Finding groups in virtual communities

Pekka Maksimainen

Helsinki May 20, 2010

Pro gradu

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author
Pekka Maksimainen

Työn nimi — Arbetets titel — Title

Finding groups in virtual communities

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level Pro gradu	Aika — Datum — Month and year May 20, 2010	Sivumäärä — Sidoantal — Number of pages 44 pages

Tiivistelmä — Referat — Abstract

It is currently a trend to study means of learning in virtual envinronments. This field of research did not exist just over a decade ago due to technical limitations. Nowadays it is technically possible for thousands of users to be online at the same time in a 3-dimensional virtual world and to interact with each other. We explore ways to find user groups in the virtual communities by exploiting relationships in users' chat behaviour and location inside the virtual environment. Virtual worlds have limitations and restrictions in interaction that guide how users behave. With different set of rules people are not expected to behave similarly. Yet it may be possible to find algorithms that are generic enough to capture interesting properties of the virtual world participants.

Understanding user behaviour in virtual environments is interesting at many levels. In social sciences virtual environments are perfect platforms for studies on human behaviour. Collecting data is relatively easy and vast amount of information can be quickly acquired. Gathering and analysing the data is a two-fold thing. Firstly effective data structures must be used to store the collected data for a later use. Secondly effective algorithms must be found to be able to process large collections of data. We will explore a data set collected from a three month project period in a single virtual world with 57 students from several different universities around the world. The students were divided to groups and given a project to complete during the course utilising the virtual envinroment as their main communication channel.

Networks are a suitable structure for storing interactions and properties of users in virtual environments. As the amount of users grows, the complexity of the network is likely to grow as well. The study on virtual worlds thus closely relates to the study on complex networks. We evaluate two approaches on how to find groups in large communities. A topic modelling algorithm LDA is evaluated on the chat corpus of over 10000 chat messages. We try to find groups based on the topics that each project group discusses. Then we build a network based on the spatial closeness of speaker-listener relationship between users. Using the Walktrap algorithm we are able to accurately and quickly find group structures from the speaker-listener network we built.

ACM Computing Classification System (CCS):
H.3 [Information Storage and Retrieval]
H.3.4 [Information Networks]

Avainsanat — Nyckelord — Keywords
Networks, social networks

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Collaborative virtual environments (CVEs) have become a hot topic in social and computer science research. CVEs provide a new medium for people to communicate that practically hasn't existed ever before. For computer scientists CVEs provide an interesting area to study mechanisms of large real-time online systems in which hundreds or thousands of people need to be able to interact simultaneously in one virtual world. There is a strong link between social scientists and computer scientists who need to understand each other in order to create and study CVEs that satisfy the ambitions of both fields. Research results from social studies give guidelines to people building VEs. Vice versa computer scientists create tools to study and analyze virtual environments thus providing tools for sociologists to deeply understand the human behaviour in a world that didn't exist before. This feedback loop is likely to carry on to the next decade as VEs continue to develop and expand. In this paper, we mine the (hidden) information in the data that was gathered from a three month studying period of a class of 30 international students in a popular virtual environment called Second Life.

Virtual environments have developed very quickly after the invention of the modern internet. High availability of the internet made it possible for lots of people to communicate with each other simultaneuosly in real-time over long distances. Various implementations of virtual environments have then become popular with different set of features. Common features for most virtual environments include interaction with other users and possibility for communication. In the early days of the internet interaction and communication were mostly restricted to textual exchange of data. Nowadays it is technically possible to interact in virtual three dimensional worlds with unique avatars with hundreds of other online users.

As people interact with each other in virtual environments they instinctively behave as they would in any other socially familiar situation. The presence of virtual environment does not remove the social interaction that people are used to in the real life. This leads to interactions between people that are common in real life as well – for example close friends communicate with each other more often than with strangers, work colleagues discuss about work related issues rather than leisure time activities, people with similar interests find themselves in the same chat rooms. As people interact between each other they tend to form groups in the virtual communities. Finding these groups effectively from virtual environments is not trivial. It is important especially for social studies or for marketing purposes. Identifying the binding reason between a group of people (e.g. friends, colleagues) is a challenging task as well. Knowing the reasons why people are in contact with each others is likely to be useful for knowledge-based systems.

We present two methods for approaching these issues. Text analysis can be used to derive why a group of people are connected if there is a communicational reason for the existence of a group. Using the Latent Dirichlet Allocation (LDA) topic modelling algorithm groups of same interests can be found. The other group feature that we are interested in is spatial closeness. When people interact with each other and spatial distance is a factor, it is possible to use spatial methods to find groups. We approach the problem by building an undirected network of interactions between users. The network can be then analyzed with conventional network algorithms. We use a random walk based algorithm Walktrap for this task. Using spatial restrictions between listener-speaker pairs we are able to greatly simplify the complexity of the network and thus effectively find groups even in large communities.

We use a raw data set collected from a virtual envinronment Second Life to evaluate our methods. Both LDA and Walktrap require pre-processing of the data in order to be used. For LDA we use conventional text analysis methods and filters to produce a good learning corpus for the algorithm. The pre-processing steps that we take prove to be very important. We analyze how different pre-processing filters affect the result that we get. With trial-and-error a suitable topic model is found. The topic model finds distinct topics but does not fully identify the participants.

Pre-processing of Walktrap requires combination of multiple collected databases to be effectively used. We use timestamp based aggregation to build a combined database of chat utterances and spatial locations with a margin of error in the timestamps. We are able to visualize the spatial groups found in the data using a kernel density algorithm. Then Walktrap algorithm is used on the resulting network from the pre-processing phase and we quickly and confidently find groups that are spatially related.

We first introduce the history of virtual environments and distance education in Chapter 2. In Chapter 3 we present the problem description. We present two possible solutions in Chapter 4 – namely topic modelling algorithm LDA and random walk algorithm Walktrap. We test our approaches with an empirical test setup that we explain in Chapter 5. Results of our methods are presented in Chapter 6. Finally we discuss the merit of the methods and probable future of CVEs and research interest in the field in Chapter 7.

2 History of virtual environments

Virtual environments are a relatively new topic in computer science. The notion of virtual environment (VE) reflects an artificial world within the three dimensional world that people are used to live in. The concept of virtual environment has gradually evolved as communication methods have improved mainly during the years of digital transmissions. It is important to understand how current virtual worlds have evolved and what have been the intermediate steps towards ambiguously defined field of virtual environments. By understanding from where the virtual environments evolved we will be better able to judge the weaknessess and strenghts of the current evolution cycle of VEs.

As we are interested in the learning part of the virtual environments as well we will begin from the concecpt of distance education (DE). First recorded concept of distance education is probably an advertisement in the Boston Gazette on 1728 in which a shorthand teacher offered to teach shorthand by sending weekly lessons to students across the country. In 1833 a Swedish university offered a chance to learn “composition through the medium of post”. Over the years similar courses were arranged through correspondence over regularly issued newspapers. In the 20th century more sophisticated courses were introduced, from phonographic audio lessons to courses for blind people to engineering with special kits. By the 1920’s almost two hundred American radio stations broadcasted distance education to listeners. Couple decades later television courses started to get more popular. In 1969 the British Open University (BOU) was founded which many consider marks the beginning of the modern distance education. After the BOU’s implementation of distance education the topic started to become a new paradigm of education [BH, Hol, MK89].

As for technical development of CVE’s we could start from the early visual signal passing techniques of the 4th century BCE Greek to the similar signal passing techniques of the British in the 19th century. However we concentrate on the electronic and technical evolution of the technologies that are closely related to the concept of virtual environment.

There is no single point of invention for a virtual environment. The first virtual environments might have occurred between Morse code transmitters in the early 19th century who could communicate almost in real-time with other operators even between long distances. The operators did not communicate directly to each other but instead used a Morse coding device to send electric signals over a wire which were then interpreted at the other end. The operators at each end only know about each other because they choose to use the available communication channel. There is no direct way for either party to confirm who exactly is at the other end of the line. This is quite much in agreement with the thought of what a modern virtual environment is. In fact, in 1848 a wedding was arranged over a telegraph line with all the operators along the line "present" at the wedding [Sta00].

In the 20th century communication mediums, mainly telephone lines, were developed to cover vast amount of people. This created a possibility for people to participate in the "virtual" community by calling their friends (through operators). When intranets and internet became popular the concept of modern virtual environment began.

