the study we use five different Neural Networks with the following architectures:

**NNT Testing**: We apply NNs model to the test dataset to predict if a user is going to be a fast responder. The testing set, consisting of 40% data points with 8 features, show Area Under Curve (AUC): 0.930 for Network 1, 0.921 for Network 2, 0.923 for Network 3, 0.933 for Network 4, 0.919 for Network 5.

*Tab. 6.2:* Parameters of Neural Network models

| Model | Neuron | Learning rate | Momentum | Acc | AUC | Recall | Precision | F-score |
|---|---|---|---|---|---|---|---|---|
| Network 1 | 3 | 0.1 | 0 | 0.925 | 0.930 | 0.673 | 0.930 | 0.781 |
| Network 2 | 3 | 0.2 | 0.1 | 0.924 | 0.921 | 0.660 | 0.933 | 0.774 |
| Network 3 | 3 | 0.63 | 0.73 | 0.928 | 0.923 | 0.677 | 0.928 | 0.783 |
| Network 4 | 7 | 0.57 | 0.76 | 0.927 | 0.933 | 0.712 | 0.880 | 0.787 |
| Network 5 | 16 | 0.61 | 0.77 | 0.926 | 0.919 | 0.673 | 0.933 | 0.782 |

**Case 2**

In this case, we used only three inputs: the number of message exchanged, the sentiment status, the best fit model which describes the the individual's RWT for the past 6 Months. Also we trained the data with two Machine learning algorithms a Neural Network (NN) and a Support Vector Machine (SVM).

**SVM Training**: Similarly, the SVMs are trained with our dataset which consists of 35,000 data points: 60% for training and 40% for testing.

**SVM Testing**: The SVMs model was tested using the test dataset. The model was used to predict if a user is going to be a fast responder. The testing set, consisting of 40% data points with 8 features which resulted in AUC of 0.892

**NN Training**: Multi-layer, feed-forward networks are used. The scaled conjugate gradient descent algorithm is used for training. Our data consists of the same set of 35,000 data points: 60% for training and 40% for testing. In the study we also used five different Neural Networks with the following architectures:

*Tab. 6.3:* Parameters of Neural Network models

| Model | Neuron | Learning rate | Momentum | Acc | AUC | Recall | Precision | Fscore |
|---|---|---|---|---|---|---|---|---|
| Network 1 | 3 | 0.1 | 0 | 0.925 | 0.924 | 0.661 | 0.946 | 0.778 |
| Network 2 | 3 | 0.2 | 0.1 | 0.926 | 0.925 | 0.663 | 0.946 | 0.780 |
| Network 3 | 3 | 0.63 | 0.73 | 0.926 | 0.894 | 0.662 | 0.946 | 0.779 |
| Network 4 | 7 | 0.57 | 0.76 | 0.923 | 0.926 | 0.705 | 0.884 | 0.784 |
| Network 5 | 16 | 0.61 | 0.77 | 0.923 | 0.926 | 0.705 B | 0.884 | 0.784 |

**NN Testing**: In this stage we apply NNs model to the test dataset to predict if a user is going to be a fast responder. The testing set, consisting of 40% data points with 3 features, show an area under curve(AUC) as follows: 0.924 for Network 1, 0.925 for Network 2, 0.894 for Network 3, 0.926 for Network 4, 0.926 for Network 5.

### 6.3.2 Results compared

In the case 1 where all the topics parameters were considered as one of the input, NNT(AUC of 0.933) outperformed SVMT (AUC of 0.905) while in Case 2 where the topic parameters were not considered as one of the inputs, NN(AUC of 0.926) performed better than SVM(AUC of 0.892). However, the best classifier is the Neural Network with topics (NNT) with an Area Under the Curve(AUC) of 0.933. The model's performances improved by incorporating the topics. This suggests that the subjects under discussion during online chat may be a driving factor in responding to a post message.

### 6.3.3 How robust is the best model?

Having determined that the champion model is NNT, we carried out sensitivity analysis on the NNT. To ensure consistency with the accuracy, different models of different partitions were trained. These models are presented in table 7.1. The standard deviation for Recall, Precision, F-Score and Accuracy are 0.01315, 0.01135, 0.00878 and 0.00273 respectively which reflects the consistency of model performance.

*Tab. 6.4:* Parameters of Neural Network models

| Partition | Recall | Precision | F-Score | Accuracy |
|---|---|---|---|---|
| 60:40 | 0.663 | 0.948 | 0.78 | 0.9253 |
| 70:30 | 0.667 | 0.945 | 0.782 | 0.9254 |
| 75:25 | 0.659 | 0.942 | 0.778 | 0.9235 |
| 50:50 | 0.663 | 0.947 | 0.779 | 0.9249 |
| 65:35 | 0.687 | 0.948 | 0.7966 | 0.9297 |
| 55:45 | 0.701 | 0.948 | 0.806 | 0.9322 |
| 85:15 | 0.672 | 0.9487 | 0.7867 | 0.9269 |
| 80:20 | 0.675 | 0.9503 | 0.7893 | 0.9279 |
| 90:10 | 0.686 | 0.9117 | 0.7827 | 0.9236 |
| Mean | 0.6747 | 0.9431 | 0.7867 | 0.9266 |
| **S Dev** | **0.01315** | **0.01135** | **0.00878** | **0.00273** |

To decide which of the different features/parameters are more relevant for the prediction of the RWT we used a decision tree algorithm. The algorithm is used in machine learning and it is provided by the SAS mining tool.

The algorithm uses a series of simple rules to build a decision tree from training data. Splitting rules are applied one after another based on the attribute which most effectively distinguishes between the available classes, resulting in a hierarchy of branches within branches that produces the characteristic inverted decision tree form.

The most relevant feature is the number of message exchange during chat session (Message

Exchange Number). This is quite evident in Figure 6.6, an increase in the number of message exchange between pairs of people may rapidly lead to a shorter response waiting time.

Second in importance is the values for the best fit model which describes the individual's RWT for the past 6 Months. This parameter captures the dynamic nature of RWT which occurs during chat sessions for pairs of people. Thereafter, four of the topics are relevant and sentiment analysis appears next to last. The low importance of the utterance sentiments may be attributed to the difficulties when trying to identify sentiment status of an utterance. Only about 1.5% of the utterances were classified as a positive or negative sentiment; the remaining utterances were classified as neutral sentiment. The possible reasons may be that utterances are unstructured and sentiments have the tendency of changing over time.

## 6.4 Conclusion

We used a Neural Network and a Support Vector Machine to predict if a user is going to be a fast responder. The parameters used for the prediction were the total number of messages exchanged between two people, Statistical model describing Individual RWT, the sentiment of the utterances exchanged and the topics of conversation. The most relevant parameter for a fast response is the total number of messages exchanged between individuals. This is not surprising as more messages exchanged requires faster responses if the chat occurs in a finite time, say in an afternoon. Interestingly, individuals that exchanged very few messages also tend to take a longer time to respond.

# 7. USER'S BEHAVIOUR DYNAMICS FOR GROUP CONVERSATION: THREAD DETECTION

In chapter 6, we characterized the dynamics of RWT by considering pairs of people in conversation. However, this chapter studies the dynamics of a group of people in conversation. We focus on the disentanglement of a chat room network. Disentanglement is a task that extracts the different interposed utterances in a chat log and separates them into distinct conversations. One of the challenges in chat disentanglement is the temporal variation and dynamics of the network system. Hence, we first explore the temporal changes that occur as the network evolves.

## 7.1 Temporal Behaviour

Network systems are dynamic; links exist for only short periods, disappear and reappear again [12]. The arrow of time-ordering is vital to understand the variations and fluctuations in the linking patterns of a social network [35]. Researchers often build graphs from the time-varying system by simply aggregating all the interactions as if they occur simultaneously in time. As a result, the links of these graphs do not change over time [60].

Often, the solution has been to slice the dataset into windows and then, investigate how the network structure evolves over time. However, this method does not capture all of the characteristics of the temporal structure of the interaction patterns [12]. The concept and algorithm that has been proposed in network theory are specifically built for the statics graph [35], i.e. a graph whose edges remain constant over time.

In static networks, if X is connected to Y directly, and Y is connected to Z directly, then, X is connected to C directly via a path over Y. However, in regard to temporal networks, if at a point in time X is connected to Y directly later than when Y is connected to Z directly, then X and Z will not be connected, hence no information can flow from X through Y to Z. This happens in both direct and undirected networks [35].

In another example, suppose in the morning, Q and R had a conversation, then later in the afternoon R and C had a conversation; the message may be propagated from Q to C but not from C to Q. This important feature is lost if we simply aggregate all the daily network contacts in our network analysis. Building group conversation graphs from a time-varying system by simply slicing and aggregating all of the interactions as if they are occurring simultaneously in time does not reveal

these time varying networks behaviours [51]. So, we need a better way of capturing these interesting real-world network behaviours in a time-varying graph.

### 7.1.1 Time-respecting paths

In a static network, the temporal dimensions of human activities are not considered. Rather, it assumes that the nodes and edges do not change over time and the distributions of human activities are random in time, hence can be estimated by Poisson processes. In contrast to static assumptions, social interactions are dynamic in nature; while some ties are decaying others are forming, while some participants are joining the conversation others are leaving the network. Since interactions often occur over a period of time, turning time on and off in a social network shows that human activities and patterns of individual communication change continuously.

So, a path in a static graph, can be defined as a series of edges that links a chain of nodes in such a way that the ending of one edge at a node is the beginning of another edge of the same path. However, in the case of temporal graphs, the definition of a path is often with non-decreasing time that connects sets of nodes [39]. The restriction of following time-ordered sequences of contacts differentiates temporal paths and paths in static networks. Likewise, the difference between static directed networks and temporal networks is that the paths of temporal networks are not transitive:

> "The presence of time-respecting paths from $i$ to $j$ and $j$ to $k$ does not mean that there is a path from $i$ to $k$. Just like the basic property of time-respecting paths (beginning and ending at certain points in time), the presence of a time-respecting path that begins at $i$ at time $t_0$ and leads to $j$ does not assure that such a path between $i$ and $j$ exists for $t > t_0$; moreover, in future temporal path connecting $i$ and $j$ might take a different route [39]."

### 7.1.2 Example of temporal variation

In Fig 7.1.2, we show the major fluctuation and variation that is going on as the network evolves. This is an example of a real-world time-varying network from our chat room communication. The network nodes are represented by their ID, starting from participant A and ending with participant W. The time of interaction is denoted by $t_1$ to $t_{51}$. The conversation was started by node D, who is the current speaker at $t_1$ and the audience includes A B C E F J H I. At $t_2$, participant A assumes the next current speaker because A was one of the audiences of D. It is very important to note that only participant D received the message sent by the current speaker A at $t_2$, while the rest were blocked from viewing the messages. At $t_3$ and $t_4$, D as the current speaker still spoke to A, B, C, E, F, J, H, I. But between $t_5$ and $t_{10}$ participant A, B, C, E, F, J, H, I disappeared and it looks like a

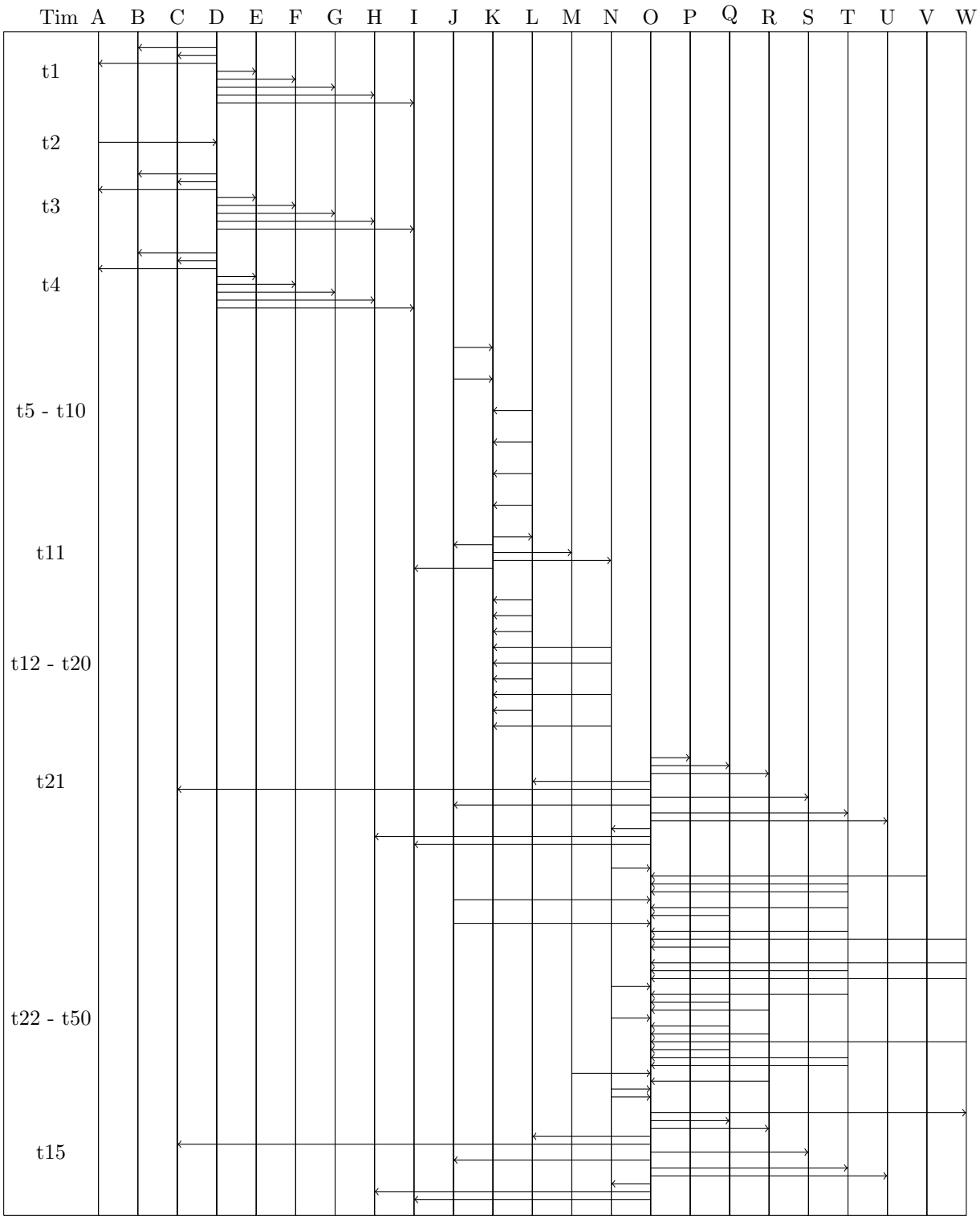



*Fig. 7.1:* Temporal variation in Time-varying graph

new conversation emerged. Here participant K seems to be an audience of two speakers J and L. J and L do not see each others messages.

At $t_{11}$, participant K assumes the current speaker role with I, J, L, M, and N as the audience. Also, notice that participant I reappeared as well. Between $t_{12}$ and $t_{20}$, participant K again became an audience of many speakers who do not see themselves. It is probable that these participants were responding to Ks messages, which were spoken at $t_{11}$. Notice that, at $t_{21}$, participants C and H reappear. The current speaker at this time is O with C, H, I, J, L, N, P, Q, R, S, T, and U as the audience. Also, between $t_{22}$ and $t_{50}$, participant O was an audience of many speakers who blocked themselves from seeing each others messages. One important thing is that participant O can sometimes be an audience of two or more speakers simultaneously. At $t_{51}$ where participant O is the speaker, the audience includes C, H, I, J, L, N, W, Q, R, S, T and U.

Another interesting feature in this interaction is that some participants responded after receiving two or more messages from the sender, thereby increasing the RWT while others reduced the waiting time by responding immediately. Another behaviour we observed as the network evolved was that some people joined the group conversation at a different time and often appeared in the conversation path.

We have shown that relationships among the participants in the time-varying network is dynamic and fluctuates over time. Hence, models and algorithms that aim to describe or extract information from dynamic networks must respect the time dependency of the links [51]. Automatic separation of interposed sequence of utterances into distinct conversation can be considered as a precondition for having effective high-level dialogue analysis. In this chapter, we pose and answer the following question:

**How can we disentangle chat using a computationally less intensive method?** Most disentanglement models involve highly computationally intensive methods such as clustering techniques, fuzzy algorithm, etc. These methods often led to deterioration in the results accuracy.

**How can we dynamically model real social network?** Social interactions are dynamic, experience time decay and form in social networks; also, nodes enter and exit via social networks. Interactions between pairs of people or groups are bursty as a result of long dormant periods separated by strong bursts of activity.

## 7.2 Methodology

The proposed approach uses a simple and effective method for chat disentanglement. The algorithm involves a two-pass process. In the first pass, the algorithm divides the text stream, after the

messages were sorted based on time of posting, into a coherent short segment of conversation using the waiting time (time gap) and turn-taking allocation rule. In the second pass, the algorithm recovers a complete distinct conversation thread from the short segments of conversations by looking at the participant-based features and the content similarity features.

### 7.2.1 Detecting a Coherent Short Segment (CSS)

Given a text stream in which the messages are ordered based on the time of posting, the basic idea of a CSS algorithm is as follows: Let $P_1$ and $P_2$ be participants in the text stream. Take the first utterance from the text stream say $U_1$ made by $P_1$ as a segment. The next utterance immediately after $U_1$, say $U_2$, could come from the same speaker (in this case $P_1$ ) or a different participant, say $P_2$ .

If $U_2$ comes from the same speaker, we will only work out the likelihood of the utterance belonging to the same segment as $U_1$. On the other hand, if $U_2$ comes from another participant, say $P_2$, it will involve two steps: first, check if $P_2$ was an audience of the immediate past speaker (in this case $P_1$). Secondly, work out the likelihood of the utterance (in this case $U_2$ ) belonging to the same segment as $U_1$. However, if it is unlikely, a new segment of conversation will start. For more clarity on the algorithm see Figure 7.2. To work out the likelihood we use the following approach:

- Extract the distribution of the RWT in the current short segment.
- Then from that distribution work out the likelihood of the current RWT (which you have just observed) belonging to the extracted distribution.
- If it is very likely, then we classify the current utterance to the current short segment; if it is unlikely, a new segment of conversation will start.

### 7.2.2 Reconstructing multi-party conversation

This second stage involves building a coherent complete conversation from the short segment. Here we developed two algorithms: first model and second model. The first model is based on content and participant features while the second model is based on content feature and participant adjustment features.

