Mining Social Interaction Data in Virtual Worlds

Syed Fahad Allam Shah[1] and Gita Sukthankar[2]

[1] Microsoft Corporation, Bellevue, WA, USA
fashah@microsoft.com
[2] University of Central Florida, Orlando, FL, USA
gitars@eecs.ucf.edu

Abstract. Virtual worlds and massively multi-player online games are rich sources of information about large-scale teams and groups, offering the tantalizing possibility of harvesting data about group formation, social networks, and network evolution. However these environments lack many of the cues that facilitate natural language processing in other conversational settings and different types of social media. Public chat data often features players who speak simultaneously, use jargon and emoticons, and only erratically adhere to conversational norms. This chapter presents techniques for inferring the existence of social links from unstructured conversational data collected from groups of participants in the Second Life virtual world.

Keywords: Network Text Analysis · Longitudinal Analysis · Virtual Worlds · Community Detection

1 Introduction

Massively multi-player online games (MMOGs) and virtual environments provide new outlets for human social interaction that are significantly different from both face-to-face interactions and non-physically-embodied social networking tools such as Facebook and Twitter. We aim to study group dynamics in these virtual worlds by collecting and analyzing public conversational patterns of Second Life users.

Second Life (SL) is a massively multi-player online environment that allows users to construct and inhabit their own 3D world. In Second Life, users control avatars, through which they are able to explore different environments and interact with other avatars in a variety of ways. One of the most commonly used methods of interaction in Second Life is basic text chat. Users are able to chat with other users directly through private instant messages (IMs) or to broadcast chat messages to all avatars within a given radius of their avatar using a public chat channel.

The physical environment in Second Life is laid out in a 2D arrangement, known as the SLGrid. The SLGrid is comprised of many regions, with each region hosted on its own server and offering a fully featured 3D environment shaped by the user population. The total number of SL users is approximately 16 million, with a weekly user login activity reported in the vicinity of 0.5 million [26]. Second Life contains users of widely divergent expertise levels, ranging from complete novices who congregate in the orientation areas practicing basic controls to highly skilled scripters who craft objects

and storefronts to sell within Second Life. There is a broad spectrum of group persistence. One can observe rapidly-formed crowds gathered around a temporary attraction, and also semi-permanent groups of people who share interests either within or outside of the virtual environment. Similar to real-life, these differences are somewhat correlated with SL regions, since each SL region contains a different mix of entertainment opportunities.

Although Second Life provides us with rich opportunities to observe the public behavior of large groups of users, it is difficult to interpret who the users are communicating to and what they are trying to say from public chat data. Network text analysis systems such as Automap [3] that incorporate linguistic analysis techniques such as stemming, named-entity recognition, and n-gram identification are not effective on this data since many of the linguistic pre-processing steps are defeated by the slang and rapid topic shifts of the Second Life users. This is a hard problem even for human observers and it was impossible for us to unambiguously identify the recipient of many of the utterances in our dataset. In this article, we present an algorithm for addressing this problem, Shallow Semantic Temporal Overlap (SSTO) [27], that combines temporal and language information to infer the existence of directional links between participants. One of the problems is that using temporal overlap as a cue for detecting links can produce extraneous links and low precision. To reduce these extraneous links, we propose the use of community detection. Optimizing network modularity reduces the number of extraneous links generated by overly generous temporal co-occurrence assumption but does not significantly improve the performance of SSTO.

There has been increasing interest in mining community structure in these networks. In general, network sections exhibiting denser linkages among themselves are classified as part of the same community. This phenomena has been studied in social networks, biochemical networks and the WWW [23, 11, 9, 13, 5]. Understanding the community structure of a network can reveal interesting trends and increase our knowledge of the function and evolution of the system.

To examine the influence of the extracted groups found with community detection on network evolution, we analyze the system using the dynamic actor-oriented model for network evolution [32]. We use this model to explore the evolution of the network (mined from the dialog exchanges) considering the community membership from previous time period as an actor attribute. This gives us statistical evidence whether the community membership persists over time and provides additional support on the accuracy of our community detection. Using longitudinal network data analysis [33], we consider sequences of network observations extracted from dialog exchanges, along with attributes of the SL avatars, and model them in an actor-oriented model using RSiena (Simulation Investigation for Empirical Network Analysis) [35]. The methodology has been successfully employed in a number of sociological studies on the influences of different factors on group behavior [18, 24, 8, 14, 16].

2 Prior Work

Second Life is a unique test bed for research studies, allowing scientists to study a broad range of human behaviors. For instance, social scientists have used Second Life

to study norms and etiquette in dressing and meeting people [7]. Several studies on user interaction in virtual environments have been conducted in SL including studies on conversation [37] and virtual agents [2]. Zhao and Wang describe a technique for simulating multiple agents in a virtual environment using a hierarchical model of cognition and decision making [39]. Second Life has also been used to recreate many real-world environments. Physical versions of libraries, art galleries, universities, and corporate meeting facilities have been developed for SL to serve as virtual portals for meetings and information access.

In this chapter, we address the problem of constructing social network linkages from public chat exchanges. This is simultaneously useful for analyzing the group dynamics in different Second Life regions and has the potential practical benefit of allowing Second Life land owners to analyze the relative utility of various attractions. Dialog analysis has been previously explored within the Restaurant Game [22], where a corpus of human dialog is collected and leveraged to improve the realism of the bot's dialog in a social situation. There has been research on the problem of constructing social networks of MMOG players, for example, Shi and Huang [30] demonstrate that concepts from social network analysis and data mining can be used identify MMOG tasks. In this article our social network analysis is focused toward revealing network characteristics rather than actor characteristics, which is significantly different from prior work at mining social networks from multi-player game data. We wish to identify differences between *groups* of participants rather than between different *actors* within the same social network. Communities within USENET have been analyzed by comparing structures of induced social networks for each group using metrics such as size, degree, and reciprocity [17]. Our analysis of Second Life communities is similar in concept but uses different techniques for constructing linkages. Kahanda and Neville [15] have compared the relative utility of different types of features at predicting friendship links in social networks; in this study we examine direct conversational data and do not attempt to predict unobserved links based on other types of events.