Virtual environment in essence is a medium where one or more humans can interact within the virtual world. The interaction often means communication with other participants in the virtual environment - being either human or an artificial intelligence.

Bulletin board systems (BBS) in the early 1990's provided a text based world for people to meet. It was possible for multiple users to send and receive messages in real-time from other users. If the BBS didn't have multiple telephone lines it was still possible for a user to read public boards and send messages to these boards through the BBS interface. Users who telephoned in afterwards were able to read the board or download the contents for later reading.

In the middle of the 1980's many MUD and ASCII graphics based adventure games evolved. Graphics were still largely limited by computational power and so the user-experience was limited to flat two dimensional virtual worlds. Players in the virtual game world could communicate with each other in real-time and see each other in the virtual world.

With more computer power available and fast broadband connections it became possible to model 3D virtual environments and human interactions as in the real world. Users could interact in human-like user models in a world that imitates the real world. There is interest in understanding how these virtual environments can be used as an efficient learning environment [SN02, BG98]. In the context of learning or co-operation the virtual environment is often called collaborative virtual environment [SHT06, Smi99]. It is important to understand how people interact in the virtual environments to build better CVEs for people's needs. We introduce the basic principles of virtual environments and the basic model interactions.

Collaborative virtual environments can be categorized as a subcategory of distance education [SN02]. Even more refined category would be a subcategory of virtual learning environment (VLE) which includes interactive webpages (eg. correct / incorrect result shown to the user after submitting his answer), wikis, administration of study groups, sharing information to students, etc. Goal of CVEs is to enhance learning. There are many reasons why this would happen. Students will continue learning at home when they have possibility to participate meetings from their home computers. In meetings students will themselves figure out the best way for them to approach the given problem thus learning through self-directed learning. CVEs are meant to be used as a complementary method to traditional teaching, not replacing it. Several studies exist [Kol81, DBA89] showing that individual students benefit when they can learn in a way that matches their learning habits. By using varying teaching tools, such as CVEs in addition to traditional methods it's possible to harness the full potential of students' learning capabilities.

Difference between a "collaborative virtual environment" and a "collaborative learning environment" is often just a difference in the expected human behaviour in the world. For example many online first-person-shooter (FPS) games are capable of hosting up to 64 players simultaneously and allow users to interact and communicate. These FPS games are not designed for learning in the sense that we consider it, thus omitting the learning part of a CVE. Yet the games can involve as much collaborative coordination as any other traditional learning experience. It's feasible however to modify these games[1] from usually violent content to a something of a more noble goal, as shown by Arango et al. in [ACEC07]. They build a virtual laboratory in which students are to collaboratively solve tasks that are given to them. The benefit of this approach is that the rendering engine, AI, physics engine and development tools are readily available and of high quality – developing these components from scratch would be painstakingly slow and difficult process.

In a virtual environment each user usually uses one avatar model to interact with the world and other online users. It is also possible to use multiple avatars either simultaneously or switch between them at will. The VE provides a medium in which the users have certain restrictions and actions. All users follow the same guidelines and have the same set of actions at their use. This defines what is possible for users to do in the VE in question. For example, the view distance and field-of-view

[1] Many multiplayer games allow custom user-content to be added in to the game. Popular framework is the Source graphics engine modifiable by Source SDK tools [http://developer.valvesoftware.com/wiki/SDK_Docs].

of a user is often limited so the users need to move and turn around to see their surroundings.

All objects in the virtual world have an ability to interact with other objects. Mainly we are interested in how the human users interact in the world but it's not impossible for two inanimate objects to be in interaction in the world as well. The interactions in any VE can be described through the spatial model of interaction [BF93]. The model defines *medium*, *awareness*, *aura*, *focus*, *nimbus* and *adapters* which define the relationship between any two objects. These concepts are explained in Table 1 and apply to all objects in VEs.

Table 1: Spatial model of interaction, terms explained

Term	**Description**
Medium	The ether in which all communication and interaction occurs. Objects in the VE need to be able to communicate in the same medium for succesful interaction to happen.
Awareness	Objects that exist in the VE can be observed only if there is awareness of their existence. For example user becomes aware of a wall when he sees or touches it. However the wall does not become aware of the user – unless the user tries to break the wall or interact in some other way with it.
Aura	Each object in the VE has an aura. The aura enables two objects to become aware of each other. When two auras collide in a compatible medium the two objects can become aware of each other. Awareness does not directly follow but a direct focus of either object is required.
Focus	The more an object observes another the more aware it is of the target. By focusing to an object the level of awareness of the observer becomes higher.
Nimbus	When an object is being observed the observer is in the nimbus of the target. In layman terms it could be described as sense of being watched.
Adapters	Adapter is an extension to an object which, when used, amplifies or attenuates aura, focus or nimbus. The object may be, for example, a microphone or table.

Each object has its own set of relationship values in each medium. If an object is not capable to communicate in some medium then it has no effect in that medium at all. The object can still be able to communicate through other mediums. The values are used to control how effective communication in the mediums is.

One of the key aspects of virtual environments is the sense of presence of the user in the environment [Ste92]. User feels himself present in the virtual medium. This is essentially different from chat rooms or telephone discussion in which a user communicates via an intermediate channel. Technically a chat room and a chat in virtual environment might not be significantly different. However it is not only the chat itself but the environment along with its restrictions that provides context information in the communication in the virtual environment. Subleties of human behaviour in physical or virtual world are not in the scope of this thesis. We will consider virtual environments from now on as 3-dimensional worlds in which users can move and interact, being only limited by the restrictions of the virtual world.

Audio and video conferencing have been considered as the main alternative for CVEs. However problems in audio and video conferences are considered greater than those of virtual environments by Tromp et al. in [TSW03]. Use of multiple cameras pointed to documents or other objects of interest is referred as media space conferencing. The main difficulties in audio and video conferencing are listed in table 2.

Table 2: Comparison of audio only and AV conferencing

Audio only	Audio + video
Difficult to identify who is talking	Dialogs significantly longer
Impossible to share documents	Lack of detail in sharing documents

Media space conferences only remedy these problems to a certain degree, still leaving the underlying problems unresolved. Tromp et al. consider [TSW03] constraints of VEs to be clumsy object interaction, limited field of view and limited set of gestures. These problems are mostly technical or VE specific rather than fundamental problems in the medium itself. Audio and video conferences however are restrained by the mediums themselves – it is impossible to share any textual document feasibly through audio or video. Even sharing audio or video content can be cumbersome due to difficulties in saving the transmission at the receiving end.

3 Finding groups in communities

Community is a subgroup of a network which is more tightly knit together than to the rest of the network. Community structures can be found in many social, biological and technological networks. Nodes in such graphs represent entities in their own domain, for instance individuals in social system, genes in gene regulatory network or servers in the Internet. The edges represent interactions, social links, gene modulation and traffic respectively between any two nodes in the network. Detecting communities fast and accurately is useful in many fields [Sco88]. In social networks it is often easy to intuitively identify what communities are, for example a network of close friends, scientists of a particular field or work place relationships. In biological and other complex networks similar community naming convention is not as trivial. Finding community properties from complex networks can guide researchers towards interesting subgroups which might remain undiscovered were these community finding methods left unused. Although social networks can become similarly large and complex there are often social features that limit the overall complexity of the network. In social interaction people have tendency to form groups [Dun98]. Intrinsically assortative mixing and clustering properties dissociate social networks from the most other types of networks [NP03]. Assortative mixing means that two nodes are likely to be connected if they share similar characteristics.

Community finding problem is closely related to the graph partitioning problem. In graph partitioning a graph is divided in to a given number of partitions while minimizing the edge cuts. Edge cut is an operation that removes one edge between two vertices. Edge cuts are required in graph partition to make subgraphs completely disjoint. Community finding aims at finding a set of highly connected subgraphs while minimizing the connections to other communities. Difference to graph partitioning is that the amount of subgraphs is not given and the goal is not to find communities of the same size. Another distinct feature is that there can be overlapping communities in the network which brings extra complexity to the community finding problem.