#### Content features (CF)

Word recurrence is an important feature for segmentation or coherence. For example, the number of words shared between segments X and Y suggest that the two utterances may belong to the same conversation. The content-based features involve comparing the amount of word similarity between segments of conversations and the approach is as follows:

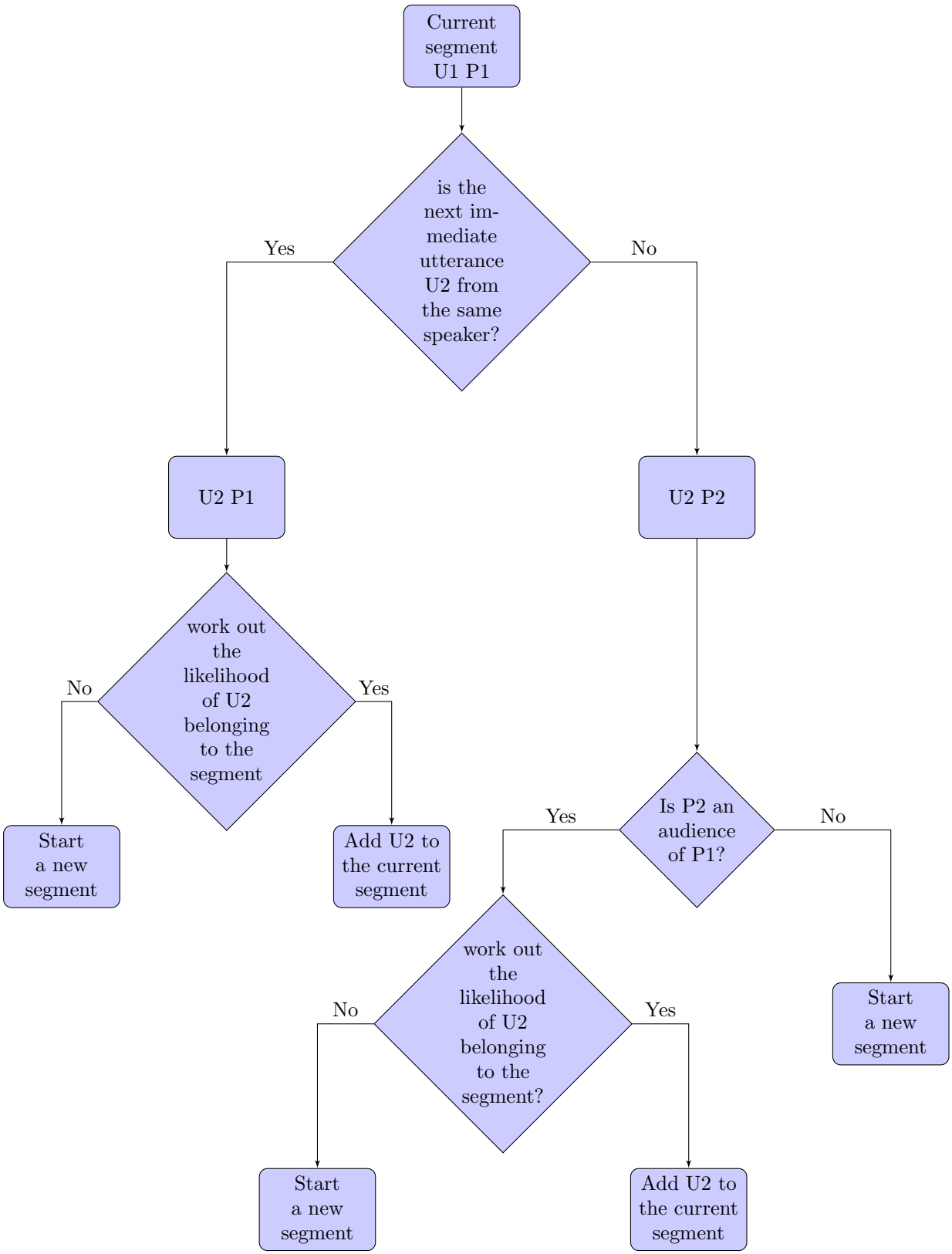



*Fig. 7.2:* Flow chart for Coherent Short Segment.

First, take the first segment and use it to form a thread, T. Next, for each of the remaining segments, M, we compute its similarity with the nearest neighbour segment in the existing thread

T using cosine measure. If the cosine measure is greater than a threshold and there is a likelihood of the current RWT (time difference between the average time of the nearest neighbour segment in the existing thread T and the average time of the current segment) belonging to the current thread (T) RWT distribution, then add M to T. This pass is efficient as it considers each segment once.

**Participant features (PF)**

Similarly, take the first segment and use it to form a thread, T. Next, for each of the remaining segments, M, we compute its similarity with the nearest neighbour segment in the existing thread T using the following principles:

- A pair of segments X and Y may be closely connected in the discourse and are likely to be directly related if those participating in segment X are the same people participating in segment Y, and there is a likelihood of the current RWT belonging to the current Thread (T) response waiting time distribution.
- A pair of segment X and Y may be widely separated in the discourse and are unlikely to be directly related if those participating in segment X are totally different from those people participating in segment Y.

*Tab. 7.1:* Process of selecting the best Parameters for our models

| Cosine measure | Max reduction of users | % of Words similarities | Precision | Recall | F-Score |
|---|---|---|---|---|---|
| 0.1 | 1 | 2 | 0.67742 | 0.65625 | 0.66667 |
| 0.2 | 3 | 10 | 0.50667 | 0.63333 | 0.56296 |
| 0.3 | 2 | 5 | 0.8248 | 0.9135 | 0.8667 |
| 0.4 | 5 | 20 | 0.49351 | 0.51351 | 0.50331 |
| 0.5 | 4 | 50 | 0.32692 | 0.22973 | 0.26984 |

Before choosing the threshold for different parameters, we conducted a test with range parameter settings as shown in Table 7.1 and we found that setting the cosine measure value to 0.3 along with other parameters produced the best result. Visualising the relationship of cosine measure values and their performance in terms precision, recall and F-Score is presented in Figure 7.3 and it is clearly shows that cosine measure value 0.3 is the best.

**Participant Features Adjustment(PFA)**

As mentioned in section 6.1.1, in traditional networks models the assumption is that edges and nodes are alive forever within a network; however, in a real-world system participants continue to leave while others continue to join the network [42]. This decreases or increases, in time, with the total number of nodes, therefore, continuously altering the properties of network structure [24]

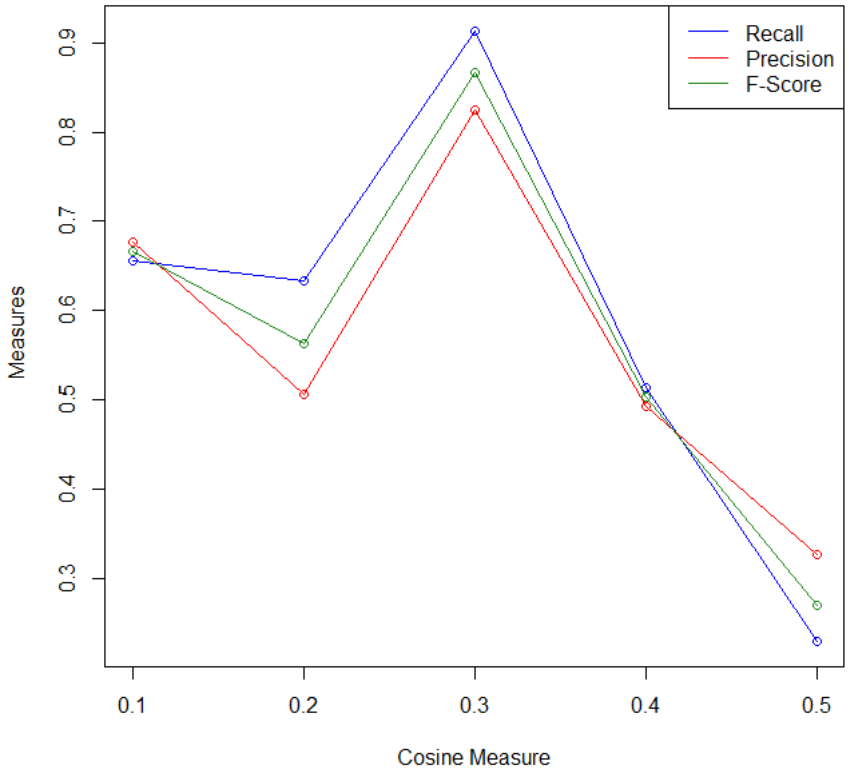



*Fig. 7.3:* Test for the best cosine measure value

In a group conversation, sometimes participants can be part of a conversation for a short period, disappear and reappear again. In order to capture the fluctuation and variations that occur in the number of participants as the conversation grows over time, there is a need to adjust for the temporal properties of human interaction. We studied a random sample of our dataset and observed the following principles:

Suppose X is the first current segment that is to be added to the thread and Y is the nearest neighbour segment in the existing thread T. Then in a situation where either of the following occurs:

1. The users who are involved in segment X (for example users A, B, C and D) are the same users participating in segment Y with a maximum addition of 2 new users in Y (for example users increase to A, B, C, D, E and F in Y)

   OR

2. The users who are involved in segment X (for example users A, B, C and D) are the same users participating in segment Y with a maximum reduction of 2 users in Y (for example, users reduced to only A and B in Y)

Segments X and Y tend to be related if:

- There is a likelihood of the current RWT belonging to the current Thread (T) RWT distribution.
- The relative percentage of the words that are similar between the two segments is greater than 5%

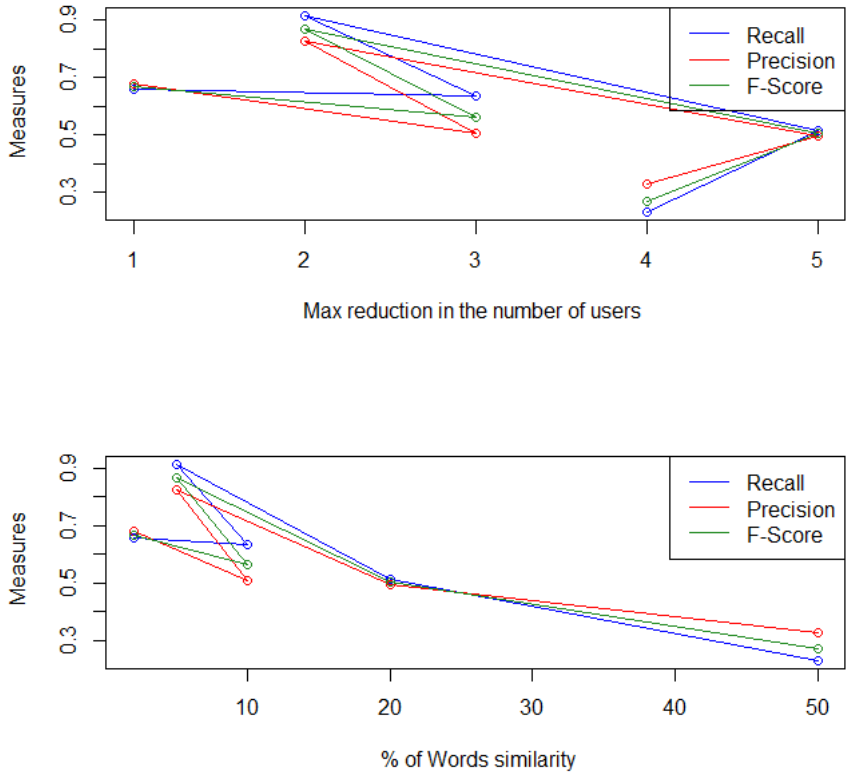



*Fig. 7.4:* Test for the best % of word similarity

Similarly Table 7.1 shows that a maximum addition or reduction of 2 new users produced the best result along with 5% as the relative percent for the number of similar words between two utterances. This is why 2 for the maximum addition or reduction of new users and 5% for the number of similar words between two utterances were selected. Also visualising the relationship between the maximum addition or reduction of new users, percentage of word similarities and their performance in terms precision, recall and F-Score is presented in Figure 7.4. Our annotation system was developed using SAS Code Node of SAS Enterprise Miner 12.1. software.

## 7.3 Experiments

We have described the three passes, which were based on RWT, content and participants features. In this section, we empirically evaluate the performance of this algorithm using IRC and Walford

chat logs. We will introduce our evaluation metrics and display the results from our experiment. We compared the results with the baseline algorithms and between the two chat rooms.

### 7.3.1 Evaluation methods

The accuracy of a classifier can be visualised in a matrix form called a confusion matrix. It demonstrates the association between the actual outcomes and the predicted classes [73]. This measures the effectiveness of the classification model by computing the number of correct and incorrect classifications for each possible value of the targeted variable [50]. Table 7.2 shows the confusion matrix used to calculate the performance of the classifier.

- True Negative (TN) a negative class data point was identified as negative.
- False Negative (FN) a positive class data point was identified as negative;
- False Positive (FP) a negative class data point was identified as positive;
- True Positive (TP) a positive class data point was identified as positive;

*Tab. 7.2:* Confusion matrix

| | | Predicted class | |
|---|---|---|---|
| | | Yes | No |
| Actual class | Yes | True Positive | False Negative |
| | No | False Positive | True Negative |

We can deduce from the above confusion table various matrices for evaluating classification performance. The matrices are Precision (P), Recall (R) and F-score (F). In Precision we calculate the fraction of objects that are really relevant in the result set while Recall is concerned with the fraction of all the relevant objects in the collection that are in the result set [50]. Metrics can be computed as follows:

$$\text{Precision}(P) = \frac{TP}{TP + FP}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN}$$

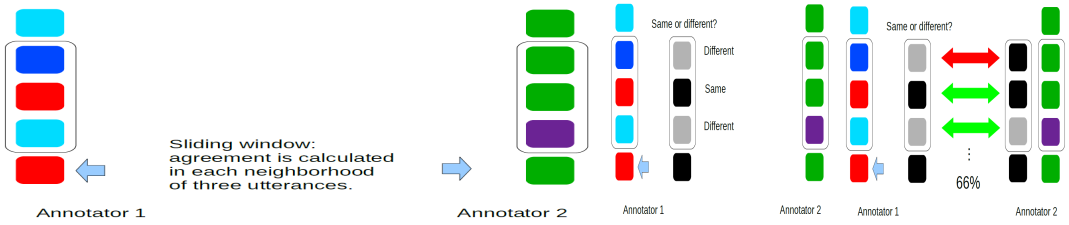
$$\text{F-score (F)} = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

### Metric: One-to-Ome, Many-to-One and Local Agreement

In addition, we explored Elsner and Charniaks entropy evaluation metric tools [50] to calculate the accuracy, precision, recall, F-score, many-to-one (m-1) and the local error (loc-N). One-to-one overlap (1-1-g) is computed by a greedy algorithm, while one-to-one overlap (1-1-o) is computed optimally (with the Hungarian algorithm). This process is displayed in Figure 7.5-7.7


(a) (b) (c)

*Fig. 7.5:* One-to-One Metric




(a) (b) (c)

*Fig. 7.6:* Local Agreement Metric

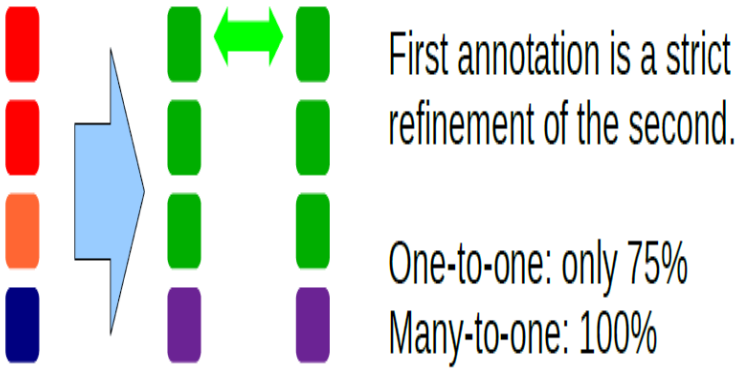



*Fig. 7.7:* Many-to-One Metric

### 7.3.2 Experiment 1: Human annotation of Walford chat logs

For the human annotation, four PhD researchers in the field of semantic and pragmatic modelling volunteered to annotate the test section of the corpus manually. The datasets were 500-line transcript (500 utterances; 1:45hr) and in total, we had four annotated test datasets of the same transcript (one from each volunteer), which serves as a validation corpus for our system. With the help of these PhD students who are working on conversation analysis, aesthetics and interaction, we adapted Elsners coding annotation scheme [50]:

- Mark each utterance as part of a single conversation.
- Distinct conversations that are not related in any way.
- Create as many or as few conversations as they need to describe the data.
- A conversation can be between any number of people,
- It should be clear that the comments inside a conversation fit together.
- If a schism occurs, we have two options:
  1. if it seems short, they may view it as a mere digression and label it as part of the parent conversation.
  2. If it seems to deserve a place of its own, they will have to separate it from the parent.
- Participants between the current and next conversation can either be the same people or increase or reduce by a maximum of half. This gives room for people who are leaving or joining the on-going conversation.
- If two utterances occur at the same time between two different pairs of people, we will use the content-based approach to determine which conversation the utterances belong to.

Examining the inter-annotator agreement is a crucial factor in knowing the classification performance of a model [50]. The inter-annotator agreement value often becomes an upper bound on what we can expect from an automatic classification models performance [2]. Each annotated dataset were analysed and we applied a pairs wise comparison to choose the best with the highest score. A summary of the inter-annotator agreement results for the Walford chat logs is displayed in Table 7.3.

Table 7.3 shows that the average number of active conversations at a time are 1.18 and the average inter-annotator agreement for local and 1-to-1 metrics are high, which is an indication that the annotators agree more. The high value of the local metric indicates good local correlations, which implies that, in the three-sentence window ahead of each sentence, the annotators are generally in

*Tab. 7.3:* Inter-annotator agreement

| Parameters | Mean | min | max |
|---|---|---|---|
| Avg.Length | 16.41 | 13.70 | 20.20 |
| Avg.Density | 1.18 | 1.05 | 1.28 |
| Avg.Entropy | 2.52 | 2.20 | 2.70 |
| 1-to-1 | 80.98 | 75.63 | 84.50 |
| loc3 | 92.19 | 88.75 | 96.53 |
| M-to-1 | 93.20 | 89.10 | 97.53 |

agreement. Local metric has a mean overlap of 92.19%, minimum of 88.75% and a maximum of 96.53%, while 1-to-1 metric has a mean overlap of 80.98%, minimum of 75.63% and a maximum of 84.50%.

*Tab. 7.4:* Classification performance

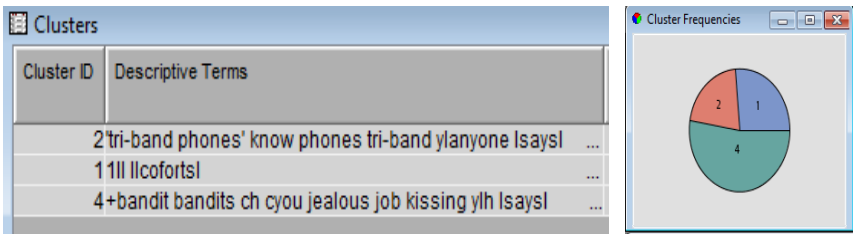
| Parameters | Avg.Value |
|---|---|
| Accuracy | 0.88202 |
| Precision | 0.8922 |
| Recall | 0.9652 |
| F-score | 0.9273 |

The entropy, which indicates where the conversation utterances belong to, ranges from 2.20 to 2.70. This is a small variation. However, the quantity increases as the number of conversations grows [26]. Many-to-one accuracy measures how much the annotators agree on the general structure [26]. Besides the above techniques, we apply another simple method to compare the agreement between different annotated corpuses. The method is unsupervised one-to-one accuracy. The approach involves clustering each distinct group conversation extracted from the corpus and applies one-to-one accuracy to measure the global similarity between annotations. The classification performance is also shown in Figure 7.4.