The problem of analyzing environmental effects on groups of users has been explored in other types of social media. Hogg and Lerman [12] describe a general stochastic process-based approach to modeling user-contributory web sites in an attempt to analyze how the design of the website affects user behavior. Fisher et al. [4] perform community analysis from the organizer's point of view and address problems such as assessing the value of online community and monitoring social activity within space. Our work can be seen as a similar effort in virtual worlds where owners are interested in attracting users, performing usage analysis and learning user activity models.

Although Second Life provides us with rich opportunities to observe the public behavior of large groups of users, it is difficult for even humans to identify who a user is communicating with at a given moment; for instance, it was impossible for us to unambiguously identify the target for many of the utterances in our dataset even with human labelers. Much of the previous work on analyzing chat has been restricted to a small number of users and is topic-specific. One notable exception is the work by the Naval Postgraduate School on collecting and analyzing the NPS corpus [6], which is based on chat dialogs from online chat rooms. Using a combination of manual annotations and filtering techniques, Wu et al. [38] divide the utterances into semantic classes; in con-

trast we use semantic cues to identify social connections. Compared to our SL dataset, the NPS corpus contains more discussion about the participants' real-life interactions whereas the SL data is more heavily slanted toward discussions of the virtual world (e.g., scripting, shopping). The NPS corpus has additionally been used in a study of topic identification [1], which is something that we hope to do in future work.

Another similar effort done within a controlled environment on a smaller corpus is that of [29]. They employed manual annotation at four levels: communication links, dialog acts, local topics and meso-topics, whereas in our case we are concerned with the automation of the first level (communication links). Another important difference is that they imposed structure to their communications by directing the conversation towards a topic and using an arbiter. Our dataset is unique in its size, lack of communication structure, and dynamic groups.

3 Approach

To conduct our study of group social interactions in Second Life, we had to address the following issues:

1. partitioning unstructured dialog into separate conversations;
2. identifying links from the partitioned data;
3. performing longitudinal network data analysis to validate communities.

Figure 1 shows the overall data collection architecture. Multiple bots, stationed in different SL regions, listen to all the messages within their hearing range on the public channel. The bots forward chat messages to the server, which parses and conditions messages for storage in the dialog database. Occasionally the server sends the bots navigational commands and optional dialog response if the communication was directed to the bot. Linkages between SL actors are inferred offline by partitioning the unstructured data into separate conversations; these linkages are used to construct the graphs used in the social network analysis.

3.1 Bot Construction

Instead of being controlled by a human user, Second Life avatars can be controlled by an automated agent known as a bot. A bot connects to the SLGrid like a normal user, but is controlled by a program that does not require user interaction. Our bots were implemented using LibOpenMetaverse (LibOMV) [21], an open source .NET based library that allows applications to be able to simulate much of the functionality of the official Second Life client software. Using this library, we were able to build multiple bots that log in at a given location and collect all desired data for chat messages within the bot's hearing range on the public channel.

The bot application begins execution by passing login info for a Second Life account to LibOMV. Once LibOMV successfully logs into Second Life, the application enters its main execution loop. Here, the application waits for notification from LibOMV that an event has occurred involving the bot. When a chat message is received, LibOMV passes the following information to the application: the name of the user who sent the

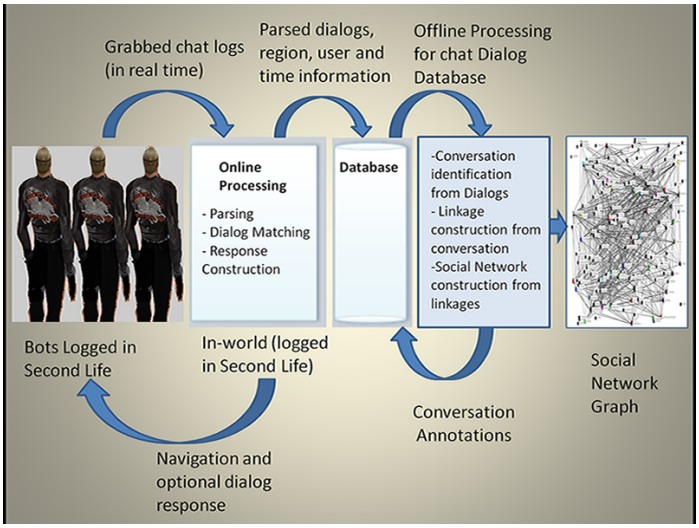


Fig. 1. Multi-agent architecture for Second Life data collection

message, the time and date the message was received, the region and local coordinates (relative to the bot's current region) that the message was sent from, and the text of the message itself. After this information is recorded in the database, the application returns to the main loop, waiting for the next event to occur. Figure 2 shows a bot harvesting data in Second Life.

Fig. 2. Bot collecting public chat messages

3.2 Data Collection

We obtained conversation data from eight different regions in Second Life over fourteen days of data collection; the reader is referred to [28] for details. To study user dialogs, we examined daily and hourly data for five randomly selected days in the eight regions. In total, the dataset contains 523 hours of information over the five days (80,000 utterances) considered for the analysis across all regions. We did a hand-annotation of one hour of data from each of the regions to serve as a basis for comparison. Table 1 gives an example of dialog exchanged between users in the RezzMe region.

Table 1. Anonymized transcript of a public conversation collected in Second Life's RezzMe region.