Since the beginning of social network analysis the main concern has been the definition and the identification of subgroups of individuals within a network. As community structures are found in other networks than just social the terms are quite varying. Instead of community we might be discussing about cluster, cohesive subgroup or group but we would still mean the same thing, a subset of nodes of a network that have strong, intense and direct ties to each other. The term community is often used in the context other than just social networks, thus using a term such as group more precisely connotes the domain of social network, involving interactions between individuals. We will use this separation in this thesis where appropriate. Terminologically we emphasize that a group does not (necessarily) include the edges connecting the nodes. If there is a risk of being ambiguous it is advised to explicitly clarify whether the edges are to be included in the term. Term subgraph can be used to implicitly include any collection of nodes selected from the nodes of the whole graph, together with a subset of the edges connecting those nodes.

We present how the concept of community [FLM04] can be eased from a fully connected network – which would be a perfect community. In a fully connected network (or subgraph) each node is connected to every other node in the network. The structure is tight because the distance between any two nodes is one, assuming an unweighted network. Requiring this property of mutuality of ties is far too strict to be useful for finding actual communities. We can ease the requirement by allowing two nodes to be connected via longer paths. This closeness or reachability defines that any node of the subgroup can be reached in n steps from any other node. This captures the notion of $n-cliques$ where any node in a *clique* is reachable through at most n intermediaries. This requirement relaxes the degree of nodes as same paths can be used to connect nodes – each node does not need to be directly connected to every other node. Further relaxation in the degree is set by frequency of ties which defines that each node need not be connected to every other node in the subgraph. A $k - plex$ [HBBS08] is such a subgraph of n nodes where any node is connected to at least $n - k$ other nodes. These properties have set requirements within the subgroup. To extend or connect such subgroups to the rest of the network we define a relative frequency of ties which compares the links within the subgroup to links outside the subgroup. The relative frequency of ties means that there are more links from a node within a subgroup S to the other nodes in S than to nodes that are not within S. Such set of nodes is called an LS set which can be extended to the definition of a *lambdaset* if edge connectivity between any pair of nodes in S is larger than edge connectivity between any node within S and a node outside S. The idea captured in this definition is that cohesive groups are hard to make disjoint by removal of edges.

Social analysts were the first to formalize the idea of communities and how to mathematically measure properties that define a community. In essence we want to find a partition $\mathcal{P} = \{C_1, \ldots, C_k\}, (\forall i, C_i \subseteq V)$ in a graph $G = (V, E)$ where C are the k groups, V are the nodes and E the edges. If the proportion of edges inside the

C_i is high compared to the proportion between them then the partition represents a good community structure (Figure 1).

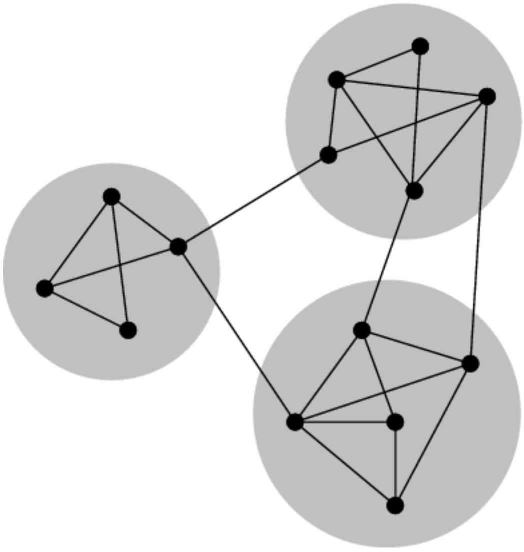


Figure 1: Three highly connected groups in a network. Connections between groups are sparse but internally there are many connections.

Social networks are not always based on the direct relationships between two individuals. Instead they are built upon indirect relationships such as through bureaucracies, markets, or impersonal information technology. In indirect relationships the participants may not recognize the connections as a social relations. A direct social link between two individuals would be direct communication from one person to another where each party is aware of the social interaction. An authority can also set more weight to social relationships that are considered more important than others.

4 Methods for finding groups

There exist several approaches for finding communities in graph structures. Remember that community finding is related to a graph partitioning problem. Graph partitioning is known to be an NP-complete problem [Wik09] and so approximation algorithms are necessary. Divisive algorithms detect links between communities and remove them from the network, agglomerative algorithms merge similar nodes or communities recursively and optimization methods try to maximize an objective function. Divisive and agglomerative algorithms can be categorized as hierarchical methods. Agglomerative algorithms focus on the addition of edges and divisive on removal of edges from the network. Agglomerative methods start from an initial network which has no edges. By calculating similarities between vertices new edges are added in the network between two vertices. The vertex pair with the highest similarity is connected with an edge – method for finding the two most similar vertices depends on the problem domain. Continuing the addition of new edges between vertices the network is built bottom-up. Final step in the process connects the last two disjoint components by connecting two vertices in the remaining components, resulting in a single connected component. The intermediate steps, graph partitions, represent the communities that were found in the process. Divisive methods work in the different direction, top-down. Division begins from a full network with all edges in the graph. Then the two least similar connected pairs of vertices are located and the edge between them is removed. As the removal of edges continues parts of the network become disjoint. Finally all the edges are removed and the network of n nodes consists of n communities. The intermediate partitions represent the different community divisions. Hierarchical methods have been found to be very good in terms of computation speed and resulting communities [BGLL08]. This recursive community building method produces a hierarchical tree, a dendrogram. Because of the bottom up building process it is possible to analyze the network communities at different levels by cutting the dendrogram tree horizontally at different heights. Yet, there is no guarantee that an optimal partition is found.

Hierarchical algorithms always produce some division of the network into communities. The resulting division may or may not be good so we need to be able to measure the goodness of a given partition. In community finding it's not required to do cuts as in graph partitioning problem and the goal is different so the same measure of goodness is not applicable. Often used measure of quality is so called modularity [NG03] of the partition. The idea behind this measure is to give a numerical value to the community properties discussed in Chapter 3. Consider a $k \times k$ symmetric matrix $\mathbf{e}$ whose element e_{ij} is the fraction of all edges in the network that link vertices in community i to vertices in community j. Fraction of edges in the network that connect vertices in the same community is given by the trace $Tr\mathbf{e} = \sum_i e_{ii}$. A partition that has a good community structure should have a high value of this trace. However the trace is not a good measure because it would be possible to assign all nodes in the same community which would give the maximimal value of $Tr\mathbf{e} = 1$ but would not give any information about the communities. We define the row sums $a_i = \sum_j e_{ij}$, which represent the fraction of edges that connect to vertices in community i. In a random network, without community structure we would have $e_{ij} = a_i a_j$. Modularity $Q(\mathcal{P})$ of a partition $\mathcal{P}$ is

$$\begin{aligned} Q(\mathcal{P}) &= \sum_i (e_{ii} - a_i^2) = Tr\mathbf{e} - \|\mathbf{e}^2\| \\ &= \sum_{C \in \mathcal{P}} e_C - a_C^2, \end{aligned} \tag{1}$$

where $\|x\|$ indicates the sum of the elements of the matrix $\mathbf{x}$, e_C is the fraction of edges inside community C and a_C the fraction of edges bound to community C. The best partition $\mathcal{P} = \{C_1, \ldots, C_n\}$ is the one that maximizes Q. This equation captures the idea that in a good community structure the proportion of internal edges in communities is high compared to the edges between them. We can rewrite Eq. 1 with edge weights as

$$Q(\mathcal{P}) = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

where A_{ij} represents the weight of the edge between i and j, $k_i = \sum_j A_{ij}$ is the sum of the weight of the edges attached to vertex i, c_i is the community to which vertex i is assigned, the δ-function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise and $m = \frac{1}{2} \sum_{ij} A_{ij}$. An interesting result found by Fortunato and Barthelemy [FM07] is that modularity optimization method does not recognize small communities – this is the so-called resolution limit. We don't present the details of the proof but mention it here because of the unexpected result. The resolution limit yields from the fact that finer details (small communities) can be hidden in an overlapping large community. The properties of a community are dictated by the size of the whole network rather than local properties of community modules. This means that the definition of a community is not consistent with its optimization. Although mathematically the structure is optimal it is possible that actual communities are hidden beneath. Fortunato and Barthelemy suggest that other test methods are used to confirm that no actual communities are left hidden. The Walktrap algorithm that we evaluate does not have this problem as it builds communities bottom up, first combining single nodes to small communities and continuing until there is a single community left. One might now think that why use a heuristic algorithm rather than choosing the partition that maximizes Q. As already mentioned the problem is known to be NP-complete, and thus known to be very difficult to solve. Number of ways to divide

n vertices into g non-empty groups is given by the Stirling number of the second kind $S_n^{(g)}$ so the number of distinct partitions is $\sum_{g=1}^{2} S_n^{(g)}$. Exhaustive search over this space would take at least exponential amount of time and is thus impractical for networks larger than 20 or 30 vertices [New04].