To compute one-to-one accuracy, we compared the cluster results from the annotations and counted the total number of overlaps, then presented the percentage of overlaps found.

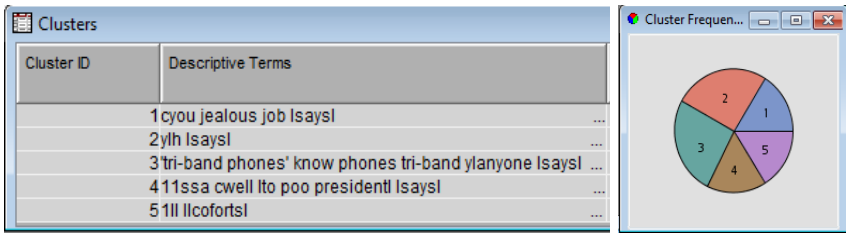
The sample results of our evaluation process are shown below. Fig 7.8 and 7.9 display a sample of the cluster results from the two annotators. A close look at the two clusters shows some degree of overlap; for instance, clusters 1, 2 and 4 in Fig 7.8 and clusters 5, 2 and 1 in Fig 7.8 appear to be similar, respectively. This shows 60% agreement between the two annotations, making it a better and encouraging result. The metrics results are displayed below.

In Table 7.5, on average the annotators have a high degree of similarity with each other, which is encouraging. With respect to our model, the result shows a big range between the maximum and




(a) (b)

*Fig. 7.8:* First human annotation




(a) (b)

*Fig. 7.9:* Second human annotation

minimum scores; however, it is interesting to know that our model average score is very close to the human annotation average score, which is encouraging. We believe that the reason for this high level of inter-annotators agreement can be traced to how the Walford chat room operates.

In Walford, the participants can construct a friend list. Walford has a tool that permits users to send a direct message to all the members in their friends list who are online at the same time[38]. So the ability to reach everyone in your friends list simultaneously helps Walford users to engage in a kind of group chat.

Annotators also observed that when users with many friend lists send a message to everyone in the group, the participants who are online receive it at the same time. At least, no less than three users reply to the message, thereby reducing the length of RWT. As a result, we can find many continuous blocks of conversation segment in the logs, which improve our thread detection approach. The same methods will be employed to compare the automatic results from our system with the manually annotated corpus. Please note that the second dataset (Elsners IRC chat logs) used for evaluation has been manually annotated by the provider. The author provided the raw dataset as well as the annotated one. Unlike Walford, we used the annotated IRC chat logs to evaluate our system.

### 7.3.3 Experiment 2: Walford chat logs

We tested our system using the same 500-line transcript (500 utterances; 1:45hr) test corpus from Walford chat logs. The result of the annotation yields 55 conversations with an average length of

*Tab. 7.5:* Metric values between proposed annotations and human annotations

| Parameters | Annotators. | Model |
|---|---|---|
| Max one-to-one. | 75.10 | 65.30 |
| Min one-to-one. | 36.60 | 34.40 |
| Mean one-to-one. | 56.20 | 49.10 |

10.20. Firstly, we investigate some of the features of multi-party conversations. Our examination focuses on the pattern of turn-taking with group conversation and the behaviours of RWT as the number of participants and the communication count increases.

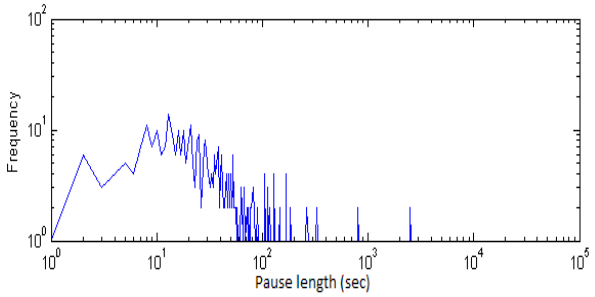
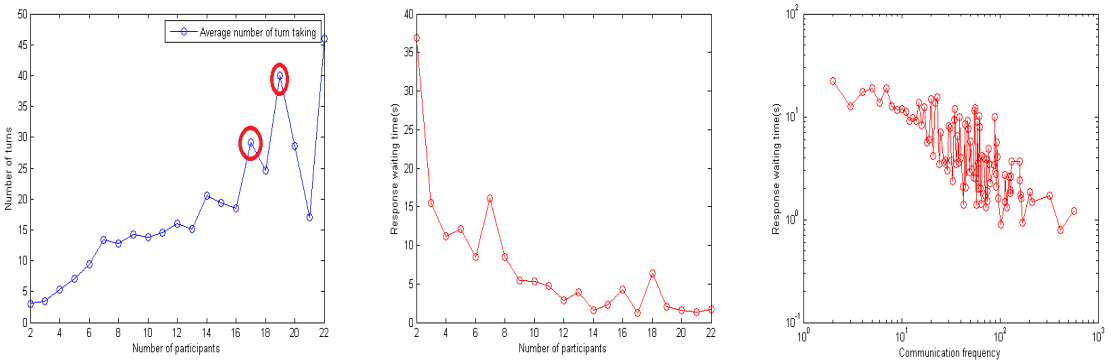



*Fig. 7.10:* Distribution of pause length between utterances in the same conversation

Figure 7.10 shows the distribution of the pauses between utterances in the same conversation. The highest point of the curve occurs at 1 to 3 seconds and there are few shorter pauses, less than seconds. This shows that, in a given group conversation, users like to receive each others responses before they will speak again. Also, the curve reveals a heavy-tailed to the right.

Figure 7.11(a) shows the average amount of turn-taking. It appears that the number of turns in each group conversation grows as the number of participants increase. This is expected in a growing community. Moreover, the points circled with red in Figure 7.11(a) were investigated further. We found that none of the users within this group had a friend list of less than 7. This means that when a user in this group conversation sends a message, no less than 7 of other users receive it at the same time. As such, one or more users may reply to the message, thereby reducing the length of RWT.

Figure 7.11(b) suggests that the average RWT depends on the number of participants in a group conversation. Groups with less participants have a longer waiting time while groups with a higher number of participants have a short RWT. This reflects the pauses in turn-taking behaviour among participants; in a given group conversation, users would rather wait to receive each others response before they will speak again. In Figure 7.11(c), we plotted the number of communication counts in each group conversation and the graph shows that the RWT decreases with an increasing number of communication counts in a group conversation. This suggests that the amount of turn-taking increases with a reduction in RWT as the number of participants and communication count grows in




(a) Number of turn-taking (b) Number of Participant (c) No. of communication

*Fig. 7.11:* Some of the feature of Walford chat room

a multi-party conversation. This kind of behaviour is expected in a group conversation and highlights the level of accuracy in our disentanglement model.

### Baselines

As a standard of comparison for our system, we provide results for several baselines trivial systems which any useful annotation should out-perform.

- All different: Each utterance is a separate conversation.
- All same: The whole transcript is a single conversation

For each particular metric, we calculate the best baseline

*Tab. 7.6:* Classification performance

| Parameters | First model | Second model |
|---|---|---|
| Precision | 0.6947 | 0.8248 |
| Recall | 0.4035 | 0.9135 |
| F-score | 0.5105 | 0.8667 |

To evaluate how well our system performed we used Elsner and Charniaks entropy evaluation metric tools. As a standard of comparison for our system, we provide results for baselines trivial systems which any useful annotation should outperform. We considered the two approaches in calculating the baselines and for each particular metric, we calculate the best baseline

- All different: Each utterance is a separate conversation.
- All same: The whole transcript is a single conversation

*Tab. 7.7:* Experimental results

| Annotators | Mean | min | max |
|---|---|---|---|
| 1-to-1 | 80.98 | 75.63 | 84.50 |
| loc3 | 92.19 | 88.75 | 96.53 |
| Our Model | | | |
| 1-to-1 | 76.10 | 72.52 | 82.60 |
| loc3 | 85.60 | 79.20 | 90.12 |
| Elsner Model | | | |
| 1-to-1 | 40.62 | 33.63 | 51.12 |
| loc3 | 72.75 | 70.47 | 75.16 |
| Baseline | | | |
| 1-to-1 | 47.13 | 29.07 | 57.78 |
| loc3 | 70.40 | 66.12 | 75.69 |

A summary of our model results for the Walford chat logs is displayed in Tables 7.6 and 7.7. In Table 7.6, we can see that the F-score of our model is 51% for the first algorithm and 87% for the second.

**Comparing the F-value for the two algorithm performance**

Table 7.6 shows that the second algorithm with extra tuning of the participants parameters (based on CF, PF and PFA) achieved obvious improvement over the first algorithm, which was based on CF and PF only. The second algorithm increases the performance relatively by 36% in terms of F-value compared with the first algorithm. This observation validates the effect of introducing the temporal fluctuation and variation in the number of participants over time. The reason for the improvement is explained as follows: when taking temporal fluctuation into consideration, the algorithm was able to capture a situation where one or two participants left or joined the group conversation, which resulted in decreasing or increasing of the number of participants respectively at a particular time. These kinds of features may not be captured by standard graph metrics, which are mainly suitable for static graphs.

Having shown that a model with extra participant tuning performed better due to the dynamic nature of a text stream, our analysis will focus on this model. In Table 7.7 we show other metrics of the second model. The local metric has a mean overlap of 85.60%, a minimum of 79.20% and a maximum of 90.12% while the 1-to-1 metric has a mean overlap of 76.10%, a minimum of 72.52% and a maximum of 82.60%. On average the annotators have a high degree of agreement with each other.

With respect to our model, it is interesting to know that our models average score is closer to

the human annotation average score than the baseline score. As explained earlier, in Walford, the participants have the ability to reach everyone in their friend list simultaneously and engage in a group chat.

### 7.3.4 Experiment 3: IRC chat logs

In this second experiment, we tested our system using the IRC chat dataset published online by Elsner [26]. This dataset is divided into three parts: development, test and pilot with 359, 706 and 800 utterances, respectively. Each part contains the raw dataset as well as the human annotated dataset of the same part. However, we tested our model using the raw test dataset and since the manually annotated of the same test dataset was also provided by the author, it serves as a validation corpus for our system. Figure 7.12 shows the number of conversations and the corresponding number of posts participated in per speaker, suggesting that those who post more are more likely to be involved in multiple conversations [2].

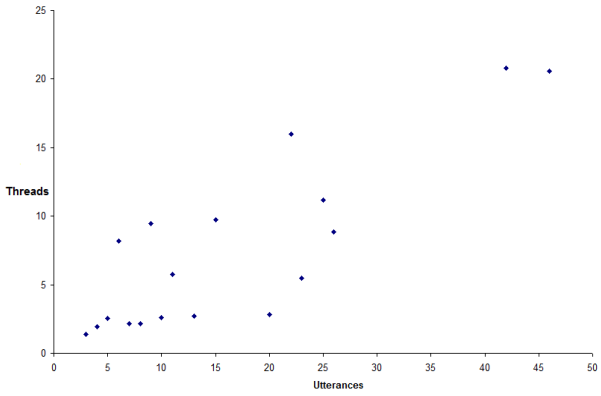



*Fig. 7.12:* Thread and Uttrance

A summary of our model results for IRC chat logs is displayed in Tables 7.9 and 7.8. Classification performance of our system for IRC chat logs is shown in Table 7.9. We can see that the F-score is 39.05% for the first algorithm (which is based on CF and PF) and 51.42% for the second algorithm (which is based on CF, PF and PFA). Similarly, the F-scores result is relatively lower in the first model compared to the second model. Again, we see the effect of adjusting for the temporal fluctuation of participants in multi-participant chat analysis. PFA in the second algorithm increased the performance relatively by 12.36% in terms of F-value compared with the first algorithm.

Table 7.8 shows that the local metric has a mean overlap of 75.23%, a minimum of 71.67% and a maximum of 78.77% while the 1-to-1 metric has a mean overlap of 53.60%, a minimum of 44.09% and a maximum of 68.90%.

The reason we have low performance in IRC chat logs can be traced to two points: primarily,

*Tab. 7.8:* Experimental results

| Parameters | Mean | min | max |
|---|---|---|---|
| Avg.Length | 13.41 | 3.0 | 17.0 |
| Avg.Density | 4.18 | 3.4 | 4.80 |
| Avg.Entropy | 3.22 | 2.80 | 3.90 |
| Annotators | | | |
| 1-to-1 | 52.98 | 35.63 | 63.50 |
| loc3 | 81.09 | 74.75 | 86.53 |
| Our Model | | | |
| 1-to-1 | 35.10 | 34.30 | 50.12 |
| loc3 | 69.60 | 65.40 | 73.13 |
| Elsner Model | | | |
| 1-to-1 | 40.62 | 33.63 | 51.12 |
| loc3 | 72.75 | 70.47 | 75.16 |
| Baseline | | | |
| 1-to-1 | 27.13 | 24.07 | 48.78 |
| loc3 | 53.40 | 49.12 | 60.69 |

*Tab. 7.9:* Classification Performance

| Parameters | First model | Second model |
|---|---|---|
| Precision | 0.8055 | 0.7206 |
| Recall | 0.2577 | 0.3971 |
| F-score | 0.3905 | 0.5142 |

at the model development stage, we only used the Walford dataset to develop our model. The data structure of Walford chat logs is slightly different from the data structure of IRC chat logs.

Secondly, the presences of schisms in the chat logs pose interesting and challenging problems. As a result, the overall classification performance is affected.

### 7.3.5 IRC and Walford results compared

In comparing the results from our model for Walford and IRC chat logs, it is clear that our model performed much better with the former than the latter. This is because, unlike IRC, Walford has a tool that permits users to send a direct message to all of the members in their friend list who are online at the same time. So the ability to reach everyone in your friend list simultaneously helps Walford users to engage in group chat. However, the occurrence of schisms in both Walford and IRC chat logs has an effect on the performance of our model. According to [71], Schism occurs when a conversation splits into two conversations; the new conversation is formed due to certain

participants branching off from a specific message and refocusing their attention upon each other. Schisms impose serious difficulty in identifying conversation threads.

Lastly, our annotation system was developed using SAS Code Node of SAS Enterprise Miner 12.1. software. The flow diagram for recovering multi-party conversation is displayed in the appendix.

### 7.3.6 Developing the second model using IRC Dataset Experiment 4: IRC chat logs

As we mentioned earlier, the data structure of Walford chat logs is slightly different from the data structure of IRC chat logs The model we developed using Walford logs had a low performance in IRC chat logs, which indicates that the model may be data-dependent. At the model development stage, we only used the IRC dataset to re-develop the second model then test it with both IRC and Walford logs. The second model is the model with an extra tuning of the participants parameters ( Model that is based on CF, PF and PFA)

The model was tested with both walford and IRC logs. The output of the model result is shown in Table 5. Testing this new model shows that the results from IRC indicate an obvious improvement and is better than the results from Walford logs. This is an evidence that the models are data dependant. Compare to the previous mode on IRC logs, this new model increases the performance relatively by 36% in terms of F-value. This observation further validates that the model is data dependant.

*Tab. 7.10:* Classification Performance

| Parameters | Walford | IRC |
|---|---|---|
| Precision | 0.8055 | 0.7206 |
| Recall | 0.2577 | 0.3971 |
| F-score | 0.3905 | 0.5142 |

### 7.3.7 Time Complexity of Algorithms

Time complexity of an algorithm signifies the total time required by the program to run to completion. The time complexity of algorithms is most commonly expressed using the big O notation. This removes all constant factors so that the running time can be estimated in relation to N, as N approaches infinity. Time complexity can be considered to be Constant, Linear, Logarithmic and Quadratic.

- Constant Time: a constant-time method is order 1: O(1) An algorithm is said to run in constant time if it requires the same amount of time regardless of the input size.
- Linear Time: a linear-time method is order N: O(N. An algorithm is said to run in linear time if its time execution is directly proportional to the input size, i.e. time grows linearly as input

size increases.

- Logarithmic Time(O(log n)): An algorithm is said to run in logarithmic time if its time execution is proportional to the logarithm of the input size.
- Quadratic Time: a quadratic-time method is order N squared: O(N2). An algorithm is said to run in quadratic time if its time execution is proportional to the square of the input size.

After carrying out a time complex analysis of all the codes in our algorithm, the result shows a time complex of O(logn) which is good. A sample of how it was done is show below.

```
Here is a sample from our algorithm

 %Do  i = 1 %to n

      %do j = 1 %to   (n/i)

         data Abc.datamer2;
         merge Abc.datamer2 Abc.hev&i;
         RUN;

      %end
 %end


Time Complexity Analysis:

For i = 1, the inner loop is executed n times.
For i = 2, the inner loop is executed approximately n/2 times.
For i = 3, the inner loop is executed approximately n/3 times.
For i = 4, the inner loop is executed approximately n/4 times.
.
For i = n, the inner loop is executed approximately n/n times.



So the total time complexity of the above algorithm is (n + n/2 + n/3 +  + n/n), Which become
```

The important thing about series (1/1 + 1/2 + 1/3 +  + 1/n) is equal to O(Logn).

So the time complexity of the above code is O(nLogn).

## 8. EVALUATION AND DISCUSSION

In chapter 3, we talked about the basics of social networks and defined some network terms. Chapter 4 showed that our chat room represents human interactions. We time-sliced the network and studied its evolution across the years. We found that most of them have power-law distributions with varying exponents. This is important because it shows that temporal distribution of human communication is generally very bursty. Furthermore, we explored the temporal difference that occurs in users' behaviour as the network evolves. In temporal difference, we study the communication behaviour patterns of weekday and weekend chat room users in terms of information flow, degree distribution and the clustering coefficient distribution. Both weekday and weekend users' behaviours exhibit power-law distribution, however, weekday participants have higher number of triangles (3-cycle) than weekend participants, who have a higher betweenness value, i.e. participants who are exercising influence over the interaction of others.

Evaluating the similarity between the content of weekday and weekend chats, we apply cosine similarity measure. The result shows that the cosine measure is 0.00058, indicating a wide range of dissimilarity between the content in weekday and weekend chat. We tested the power law hypothesis through a bootstrapping procedure. The p-value is the goodness-of-fit and ratio test metric are 0.72 and 0.74 for Walford; 0.64 and 0.90 for IRC; 0.62 and 0.38 for T-REX respectively. This suggest that an existence of Power-law.

Chapter 5 investigated the dynamics of pairs of people in conversation with respect to their response waiting time (RWT). We considered the Response Waiting Time in a chat room communication in regard to the time difference between successive messages sent between two people. Examining the distribution of the RWT reveals a graph with several distinct regions and this is a significant shift from the current views on the nature of RWT as a simple power-law distribution to a more complex pattern.

To evaluate our result we use the bootstrap procedure which shows a goodness-of-fit of 0.110 and ration test of 0.018 for Walford, goodness-of-fit of 0.110 and ratio test of 0.018 for IRC and goodness-of-fit of 0.102 and ratio test of 0.006 for T-REX. This suggest that the model does not provide a plausible fit to the data and another distribution may be more appropriate. Further test shows that the distribution of response waiting time is quite closer to burr than pareto (power law) which suggest that Burr as the best distribution to describe response waiting time in an on-line chat

room.