User Name	Dialog
user1	anyone know if there's a way to turn off notifications in local chat for shields or any other objects when you're in a no-rez zone?
user2	brb need to get drink :)
user3	lol I put the pengiun in the trash can
user4	not too many who knows what it actually stands for
user5	user1, pls can you explain in more detail what you ask? mute it?
user3	GRR YOU DARN PENGUIN
user4	/status
user1	i can paste it in for you:
user5	user3 pls dont pushy ppl
user3	ok sorry
user1	Can't rez object 'animcept4' at {55.9452, 35.1487, 23.4774 } on parcel 'Help People Island'! in region Help People Island because the owner of this land does not allow it Use the land tool to see land restrictions

Second Life's multi-user, open-ended setting poses unique challenges to dialog analysis. In such situations it is imperative to identify conversational connections before proceeding to higher level analysis like topic modeling, which is itself a challenging problem. We considered several approaches to analyzing our dialog dataset, ranging from statistical NLP approaches using classifiers to corpus-based approaches using tagger/parsers; however we discovered that there is no corpus available for group-based online chat in an open-ended dialog setting. It is challenging to label the conversations themselves for the large size of the dataset and the ambiguity in a multi-user open-ended setting makes it difficult even for a human to figure who is talking to whom. Furthermore, the variability of the utterances and the nuances such as emoticons, abbreviations and the presence of emphasizers in spellings (e.g., "Yayyy") makes it difficult to train appropriate classifiers. As for the parser/tagger-based approaches, since there is no corpus available and the vocabulary is not restricted to English words, the parser/tagger performs poorly.

Consequently, we decided to investigate approaches that utilize non linguistic cues such as temporal co-occurrence. Although temporal co-occurrence can create a large

number of false links, many aspects of the network group structure are preserved. Hence we opted to implement two-pass approach: 1) create a noisy network based solely on temporal co-occurrence 2) perform modularity detection on the network to detect communities of users 3) attempt to filter extraneous links using the results of the community detection.

3.3 Modularity Optimization

In prior work, community membership has been successfully used to identify latent dimensions in social networks [36] using techniques such as eigenvector-based modularity optimization [20] which allows for both complete and partial memberships. As described in [20], modularity (denoted by Q below) measures the chances of seeing a node in the network versus its occurrence being completely random; it can be defined as the sum of the random chance $A_{ij} - \frac{k_i k_j}{2m}$ (where A_{ij} is the entry from adjacency matrix, k_i the node's degree, and $m = \frac{1}{2}\sum_i k_i$ the total edges in the network) summed over all pairs of vertices i, j, where s equals 1 if a vertex falls in community 1 and -1 if it falls in community 2:

$$Q = \frac{1}{4m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]s_i s_j \tag{1}$$

If B is defined as the modularity matrix given by $A_{ij} - \frac{k_i k_j}{2m}$, which is a real symmetric matrix and s column vectors whose elements are s_i then Equation 1 can be written as $Q = \frac{1}{4m}\sum_{i=1}^{n}(u_i^T s)^2\beta_i$, where β_i is the eigenvalue of B corresponding to the eigenvector u (u_i are the normalized eigenvectors of B so that $s = \sum_i^n a_i u_i$ and $a_i = u_i^T s$). We use the leading eigenvector approach to spectral optimization of modularity as described in [19] for the strict community partitioning (s being 1 or -1 and not continuous). For the maximum positive eigenvalue we set $s = 1$ for the corresponding element of the eigenvector if it is positive and negative otherwise. Finally we repeatedly partition a group of size n_g in two and calculate the change in modularity measure given by $\Delta q = \frac{1}{4m}\sum_{l=1}^{c}\sum_{i,j\epsilon g}[B_{ij} - \delta_{ij}\sum_{k\epsilon g}B_{ik}]s_{il}s_{jl}$, where l is the number of communities from 1 to c and δ_{ij} is the Kronecker δ symbol, terminating if the change is not positive and otherwise choosing the sign of s (the partition) in the same way as described earlier.

3.4 Shallow Semantics and Temporal Overlap Algorithm (SSTO)

Because of an inability to use statistical machine learning approaches due to the lack of sufficiently labeled data and absence of a tagger/parser that can interpret chat dialog data, we developed a rule-based algorithm that relies on shallow semantic analysis of linguistic cues that commonly occur in chat data including mentions of named entities as well as the temporal co-occurrence of utterances to generate a to/from labeling for the chat dialogs with directed links between users. Our algorithm employs the following types of rules:

salutations: Salutations are frequent and can be identified using keywords such as "hi", "hello", "hey". The initial speaker is marked as the *from* user and users that respond within a designated temporal window are labeled as *to* users.

questions: Question words (e.g., "who", "what", "how") are treated in the same way as salutations. We apply the same logic to requests for help (which are often marked by words such as "can", "would").

usernames: When a dialog begins or ends with all or part of a username (observed during the analysis period), the username is marked as *to*, and the speaker marked as *from*.

second person pronouns: If the dialog begins with a second person pronoun (i.e., "you", "your"), then the previous speaker is considered as the *from* user and the current speaker the *to* user; explicit mentions of a username override this.

temporal co-occurrences: Our system includes rules for linking users based on temporal co-occurrence of utterances. These rules are triggered by a running conversation of 8–12 utterances.

This straightforward algorithm is able to capture sufficient information from the dialogs and is comparable in performance to SSTO with community information, as discussed below.

3.5 Temporal Overlap Algorithm

The temporal overlap algorithm consists of using the temporal co-occurrence to construct the links. It exploits the default timeout in Second Life (20 minutes) and performs a lookup for 20 minutes beginning from the occurrence of a given username and constructs an undirected link between the speakers and this user. This process is repeated for all users within that time window (one hour or day) in 20 minute periods. This algorithm gives a candidate pool of initial links between the users without considering any semantic information. Later, we show that incorporating community information from any source (similar time overlap or SSTO based) and on any scale (daily or hourly) enables us to effectively prune links, showing the efficacy of mining community membership information.

3.6 Incorporating Community Membership

The dataset was separated from daily logs into hourly partitions, based on the belief that an hour is a reasonable duration for social interactions in a virtual world. The hourly partitioned data for each day is used to generate user graph adjacency matrices using the two algorithms described earlier (Sections 3.4 and 3.5). The adjacency matrix is then used to generate the spectral partitions for the communities in the graph, which are then used to back annotate the tables containing the to/from labeling (in the case of the SSTO algorithm). These annotations serve as an additional cue capturing community membership. Not all the matrices are decomposable into smaller communities so we treat such graphs of users as a single community.