Apart from traditional approaches in graph partitioning as described earlier we try to find groups from data using topic modelling (LDA). This approach is based on the the similar chat behaviour of the users. We explain the process in Chapter 4.2.

4.1 Walktrap - modularity optimization

Walktrap [PL05] is a heuristic algorithm that can be used to find communities in an undirected weighted graph. Walktrap uses random walks of fixed length t to find densely connected areas in the graph where the random walks tend to get “trapped”. Each random walk starts from a randomly chosen node i and ends in some reachable node j. Probability of going from node i through t steps and stopping in node j is denoted by P_{ij}^t. In each node the random walk chooses the edge to travel by probability $P_{ij} = \frac{A_{ij}}{d(i)}$ where A_{ij} is the adjacency matrix and $d(i) = \sum_j A_{ij}$ is the degree of node i. The adjacency matrix A can be weighted which is an advantage of this method.

There are two properties in random walks that we are interested in:

Property 1. *When the length t of a random walk starting at vertex i tends towards infinity, the probability of being on a vertex j only depends on the degree of vertex j and not on the starting vertex i:*

$$\forall i : \lim_{t \to \infty} P_{ij}^t = \frac{d(j)}{\sum_k d(k)}.$$

Property 2. *The probabilities of going from i to j and from j to i through a random walk of a fixed length t have a ratio that only depends on the degrees d(i) and d(j):*

$$\forall i, j : d(i) P_{ij}^t = d(j) P_{ji}^t.$$

These properties are common for all random walks. Each random walk can be thought to reveal a little bit of information about the network structure. Property 1 tells us not to set the path length t too long as then the probabilities would only depend on the degree of the vertices. Property 2 defines that P_{ij}^t and P_{ji}^t encode the same information – each random walk from i to j or j to i reveals the same information. For nodes that lie in the same community the probability P^t_{ij} will be high – however high P^t_{ij} does not imply that the nodes are in the same community.

The algorithm proceeds iteratively merging two nodes or communities in each step. Merging is done by merging the two closest communities defined by a distance measure r in Definition 1.

Definition 1. Let i and j be two vertices in the graph. The distance r_{ij} between i and j is given by

$$r_{ij}(t) = \sqrt{\sum_k = 1^n \frac{(P^t_{ik} - P^t_{jk})^2}{d(k)}} = ||D^{-\frac{1}{2}}P^t_{i\bullet} - D^{-\frac{1}{2}}P^t_{j\bullet}||,$$

where $\|.\|$ is the Euclidean norm of $\mathbb{R}^n$.

For clarity we will denote $r_{ij}(t)$ as r_{ij} to simplify the notations. The distance r is a distance between two nodes, we now generalize this to a distance between two communities.

We can randomly and uniformly choose a vertex from a community C to define the distance between a community C and a vertex j as

$$P^t_{Cj} = \frac{1}{|C|}\sum_{i\in C} P^t_{ij}.$$

This defines a probability vector $P^t_{C\bullet}$ that allows us to generalize our distance between communities as described in Definition 2.

Definition 2. Let $C_1, C_2 \in V$ be two communities. We define the distance $r_{C_1C_2}$ between these two communities by

$$r_{C_1C_2} = \|D^{-\frac{1}{2}}P^t_{C_1\bullet} - D^{-\frac{1}{2}}P^t_{C_2\bullet}\| = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{C_1k} - P^t_{C_2k})^2}{d(k)}}.$$

We can now define the distance between a vertex i and community C as $r_{iC} = r_{\{i\}C}$. These definitions allow us to find and merge the communities iteratively until a single component is left. The algorithm is hierarchical so that the final iteration produces one connected component. From the intermediate steps it is possible to select the partition that maximizes connections inside communities and minimizes the link weights between communities.

The evaluation of Walktrap algorithm is presented in Chapter 6.2.

4.2 LDA - topic modelling

Topic modelling is a concept in which a corpus $\mathcal{D} = \{\mathbf{w}_i, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ is given which consists of M documents $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ where each document is a sequence of N words, w_n is the nth word in the sequence and the task is to identify underlying k topics in each document. A word is not restricted by language constructs (i.e. space separates words) but may be any basic unit of discrete data. A topic modelling algorithm is given the corpus from where it then learns the model. The model can be used for common tasks such as classification, novelty detection, summarization, and similarity and relevance judgments.

Commonly the documents are human readable text articles from magazines, newspapers, scientific publications or other large text collections. There exist vast archives of texts that would be practically impossible for humans to read through and classify. An automated process is required to handle such collections of data. On a single document level the contents should be coherently about one or few topics but may vary between different documents. Human written text articles naturally conform to this structure – for example movie reviews, news articles of different topics and scientific publications. Although articles are often concentrated on a single topic it is very likely that portions of the articles include chapters that woud be better classified under different topics. For this reason a single document may be classified under multiple topics even by a human supervisor. An algorithm that strives to classify documents in text corpora can assign each document a set of topic probabilities. Topic modelling methods can be used in other data sets as well, from bioinformatics to collaborative filtering to content-based image retrieval. In this thesis we attempt to use topic modelling in the context of chat text corpus in order to find distinct user groups in the data.

Intuitively the concept of a topic model is easy to understand if we consider how humans naturally would label a set of words. From a topic model learned by a modelling algorithm it is possible to draw the most representing words for each of the topics. If the representative words for a topic include terms such as *ball*, *goal* and *player* then one label that could be assigned to that single topic would be *sports*. We could expect other words like *win*, *lose* or *tie* be among the most representative words in the model. If such is the case then the model can quite well extract sports articles from a corpus even though there is no understanding of the language involved in the extraction.

Topic modelling in the context of chat corpus is intrinsically different from traditional text corpora such as news article collections or scientific journals. Chat messages can include variety of spelling mistakes, ascii graphics to express emotions, dynamically changing amount of chatters and possibly multiple interleaving on-going discussions. These characteristics make chat topic modelling a challenging task. In [TT04] topic modelling is done specifically on the terms of social interactions between chatters and the resulting social network.

In Chapter 6.1 we evaluate an approach that is not generally used for community finding but could be exploited to reveal information if community structure exists. The idea is to use topic modelling to create a model of the complete chat data available. The model should be good enough to distinctly recognize discussions of certain topics. We don't expect single utterances to be trustfully classified to a certain topic but it should be possible to quite reliably assign each conversation to some topic. We assume that groups that work on a problem – even if the same problem is shared with other groups – develop a vocabularity that is distinct from the other groups. It might be abbreviations or misspellings that give out information about the group or any other feature that all group members adapt to. We assume that working in a group exposes all members to similar vocabularity that other group members then involuntarily or not conform to. Because the groups involve several members it is likely that one or few have a vocabularity that is distinctively their own. If this adaptation of similar vocabularity happens [MSLC01] then each group could be identified by identifying the vocabularity and thus the topics that are unique. Inspecting users' utterances would then reveal which topics they discuss and thus in which group they are most adapted to. In [NRT04] text analysis was used to detect similar vocabularity between different user aliases on an internet message board and quite succesfully recognize which aliases actually belonged to the same person. The risk is that the vocabularities are not distinct enough and the topics that are found are common for each group.

If we have n distinct groups we assume to find at least n distinct topics. The n topics would distinctively identify each of the groups as they would share the similar vocabularity. Even if we were to search for more topics, say $n + 10$, it would be feasible that n of those topics are identifiable to certain groups. The idea of topic modelling is that we can assign the same discussion to multiple topics with a certain degree of belief - for example to a group topic and to a topic of a generic discussion (e.g. project, schedule). If we were to analyze chat data with an online algorithm it might be possible to dynamically build belief over which topic and in which group the person is discussing. By recognizing some popular words in topics (e.g. schedule, project) we may identify a generic topic and later on when a vocabularity that is more group specific is spoken we are more certain about in which group the person is most likely attending to.