For simplicity we divided the distribution into two major regions. Region one consisted of users with waiting time less or equal to one hour and can be described using power-law. In region two, the waiting time was between one hour and a month. We compared the two regions based on their word content and the way the users interacted with each other to exchange information. We started by extracting the word content of the two regions and then, used cosine measure to compare the word similarity between them. The result shows that the word content in the two regions was less similar with a cosine measure of 0.130. From this result we infer that the type of words people use when chatting appears to influence the RWT. Next, we compared the network structure in the two regions. A network is made up of users (nodes) and the links between them. The way that these links (edges) are organised has a big effect on who gets what information. In relation to the RWT we investigated the network properties in the two regions. The first network property to be examined is the degree distribution; the degree distribution of a node or user k is the number of edges that have k as a vertex. From analysis of the Cumulative Distribution graph, we find that the degree distribution is higher in region one with shorter waiting time than in region two with a longer waiting time. This indicates that the degree distribution of a network influences the RWT. The second network property we examined is the cluster coefficient. This measures the degree to which friends of friends are also befriending each other. The clustering coefficient of regions one and two are 0.54 and 0.30, respectively. This indicates that the participants in region two who have longer waiting time are less connected than participants in region one who have shorter waiting time.

In another strand of analysis we conducted on the RWT involved investigating the waiting time in relation to communication count. For each pair, we counted the number of times they sent messages between each other and the average of their RWTs. A plot of the averaged RWT and the communication count for pairs of people revealed that the RWT decreased as the communication count increased. Thus, we can infer that the communication frequency influences the RWT. Finally, this section ended with an investigation of the behaviour of the RWT by considering one user with other participants. Our result shows that an individual can have several waiting times depending on the interference factors, which may be fatigue, lack of interest, lack of communication, dullness, lack of interest or concentration etc. This suggests that communication dynamics depends on the group or pairs rather than being simply about the individual.

However, it is important to note that the different technologies such as e-mail, Twitter, Walford, IRC and T-REX chat logs appear to have wild variation in their results, indicating that the results may be technology-dependent. E-mail shows simple scaling behaviour with an alpha value of 1 while in Twitter, Walford, IRC and T-REX chat logs revealed multi-scaling behaviour. With twitter, a user's waiting time behaviour changed after one day while in Walford, IRC, and T-REX chat logs a

user's waiting time behaviour changed after one or a few hours. It suggests that technology has some influence on the user's RWT. This shows that there are different factors that influence the RWT at different time scales.

In chapter 6, we used chat room characteristics to predict response waiting time. Utilising over 35,000 records of online chat sessions, we trained Neural Network(NN) and Support Virtual Machine (SVM) with a 60% train, 20% validation and 20% test split. Our best classifier is Neural network(NNT) which achieved 97.17% accuracy, however, Support Virtual Machine(SVMT) which attained 96.68% is few percent behind. The result shows that the number of message exchange and previous RWT distributions for pairs plays a significant role in reducing RWT during chat sessions. Suggesting that increasing the number of message exchange between pairs minimise the RWT and may lead to more friendly connection. Knowing in advance the response rate of a chat user will assist in making appropriate decision on which medium of communication will be convenient for the pairs of people and hence, reducing unfriendly connections.

To evaluate the model performance, we use the concepts of Receiver Operating Characteristic (ROC) to analyse the accuracy of the models and provide diagnostic on the best model. Nine data partition were done and in each partition we recorded the Recall, Precision, F-Score and Accuracy. The standard deviation for Recall, Precision, F-Score and Accuracy are 0.01315, 0.01135, 0.00878 and 0.00273 respectively which reflects the consistence of model performance

Chapter 7 focused on the dynamics of group conversation and temporal behaviours. We presented the challenges as well as the gap in analysing time-varying graph with an algorithms for static networks. We noted that relationship among the participants in a time-varying network is dynamic and fluctuates over time. Therefore, models and algorithms that aim to describe or extract information from dynamic networks must respect the time dependency of the links.

In this chapter, we developed an algorithm for chat thread detection that is time dependant. We proposed a simple and effective technique that utilised simple statistics information, such as utterance similarities, RWT, turn-taking and the participant-based feature for thread detection in chat logs. This supplemented the traditional qualitative method, which has been proven to be difficult due predict to the contextual nature of meaning. Unlike most existing thread detection models, our approach is less computational intensive.

To verify their applicability, the proposed algorithm was applied to two different real-world data sets (Walford and IRC chat logs) and the obtained results were evaluated using the following matrices: Precision (P), Recall (R) and F-score (F). we also explored Elsner and Charniaks entropy evaluation metric tools [49] such as many-to-one (m-1) and the local error (loc-N). One-to-one overlap (1-1-g) is computed by a greedy algorithm, while one-to-one overlap (1-1-o) is computed optimally (with the Hungarian algorithm. Our proposed novel algorithms outperformed the basic algorithm.

## 8.1 Conclusion

We investigated users' behaviours in chat rooms with respect to RWT. This study focused only on pairs of people chatting. Interestingly our result clearly shows an existence of multi-scaling behaviour (more than one behaviour in a user's pattern) in the RWT for pairs of people in a chat network. Unlike most empirical data in nature, and in contradiction to previous reports, the RWT distribution is not a pure power-law; rather it is a graph with several distinct regions. We also show that the pairs of people involved in conversation A B may not necessarily have the same RWT distribution or behave alike. An individual can have several waiting times. This suggests that communication dynamics depends on the groups or pairs rather than being simply about the individual.

Lastly, we presented two different approaches of cluster algorithm to detect a thread in a dynamic text message stream. Our novel approach depends on content features, participants' features and the amount of participant adjustment and it takes temporal information such as fluctuations or variations in the number of participants into consideration. As a result the performance improved by at most 34.6% in terms of F-values when compared with the basic algorithm. While the basic algorithm is based on content and participant features, the novel algorithm captured the temporal fluctuation in the numbers of participants during multi-participant conversation. Our proposed novel algorithm outperformed the basic algorithm and achieved results that are nearer human performance on Walford's annotated corpus.

### Our contribution to knowledge

- We demonstrated the real distribution of the RWT during on-line chat, which significantly affects the current views on the nature of RWT. This is a shift from simple power-law distribution to a more complex pattern.
- We also show that an individual can have several waiting times which suggests that communication dynamics depends on the groups or pairs rather than being simply about the individual.
- We proposed a simple and less computational intensive approach for thread detection in chat logs.

### Future work

There are two ways in which new conversations can start: one is through a schism and the other is through a conversation initiating statement. Disentangling chat logs in the absent of a schism may improve model performance and yield better results. So, given the structure of Walford chat logs - the participants can construct a friend list and send direct messages to those on it who are online at the same time while also being able to do a group chat - it may be possible to detect a schism

and remove them before disentangling the remain Walford chat logs. Since a schism occurs when a conversation splits into two, this implies that the users who are involved in a schism were once an audience of the current speaker in the main conversation before the schism occurred and, secondly, the two conversations seem to occur at the same time. With these features we can detect when and where schisms occur and remove them in the chat logs. It is important to note that this will work for Walford chat logs because of their structure but not for IRC chat logs.

# APPENDIX

# A. CODES FOR CHAPTER 5

Code for slicing the dataset by week of day

```
1
2
3  options mprint symbolgen mlogic;
4
5  data dataset2;
6  set dataset;
7  CST = catx('=',Dsender,DTarget);
8  CTS = catx('=',DTarget,Dsender);
9  CST2 = CATS("'",CST,"'");
10 run;
11
12 proc sql noprint;
13 select distinct CST2 into: GIB separated by ' '
14 from dataset2;
15 quit;
16
17 data merged;
18 run;
19
20 %let mylist = &GIB;
21 %macro Track
22 (
23 inputds =
24 , variabl =
25 )
26 ;
27
28 data rev;
29 F = put(scan(&variabl,1,'='),6.);
30 S = put(scan(&variabl,2,'='),6.);
31 C = catx('=',S,F);
32 run;
33
34 data rev;
35 set rev;
36 call symput('F',F);
37 call symput('S',S);
38 call symput('C',C);
39 run;
40
41
42 data data3;
43 set &inputds;
```

```
44  if CST = "&C" then
45     do;
46        if find(CST,"&S") > 0 | find(CTS,"&S") > 0 then  output;
47     end;
48  else
49     do;
50       if find(CST,"&S") > 0 | find(CTS,"&S") > 0 then  output;
51    end;
52  run;
53
54  data merged;
55  set merged data3;
56  run;
57
58  proc sort data = merged;
59  by date;
60  run;
61
62  %mend Track;
63
64  %iterlist( list = &mylist.,
65  code = %nrstr(
66  %Track(inputds = dataset2, variabl = ?);
67  )
68  );
69
70  data merged;
71  set merged;
72  where date > .z;
73  run;
74  proc sort data = merged;
75  by date;
76  run;
77
78
79  %macro cal_sec;
80  *Perfoming calculation for sec;
81
82  proc sql;
83  create table ComDate as
84  select Date as Date2
85  from merged;
86  quit;
87
88  data comdate2;
89  set ComDate(firstobs = 2);
90  data MergDC;
91  merge merged comdate2;
92  run;
93
94  proc sort data = MergDC;
95  BY Date;
```

```
96
97  data MergDC;
98  set MergDC;
99  hr=intck('hour',Date,Date2);
100 min=intck('minute',Date,Date2);
101 sec=intck('second',Date,Date2);
102 run;
103 %mend cal_sec;
104
105 %cal_sec
106
107
108
109 %Macro InsertValue;
110 proc sort data = Mergdc;
111 by descending Date;
112 proc sql;
113 create table dsec as
114 select sec
115 from Mergdc;
116 proc sql;
117    insert into dsec
118     set sec= 0;
119 quit;
120
121 data MergDC2;
122 merge Mergdc dsec;
123 run;
124
125 proc sort data = Mergdc2;
126 by Date;
127 run;
128
129 proc sql;
130 create table ddsec as
131 select sec
132 from Mergdc2;
133 data Mergdc3(drop = sec);
134 set  Mergdc2;
135 where date > .z;
136 run;
137
138 data MergDC4;
139 merge Mergdc3 ddsec;
140 run;
141
142 proc sql;
143 create table datasec as
144 select Dsender, DTarget, sec
145 from MergDC4;
146 quit;
147
```

```
148 data Datasec2(drop = Dsender DTarget sec);
149 set Datasec;
150 Sender = input(Dsender,BEST12.);
151 Target = input(DTarget,BEST12.);
152 Sec2 = sec;
153 Seq = _n_;
154 run;
155
156 %Mend InsertValue;
157 %InsertValue
158
159 data secnozero;
160 set ddsec;
161 where sec > 1;
162 run;
163
164 proc univariate data = secnozero;
165 var sec;
166 histogram sec;
167 output out=modesec2 mode = mode;
168
169 run;
170
```

Code for slicing the dataset by week of day

```
1
2  data datasetw2;
3  set Datasetw;
4  datepart = datepart(Date);
5  wk1 = weekday(datepart);
6  where sec > 0;
7  run;
8
9  proc sql;
10 create table temp1day as
11 select wk1,Sec
12 from datasetw2;
13 quit;
14
15 proc sort data= temp1day;
16  by wk1;
17 run;
18
19 proc means data=temp1day mean noprint;
20  by wk1;
21   output out = Avg;
22 run;
23
24 data walfavgsec(drop = _TYPE_ _STAT_ _FREQ_);;
25 set Avg;
26 if _STAT_ = 'MEAN';
27 RUN;
28
```

Code for slicing the dataset by time of day

```
data datasetsec2;
set Datasetw2;
datepart = datepart(Date);
wk1 = weekday(datepart);
hhr = hour(date);
where sec > 0;
run;

Data datam;
run;

%macro wksec;
%DO k = 1 %TO 7;

data walfdatasec1(rename = (sec = sec&k));
set datasetsec2;
where wk1 = &k;
run;

Data walfdatasec2;
set walfdatasec1;
if hhr = 0  or hhr le 3 then perd = 1;
else if hhr = 3  or hhr le 6 then perd = 2;
else if hhr = 6  or hhr le 9 then perd = 3;
else if hhr = 9  or hhr le 12 then perd = 4;
else if hhr = 12  or hhr le 15 then perd = 5;
else if hhr = 15  or hhr le 18 then perd = 6;
else if hhr = 18  or hhr le 21 then perd = 7;
else perd = 8;
run;

proc sql;
create table temp1day as
select perd,Sec&k
from walfdatasec2
where sec&k > 0;

quit;

proc sort data= temp1day;
 by perd;
run;


 proc means data=temp1day mean noprint;
 by perd;
  output out = Avg;
run;

```

```
data data&k(drop = _TYPE_ _STAT_ _FREQ_);
set Avg;
if _STAT_ = 'MEAN';
RUN;

Data datam;
merge datam data&k;
run;
%end;
%mend wksec;
%wksec
```

Code for RWT,time of day and week of day interaction

```
data datasetR;
set Datasetw2;
datepart = datepart(Date);
wk1 = weekday(datepart);
hhr = hour(date);
where sec > 0;
run;

Data datam;
run;


proc sql noprint;
select distinct CST2 into: GIB separated by ' '
from datasetR;
quit;

data merged;
run;

%let mylist = &GIB;
%macro Track

(
inputds =
, variabl =
)
;

data walfdatasec;
set &inputds;
where CST = &variabl;
RUN;


%DO k = 1 %TO 7;

data walfdatasec1;
set walfdatasec;
```

```
40  where wk1 = &k;
41  run;
42
43  Data walfdatasec2;
44  set walfdatasec1;
45  if hhr = 0  or hhr le 3 then perd = 1;
46  else if hhr = 3  or hhr le 6 then perd = 2;
47  else if hhr = 6  or hhr le 9 then perd = 3;
48  else if hhr = 9  or hhr le 12 then perd = 4;
49  else if hhr = 12  or hhr le 15 then perd = 5;
50  else if hhr = 15  or hhr le 18 then perd = 6;
51  else if hhr = 18  or hhr le 21 then perd = 7;
52  else perd = 8;
53  run;
54
55  proc sql;
56  create table temp1day as
57  select perd,Sec
58  from walfdatasec2
59  where sec> 0;
60
61  quit;
62
63  proc sort data= temp1day;
64   by perd;
65  run;
66
67
68   proc means data=temp1day mean noprint;
69   by perd;
70    output out = Avg;
71  run;
72
73  data data(drop = _TYPE_ _STAT_ _FREQ_);
74  set Avg;
75  weeks = &k;
76  if _STAT_ = 'MEAN';
77  RUN;
78
79  Data datam;
80  set datam data;
81  run;
82  %end;
83  data datam;
84  set datam;
85  ID = &variabl;
86  run;
87  %mend Track;
88
89  %iterlist( list = &mylist.,
90  code = %nrstr(
91  %Track(inputds = datasetR, variabl = ?);
```

```
92   )
93   );
94
```

## B. CODES FOR CHAPTER 6

Code 1

```
1
2  data mergedk(rename = (Spk = Sender Recipient = Target Text = Message));
3  set &EM_IMPORT_DATA(obs = 1000);
4  datek = input(strip(date1),datetime34.);
5  format datek datetime25.;
6  run;
7  proc sql;
8  create table merged as
9  select Sender, Target, Message, datek as Date
10 from mergedk;
11 quit;
12
13 %macro cal_sec;
14 *Perfoming calculation for sec;
15
16 proc sql;
17 create table ComDate as
18 select Date as Date2
19 from merged;
20 quit;
21
22 data comdate2;
23 set ComDate(firstobs = 2);
24 data MergDC;
25 merge merged comdate2;
26 run;
27
28 data MergDC;
29 set MergDC;
30 hr=intck('hour',Date,Date2);
31 min=intck('minute',Date,Date2);
32 sec=intck('second',Date,Date2);
33 Hrd = Hour(Date);
34 Mind = Minute(Date);
35 Secd = Second(Date);
36 SHMS = catx('=',sender,Hrd,Mind,Secd);
37 THMS = catx('=',Target,Hrd,Mind,Secd);
38 HMSS = catx('=',Hrd,Mind,Secd);
39 HMST = catx('=',Hrd,Mind,Secd);
40 run;
41 %mend cal_sec;
42 %cal_sec
43
```

```
%Macro InsertValue;
proc sql;
create table datasec as
select Sender, Target as Reciever,sec, SHMS, THMS, HMSS, HMST, Message, Date
from MergDC;
quit;

data Datasec;
set Datasec;
Seq = _n_;
run;

data Datasec2(drop = Seq Sec);
set Datasec;
Sec2 = put(Sec,8.);
Seq2 = put(Seq,8.);
run;

proc sql;
create table Abc.Datasec as
select Seq2 as ID, Sender, Reciever, Message,Sec2, SHMS, THMS, HMSS, HMST, Date
from Datasec2;
quit;

data Abc.Datasec(rename = (Sender = Senders Reciever = Receivers));;
set Abc.Datasec;
run;

proc sql;
create table Abc.showData as
select  Sender, Reciever, Message, SHMS, THMS, HMSS, HMST, Date
from Datasec2;
quit;

proc print data = Abc.Datasec(obs = 20);
run;
%Mend InsertValue;
%InsertValue
```

Code 2

```

data Abc.Data1;
set Abc.datasec;
run;

proc sql noprint;
select count(*)
into :OBSCOUNT
from Abc.Data1;
quit;
```

```
11 data Abc.lastob2;
12 Slice_point1 = &OBSCOUNT;
13 RUN;
14
15 %Macro procIML1;
16
17  %DO i = 1 %TO &OBSCOUNT;
18
19      proc iml;
20      use Abc.Data1;
21         read all varSHMS THMS HMSS HMST ID Senders Receivers Sec2 INTO ST;
22         n=NROW(ST);
23         m = &i;
24
25
26         F1 = ST[m,1];
27                 T1 = ST[m,2];
28         k = m+1;
29            if k > n then k = n;
30             F444 = ST[k,2];
31         F4 = ST[k,1];
32                 T4 = ST[k,2];
33             F8 = ST[m,5];
34             Fg = ST[k,8];
35         TR = num(Fg);
36             call symput('F8',F8);
37             %let Y = T;
38             %let U = &i;
39
40
41             F11 = scan(F1,1,'=');
42                 T11 = scan(T1,1,'=');
43             F44 = scan(F4,1,'=');
44                 T44 = scan(T4,1,'=');
45             if F1 ^= F4 then
46            do;
47
48
49                 F&U = j(1,1,0);
50                                        FT&U =  &f8;
51                                  store FT&U;
52                           create Abc.hev&U varFT&U;
53                           append;
54                           show contents;
55                                       close ratio&U;
56                   end;
57
58
59
60 /*
61                   find all where(SHMS=F1) into p;
62
```

```
63                      h1 = max(p);
64                  h2 = min(p);
65                  h3 = ST[h1:h2,2];
66
67                      h4 = scan(h3,1,'=');
68
69                      F5 = scan(F4,1,'=');
70
71                      G =  find(h4,F5);
72                      GG = ANY(G);
73
74                  if  GG = 0 then
75                    do;
76                                          F&U = j(1,1,0);
77                                          FT&U =  &f8;
78                                  store FT&U;
79                          create Abc.hev&U varFT&U;
80                          append;
81                          show contents;
82                                      close ratio&U;
83                          end;
84             end;*/
85
86           %end;
87
88     quit;
89
90  %Mend procIML1;
91  %procIML1
92
93  data Abc.datamer2;
94  FT1 = 1;
95  RUN;
96
97
98  %macro checkds;
99  %Do i = 1 %TO &OBSCOUNT;
100   %if %sysfunc(exist(Abc.hev&i)) %then %do;
101       data Abc.datamer2;
102               merge Abc.datamer2 Abc.hev&i;
103           RUN;
104   %end;
105 %end;
106 %mend checkds;
107 %checkds
108
109 proc transpose data= Abc.Datamer2
110 out=Abc.Transp
111 name=VAR
112 prefix=Slice_point;
113 run;
114 data Abc.Transp2;
```

```
115           set Abc.Transp Abc.Lastob2;
116           RUN;
117   data Abc.rangee1(drop = VAR);
118   set Abc.Transp2;
119   run;
120
121   data &EM_EXPORT_TRAIN;
122   set Abc.rangee1;
123   run;
124
125   %em_register(key=Class3, type=data);
126   data &em_user_Class3;
127   set &EM_EXPORT_TRAIN;
128   run;
129
130
131   %EM_REPORT(key=Class3,
132              viewtype=Data,
133              description= Conversation slice points);
134
135
136   proc print data = Abc.rangee1;
137   run;
138
139
140
```