There are multiple options for using the community information: it can be calculated on an hourly or daily basis, using the initial run from either SSTO or the temporal

overlap algorithms. The daily data is a long-term view that focuses on the stable network of users while the hourly labeling is a fine-grained view that can enable the study of how the social communities evolve over time. The SSTO algorithm gives us a conservative set of directed links between users while the temporal overlap algorithm provides a more inclusive hypothesis of users connected by undirected links.

For the SSTO algorithm, we consider several variants of using the community information:

SSTO: Raw SSTO without community information;
SSTO+LC: SSTO (with loose community information) relies on community information from the previous run only when we fail to make a link using language cues;
SSTO+SC: SSTO (with strict community information) always uses language cues in conjunction with the community information.

For the temporal overlap algorithms, we use the community information from the previous run.

TO: Raw temporal overlap algorithm without community information;
TO+DT: Temporal overlap plus daily community information;
TO+HT: Temporal overlap plus hourly community information.

4 Results

In this section we summarize the results from a comparison of the social networks constructed from the different algorithms. While comparing networks for similarity is a difficult problem [25], we restrict our attention to comparing networks as a whole in terms of the link difference (using Frobenius norm) and a one-to-one comparison for the *to* and *from* labelings for each dialog on the ground-truthed subset (using precision and recall).

4.1 Network Comparison Using the Frobenius Norm

We constructed a gold-standard subset of the data by hand-annotating the to/from fields for a randomly-selected hour from each of the Second Life regions. It is to be noted that there were instances where even a human was unable to determine the person addressed due to the complex overlapping nature of the dialogs in group conversation in an open ended setting (Table 3).

To compare the generated networks against this baseline, we use two approaches. First we compute a Frobenius norm [10] for the adjacency matrices from the corresponding networks. The Frobenius norm is the matrix norm of an $M \times N$ matrix A and is defined as:

$$\|A\| = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|a_{ij}|^2}. \tag{2}$$

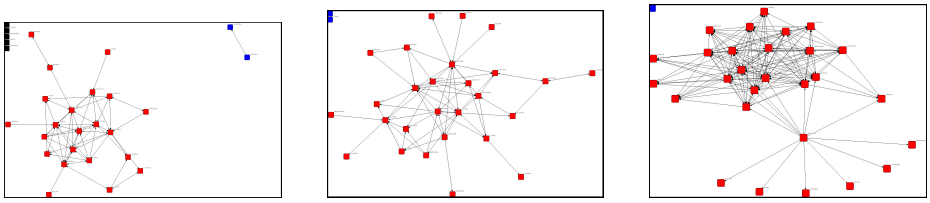
The Frobenius norm directly measures whether the two networks have the same links and can be used since the networks consists of the same nodes (users). Thus, the norm serves as a measure of error (a perfect match would result in a norm of 0). Table 4.1 shows the results from this analysis.

Table 2. Frobenius norm: comparison against hand-annotated subset

	SSTO	SSTO+LC	SSTO+SC	TO	TO+DT	TO+HT
Help Island Public	35.60	41.19	46.22	224.87	162.00	130.08
Help People Island	62.23	60.50	66.34	20.29	20.29	54.88
Mauve	48.45	45.11	51.91	58.44	58.44	49.89
Morris	24.67	18.92	20.76	43.12	37.54	38.98
Kuula	32.12	30.75	32.66	83.22	73.15	77.82
Pondi Beach	20.63	21.77	21.56	75.07	62.62	71.02
Moose Beach	17.08	18.30	21.07	67.05	53.64	50.97
Rezz Me	36.70	39.74	45.78	38.72	39.01	41.10
Total error	277.48	**276.28**	306.30	610.78	507.21	514.74

4.2 Direct Label Comparisons

The second quantitative measure we present is the head-to-head comparison of the to/from labelings for the dialogs using any of the approaches described above (for SSTO) against the hand annotated dialogs. This gives us the true positives and false positives for the approaches and allows us to see which one is performing better on the dataset, and if there is an effect in different Second Life regions. Table 3 shows the results from this analysis.



(a) Hand labeled network. (b) SSTO labeled network. (c) TO labeled network.

Fig. 3. Networks from different algorithms for one hour in the Help Island Public region

For the temporal overlap algorithm (TO), the addition of the community information reduces the link noise, irrespective of the scale — be it hourly or daily. This is shown by the decreasing value of the Frobenius norm in all the cases as compared to the value obtained using temporal overlap algorithm alone. In general shallow semantic approach (SSTO) performs the best and is only improved slightly by the loose incorporation of community information. For the SSTO algorithm, the daily or hourly community partition also does not affect the improvement. Table 3 shows how the dialog labeling generated from various algorithms agrees with the ground truth notations produced by a human labeler. Since TO only produces undirected links, we do not include it in the comparison. Plain SSTO generally results in a better precision and recall than SSTO plus either strict or loose community labeling. These results are also confirmed from

Table 3. Precision/Recall values for one-to-one labeling comparison

		Help Island Public	Help People Island	Mauve	Morris	Kuula	Pondi Beach	Moose Beach	Rezz Me
Total Dialogs		360	184	128	179	227	144	128	97
Hand	recall	0.6278	0.9076	0.9453	0.6983	0.8370	0.6944	0.6797	0.8866
Labeled	total	226	167	121	125	190	100	87	86
SSTO+SC	match	61	59	49	43	63	27	12	23
	precision	0.2607	0.6629	0.6364	0.4216	0.4632	0.3971	0.2105	0.4600
	recall	0.2699	0.3533	0.4050	0.3440	0.3316	0.2700	0.1379	0.2674
	F-Score	0.2652	0.4609	0.4204	0.3789	0.3865	0.3214	0.1667	0.3382
	total	234	89	77	102	136	68	57	50
SSTO+LC	match	61	51	37	39	52	26	12	15
	precision	0.3005	0.6456	0.6607	0.4643	0.4561	0.4194	0.2667	0.4688
	recall	0.2699	0.3054	0.3058	0.3120	0.2737	0.2600	0.1379	0.1744
	F-Score	0.2844	0.4146	0.4181	0.3732	0.3421	0.3210	0.1818	0.2542
	total	203	79	56	84	114	62	45	32
SSTO	match	76	68	51	45	66	30	20	27
	precision	0.3065	0.7083	0.6145	0.4500	0.4748	0.4225	0.3077	0.4576
	recall	0.3363	0.4072	0.4215	0.3600	0.3474	0.3000	0.2299	0.3140
	F-Score	0.3207	0.5171	0.5000	0.3617	0.4012	0.3509	0.2299	0.3724
	total	248	96	83	100	139	71	65	59