Latent Dirichlet Allocation (LDA) [BNJ03] is a topic modelling schema that has recently gained much interest. In information retrieval it is important to be able to store vast collections of text data in compact but yet informative form. For this purpose tf-idf scheme [SB88] is a popular and good choice – a collection of documents can be reduced to a term-by-document matrix which can be presented in fixed-length lists of numbers. Although tf-idf can identify discriminative words for documents in collections it does not reveal much else about broader concepts. To put single terms in to a context, a latent semantic indexing (LSI) was proposed in [DDF$^+$90]. Evaluating LSI's capability with a generative probabilistic model [WdVdJ07] of text corpora reveals how well LSI is able to recover aspects of the generative model from data. A direct way of fitting the generative model in the data was introduced in [Hof99], a probabilistic LSI (pLSI) which models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of topics. Both LSI and pLSI assume a bag-of-words model which defines that any word in a document can be exchanged. The order of words is not important and so any two words can be exchanged – a principle that is known as exchangeability in probability theory. Both models also assume exchangeability of documents which means that the order of the documents can be neglected. These observations lead to the LDA model that captures the exchangeability of both words and documents.

The LDA model was further extended to correlated topic model (CTM) ([BL07]) which measures the similarity between topics. The main idea in CTM is to take latent topics in the same document into consideration. For example an article about genetics may very well be about health and disease but very unlikely to involve X-ray astronomy. Measuring the correlation of topics in articles gives out information about topic groups. Presence of one topic is likely to correlate with another topic. Some topics may be repelling while others attracting. LDA assumes independence between each topic which is an unrealistic assumption. While CTM yields better results we use LDA in our experiments because of the simplicity and good results obtained by Blei et al.

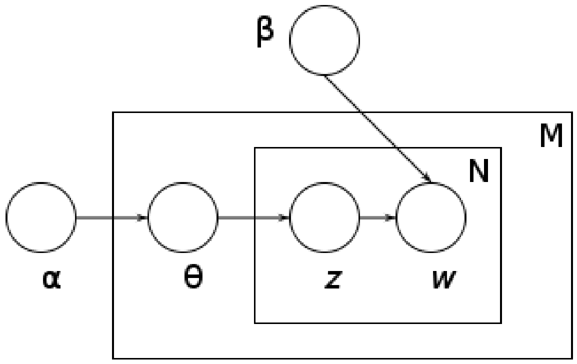


Figure 2: Plate model describing how the LDA model works. Outer plate represents documents, inner plate topics and words within a document.

LDA is a generative probabilistic model of a corpus. Documents are represented as random mixtures over latent topics where each topic is modelled by a distribution over words. Graphical representation of the model is shown in Figure 2 where relations between documents, topics and words are shown at different levels. The parameters α and β are corpus-level parameters sampled once in the process of generating a corpus. The variable α is the parameter for the Dirichlet prior for document topic distributions and β is the parameter for topic word distributions. Variables θ_d are document-level variables for each document d, sampled once per document. Variables z_{dn} and w_{dn} are word-level variables and are sampled once for each word in each document, where z_{dn} is the topic for nth word in dth document and w_{dn} is the actual word.

The generative process follows the following steps. First a distribution for document lengths is chosen by $N \sim Poisson(\xi)$ where the Poisson assumption is not critical. Other, more realistic document length distributions can be used. Then we choose a k-dimensional Dirichlet random variable $\theta \sim Dir(\alpha)$, where k is the dimensionality of the Dirichlet distribution and the topic variable z. For each of the N words w_n we choose a topic $z_n \sim Multinomial(\theta)$ and choose a word w_n from $p(w_n|z_n, \beta)$

(a multinomial probability conditioned on the topic z_n). Word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$, where w^j denotes a V-vector such that $w^v = 1$ and $w^u = 0$ for $u \neq v$ and z^i is the ith topic of the topic vector z.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta),$$

where $p(z_n|\theta)$ is θ_i for the unique i such that $z_n^i = 1$. Then to obtain the marginal distribution of a document we integrate over θ and sum over z:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta.$$

Probability of a corpus is obtained by taking the product of the marginal probabilities of single documents:

$$p(\mathcal{D}|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d.$$

We wish to find parameters α and β that maximize the log likelihood of the data in document corpus $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$. Using variational EM it is tractable to estimate the maximum likelihood $\ell(\alpha, \beta) = \sum_{d=1}^{M} \log p(\mathbf{w}_d|\alpha, \beta)$. We are not interested in inference so LDA model estimation will suffice for our needs. We use complete chat corpus to build the LDA model.

The evaluation of LDA topic modelling is presented in Chapter 6.1.

5 Empirical setup

We evaluate the community finding algorithms in a real world social network[2] that sprung into existence in a virtual environment called Second Life. Second Life is a virtual environment that is open for everyone through a free registration and the

[2] The Global Virtual Education (GloVEd) project, http://simlab.tkk.fi/GloVEd. We wish to thank the SimLab team for providing the data.

client-side application. Once the user has logged in he is then free to roam the virtual world. The user can travel long distances by jumping high in the air and then fly to his destination or walk short distances for example inside buildings. In our research the users interact in group rooms and only move by walking. Because the space is confined and clearly limited by walls it is feasible to employ spatial features in our methods. The users enter the conference place by walking in to a portal and exit the same way. While inside the conference hall all users only enter their own group's room or the public lobby. Overview of the conference hall and the residing group rooms is shown in Figure 3. Each room was decorated by the group members themselves which might guide how group members move in their rooms. Each room had mandatory team wall for sharing content with others and a launch screen for launching utilities.

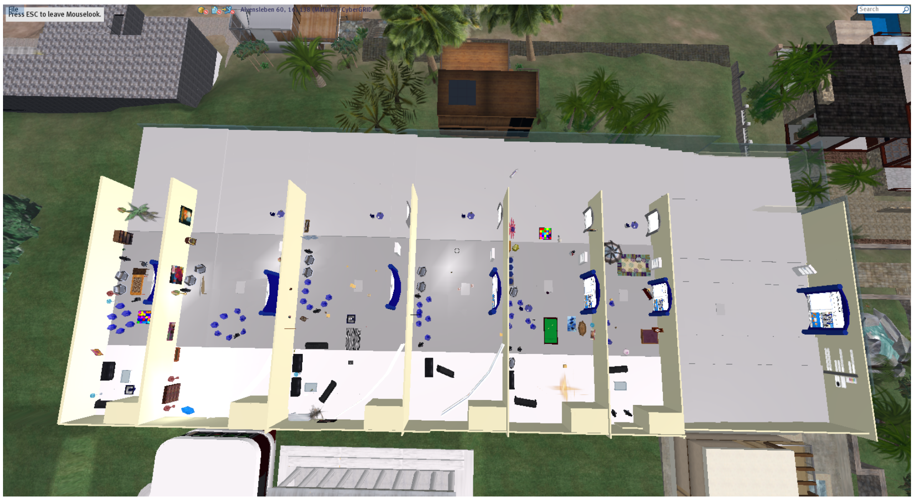


Figure 3: Aerial view of the group rooms in Second life. Group 7's room is at the far left and other rooms are ordered in numerical order.

The virtual world provides users a set of common tools that are used in most other virtual environments as well. Users can chat with each other by using text messages. Length of the messages is not limited but usually they are kept short, 10 – 200 characters long. Text messages appear next to a character who speaks it. Persons who attend the discussion need to be in viewing distance to be able to see the chat messages from others. It was also possible for users to use voice communication but due to technical problems the use was very minimal.

In our CVE setting the groups are known a priori and thus we have a good test network for evaluating community finding algorithms. A data set of chat messages linked to spatial locations were collected from a three month period. In the Second Life virtual environment six groups regularly held meetings. Each group had their own room which they used to hold the meetings. Group members could freely join the meetings by walking in to the room and leave at any time as well. Meeting times were usually held at certain hour or occasionally agreed upon via e-mail prior the meeting. Each group consists of 6 to 10 users. Groups did not communicate with members in the other groups. Group size was kept small as is strongly suggested by Bernard et al. [BRSP00].

We will describe the collected dataset through common text analysis methods and ways the data was stored in the database. The data includes textual representation of users' chat messages and spatial information linked to each event. Description of the chat database is presented in Table 3.