Code 3

```
1
2
3
4   options minoperator mindelimiter=',';
5   %Macro Constr;
6   data outputtracking;
7   set mysas.outputtracking(firstobs = 2);
8   run;
9
10  data Innerdata;
11    set outputtracking(Firstobs = 2);
12  run;
13   proc sql noprint;
14     select distinct chatnum into: nxtpt separated by ','
15     from outputtracking;
16  quit;
17  proc sql noprint;
18     select distinct chatnum into: nxtpt2 separated by ','
19     from Innerdata;
20  quit;
21
22
23   proc sql noprint;
24       select count(*)
25       into :OBSOUTPUTCOUNT
```

```
26      from outputtracking;
27   quit;
28
29  data mysas.classified;
30  run;
31
32   %DO j = 1 %TO &OBSOUTPUTCOUNT;
33    %if &j in(&nxtpt) %then
34     %do;
35
36       data mysas.Fout&j;
37       run;
38       data mysas.Fout&j;
39        set mysas.Fout&j mysas.out&j;
40       run;
41
42       proc tgparse data=mysas.out&j
43              out=parseOut key=key
44       stemming=yes tagging=no
45       entities=no ng=std
46       stop =sasuser.Gibstoplist
47       IGNORE = sasuser.Gibstoplist;
48       var message;
49       run;
50
51       PROC SQL;
52       create table BaseTerms as
53       select Term  as Term1
54       from work.Key;
55       quit;
56
57           proc sql;
58       create table senderData as
59       select sender
60       from mysas.out&j;
61
62       proc sql;
63       create table ReceiverData as
64       select Receiver as sender
65       from mysas.out&j;
66
67       data SR;
68       set Senderdata  Receiverdata;
69       run;
70
71       proc sort data = SR nodupkey;
72       by sender;
73       quit;
74
75          %DO i = 1 %TO &OBSOUTPUTCOUNT;
76                    %if &i in(&nxtpt2) %then
77                 %do;
```

```
78                              proc tgparse data=mysas.out&i
79                      out=parseOut key=key
80                      stemming=yes tagging=no
81                      entities=no ng=std
82                                    stop=sasuser.Gibstoplist
83                                    IGNORE = sasuser.Gibstoplist ;
84                                    var message;
85                                    run;
86
87              proc tgparse data= mysas.out&i
88                      out=parseOut2 key=key1
89                      stemming=yes tagging=no
90                      entities=no ng=std ;
91                                    *stop=sasuser.Gibstoplist
92                                    *IGNORE = sasuser.Gibstoplist ;
93                                    var message;
94                                    run;
95
96                                       proc sql noprint;
97                      select count(*)
98                      into :KeyCOUNT
99                      from Key;
100                                      quit;
101
102                                      %put KeyCOUNT = &KeyCOUNT;
103
104                                      %if &KeyCOUNT = 0 %then
105               %do;
106                      data Key;
107                      set key1;
108                      run;
109                     %end;
110
111                                   PROC SQL;
112                                   create table CompareTerms as
113                                   select Term  as Term2
114                                        from work.Key;
115                                   quit;
116
117                                          %let maxscore=2000;
118                                           proc sql;
119                                             create table Scoredata as
120                                   select Term2, Term1 ,
121                                   compged(Term2 ,Term1,&maxscore,'iLN' ) as gedscore,
122                                   (length(Term2) + length(Term1)) / 2 as  ablen
123                                   from CompareTerms,BaseTerms
124                                   order by calculated gedscore;
125                                            quit;
126
127             /*Finding out the different in number of users between chats */
128                                          proc sql;
129                 create table senderData2 as
```

```
130                      select sender as Receiver
131                      from mysas.out&i;
132                      proc sql;
133                      create table ReceiverData2 as
134                      select Receiver
135                      from mysas.out&i;
136                      data SR2;
137                      set Senderdata2  Receiverdata2;
138                      run;
139                      proc sort data = SR2 nodupkey;
140                      by Receiver;
141                      quit;
142
143                       data SR22;
144                       set SR2;
145                                                Receiver2 = CATS("'",Receiver,"'");
146                                               RUN;
147
148                                               data SR11;
149                       set SR;
150                                               Sender2 = CATS("'",Sender,"'");
151                                               RUN;
152
153                      proc sql noprint;
154                      select distinct Receiver into: Rcv separated by ','
155                      from SR22;
156                      quit;
157
158                      %put &Rcv;
159
160                      data BASEDATA(drop=k);
161                        set SR;
162                        array test*_ALL_;
163                          do k =1 to dim(test);
164                             if testk not in (&Rcv) then output;
165                          end;
166                      run;
167
168
169
170                      proc sql noprint;
171                       select distinct Sender into: Rcv separated by ','
172                       from SR11;
173                      quit;
174
175                     %put &Rcv;
176
177                     data COMPAREDATA(drop=k);
178                      set SR2;
179                       array test*_ALL_;
180                        do k=1 to dim(test);
181                            if testk not in (&Rcv) then output;
```

```
182                       end;
183                     run;
184
185                                          proc sql noprint;
186                       select count(*)
187                       into :BASECOUNT
188                       from BASEDATA;
189                     quit;
190                     %put Count=&BASECOUNT.;
191
192                     proc sql noprint;
193                        select count(*)
194                        into :COMPARECOUNT
195                        from COMPAREDATA;
196                     quit;
197                     %put Count=&COMPARECOUNT.;
198
199                      %let DiffCount = %eval(&BASECOUNT + &COMPARECOUNT);
200
201                      %put TDiffCount= &DiffCount,;
202
203
204                                   /*conversations are far apath from each other if DiffCount is eaull to DiffCoun
205
206                                             proc sql noprint;
207                        select count(*)
208                        into :BASECOUNT_SR
209                        from SR;
210                      quit;
211                      %put Count=&BASECOUNT_SR.;
212
213                      proc sql noprint;
214                         select count(*)
215                         into :COMPARECOUNT_SR2
216                         from SR2;
217                      quit;
218                       %put Count=&COMPARECOUNT_SR2.;
219
220                      %let DiffCount_SORT = %eval(&BASECOUNT_SR + &COMPARECOUNT_SR2);
221
222                      %put TDiffCount_SORT = &DiffCount_SORT,;
223
224
225                   proc sql;
226                   create table MWID as
227                   select gedscore
228                   from Scoredata
229                   where  gedscore = 0;
230                   quit;
231                   proc sql noprint;
232                        select count(*)
233                        into :WMPERCCOUNT
```

```
234                          from MWID;
235                    quit;
236
237                    %put WordPercent = &WMPERCCOUNT;
238
239
240                                              proc iml;
241                                                     use Scoredata;
242                                                read all vargedscore INTO STG;
243                                                list all;
244                                                GS = ALL(STG);
245                                                print(GS);
246
247
248                                                if  &BASECOUNT = 0  &  &COMPARECOUNT = 0  then
249                                               do;
250                                                %include 'C:2012 SAS Files_working.sas';
251                                              end;
252
253                                                else if  GS = 0  &  &DiffCount ^= &DiffCount_SORT then
254                                                 do;
255                                                %include 'C:2012 SAS Files_working.sas';
256                                              end;
257
258
259
260                                                else if GS = 0  &  &BASECOUNT = 0  &  &COMPARECOUNT <= 3  then
261                                                 do;
262                                                %include 'C:2012 SAS Files_working.sas';
263                                              end;
264
265
266                                                else if  GS = 0  &  &BASECOUNT <= 3  &  &COMPARECOUNT = 0 then
267                                                 do;
268                                                %include 'C:2012 SAS Files_working.sas';
269                                              end;
270
271                                         quit;
272                                    %end;
273               %END;
274
275                    proc iml;
276              %include 'C:2012 SAS Files_working2.sas';
277                    quit;
278
279
280              proc sql noprint;
281                         select distinct cl into: cl2 separated by ' '
282                         from mysas.Classified;
283           quit;
284
285          DATA Tgp TGP2;
```

```
286                 do p = 1 to &OBSOUTPUTCOUNT;
287             if p not in(&cl2 ) then OUTPUT Tgp;
288             else OUTPUT Tgp2;
289                 end;
290                 run;
291                  data TgpInner;
292                  set Tgp(Firstobs = 2);
293                  run;
294                     proc sql noprint;
295                          select distinct p into: nxtpt separated by ','
296                          from Tgp;
297                 quit;
298
299                  proc sql noprint;
300                          select distinct p into: nxtpt2 separated by ','
301                          from TgpInner;
302                 quit;
303                  %put &nxtpt;
304                  %put &nxtpt2;
305       %end;
306     %END;
307   %Mend Constr;
308   %Constr
309
```

## C. ADDITIONAL RESULTS

00:09 Sunday, September 22, 2013 1

***Vocabulary in long group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| 1 | abotu | energi | jp | lsee | rlyou | wsaysw |
| 2 | ac | equal | kettl | lsexi | rmattl | wsmooch |
| 3 | accuraci | error | kind | lshe | rod | wsnuggl |
| 4 | addictl | eval | ko | lsnarf | rong | wstand |
| 5 | admin | even | laaaaaaa | lso | rover | wto |
| 6 | afk | excel | lactual | lsome | rpg | wyou |
| 7 | agent | except | laff | lstill | rule | xenosaga |
| 8 | ahh | excus | lah | lsweet | saturdai | xfnat |
| 9 | ahhh | extremli | lairship | lswordfi | sausag | ye |
| 10 | alarm | ey | lan | ltell | saw | yell |
| 11 | alt | face | land | lthat | saysl | york |
| 12 | andl | farmer | landi | lthe | schedul | yuou |
| 13 | ann | fbil | lariel | lthough | score | dog |
| 14 | appi | feel | lask | lto | second | dolc |
| 15 | asksl | fei | lat | ltraxl | serpent | drool |
| 16 | ass | ferrari | laught | luh | sesh | druid |
| 17 | assist | fifti | law | lunch | set | dsai |
| 18 | assistan | file | lbalrog | lwe | sheep | dump |
| 19 | astonish | find | lbellow | lwell | shower | emerelda |
| 20 | atl | finger | lbilbo | lwhat | shrug | jasond |
| 21 | atm | flag | lbtw | lwhee | size | jasonl |
| 22 | attemp | flail | lbut | lwhen | skill | jaw |
| 23 | bad | flatland | lcaesard | lwill | slap | jazz |
| 24 | bank | fof | lcheer | lwto | slaw | jcb |
| 25 | bastard | foreach | lchelder | lye | slimi | jeez |
| 26 | bb | friend | lchera | lyep | slobber | joei |
| 27 | bbl | fuck | lcherawl | lyou | smirk | lpoint |
| 28 | beer | fuel | lchivon | lyour | smooch | lpoor |
| 29 | beeyotch | fuse | lcool | lyup | snicker | lquiet |
| 30 | bestl | gai | ldai | lzero | sniper | lrlto |
| 31 | bet | gasp | ldamn | magic | snofox | lsai |
| 32 | bigiu | giggl | ldamnit | map | snort | lsailor |
| 33 | black | glask | ldo | math | somethig | lsaysl |

*Fig. C.1:* Long conversation

00:09 Sunday, September 22, 2013 **1**

***Vocabulary in long group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| **1** | blink | glat | ldon | mean | sooooooo | red |
| **2** | blood | glbeam | ldyason | messag | sound | rlchildr |
| **3** | blue | glbut | learn | meyer | space | rlokai |
| **4** | bodi | glha | leg | mike | spank | rlponder |
| **5** | boi | gli | lemerald | mind | starac | rlsai |
| **6** | boot | gllha | ler | mine | stat | rlsure |
| **7** | botspot | gllike | let | minut | stretch | rlyai |
| **8** | bout | gllol | letter | miss | stroke | whors |
| **9** | bug | glmentio | level | mistak | suck | wil |
| **10** | bye | glnjoi | leven | mmmmm | supris | wldinner |
| **11** | caesar | glno | lfei | mmmmmmmm | sweet | wnod |
| **12** | cage | glnot | lfloydd | monsti | system | woman |
| **13** | call | glput | lfrozenf | moon | talk | wonder |
| **14** | calporni | glreturn | lgame | muahahah | tcz | word |
| **15** | car | glrun | lgrin | muahahah | teh | |
| **16** | card | glsai | lhe | music | tell | |
| **17** | care | glseeya | lheh | myconid | temp | |
| **18** | carri | glstupid | lhehe | nake | term | |
| **19** | case | glthink | lhug | neer | test | |
| **20** | cawwwwww | glwabe | li | neig | thean | |
| **21** | cell | glwave | lif | new | thing | |
| **22** | cenciusl | glwhoa | lightn | night | think | |
| **23** | chairman | go | line | nod | topic | |
| **24** | charact | god | list | nostolgi | total | |
| **25** | charg | gold | lixalon | number | totp | |
| **26** | charger | gotta | ljust | object | transpor | |
| **27** | chick | green | lkinklad | onder | traxwll | |
| **28** | chili | grin | llaff | onion | trevor | |
| **29** | cial | gropu | lland | oper | tripl | |
| **30** | citan | gropus | llater | outta | triumpha | |
| **31** | civilisl | group | llayawai | owner | try | |
| **32** | ckinki | groupsiz | llhold | page | unicorn | |
| **33** | clmonsty | gun | llmost | pain | unknown | |

*Fig. C.2:* Long conversation

00:09 Sunday, September 22, 2013 **1**

***Vocabulary in long group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| **1** | clnaaaaa | gurn | llog | part | upright | |
| **2** | code | gutter | llorax | parti | upwhen | |
| **3** | columel | habit | llthei | pattern | usag | |
| **4** | command | haha | llwe | peep | user | |
| **5** | connect | head | llyour | person | video | |
| **6** | cool | headfuck | lmake | pickup | villag | |
| **7** | copi | heh | lmike | pilot | violent | |
| **8** | coug | help | lmonster | pint | vita | |
| **9** | crap | high | lnah | place | vock | |
| **10** | creat | hiiiiiii | lno | plai | vockl | |
| **11** | ctrl | hit | lnot | player | voic | |
| **12** | cyai | hjavet | lof | plug | vote | |
| **13** | cycl | hmmmmmmm | log | poet | vowel | |
| **14** | daftiu | hour | logfil | point | wait | |
| **15** | damnit | hsi | loh | pointels | walt | |
| **16** | dask | hun | lok | poke | want | |
| **17** | dave | hunter | lokai | pontoon | war | |
| **18** | del | icq | lol | poo | wave | |
| **19** | demand | idl | lomg | popular | wbore | |
| **20** | dfire | infin | look | post | wcan | |
| **21** | director | intersec | loooo | pot | wchuckl | |
| **22** | displai | isa | lor | power | weather | |
| **23** | dissi | italian | lorax | press | websit | |
| **24** | dkathryn | ixalon | loui | preview | week | |
| **25** | dli | ixq | low | problem | wgrin | |
| **26** | dlwhell | janitor | lpage | put | whinei | |
| **27** | doe | jason | lpah | rdsai | whitman | |

*Fig. C.3:* Long conversation

00:09 Sunday, September 22, 2013 **1**

***Vocabulary in short group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | aabus | channel | dlu | geek | gonna | kinda | ltcz | black | die |
| **2** | absolut | char | dlyeah | gexclaims | goodi | know | lthink | blahblah | differ |
| **3** | accent | charact | dlyep | ggrin | gotta | known | lto | blink | dildo |
| **4** | access | check | dlyou | ghover | govern | ko | ltoss | bloodi | din |
| **5** | account | cheer | doctor | ghug | gprime | lab | lturn | blue | director |
| **6** | addi | cherub | doen | giggl | grail | ladi | lwave | blush | dlahh |
| **7** | addict | chicken | doesnt | ging | grandkid | laff | lwild | board | dland |
| **8** | address | childbirt | doghous | ginger | grimei | lalright | lwto | boggl | dlbrazil |
| **9** | admin | choic | dollar | girl | grin | lamp | mail | boi | dlbut |
| **10** | advert | chpid | domesto | gladn | grinz | lampost | mailbox | boltz | dlcan |
| **11** | aer | christian | dont | gland | ground | land | main | bond | dlcigaret |
| **12** | affect | christin | door | glannoi | grt | languag | majik | book | dlderrick |
| **13** | ag | cig | dough | glare | gsaysl | lap | manufactu | born | dleveryon |
| **14** | agian | cigarett | dragon | glask | gshake | laptop | master | boss | dlewwwww |
| **15** | ahahahaha | class | dream | glawwwww | gsnort | lardbal | mate | bottom | dlgood |
| **16** | ahh | classic | drink | glbass | gthink | lask | math | break | dlgreet |
| **17** | ahold | cliff | drug | glbite | gto | lasksi | matter | brick | dlhrm |
| **18** | aint | clokai | druid | glblink | guess | lasksl | mayb | brother | dlhrrmrmr |
| **19** | aircraft | cmake | dsai | glbow | gui | lat | meep | brow | dli |
| **20** | alcohol | cnice | duck | glbrb | guilt | laugh | meet | bruiss | dlif |
| **21** | alot | code | dumb | glbzzz | gun | law | memori | btw | dlit |
| **22** | andl | comdom | dunno | glcan | gwait | lbeeyotc | messag | buck | dllee |
| **23** | anim | comfort | dwink | glchortl | gwave | lbilbo | mhtz | bump | dlneither |
| **24** | answer | command | ear | glchuckl | gwto | lbounc | militari | bunch | dlnice |
| **25** | aol | comp | ebai | glcri | hair | lcontemp | million | butt | dlnight |
| **26** | apoor | complex | eck | gldont | half | ldisconn | min | button | dlnor |
| **27** | appoint | compound | edit | gldroll | hand | ldrool | mind | bye | dlor |
| **28** | apprentic | comput | eek | gldrool | hang | leav | minut | case | dlprick |
| **29** | approv | contracto | effect | gleat | hard | left | missil | cask | dlprod |
| **30** | aprov | control | elder | gler | hassl | leftov | mission | cat | dlscottis |
| **31** | arm | convers | element | glerr | hate | leg | missl | cattl | dlsmoke |
| **32** | arrai | convo | emp | glexclaim | head | leicest | mistook | ced | dlsometim |
| **33** | ars | cooki | employ | glexcus | heart | lemm | mlsai | cent | dlsound |