the visualizations for one of the hours of data for all the three methods in figure 3, where the SSTO network most closely resembles the hand-labeled network while the TO network contains many spurious links.

The challenging nature of this dataset is evident in the overall low precision and recall scores, not only for the proposed algorithms but also for human labelers. We attribute this largely to the inherent ambiguity in the observed utterances. Among the techniques, SSTO performs best, confirming that leveraging semantics is more useful than merely observing temporal co occurrence. We observe that community information is not reliably informative for SSTO but does help TO, showing that link pruning through network structure is useful in the absence of semantic information.

5 Evaluating Community Persistence

To evaluate the usefulness of the community detection and determine if the patterns determined by the algorithm prevail over time, we devised the following experiment utilizing the longitudinal (cross-sectional) analysis of the network in relation to the attribute information:

1. We use social networks formed from three days of data and determine the community membership for each of the actors in this set.
2. Next, we randomly select four hours worth of data from a subsequent day to be used for longitudinal analysis.
3. We use the community membership information as a constant actor-covariate. The objective here was to explore if the actors with same community membership communicate more frequently among themselves across multiple days, hence testifying to the stability of the communities and our SSTO link-mining algorithm.

There were total 98 actors across the selected four hours period; 47 actors are common across all four days of data. We use the stochastic actor-oriented model from Snijders [34, 33] to explore the co-evolution of the network behavior including the parameters for Similarity (to evaluate the hypothesis of preferential communication between actors of the same community), Ego (covariate-related activity), and Alter (covariate-related popularity).

Here we summarize the network evolution model used in RSiena (Simulation Investigation for Empirical Network Analysis) [35]. The network evolution model examines the actors' decisions to establish new ties or break existing ties (as defined by evaluation and endowment functions), and the model of the timing of these decisions (controlled by rate function). The objective function of the actor is then defined by the sum of the network evaluation function and the network endowment function as shown in Equation 3:

$$u^{\text{net}}(x) = f^{\text{net}}(x) + g^{\text{net}}(x). \tag{3}$$

The network evaluation function for actor i can be written as:

$$f^{\text{net}}(x) = \sum_k \beta_k^{\text{net}} s_{ik}^{\text{net}}(x) \tag{4}$$

where β_k^{net} denotes the parameters and $s_{ik}^{\text{net}}(x)$ the effects (discussed below).

The structural part of the network dynamics is modeled by the structural effects that depend only on the network. We considered the following two structural effects in our model:

– **out-degree** or **density effect** as given by

$$s_{i1}^{\text{net}}(x) = x_{i+} = \sum_j x_{ij} i \tag{5}$$

where the presence of a tie from i to j is indicated by $x_{ij} = 1$ and $x_{ij} = 0$ denotes the absence and

– **reciprocity effect**, defined as the number of reciprocated ties

$$s_{i2}^{\text{net}}(x) = \sum_j x_{ij} x_{ji}. \tag{6}$$

Covariates are the variables that depend on the actors (also called actor covariates). For actor-dependent covariates v_j the following effects were used for the analysis:

– **covariate-alter** or **covariate-related popularity** is the sum of the covariate over all actors with which actor i has a tie and is given by:

$$s_{i3}^{\text{net}}(x) = \sum_j x_{ij} v_j. \tag{7}$$

– **covariate-ego** or **covariate-related activity** is the actor i's out-degree weighted by his covariate value as given by:

$$s_{i4}^{\text{net}}(x) = v_i x_{i+}. \tag{8}$$

– **covariate-related similarity** is the sum of centered similarity scores sim_{ij}^{v} between the actor i and the other actors j that are tied to i as given by:

$$s_{i5}^{\text{net}}(x) = \sum_j x_{ij} (\text{sim}_{ij}^{v} - \hat{\text{sim}}^{v}) \tag{9}$$

where $\hat{\text{sim}}^{v}$ is the mean of all similarity scores given by $\text{sim}_{ij}^{v} = \frac{\Delta - |v_i - v_j|}{\Delta}$ and $\Delta = \max_{ij} |v_i - v_j|$ is the observed range of the covariate *v*.

The network rate function λ^{net} is given by:

$$\lambda_i^{\text{net}}(\rho, \alpha, x, m) = \lambda_{i1}^{\text{net}} \lambda_{i2}^{\text{net}} \lambda_{i3}^{\text{net}} \tag{10}$$

where the factors in Equation 10 depend respectively on period m, actor covariates, and actor position.

The dependence on the period can be denoted by a simple factor given in:

$$\lambda_{i1}^{\text{net}} = \rho_m^{\text{net}} \tag{11}$$

for $m = 1, ..., M-1$. If we have $M = 2$ observations, the basic rate parameter can be written as ρ^{net}. The effect of actor covariates with values v_{hi} can be denoted by a factor as shown:

$$\lambda_{i2}^{\text{net}} = \exp\left(\sum_h \alpha_h v_{hi}\right). \tag{12}$$

The actor's dependence on the position can be modeled as a function of the actor's out-degree, in-degree, number of reciprocated relations, and reciprocated degrees, given by:

$$x_{i+} = \sum_j x_{ij}, x_{+i} = \sum_j x_{ij}, x_{i(r)} = \sum_j x_{ij}x_{ji} \tag{13}$$

where $x_{ii} = 0$ for all i. The out-degree's contribution to $\lambda_{i3}^{\text{net}}$ is a factor $\exp(\alpha_h x_{i+})$ if the associated parameter is given by α_h for some h, and similarly for the in-degree and the reciprocated degree contributions.