Table 3: Chat database description.

Field	**Type**	**Description**
userid	*int*	Unique id number for each member
groupid	*int*	Member's group identificator
loc	*point*	Coordinates of event
chat message	*text*	Utterance
time	*datetime*	Event time in 5 second accuracy

Initially the data was not recorded in this format but as separate data sets of different field types. Combining the sets in to a single database with optimized field types we could significantly speed up the data mining.

Few key numbers from the data are presented in Table 4. It is interesting that about 16 % of the chat messages are not unique. This can be explained by the vast amount of short agreement, greeting and leaving messages such as "ok", "hello" and "bye". Many of the 5703 unique words are nonsensical (*pdg*), typos (*abour* ∼ about) or result of the filtering (*don* ∼ don't).

Most of the vocabulary is in English but also native languages of the participants are present. In addition there are lots of internet or computer science jargon (*lol*, *k*, *msg*, e-mail / web addresses). Due to strict filtering rules all smileys (eg. *:-)*) were removed.

There were six active groups and one administrative group which did not communicate after the initial testing period. In Figure 4 the chat activity is presented as

Table 4: Key figures describing the collected data.

Chat data	
13529 chat messages	11366 unique chat messages
93417 words	5703 unique words
6 groups	plus one inactive administrative group
57 users	53 active over the observation period
Coordinate data	
606626 spatial coordinates of user movements	

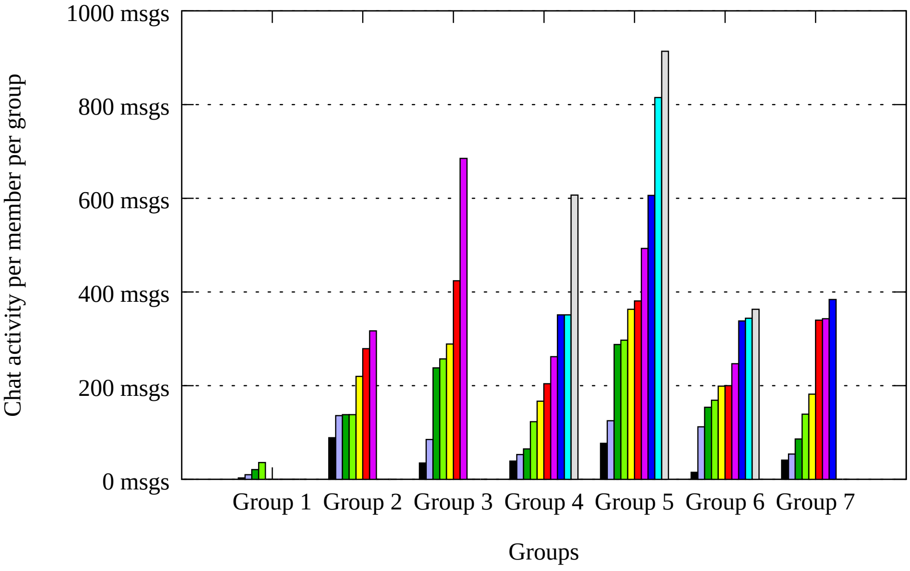
amount of chat messages and chat words. While this is not a good measure of the true activity and effort towards achieving the group goals, it reveals a bit about group dynamics. Preece [Pre01] has suggested several other measures for activity measurement. Measuring the number of on-topic messages, reciprocity or even quality of contribution might be more informative measurements. These measures require either manual classification or sophisticated semantic text analysis methods and so are not in the scope of this thesis. Preece notes that as educational communities may be strongly goal-directed it would be more reasonable to use measures of information exchange and collaborative working than measures of social chitchat. Subjective overview over the chat logs reveals that vast amount of the chat is indeed chitchat or meta-discussion about the available tools. In practice many of the participants fall back to the familiar e-mail as a communication channel – which of course forces the other group members to use e-mail too. Even though they were present in the virtual world they chose to use e-mail as the channel to share important files or thoughts. Many participants clearly expressed their frustration in the virtual environment which might have further directed the actual collaboration in the e-mails and other communication channels. Thus the chat data might not be a good sample of a collaboration in a virtual world – however we need to remember that social interaction, even that unrelated to actual work is an important part of a succesful collaboration [Cai05].

In each group there are a few users who have been more active than the group's average. As chatting always requires two or more parties it is expected that as two active chatters enter the room at the same time they will work as catalysts to each other. This in turn can provoke other, otherwise silent memembers to participate in the discussion. We can compare groups 5 and 6 which have the same amount of members but clearly group 5 is more active. It would be interesting to evaluate the actual achievements of the groups and see whether the chat activity correlates with the results. Whether the chat activity is just socializing or resourceful discussion is not taken in to account here.

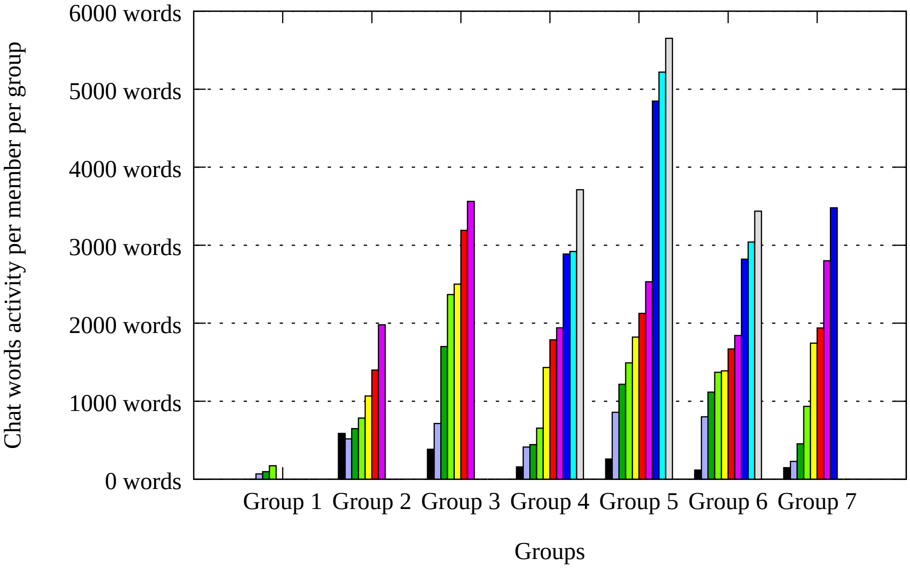
Chat data was pre-processed by following steps. First all lines were translated to lower-case. Then non-alphabets were removed (allowed characters being {a-z, }) and all lines trimmed from repeating white space. Blank lines were ignored – due to filtering chat messages such as "..." or ":-)" become blank. This is done to ease the task of text analysis. Although subleties of chatting may be lost in filtering the majority of chat data consists of alphabets only.

In our analysis it is not necessary to consider the small subleties that are lost in the filtering process. To further delineate the filtering process let's examine one of the chat messages: *ok Melissa.....take care,,,.* The filter removes the extra punctuation. After the filtering the message is briefly *ok melissa take care.* One may ponder that the filtering alters the contents too much, for example words with an apostrophe (*don't*, *can't*, *i'm*, *i'll*) are broken. We consider this a reasonable loss of accuracy.

Spatially the data is interesting as well. As there are six groups and the rooms are next to each other it is likely that occasionally members of two different groups in different rooms stand close to each others only separated by a wall. Members of different groups may be talking at the same time but they would not be communicating with each others but with other members in their groups. This is a good test setup because group members are usually close to eachothers and most of the time they aren't near members of the other groups. Still there are occasions when members of the other groups may be closer to the speaker than his own group members. If the groups were completely spatially isolated from other groups then this kind of situations wouldn't happen. Now if we only look at the spatial data it is not always trivially clear to whom a speaker is actually speaking to. Consider a virtual world in which the team members were free to go anywhere they like and keep meetings at any place at any time. There would be many situations in which it is not clear from the spatial data if users near the speaker just happen to be there or if the speaker is actually speaking to them. It would be spatially insignificant where the communication happens as group members would still be close to each other. Yet random wanderes may occasionally seem as if they were within the group. This is something that our spatial approach manages well. The data set that we have also includes many such events.