*Fig. C.4:* Short conversation

00:09 Sunday, September 22, 2013 **1**

***Vocabulary in short group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | ask | cool | end | glfrog | heat | lestah | mmmmmmmm | chanc | dlthe |
| **2** | asksl | cooli | english | glg | hee | let | mom | chang | dlthei |
| **3** | ass | cooollll | episod | glgasp | heh | letter | monei | piscin | saint |
| **4** | astonish | corner | equip | glgiggl | hehe | lexclaim | monitor | piss | sale |
| **5** | atl | cost | error | glgrin | height | lexclaim | monster | pizza | salin |
| **6** | attent | couch | escap | glha | hell | lfor | month | place | saliva |
| **7** | attept | countri | estim | glhate | hello | lfrown | moon | plan | sax |
| **8** | averag | coupl | ether | glheheheh | help | lgandalf | morn | plasma | saysl |
| **9** | avid | coupon | ev | glhell | high | lgot | moss | plastic | scan |
| **10** | aw | crafti | eval | glhug | highligh | lgrin | moth | pleaseee | scar |
| **11** | aww | crap | exam | gli | highwai | lheheh | mountain | plug | scari |
| **12** | babe | dad | excess | glick | hihi | lhug | mouth | poet | school |
| **13** | babi | dammit | excit | glii | histori | lhuggl | movi | point | score |
| **14** | bad | damn | exclaim | glit | hiya | liber | mpeg | pong | scream |
| **15** | ban | damnit | excreme | glk | hmm | lie | musta | poor | screen |
| **16** | bass | dask | excus | gllaff | hobo | light | mycl | privatli | sexual |
| **17** | batch | date | experi | gllaugh | holi | likkl | nak | priveled | shame |
| **18** | beat | dc | explor | gllike | home | lil | nbnet | prob | sharessol |
| **19** | bed | deal | extra | gllisten | homework | limit | neat | problem | sharestot |
| **20** | beddi | dear | ey | gllix | hook | lin | need | profil | shelf |
| **21** | beer | death | fact | gllol | hope | line | net | program | shift |
| **22** | begin | dec | fail | gllponder | horizon | link | nevermind | prohibit | shirt |
| **23** | behavior | defens | failur | glmy | hour | lion | newcastl | prolli | shit |
| **24** | bestl | degre | fantasi | glno | hous | list | newt | promot | shitfit |
| **25** | bet | depart | fat | glnod | howr | llamaz | nibbl | pron | shitl |
| **26** | betta | desc | fault | glnoddl | hrm | llaugh | nicotin | provok | shitload |
| **27** | big | descript | favorit | glnot | hug | llbass | night | psysic | shouldnt |
| **28** | bin | detector | favour | gloh | huggl | llbeani | nik | public | shoutsl |
| **29** | birthdai | deton | feel | glokai | hun | lldo | nod | pudgi | side |
| **30** | bitchslap | dexclaim | fellow | glor | iaido | lldpromi | nois | puppet | sigh |
| **31** | bite | didnt | fer | glow | icecream | lllaugh | nose | qmu | simpson |
| **32** | blablabla | didtn | file | glprod | idea | llokai | novemb | quarter | sing |
| **33** | gblink | gof | filet | glput | idiot | llook | nuke | quest | singl |

*Fig. C.5:* Short conversation

00:09 Sunday, September 22, 2013 1

***Vocabulary in short group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | gear | goin | film | glrealli | ill | llsai | number | question | sister |
| 2 | talk | pope | final | glreconne | illeg | llshiver | okai | quiet | site |
| 3 | talker | poplist | find | glsai | illinoi | llthat | okthank | quota | siterepor |
| 4 | tax | popsit | fine | glshut | import | llwhat | omg | wink | wnoddl |
| 5 | tcz | popular | fire | glsinc | inapprop | llwhoops | ooher | wish | woe |
| 6 | tdo | porn | flag | glsmirk | inbox | llyawn | opinion | wit | won |
| 7 | teas | port | flick | glspank | incest | llynxd | ounch | witch | wonder |
| 8 | technolog | pot | folder | glthank | index | lmayb | outcom | wizard | word |
| 9 | tell | ppl | folk | glthat | indi | lmillymo | outta | wizzi | worship |
| 10 | temp | practic | foot | glthe | individu | lnod | owner | wlaugh | worth |
| 11 | term | prefix | forc | glthink | info | locat | pack | wlick | wouldnt |
| 12 | terribl | premier | forest | gltitl | ingor | lof | page | stereo | steve |
| 13 | test | press | formatt | gltrue | instruct | loge | pain | tonight | rlso |
| 14 | thanx | price | formula | glttttthh | intellem | look | pair | tooth | rlthei |
| 15 | trace | printer | freak | glue | internet | loon | pant | total | rlwhy |
| 16 | trash | rad | freakin | glwalk | invis | looni | park | towel | rlyo |
| 17 | tree | rage | free | glwave | iri | loop | part | vacat | rlyou |
| 18 | trenchcoa | rai | fresh | glwell | iron | lord | parti | vampir | robberi |
| 19 | trick | rais | freshma | glwhat | isnt | losingli | particl | vell | room |
| 20 | troublesh | read | fri | glwhee | ixalond | lot | pastl | version | royal |
| 21 | TRUE | real | fridg | glwill | jan | love | pathet | veryon | rule |
| 22 | try | reason | friend | glwish | jazz | low | payer | video | run |
| 23 | tryin | red | fuck | glwonder | jill | lpoint | pee | voic | sabbath |
| 24 | tsk | ref | fucker | glworship | jimmi | lpout | peni | wahat | stir |
| 25 | ttyl | refin | fun | glye | job | lrais | percent | wait | stjh |
| 26 | turd | regular | fustrat | glyep | joei | lrlto | perfect | wall | stop |
| 27 | turn | remot | futur | glyou | joel | lsai | perfectio | walmart | straight |
| 28 | twix | request | fuzzi | gmischan | john | lsaysi | perform | wand | strang |
| 29 | type | resid | game | go | joke | lsaysl | person | wanna | stranger |
| 30 | typo | resolut | gank | god | karat | lshout | pic | want | strip |
| 31 | uit | respect | gasksl | goddess | kate | lshudder | picci | warfar | stuff |
| 32 | ultra | rest | gate | goesl | kiddi | lsit | pick | wasksw | stupid |
| 33 | uninforc | restuar | sound | sex | kiddin | lsmile | pictur | wasnt | subsitut |

*Fig. C.6:* Short conversation

00:09 Sunday, September 22, 2013 1

***Vocabulary in short group conversation***

| Obs | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | univers | return | sp | slap | kill | ltalk | piec | watch | substanc |
| 2 | user | revers | spain | sleep | screensh | that | rlme | watcher | substitut |
| 3 | wread | review | spank | slut | scum | thing | rlmike | watt | success |
| 4 | write | ring | speaker | smell | sea | think | rlmmm | wave | suck |
| 5 | wrong | rlandyyyy | special | smile | seeya | thought | rlmmmm | waver | summer |
| 6 | wsaysw | rlasleep | spectru | smirk | sell | tickl | rlnope | wboot | sumptin |
| 7 | wsend | rlboltzie | spell | smoke | send | tingli | rloh | webadmin | sundai |
| 8 | rlhi | rlglori | spunker | smoker | sens | titl | rlokai | webpag | sunglass |
| 9 | rljason | rlheh | squad | snicker | sentenc | todai | rloohr | week | super |
| 10 | rljust | rlhei | standar | snort | seper | tom | rlsai | wgrin | suprising |
| 11 | wimp | window | star | sock | sequenc | tomorrow | rlshe | whuggl | sweet |
| 12 | setup | statement | start | song | serpent | sort | soooooo | wil | good |
| 13 | soooooonis | server | | | | | | | |

*Fig. C.7:* Short conversation

*Tab. C.1:* Dynamics of RWT for pairs of people: user K and others with varying alpha values

| People | dist | $\chi$ | df | P-value | alpha | b | C |
|---|---|---|---|---|---|---|---|
| 1 | Zm | 4.20 | 5 | 0.52 | 0.570 | 0.015 | 2.48 |
| 2 | Zm | 5.99 | 6 | 0.11 | 0.720 | 0.065 | 0.59 |
| 3 | Zm | 7.48 | 7 | 0.38 | 0.506 | 0.020 | 3.39 |
| 4 | Zm | 5.84 | 5 | 0.32 | 0.567 | 0.022 | 2.22 |
| 5 | Zm | 4.25 | 4 | 0.37 | 0.583 | 0.026 | 1.87 |
| 6 | Zm | 0.89 | 4 | 0.83 | 0.749 | 0.025 | 0.62 |
| 7 | Zm | 6.66 | 6 | 0.35 | 0.517 | 0.024 | 2.94 |
| 8 | Zm | 1.57 | 6 | 0.95 | 0.522 | 0.017 | 3.27 |
| 9 | Zm | 1.47 | 3 | 0.68 | 0.465 | 0.018 | 4.52 |
| 10 | Zm | 0.40 | 4 | 0.94 | 0.599 | 0.018 | 1.98 |
| 11 | Zm | 0.18 | 4 | 0.97 | 0.594 | 0.041 | 1.46 |
| 12 | Zm | 1.46 | 6 | 0.16 | 0.697 | 0.055 | 1.72 |
| 13 | Zm | 1.57 | 3 | 0.66 | 0.676 | 0.028 | 1.02 |
| 14 | Zm | 1.54 | 6 | 0.95 | 0.320 | 0.017 | 3.31 |
| 15 | Zm | 4.58 | 4 | 0.33 | 0.621 | 0.031 | 1.40 |
| 16 | Zm | 17.80 | 12 | 0.12 | 0.612 | 0.014 | 2.01 |
| 17 | Zm | 79.65 | 5 | 0.32 | 0.567 | 0.022 | 2.22 |
| 18 | Zm | 4.25 | 4 | 0.37 | 0.583 | 0.026 | 1.87 |
| 19 | Zm | 0.89 | 4 | 0.83 | 0.749 | 0.025 | 0.62 |
| 20 | Zm | 6.66 | 6 | 0.35 | 0.517 | 0.024 | 2.94 |
| 21 | Zm | 1.57 | 6 | 0.95 | 0.522 | 0.017 | 3.27 |
| 22 | Zm | 1.47 | 3 | 0.68 | 0.365 | 0.018 | 4.52 |
| 23 | Zm | 5.26 | 10 | 0.87 | 0.33 | 0.023 | 3.77 |
| 24 | Zm | 0.40 | 4 | 0.93 | 0.59 | 0.018 | 1.98 |
| 25 | Zm | 13.05 | 10 | 0.22 | 0.60 | 0.024 | 1.88 |
| 26 | Zm | 1.58 | 4 | 0.66 | 0.31 | 0.010 | 15.38 |
| 27 | Zm | 5.05 | 4 | 0.168 | 0.549 | 0.018 | 2.70 |
| 28 | Zm | 13.87 | 10 | 0.178 | 0.54 | 0.026 | 2.37 |
| 29 | Zm | 4.934 | 3 | 0.17 | 0.56 | 0.014 | 2.72 |
| 30 | Zm | 1.58 | 4 | 0.66 | 0.31 | 0.011 | 15.38 |
| 31 | Zm | 2.50 | 4 | 0.64 | 0.65 | 0.020 | 1.32 |
| 32 | Zm | 7.37 | 7 | 0.40 | 0.504 | 0.020 | 3.422 |
| 33 | Zm | 13.87 | 10 | 0.20 | 0.545 | 0.026 | 2.37 |
| 34 | Zm | 8.55 | 7 | 0.28 | 0.561 | 0.032 | 1.97 |

*Tab. C.2:* Dynamics of RWT for pairs of people with varying alpha values

| People | dist | $\chi$ | df | P-value | alpha | sigma | theta |
|---|---|---|---|---|---|---|---|
| 1 | gp | 7.9329 | 6 | 0.2271 | 0.3010 | 22.3480 | 11.0 |
| 2 | gp | 6.9329 | 6 | 0.3271 | 0.4069 | 26.3480 | 12.0 |
| 3 | gp | 4.5867 | 6 | 0.60 | 3.03 | 44.2427 | 1 |
| 4 | gp | 1.7105 | 6 | 0.9443 | 2.9294 | 42.8570 | 1 |
| 5 | gp | 7.7329 | 6 | 0.2583 | 2.8811 | 50.7146 | 1.2 |
| 6 | gp | 8.0966 | 6 | 0.2311 | 2.5027 | 57.4092 | 1 |
| 7 | gp | 2.8610 | 6 | 0.8261 | 1.7176 | 49.5403 | 2 |
| 8 | gp | 1.3823 | 6 | 0.9669 | 3.3042 | 36.0939 | 3.0 |
| 9 | gp | 3.2671 | 6 | 0.7747 | 1.6854 | 27.9655 | 1 |
| 10 | gp | 4.3057 | 6 | 0.6354 | 1.8043 | 36.8140 | 0.4 |
| 11 | gp | 4.3432 | 6 | 0.6303 | 2.1364 | 45.7855 | 1 |
| 12 | gp | 4.2249 | 6 | 0.6463 | 2.4654 | 45.4721 | 0 |
| 13 | gp | 3.7483 | 6 | 0.7107 | 2.0894 | 30.5042 | 0.1 |
| 14 | gp | 12.1704 | 6 | 0.583 | 2.4147 | 53.8124 | 0.3 |
| 15 | gp | 9.5767 | 6 | 0.1436 | 2.7590 | 48.6667 | 1 |
| 16 | gp | 5.5611 | 6 | 0.4741 | 2.1265 | 66.4528 | 1 |
| 17 | gp | 7.4745 | 6 | 0.2792 | 2.7489 | 66.4528 | 1 |
| 18 | gp | 2.1980 | 6 | 0.9006 | 1.6909 | 19.6605 | 0.2 |
| 19 | gp | 8.0400 | 6 | 0.2352 | 2.1765 | 38.9536 | 0.37 |
| 20 | gp | 6.2226 | 6 | 0.3987 | 1.7250 | 23.2741 | 0 |
| 21 | gp | 8.1251 | 6 | 0.2291 | 1.8221 | 22.7837 | 0.3 |
| 22 | gp | 1.7105 | 6 | 0.9443 | 3.0294 | 42.8570 | 1 |
| 23 | gp | 7.7329 | 6 | 0.2583 | 3.001 | 50.7146 | 1 |
| 24 | gp | 4.5867 | 6 | 0.5978 | 3.5329 | 44.2427 | 1 |
| 25 | gp | 1.3823 | 6 | 0.9669 | 3.801 | 36.0939 | 3 |
| 26 | gp | 8.9168 | 6 | 0.2831 | 0.6801 | 37.4286 | 1 |
| 27 | gp | 10.1356 | 6 | 0.3801 | 0.5160 | 34.8286 | 1 |
| 28 | gp | 2.5527 | 6 | 0.863 | 2.4378 | 31.5471 | 2 |
| 29 | gp | 3.0706 | 6 | 0.799 | 1.50 | 40.4651 | 1 |

*Tab. C.3:* Dynamics of RWT for pairs of people with varying alpha values

| People | dist | $\chi$ | df | P-value | alpha | sigma | mu |
|---|---|---|---|---|---|---|---|
| 1 | gev | 4.3699 | 6 | 0.6267 | 1.3400 | 18.4542 | 12.1772 |
| 2 | gev | 12.3441 | 6 | 0.1692 | 1.6162 | 28.8902 | 17.5242 |
| 3 | gev | 4.5867 | 6 | 0.5641 | 0.5654 | 17.5517 | 15.0589 |
| 4 | gev | 3.6112 | 6 | 0.7291 | 2.9294 | 42.8570 | 32.2177 |
| 5 | gev | 11.3003 | 6 | 0.0795 | 1.6993 | 45.9668 | 26.4452 |
| 6 | gev | 2.0437 | 6 | 0.9156 | 1.4683 | 43.2448 | 28.4869 |
| 7 | gev | 6.1276 | 6 | 0.4091 | 1.2398 | 24.5933 | 17.9531 |
| 8 | gev | 1.7593 | 6 | 0.9405 | 1.7865 | 63.4121 | 34.9004 |
| 9 | gev | 2.4480 | 6 | 0.8742 | 1.4170 | 24.9496 | 16.6428 |
| 10 | gev | 10.1785 | 6 | 0.1173 | 1.7065 | 36.3287 | 20.6993 |
| 11 | gev | 10.4499 | 6 | 0.1069 | 1.5963 | 25.6037 | 15.5355 |
| 12 | gev | 3.5105 | 6 | 0.7426 | 1.7775 | 96.9960 | 53.1522 |
| 13 | gev | 4.754 | 6 | 0.6303 | 0.5606 | 12.0574 | 10.4799 |
| 14 | gev | 9.9030 | 6 | 0.1288 | 1.2783 | 35.4031 | 25.8301 |
| 15 | gev | 11.8574 | 6 | 0.0652 | 1.1648 | 29.6449 | 22.9811 |
| 16 | gev | 3.6112 | 6 | 0.7291 | 1.5700 | 52.1786 | 32.2177 |
| 17 | gev | 10.7550 | 6 | 0.0962 | 1.4767 | 37.5802 | 24.0673 |
| 18 | gev | 2.1308 | 6 | 0.9073 | 1.5137 | 46.4222 | 28.3464 |
| 19 | gev | 10.1891 | 6 | 0.1169 | 1.4141 | 33.4409 | 22.0873 |
| 20 | gev | 2.0437 | 6 | 0.9156 | 1.4683 | 43.2448 | 28.4869 |
| 21 | gev | 3.5105 | 6 | 0.7426 | 1.7775 | 96.9960 | 53.1522 |
| 22 | gev | 1.7593 | 6 | 0.9405 | 1.7865 | 63.4121 | 34.9004 |
| 23 | gev | 2.4480 | 6 | 0.8742 | 1.4170 | 24.9496 | 16.6428 |
| 24 | gev | 11.8574 | 6 | 0.0652 | 0.5648 | 29.6449 | 22.9811 |
| 25 | gev | 6.1276 | 6 | 0.4091 | 0.8398 | 24.5933 | 17.9531 |
| 26 | gev | 10.1785 | 6 | 0.1173 | 2.0065 | 36.3287 | 20.6993 |
| 27 | gev | 10.4499 | 6 | 0.1069 | 1.5963 | 25.6037 | 15.5355 |