5.1 Actor-Oriented Model

The main component of the actor-oriented model is the evaluation function [33, 34], given in Equation 4. The objective function can give an idea of the "attractiveness" of the network for a given actor. Interpretation of the values for the estimates can be helped by the objective function computations that give an idea of how attractive different tie changes are.

A variable V's effects can best be understood by considering all effects in the model on which it appears simultaneously. In our network dynamics model, the ego, alter, and similarity effects of a variable V were considered and the formula for their contribution can be obtained from the components listed in Equation 4 as

$$\beta_{\text{ego}} v_i x_{i+} + \beta_{\text{alter}} \sum_j x_{ij} v_j + \beta_{\text{sim}} \sum_j \left(\text{sim}_{ij}^{v} - \hat{\text{sim}}^{v}\right) \tag{14}$$

where the similarity score is given by $\text{sim}_{ij}^{v} = 1 - \frac{|v_i - v_j|}{\Delta_V}$ with $\Delta_V = \max_{ij} |v_i - v_j|$ denoting the observed range of the covariate v and $\hat{sim}^{v}$ being the mean of all similarity scores. Note, for simplicity, the superscript *net* is removed from the notation for the parameters.

The single tie variable x_{ij} gives the contribution of the tie from i to j; hence, the difference between the values of Equation 14 for $x_{ij} = 1$ and $x_{ij} = 0$ can be computed from this equation. Since we are using RSiena which centers the values around the mean, Equation 14 can be rewritten as

$$\beta_{\text{ego}} v_i x_{i+} + \beta_{\text{alter}} \sum_j x_{ij} v_j + \beta_{\text{sim}} \sum_j \left(1 - \frac{|v_i - v_j|}{\Delta_V} - \hat{\text{sim}}^{v}\right) \tag{15}$$

This section details the statistics obtained from running the estimation on the Ego, Alter, and Similarity parameters considered for the three covariates (age, gender, and

community).[3] First we present summary statistics for the network as shown in the Table 4. The average density for all the periods is quite low, indicating the sparse nature of the data. The average degree shows that only observation time 1 has an average close to 0.5 while the rest are low indicating the asymmetric nature of the ties. Lastly the number of ties are listed for each, where the higher number of ties in observation time 1 explains its higher density, whereas the missing fraction for all observation times being zero.

Table 4. Network density indicators

Observation Time	1	2	3	4
density	0.006	0.003	0.003	0.003
average degree	0.538	0.286	0.286	0.308
number of ties	49	26	26	28
missing fraction	0.000	0.000	0.000	0.000

Table 5 shows the changes between the observations for each period. There are no changes between periods 1-2 and 2-3 in contrast to the high number of changes from 3-4 (indicated by a higher value of the distance). This indicates that the ties that were observed in observation 1 persist in observation 2, observation 2 ties persist to observation 3, but not so for the observation 4. This might be due to the high influx of users during period 4.

Table 5. Changes between observations

Periods	0 to 0	0 to 1	1 to 0	1 to 1	Distance	Jaccard	Missing
1 to 2	8115	26	49	0	0	0.000	0 (0%)
2 to 3	8138	26	26	0	0	0.000	0 (0%)
3 to 4	8139	25	23	3	27	0.059	0 (0%)

5.2 Estimation Procedure

We used the Method of Moments (MoM) [31, 34], where the parameters are estimated in such a way that expected values of a vector of selected statistics are equal to their observed values for the network. The SIENA software implements two methods for

[3] In this study, age and gender refer to the avatar's listed age and gender, rather than the player demographics, which are not publicly available.

MoM estimation: conditional and unconditional. The difference between the two is in the stopping criteria for the simulations of the network evolution.

For unconditional estimation, the network evolution simulations for each time period continue until a predetermined time (taken to be 1.0 for each consecutive time period) has passed. In conditional estimation, the simulations for each period continue to run until a stopping criterion (calculated from the observed data) is reached. It is possible to do conditioning for each of the dependent variables. The conditioning on the network variable refers to running the simulations until the difference in entries for the initially observed network of this period and the simulated network equals the number of entries in the adjacency matrix for the difference between the initial and the final networks of this period. We used the conditional MoM for the community and age covariates and unconditional MoM for gender covariate.

5.3 Convergence Check

A convergence check can be computed from the deviations between the simulated values of the statistics and the observed values. Ideally these deviations should be as close to zero as possible for good convergence. Siena provides t-statistics computed from these averages and standard deviations. The recommendation for the t-statistics for the longitudinal analysis [35] is that the convergence is excellent when these values are less than 0.1 (absolute value), good when less than 0.2, and moderate when less than 0.3. In our case the t-ratios for all estimated parameters in the model were less than 0.1 in the absolute indicating good convergence.

5.4 Interpretation of Parameter Values

The rate parameter (ρ) for the three periods is shown in the Table 6. A value of near zero for the first two periods (1 and 2) indicates that there is very little change between these two periods, while a value of 3.25 indicates the estimated number of changes per actor between the two observations comes out to be approximately 3 ties. It is to be noted that this refers to unobserved changes, and that some of these changes may cancel, such that the average observed number of differences per actor can be actually smaller than the estimated number of unobserved changes.

We also included outdegree (density), however as the [35] points out, no definite conclusion can be made on the basis of this value alone as all the parameters depend on this parameter. It has a near constant value of -1.9304 across all our estimates.

We explored three constant actor covariates in our model: 1) community membership 2) Second Life avatar gender 3) Second Life avatar age (number of days the avatar has existed). The values for the Ego, Alter and Similarity for the three actor covariates are presented in Table 7. A positive value of similarity indicates that for the covariate the actors are more likely to make connection to other actors of the same value of the covariate as them, whereas a negative value indicates otherwise. The following can be concluded from the values for similarity in Table 7.