(a) Chat messages



(b) Chat words

Figure 4: Chat activity of each group and group members. Each bar corresponds to one user and each member belongs to one cluster. In figure 4(a) activity is measured in amount of chat messages. When activity is measured in amount of words (figure 4(b)) groups 6 and 7 seem more active.

6 Results

Visualizing the data is a useful way to approach many problems. Here we can use spatial information to plot heatmaps[3] on the blueprint of the conference hall structure. We build the heatmaps with a discrete kernel density estimation [Lev09] – that is, we sum over a set of discrete kernels instead of quantize over continuous kernels. This approximation is good enough for our purposes because a) we are mainly interested in the graphical representation and b) there is no "hidden information" lost even if our estimation is not completely accurate. We use a symmetrical normal distribution kernel with a bandwidth of about 1000 VE units. We initialize a matrix of $M = 200 \times 200$ in which we sum the kernels. The virtual world coordinates can be translated to matrix entries i and j in the matrix. For translation we need to know our observation window which is the area in the virtual world that we are interested in – in this case the rectangular conference hall. When we know the width and height of the virtual observation window we can compute where each of the virtual world coordinates is mapped in the matrix M.

It is possible to represent the matrix as 200×200 pixel picture, each cell representing one pixel. We then normalize the values on range $0 - 255$ and use a pseudo coloring algorithm to map the values to RGBA color space. The pseudo color map, i.e. heatmap can be then superimposed over the blueprint of the conference hall. Alpha channel is transparent on the non-heated areas and fully opaque on the maximum heat areas.

For normalization simple linear normalization constant (i.e. $\max M_{ij}$) is not a good choice. If this choice is done then only a few intense spots are usually found. Reason for this is long discussions among several users in the same place or people who forget to logout and sit idle in the same place for long periods of time. These events then outweight most of the other heat clusters. Instead we've normalized the matrix entries to be $M'_{ij} = \frac{\sqrt[5]{M_{ij}}}{\sqrt[5]{\max M_{ij}}}$. By taking the 5th root of a cell value and matrix maximum we essentially lower the peaks and raise the small bumps in the heatmap. This allows smaller clusters to be found while preserving the relationships between low and high locations. Choice of the 5th root of the matrix maximum was based on trial and error and on the visual pleasance. In Figure 5 the resulting heatmap is superimposed over the conference hall blueprint.

[3] We call these graphs heatmaps. These figures often appear in the context of kernel density estimation without the term heatmap being mentioned.

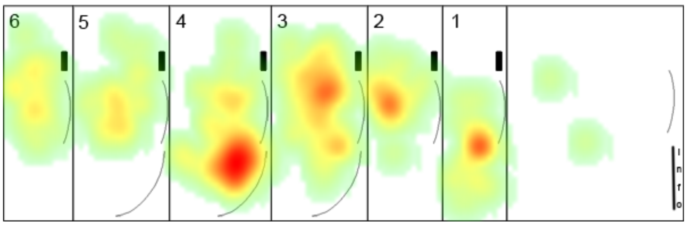


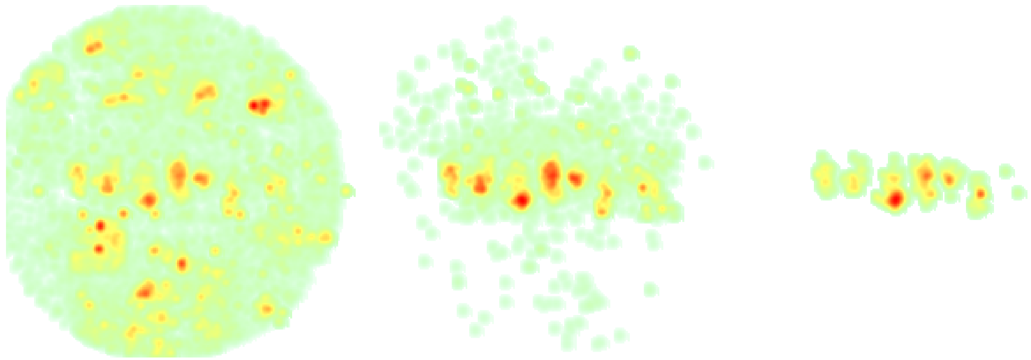
Figure 5: Heatmap of chat activity plotted on top of the blueprint of the conference hall. Room walls and some important objects of interest are shown. Curved screens are teamwalls and black rectangles launch screens.

In Figure 6 the filtering process from the complete spatial data to study group chat locations is shown. In 6(c) the study groups can be visually identified as there are clear cluster centers in each room. The heatmap is however normalized over the maximum of all rooms so minor details in the group members' internal positions are slightly blurred. Applying the same heatmap generation on each group specifically brings forth details from each study room. Figure 7 shows heatmaps for each study group. Group 1 is the administrative group which wasn't much active so there is no heatmap for that group. In other groups there are more interesting formations. The rooms weren't identical either in size or location of the furniture.

6.1 Experiences with LDA

Our attempt is to find distinct user groups by identifying topics distinct to each user group. We assume that each of the groups develops a vocabularity that can be uniquely identified.

Let us go through the important pre-processing phase. The main discovery in this chapter is actually the importance of the pre-processing of the chat data rather than the evaluation of the chosen topic modelling algorithm (LDA). We present different approaches used to filter the data and how they affect the topics found by LDA. The chat data was first pre-processed and then LDA was applied to compute the topic model. Eventually the model could have been used to identify the group of new participants who join any one of the study groups and begin to discuss with the other members of the group through inference – however such a task was left for future work. We iteratively tested multiple pre-processing methods and parameters on LDA to find a good topic model. During the iterations it became



(a) All movement data plotted (incl. outside users)

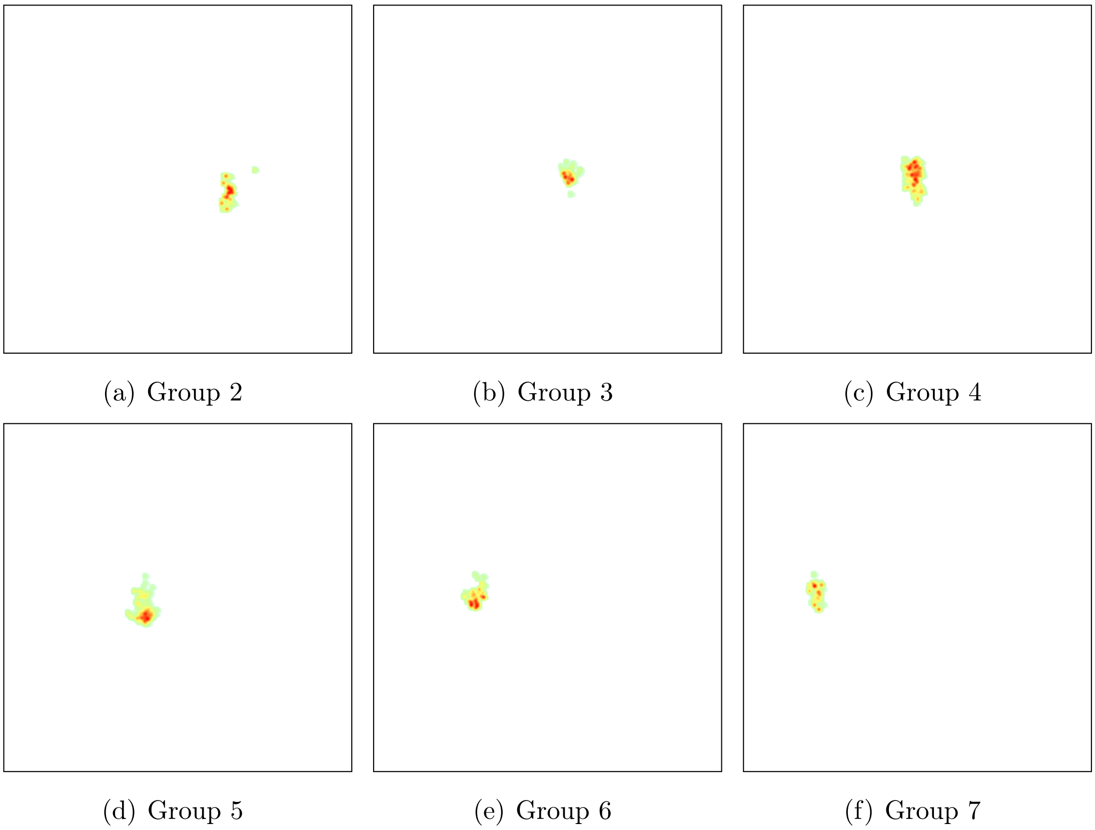
(b) Study group movements

(c) Study group chat locations

Figure 6: Coordinate data filtering process presented as heatmaps. In figure 6(a) movement is observed all around the observation window. In 6(b) the movement is much more concentrated in the conference rooms as only the study groups' movements are plotted, although roaming outside the conference centrum is still visible. When we plot the chat activity in 6(c) it is clear that most communication happened inside the conference rooms – six clusters and few outlier clusters near the entrance / exit portal can be easily seen.