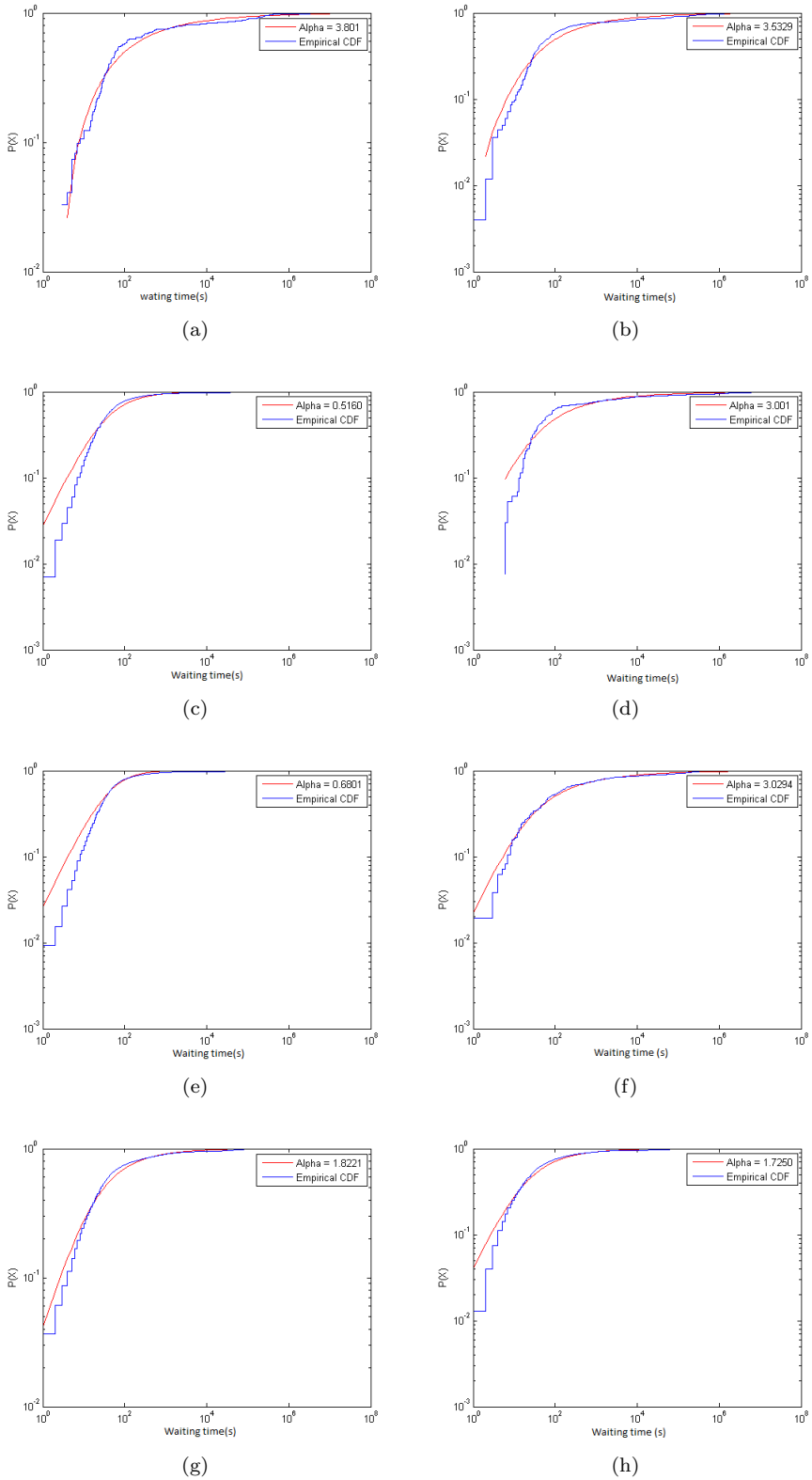



*Fig. C.8:* Pareto distribution: Dynamics of RWT for pairs of people

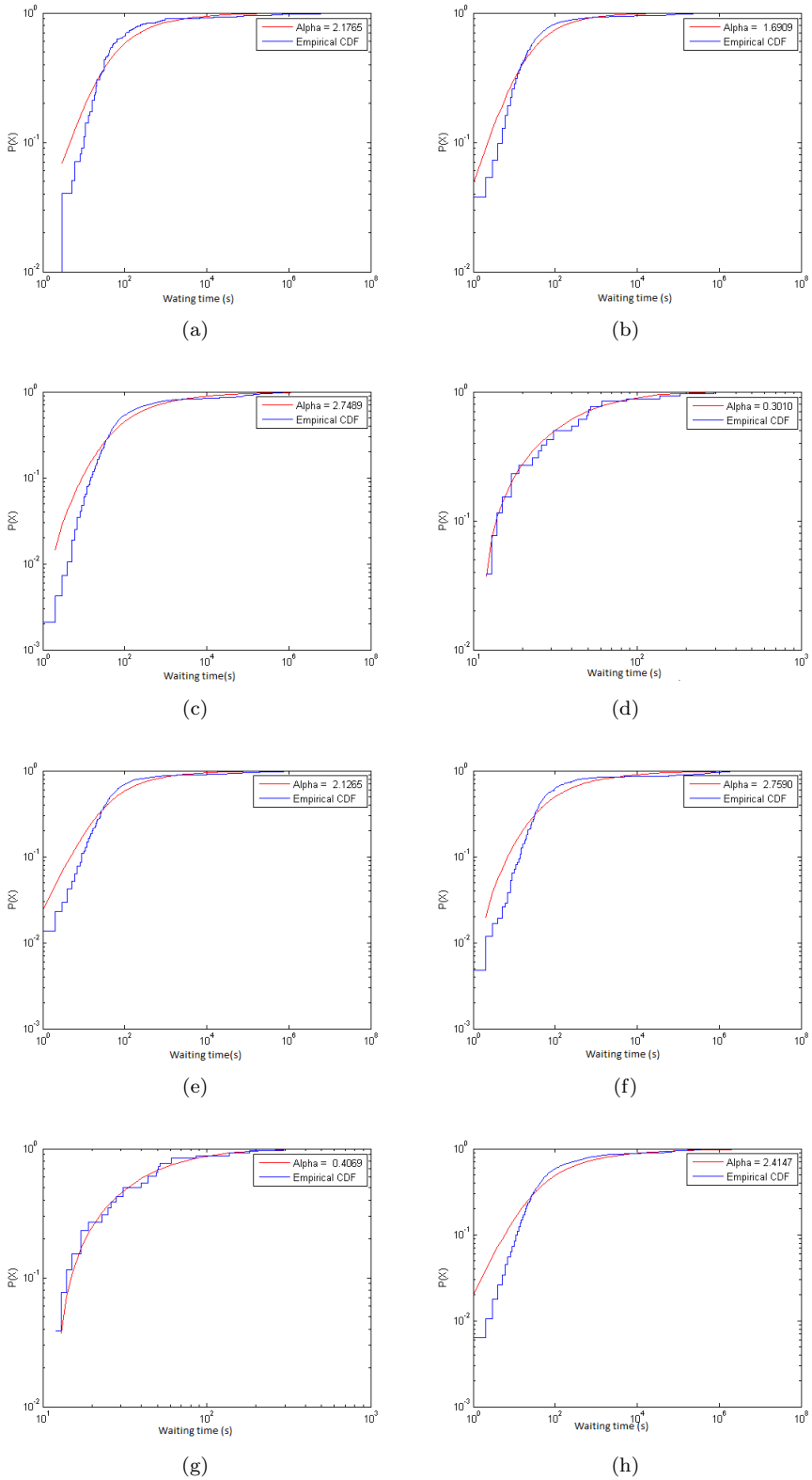



*Fig. C.9:* Pareto distribution: Dynamics of RWT for pairs of people

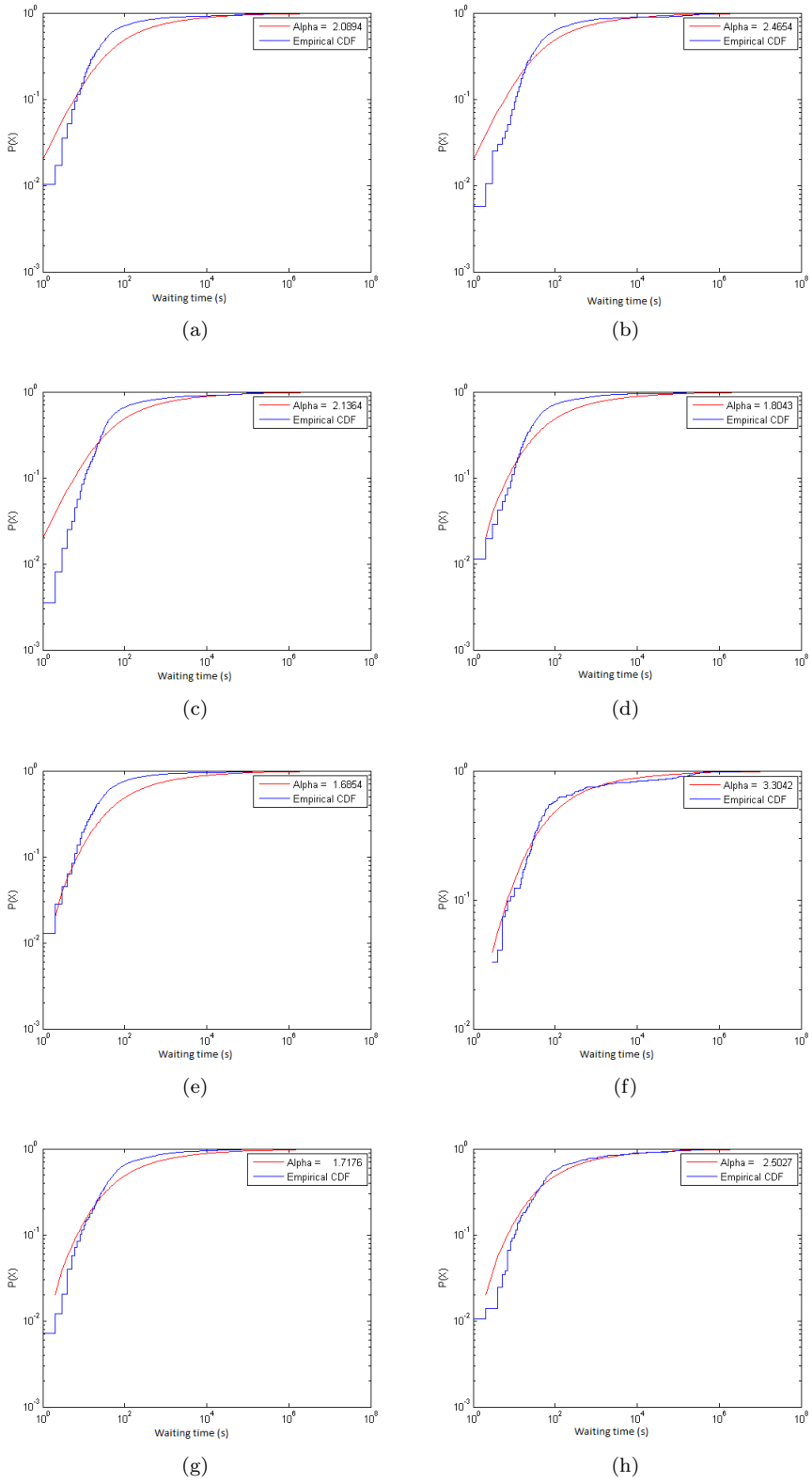



*Fig. C.10:* Pareto distribution: Dynamics of RWT for pairs of people

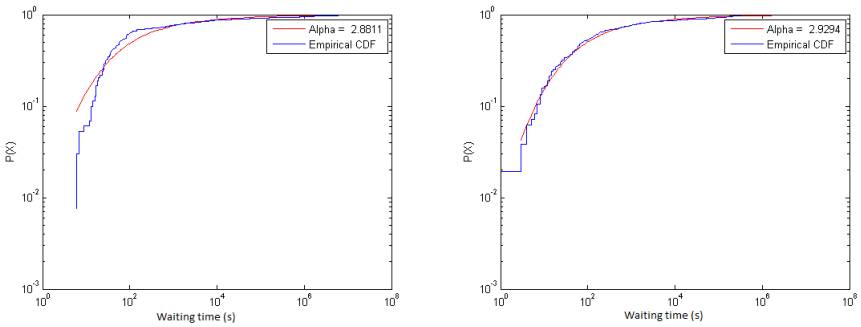



*Fig. C.11:* Pareto distribution: Dynamics of RWT for pairs of people

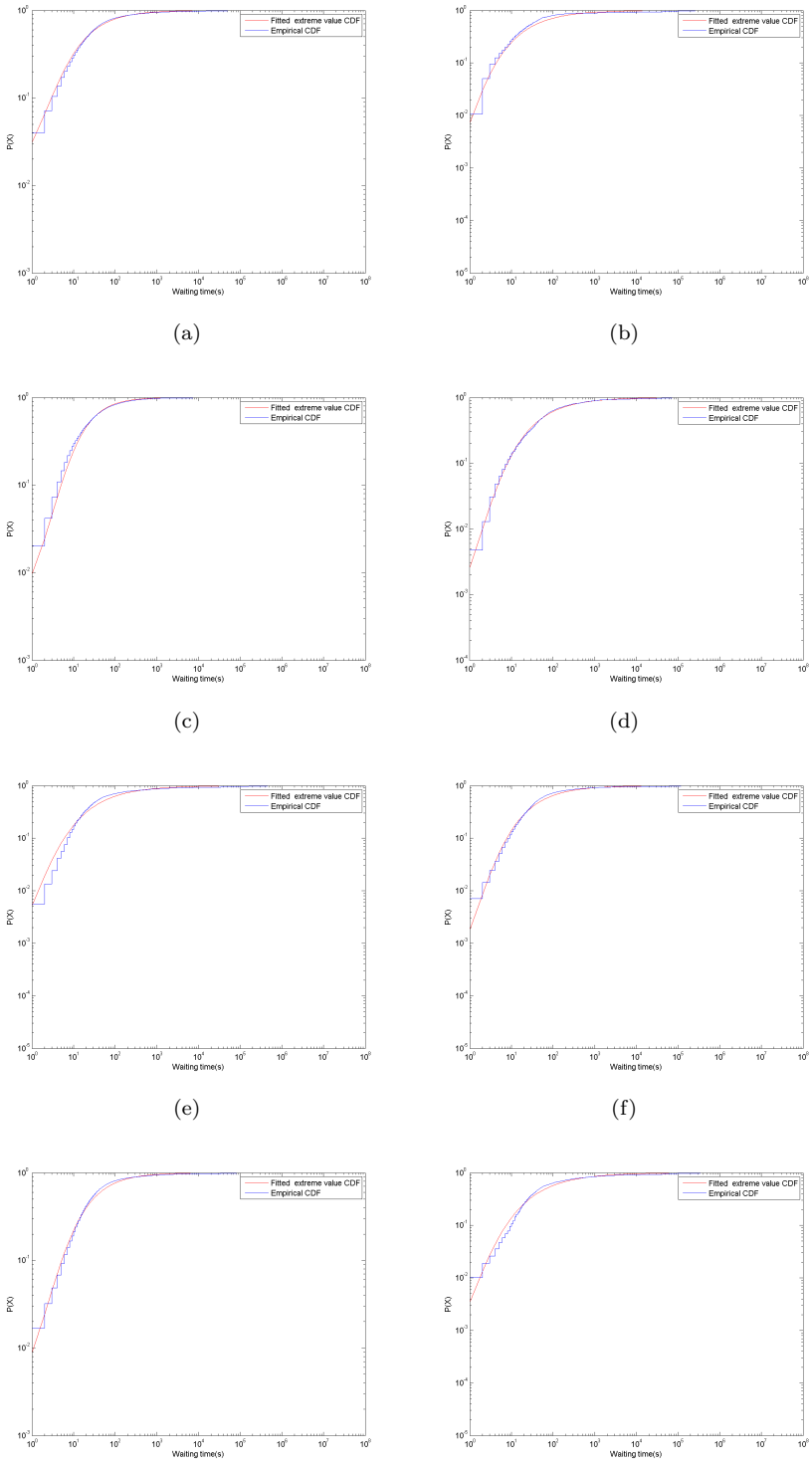



*Fig. C.12:* Generalized extreme value distribution: Dynamics of RWT for pairs of people

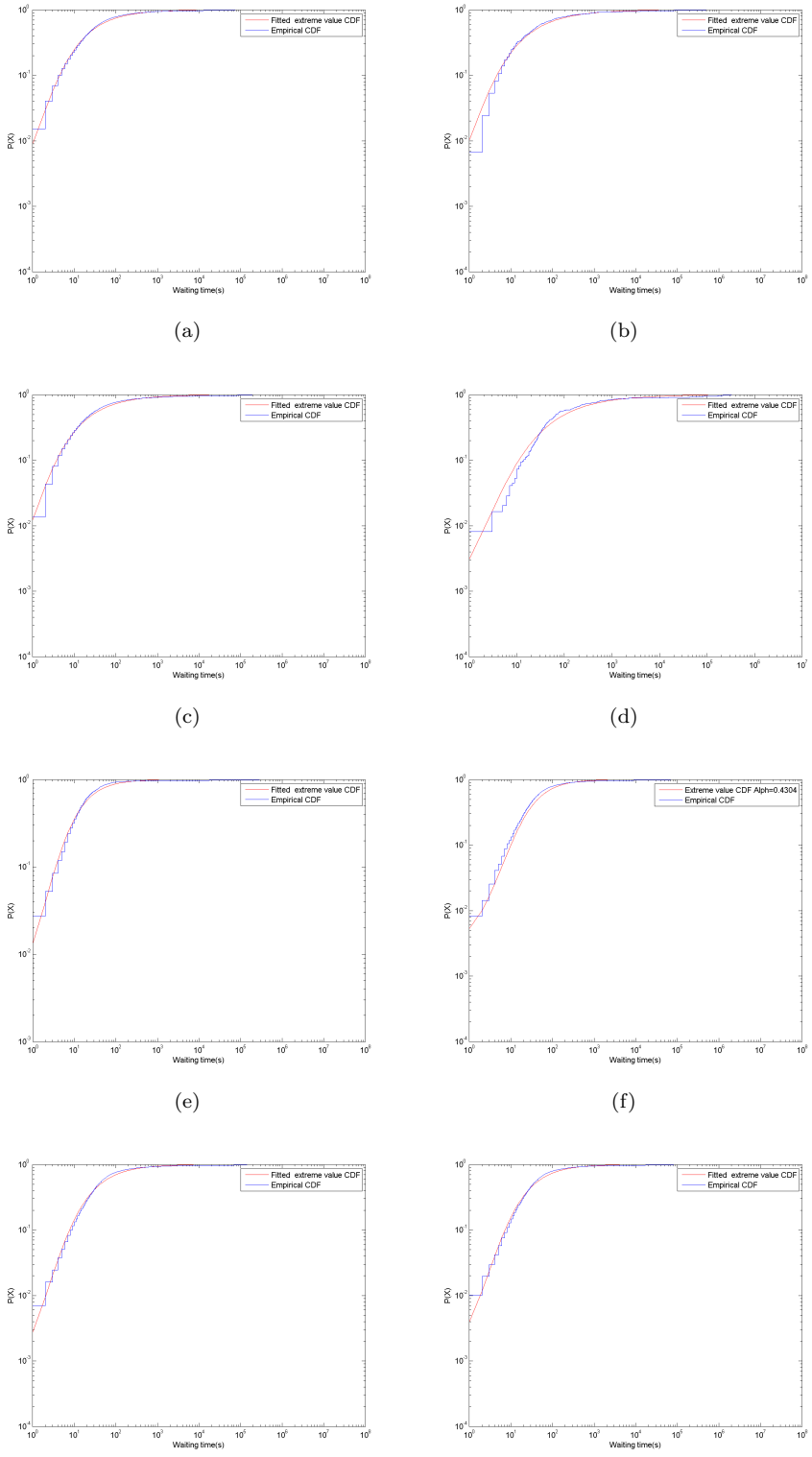



*Fig. C.13:* Generalized extreme value distribution: Dynamics of RWT for pairs of people