1. A high positive value of similarity for community means that more actors are likely to connect to other actors that have same value of community membership. This supports our hypothesis about the value of our community detection procedure.

Table 6. Rate parameter estimates

Rate Parameter	Estimate	Standard Error
Period 1	0.0247	0.0242
Period 2	0.0476	0.0508
Period 3	3.2528	0.8116

2. A slight positive value of similarity for the gender means that actors are more likely to talk to other people that are of the same SL avatar gender.
3. A negative value of similarity for SL age means that actors are more likely to communicate to other actors that are different from their own age group.

Table 7. Similarity estimates for the constant covariates

Parameter	Community	Gender	Age
Ego	-0.3770 (0.3300)	-0.0350 (0.5568)	0.0379 (0.6681)
Alter	-1.4736 (0.5547)	-0.0350 (0.5568)	-0.7767 (0.5726)
Similarity	3.8121 (3.4156)	0.4057 (0.5131)	-1.1280 (3.3675)

5.5 Model Estimates for the Community Covariate

In Section 5.4 we discussed the values for the similarity, ego and alter covariates given in the Table 7 for the three covariates (age, gender and community) and their effect on the tendency of the actors to form links. The community covariate ranges from 0-13 (values 10 and 11 were not used as they represent structural zeros and ones respectively within RSiena), with average value $\bar{v} = 1.857$ and average dyadic similarity $\hat{sim}^v = 0.8037$. Substituting these values into Equation 15 yields Equation 16 and Table 8 gives the values from the equation for each value of v_i, v_j for the covariate.

$$-0.38(v_i - \bar{v}) - 0.12(v_j - \bar{v}) + 3.81\left(1 - \frac{|v_i - v_j|}{\Delta_V} - 0.8037\right) \tag{16}$$

Table 8 shows that the highest values for each row are along the first column. The first column encodes the actors that are from the community that were not present in the three days data that were considered for the community labeling. A high value of the similarity warrants a preference for the actors that have the same community membership while a negative alter value favors the actors that have a lower value; similarly the lower membership actors are favored by the negative value of the ego (Table 7). The end result is that for all the row values the actors end up favoring ties with the actor

Table 8. Contribution from ego, alter and similarity for the community covariate

v_i/v_j	0	1	2	3	4	5	6	7	8	9	12	13
0	1.68	1.56	1.44	1.32	1.2	1.08	0.96	0.84	0.72	0.6	0.24	0.12
1	1.3	1.18	1.06	0.94	0.82	0.7	0.58	0.46	0.34	0.22	-0.14	-0.26
2	0.92	0.8	0.68	0.56	0.44	0.32	0.2	0.08	-0.04	-0.16	-0.52	-0.64
3	0.54	0.42	0.3	0.18	0.06	-0.06	-0.18	-0.3	-0.42	-0.54	-0.9	-1.02
4	0.16	0.04	-0.08	-0.2	-0.32	-0.44	-0.56	-0.68	-0.8	-0.92	-1.28	-1.4
5	-0.22	-0.34	-0.46	-0.58	-0.7	-0.82	-0.94	-1.06	-1.18	-1.3	-1.66	-1.78
6	-0.6	-0.72	-0.84	-0.96	-1.08	-1.2	-1.32	-1.44	-1.56	-1.68	-2.04	-2.16
7	-0.98	-1.1	-1.22	-1.34	-1.46	-1.58	-1.7	-1.82	-1.94	-2.06	-2.42	-2.54
8	-1.36	-1.48	-1.6	-1.72	-1.84	-1.96	-2.08	-2.2	-2.32	-2.44	-2.8	-2.92
9	-1.74	-1.86	-1.98	-2.1	-2.22	-2.34	-2.46	-2.58	-2.7	-2.82	-3.18	-3.3
12	-2.88	-3.0	-3.12	-3.24	-3.36	-3.48	-3.6	-3.72	-3.84	-3.96	-4.32	-4.44
13	-3.26	-3.38	-3.5	-3.62	-3.74	-3.86	-3.98	-4.1	-4.22	-4.34	-4.7	-4.82

with lowest value of the community membership. This agrees with the intuition as most changes in the network are likely to happen from a actor initiating communication with this new user group.

6 Conclusion and Future Work

In this article, we introduce a general framework for mining social structure from public chat data in virtual worlds and present a comprehensive analysis demonstrating the utility of our techniques for predicting social links and identifying stable communities. The principal contributions of our work are:

1. the creation of an agent architecture suitable for mining social interactions in a variety of massively multi-player online games with minimal modification;
2. introducing two new algorithms for robust conversational partitioning and social network extraction on unstructured dialog data;
3. demonstrating the effectiveness of the conversational partitioning and to/from labeling of our proposed SSTO algorithm;
4. demonstrating the persistence of dialog interaction patterns and communities over time (as mined using our SSTO algorithm) using longitudinal analysis.

Although most earlier studies on group dynamics [30] have been conducted on individuals connected by long-standing social interactions, humans can form groups that exhibit group behavior patterns and biases within a few seconds of minimal interaction, even without face-to-face contact or prior history; Second Life is an interesting research testbed since it contains a large number of groups of this nature. In future work, we plan to do a detailed comparison of the social networks mined from Second Life with those constructed from other sources of data such as blogs, social networking sites, and RSS feeds to better understand the differences between such social networks and those emerging in the virtual world of Second Life.

Acknowledgments Support for this research was provided by AFOSR YIP award FA9550-09-1-0525 and NSF IIS-08451.