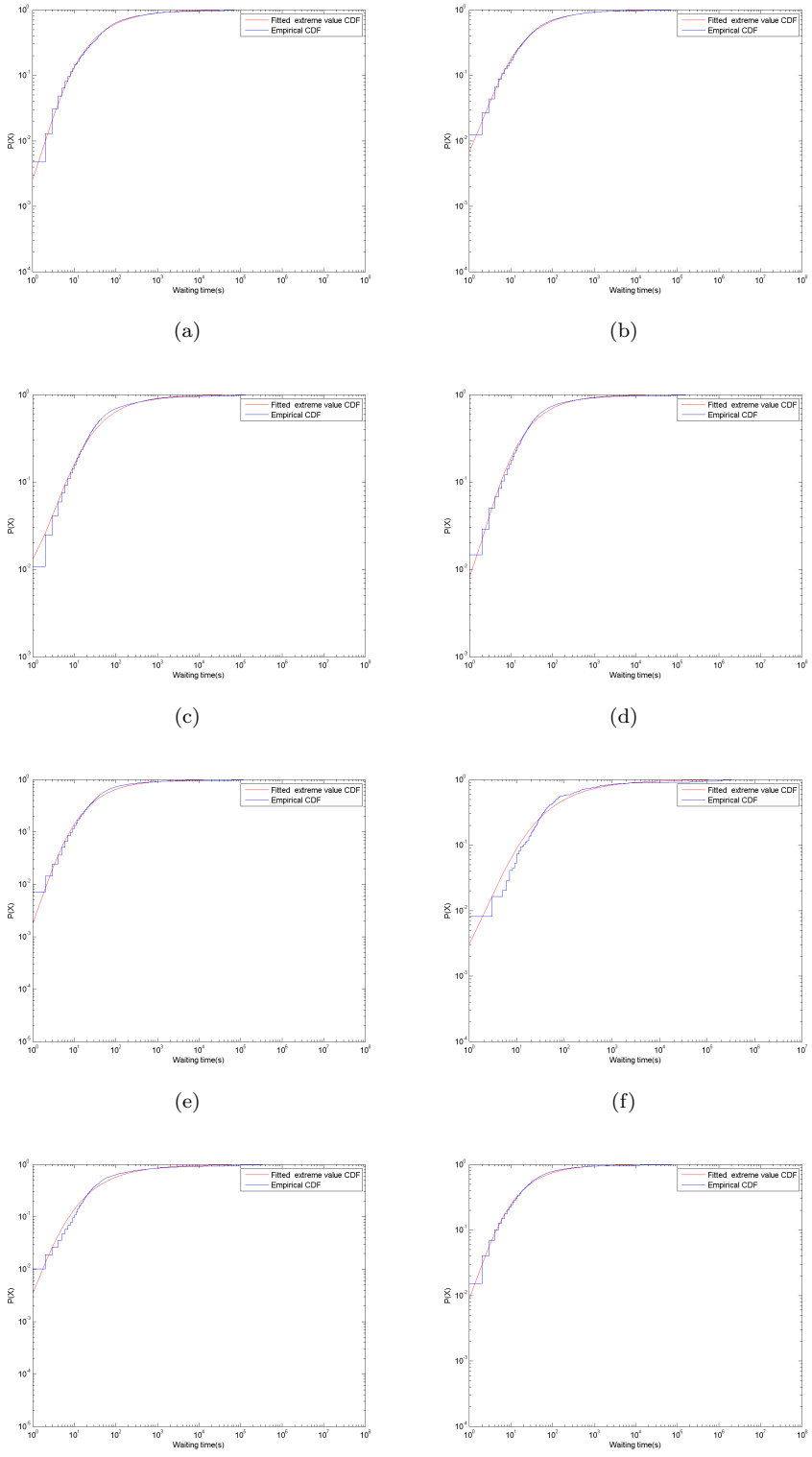



*Fig. C.14:* Generalized extreme value distribution: Dynamics of RWT for pairs of people

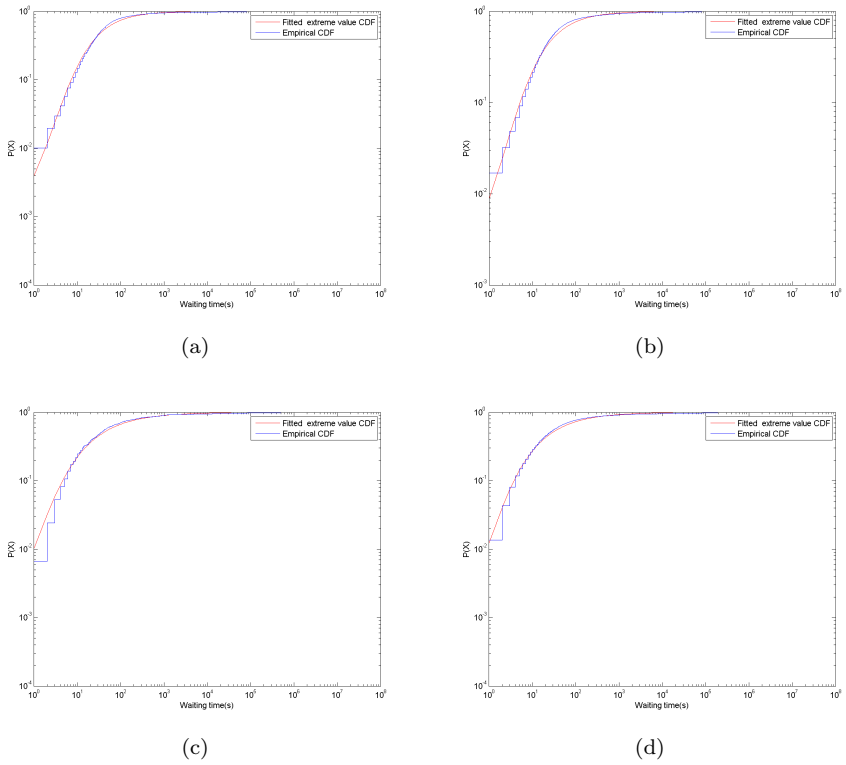



*Fig. C.15:* Generalized extreme value distribution: Dynamics of RWT for pairs of people

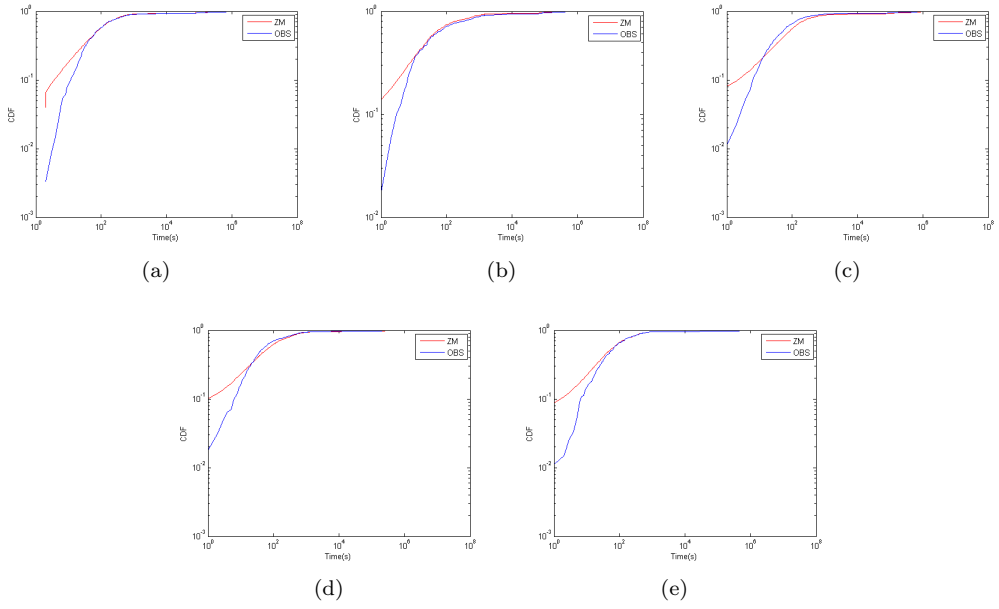



*Fig. C.16:* Dynamics of RWT for pairs of people: user K and others

## BIBLIOGRAPHY


[1] Evrim Acar, Seyit A. Çamtepe, Mukkai S. Krishnamoorthy, and Bülent Yener. Modeling and multiway analysis of chatroom tensors. In *Proceedings of the 2005 IEEE international conference on Intelligence and Security Informatics*, ISI'05, pages 256–268, Berlin, Heidelberg, 2005. Springer-Verlag.

[2] Paige Adams and Craig Martel. Conversational thread extraction and topic detection in text-based chat. *Semantic Computing*, pages 87–113, 2010.

[3] P.H. Adams and C.H. Martell. Topic detection and extraction in chat. In *Semantic Computing, 2008 IEEE International Conference on*, pages 581–588, Aug 2008.

[4] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 835–844, New York, NY, USA, 2007. ACM.

[5] Tim Althoff, Damian Borth, Jörn Hees, and Andreas Dengel. Analysis and forecasting of trending topics in online media streams. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 907–916. ACM, 2013.

[6] Anjo Anjewierden, Bas Kolloffel, and Casper Hulshof. Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes. In *International Workshop on Applying Data Mining in e-Learning (ADML 2007)*, Crete, Greece, 2007.

[7] Ioannis Antonellis and Efstratios Gallopoulos. Exploring term-document matrices from matrix models in text mining. *arXiv preprint cs/0602076*, 2006.

[8] Albert-Laszlo Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[10] Ginestra Bianconi and A-L Barabási. Competition and multiscaling in evolving networks. *EPL (Europhysics Letters)*, 54(4):436, 2001.

[11] Seyit Ahmet Camtepe, Mark Goldberg, Mukkai Krishnamoorty, and Malik Magdon-ismail. Detecting conversing groups of chatters: a model, algorithms, and tests. In *In Proceedings of the IADIS International Conference on Applied Computing*, pages 89–96, 2005.

[12] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *CoRR*, abs/1012.0009, 2010.

[13] Shu-Yan Chan, Pan Hui, and Kuang Xu. Community detection of time-varying mobile social networks. In *Complex Sciences*, volume 4, pages 1154–1159. Springer Berlin Heidelberg, 2009. 10.1007/978-3-642-02466-5_115.

[14] Yi chia Wang, Mahesh Joshi, William Cohen, and Carolyn Ros. Recovering implicit thread structure in newsgroup style conversations, 2008.

[15] H.H. Clark. *Using Language*. Cambridge University Press, 1996.

[16] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703, November 2009.

[17] Giovanni Comarela, Mark Crovella, Virgilio Almeida, and Fabricio Benevenuto. Understanding factors that affect response rates in twitter. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, pages 123–132, New York, NY, USA, 2012. ACM.

[18] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 2007.

[19] Scott Deerwester, Susan T. Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[20] Lucinda Dollman, Catherine Morgan, Jennifer Pergler, William Russell, and Jennifer Watts. Improving social skills through the use of cooperative learning. *Online Submission*, 2007.

[21] A. Duranti. *Key Terms in Language & Culture*. Journal of linguistic anthropology. Wiley, 2001.

[22] Alessandro Duranti. *Linguistic Anthropology*. Cambridge University Press, New York, 1997.

[23] Jonathan S Durham. *Topic detection in online chat*. PhD thesis, Monterey, California. Naval Postgraduate School, 2009.

[24] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale-free topology of e-mail networks. *Phys. Rev. E*, 66:035103, Sep 2002.

[25] E Elnahrawy. Log-based chat room monitoring using text categorization: A comparative study. In *The International Conference on Information and Knowledge Sharing, US Virgin Islands*, 2002.

[26] Micha Elsner and Eugene Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[27] Micha Elsner and Eugene Charniak. Disentangling chat. *Comput. Linguist.*, 36(3):389–409, 2010.

[28] Micha Elsner and Eugene Charniak. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1179–1189, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[29] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[30] Paul Erd6s and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

[31] P Erdős and A Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[32] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, August 1999.

[33] R. Ferrer i Cancho. The variation of zipf's law in human language. *The European Physical Journal B - Condensed Matter and Complex Systems*, 44(2):249–257, 2005.

[34] K.-I. Goh, E. Oh, B. Kahng, and D. Kim. Betweenness centrality correlation in social networks. *Phys. Rev. E*, 67:017101, Jan 2003.

[35] Peter Grindrod, Mark C. Parsons, Desmond J. Higham, and Ernesto Estrada. Communicability across evolving networks. *Phys. Rev. E*, 83:046120, Apr 2011.

[36] Hamed Haddadi, Damien Fay, Almerima Jamakovic, Olaf Maennel, Andrew W. Moore, Richard Mortier, Miguel Rio, and Steve Uhlig. Beyond node degree: evaluating AS topology models. Technical Report UCAM-CL-TR-725, University of Cambridge, Computer Laboratory, July 2008.

[37] D. Haichao and C. H.and Yulan H. Siu. Structural analysis of chat messages for topic detection. *Online Information Review.*, 301:496–516, Jun 2006.

[38] PatrickG.T. Healey, Graham White, Arash Eshghi, AhmadJ. Reeves, and Ann Light. Communication spaces. *Computer Supported Cooperative Work (CSCW)*, 17(2-3):169–193, 2008.

[39] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[40] C. Honey and S. C. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pages 1–10, Jan 2009.

[41] Minjoon Jun, Zhilin Yang, and DaeSoo Kim. Customers' perceptions of online retailing service quality and their satisfaction. *International Journal of Quality & Reliability Management*, 21(8):817–840, 2004.

[42] Gueorgi Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[43] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[44] Kristina Lerman and Rumi Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *ICWSM*, 10:90–97, 2010.

[45] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 177–187, New York, NY, USA, 2005. ACM.

[46] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[47] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[48] Menghui Li, Ying Fan, Jiawei Chen, Liang Gao, Zengru Di, and Jinshan Wu. Weighted networks of scientific communication: the measurement and topological role of weight. *Physica A: Statistical Mechanics and its Applications*, 350(24):643 – 656, 2005.

[49] Bang Hui Lim, Dongyuan Lu, Tao Chen, and Min-Yen Kan. # mytweet via instagram: Exploring user behaviour across multiple social networks. *arXiv preprint arXiv:1507.03510*, 2015.

[50] Jane Lin. *Automatic author profiling of online chat logs*. PhD thesis, Monterey, California. Naval Postgraduate School, 2007.

[51] Alexander V Mantzaris and Desmond J Higham. Understanding dynamic interactions. 2011.

[52] Miika J. Marttunen and Leena I. Laurinen. Secondary school students collaboration during dyadic debates face-to-face and through computer chat. *Computers in Human Behavior*, 25(4):961 – 969, 2009. Including the Special Issue: The Use of Support Devices in Electronic Learning Environments.

[53] Elijah Mayfield, David Adamson, and Carolyn Penstein Ros. Hierarchical conversation structure prediction in multi-party chat. In *In Proceedings of SIGDIAL Meeting on Discourse and Dialogue*, 2012.

[54] Sally J. McMillan and Jang-Sun Hwang. Measures of perceived interactivity: An exploration of the role of direction of communication, user control, and time in shaping perceptions of interactivity. *Journal of Advertising*, 31(3):29–42, 2002.

[55] T. Mihaljev, L. de Arcangelis, and H. J. Herrmann. Interarrival times of message propagation on directed networks. , 84(2):026112, August 2011.

[56] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM.

[57] James Moody, Daniel McFarland, and Skye Bender?deMoll. Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–1241, 2005.

[58] P. Mutton. Inferring and visualizing social networks on internet relay chat. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 35–43, July 2004.

[59] M. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351, September 2005.

[60] V. Nicosia, J. Tang, M. Musolesi, G. Russo, C. Mascolo, and V. Latora. Components in time-varying graphs. *Chaos*, 22(2):023101, June 2012.

[61] J. G. Oliveira and A.-L. Barabási. Human dynamics: Darwin and Einstein correspondence patterns. , 437:1251, October 2005.

[62] Jacki O'Neill and David Martin. Text chat in action. In *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, GROUP '03, pages 40–49, New York, NY, USA, 2003. ACM.

[63] Özcan Özyurt and Cemal Köse. Chat mining: Automatically determination of chat conversations topic in turkish text based chat mediums. *Expert Systems with Applications*, 37(12):8705–8710, 2010.

[64] Gergely Palla, Pter Pollner, Albert-Lszl Barabsi, and Tams Vicsek. Social group dynamics in networks. In Thilo Gross and Hiroki Sayama, editors, *Adaptive Networks*, volume 51 of *Understanding Complex Systems*, pages 11–38. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-01284-6-2.

[65] K. Park, J. Kim, J. Park, M. Cha, J. Nam, S. Yoon, and E. Rhim. Mining the Minds of Customers from Online Chat Logs. *ArXiv e-prints*, October 2015.

[66] Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language*, 50(4):696–735, 1974.

[67] Zon-Yin Shae, Dinesh Garg, Rajarshi Bhose, Ritabrata Mukherjee, and Sinem Guven. Efficient internet chat services for help desk agents. In *IEEE International Conference on Services Computing (SCC 2007)*, pages 589–596. IEEE, 2007.

[68] Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. Thread detection in dynamic text message streams. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 35–42, New York, NY, USA, 2006. ACM.

[69] Stefan Trausan-Matu, Traian Rebedea, Alexandru Dragan, and Catalin Alexandru. Visualisation of learners' contributions in chat conversations. *Blended learning*, pages 217–226, 2007.

[70] Myra Spiliopoulou Ulrik Brandes, Rudolf Kruse. *Community Analysis in Dynamic Social Networks*. PhD thesis, angenommen durch die Fakultat fur Informatik der Otto-von-Guericke-Universit ?t Magdeburg, 2009.

[71] David C. Uthus and David W. Aha. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199200(0):106 – 121, 2013.

[72] Le Wang, Yan Jia, and Yingwen Chen. Conversation extraction in dynamic text message stream. *Journal of Computers*, 3(10):86–93, 2008.

[73] Le Wang, Yan Jia, and Yingwen Chen. Conversation extraction in dynamic text message stream. *Journal of Computers*, 3(10), 2008.

[74] Lidan Wang and Douglas W. Oard. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 200–208, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[75] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[76] Duncan J Watts. *Six degrees: The science of a connected age*. WW Norton & Company, 2004.

[77] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.