References

1. Adams, P.H., Martell, C.H.: Topic detection and extraction in chat. In: Proceedings of the 2008 IEEE International Conference on Semantic Computing. pp. 581–588. IEEE Computer Society (2008)
2. Bogdanovych, A., Simoff, S., Esteva, M.: Virtual institutions: Normative environments facilitating imitation learning in virtual agents. In: International Working Conference on Intelligent Virtual Agents (2008)
3. Carley, K., Columbus, D., DeReno, M., Bigrigg, M., Diesner, J., Kunkel., F.: AutoMap user's guide. Tech. Rep. CMU-ISR-09-114, Carnegie Mellon University, School of Computer Science, Institute for Software Research (2009)
4. Fisher, D., Turner, T.C., Smith, M.A.: Space planning for online community. In: Proceedings of the Second International Conference on Weblogs and Social Media (2008)
5. Flake, G., Lawrence, S., Giles, C., Coetzee, F.: Self-organization and identification of web communities. Computer 35(3), 66–70 (March 2002)
6. Forsyth, E., Martell, C.: Lexical and discourse analysis of online chat dialog. In: International Conference on Semantic Computing. pp. 19–26 (2007)
7. Friedman, D., Steed, A., Slater, M.: Spatial social behavior in Second Life. In: International Working Conference on Intelligent Virtual Agents (2007)
8. Gerhard G. Van De Bunt, M.A.V.D., Snijders, T.A.: Friendship networks through time: An actor-oriented dynamic statistical network model. Computational and Mathematical Organization Theory 5(2) (1999)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821–7826 (June 2002)
10. Golub, G.H., Loan, C.F.V.: Matrix Computations. JHU Press, 3rd edn. (1996)
11. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature 433(7028), 895–900 (February 2005)
12. Hogg, T., Lerman, K.: Stochastic models of user-contributory web sites. In: Proceedings of the Third International Conference on Weblogs and Social Media (2009)
13. Holme, P., Huss, M., Jeong, H.: Subnetwork hierarchies of biochemical pathways. Bioinformatics 19(4), 532–538 (March 2003)
14. Huisman, M., Snijders, T.A.B.: Statistical analysis of longitudinal network data with changing composition. Sociological Methods and Research 32, 253–287 (2003)
15. Kahanda, I., Neville, J.: Using transactional information to predict link strength in online social networks. In: Proceedings of the Third International Conference on Weblogs and Social Media (2009)
16. Lubbers, M.J., Molina, J.L., Lerner, J., Brandes, U., vila, J., McCarty, C.: Longitudinal analysis of personal networks. the case of Argentinean migrants in Spain. Social Networks 32(1), 91 – 104 (2010)
17. McGlohon, M., Hurst, M.: Community structure and information flow in Usenet: Improving analysis with a thread ownership model. In: Proceedings of the Third International Conference on Weblogs and Social Media (2009)
18. Merckena, L., Snijders, T., Steglichd, E., Vartiainene, E., de Vriesa, H.: Dynamics of adolescent friendship networks and smoking behavior. Social Networks 32, 72–81 (2010)

19. Newman, M.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 036104 (2006)
20. Newman, M.: Modularity and community structure in networks. In: Proceedings of the National Academy of Sciences. vol. 103, pp. 8577–8582 (2006)
21. openmetaverse.org: LibOpenMetaverse (2009), retrieved July 2009 http://openmetaverse.org/projects/libopenmetaverse
22. Orkin, J., Roy, D.: The Restaurant Game: Learning social behavior and language from thousands of players online. Journal of Game Development 3(1), 39–60 (2007)
23. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818 (June 2005)
24. Pearson, M., Steglich, C., Snijders, T.: Homophily and assimilation among sport-active adolescent substance users. Connections 27, 51–67 (2006)
25. Prulj, N.: Biological network comparison using graphlet degree distribution. Bioinformatics 23(2) (2007)
26. Second Life: Second Life Economic Statistics (2009), retrieved July 2009 http://secondlife.com/whatis/economy_stats.php
27. Shah, F., Sukthankar, G.: Constructing social networks from unstructured group dialog in virtual worlds. In: Proceedings of the International Conference on Social Computing and Behavioral-Cultural Modeling. pp. 180–187. College Park, MD (Mar 2011)
28. Shah, F., Usher, C., Sukthankar, G.: Modeling group dynamics in virtual worlds. In: Proceedings of the Fourth International Conference on Weblogs and Social Media (2010)
29. Shaikh, S., Strzalkowski, T., Broadwell, A., Stromer-Galley, J., Taylor, S., Webb, N.: Mpc: A multi-party chat corpus for modeling social phenomena in discourse. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA) (2010)
30. Shi, L., Huang, W.: Apply social network analysis and data mining to dynamic task synthesis to persistent MMORPG virtual world. In: Proceedings of Intelligent Virtual Agents (2004)
31. Snijders, T., van de Bunt, G., Steglich, C.E.G.: Introduction to actor-based models for network dynamics. Social Networks 32, 44–60 (2010)
32. Snijders, T., Steglich, C.E.G., Schweinberger, M.: Longitudinal models in the behavioral and related sciences. Cambridge University Press (2007)
33. Snijders, T.A.B.: Models and methods in social network analysis. Cambridge University Press, New York (2005)
34. Snijders, T.A.B.: The statistical evaluation of social network dynamics. Sociological Methodology 31, 361–395 (2001)
35. Snijders, T.A., Ripley, R.M.: Manual for SIENA version 4.0 (2010)
36. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining. pp. 817–826. ACM (2009)
37. Weitnauer, E., Thomas, N.M., Rabe, F., Kopp, S.: Intelligent agents living in social virtual environments bringing Max into Second Life. In: International Working Conference on Intelligent Virtual Agents (2008)
38. Wu, T., Khan, F., Fisher, T., Shuler, L., Pottenger, W.: Posting act tagging using transformation based learning. In: The Proceedings of the Workshop on Foundations of Data Mining and Discovery. IEEE International Conference on Data Mining (ICDM) (2002)
39. Zhao, Y., Wang, W.: Attributions of human-avatar relationship closeness in a virtual community. In: Emerging Technologies and Information Systems for the Knowledge Society. Lecture Notes in Computer Science, vol. 5288 (2008)