clear that the pre-processing phase quite much dictates what kind of model LDA can find. This seemingly obvious issue is often largely disregarded (e.g. [TT04, BNJ03, BL07]) as a minor detail. In our findings, as in many other publications the pre-processing is clearly based on trial-and-error method. Common pre-processing filtering involves removing stop words (e.g. *a*, *the*, *with*) which are mostly language structures without any real meaning, removing rare words or removing the most frequent words. Depending on the problem domain more filtering can be done to either preserve, remove or alter the data that inherently would or wouldn't occur. This pre-processing is of the utmost importance for finding a good starting set for the actual topic modelling algorithm. In our empirical test we take the pre-processing to the point where it probably becomes more important than the chosen topic modelling algorithm itself.

Thus instead of our expected evaluation of the topic modelling algorithm and the goodness of the model we could as well evaluate the goodness of the ambiguous pre-processing phase. To understand the iterative (and subjective) evaluation of the effects of the pre-processing on the topics given by LDA see Algorithm 1. The algorithm describes the steps we've taken to ultimately achieve our resulting topic


(a) Group 2

(b) Group 3

(c) Group 4

(d) Group 5

(e) Group 6

(f) Group 7

Figure 7: Chatting locations within the group rooms. Figures 7(a) to 7(f) show each group's internal distribution of chat locations.

model. On line 2 we define the filters – single filter can include many non-trivial phases though. A filter could for instance remove stop words, rarewords, frequent words, find the stemming for all words and translate them to their base form, correct typos based on the word context and so on – tasks that may not be trivial. Now we casually pass the step as if it was a simple task. On lines $3 - 17$ is the main loop where we actually evaluate the topic model with given filters. It would be plausible to add more filters in the set inside the loop but we will omit that. For example the evaluation of the topic model might reveal some information that could be used as a guide to modify an existing filter. Lines $7 - 13$ transform our corpus in to a form that we require it to be for the actual LDA implementation. Computation of the topic model is done on line 14 where we settle on finding six different topics. We noticed that using larger amount of topics mostly scatters the most representative words to multiple topics. On line 15 we subjectively evaluate the results – something that could be done properly with perplexity measure [BDPd$^+$90]. The perplexity of a language model is the inverse of the geometric average of the probabilities of the per-word predictions in the test data. For a test set of M documents, the perplexity is:

$$perplexity(\mathcal{D}_{test}) = exp\left\{ -\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d} \right\}.$$

We do not see it necessary to deeply delve in the comparison of different models due to the nature of our exploring quest on the topic. On line 17 we either accept the model or continue until all filtering methods have been exhausted.

At first we start with the whole corpus with practically no filtering other than transforming the chat data to alphabetic words. We choose to preserve the names of the users (each user had a two-part name) in front of each chat message. Preserving the names is important because we would like to have these names to appear in the topics that LDA finds. Result is as expected with no clear topics found and stop words occupy the top positions in each topic. Following this observation it makes sense to remove the common stop words – but which ones and how many of them? We evaluate a few choices: removal of the 20 most popular words, the 100 most popular words and finally setting on a list of 319 stop words found with a "stop words" query on an internet search engine. Although stop words for different texts may vary we consider the list of 319 most popular English words a good choice. The

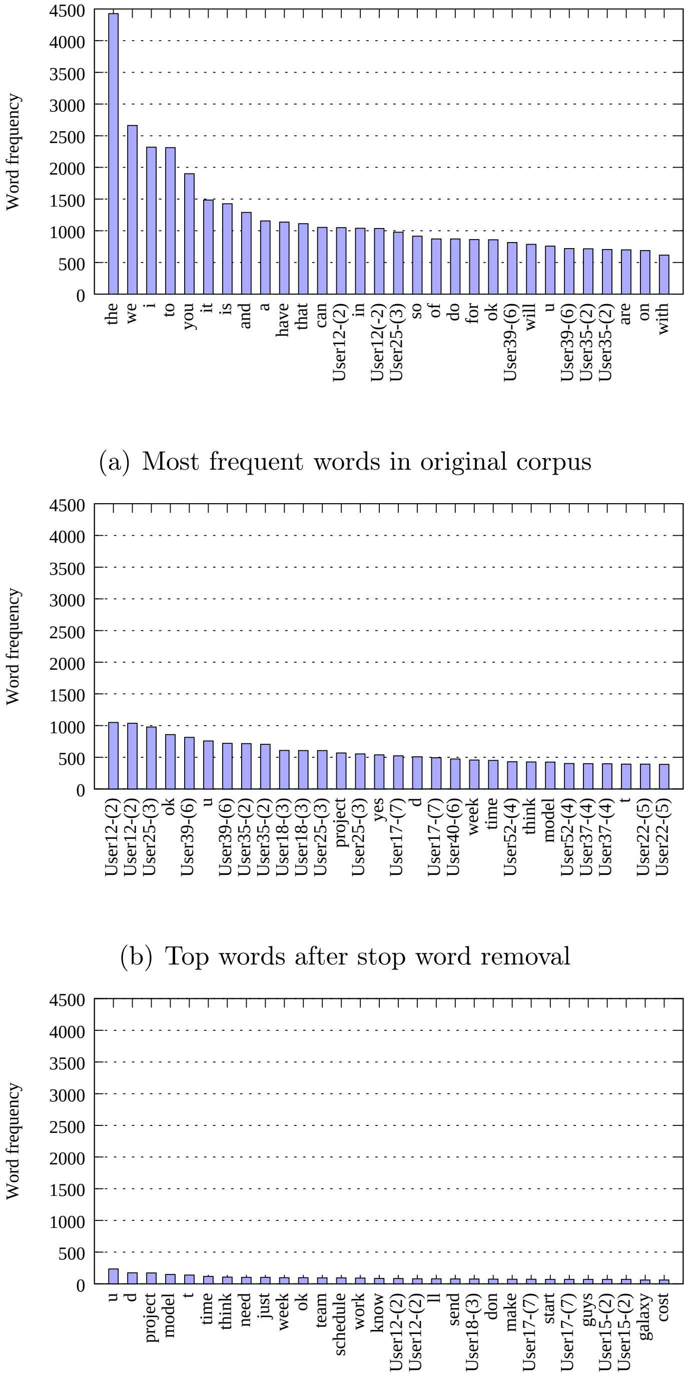
Algorithm 1 Evaluating the effect of pre-processing to the topic model discovered by LDA.

```
$i = 0$
Define a set of filters $F = \{f_1, \ldots, f_n\}$
**repeat**
    $\mathcal{D}' \leftarrow$ Apply filter $f_i$ to corpus $\mathcal{D}$
    $\mathcal{G} \leftarrow \emptyset$
    Create vocabularity $A$ from the corpus $\mathcal{D}'$ //Map 'word' to *term index*
    **for all** *documents* in $\mathcal{D}$ as *document* **do**
        **for all** *words* in *document* as *word* **do**
            $c \leftarrow$ Count *word* occurrences in *document*
            Replace *word* with string $A[word]{:}c$ in *document*
        **end for**
        Append *document* at the end of corpus $\mathcal{G}$
    **end for**
    $T \leftarrow$ Compute topic model on corpus $\mathcal{G}$
    $g \leftarrow$ Evaluate the goodness of $T$
    $i = i + 1$
**until** $g$ is satisfactory or $i = |F|$
```

most frequent terms in the chat corpus before filtering are shown in Figure 8(a)[4]. Once the stop words have been removed from the corpus the most frequent terms are mostly those found in the usernames of the members as shown in Figure 8(b).

After the removal of the stop words we are left with a slightly reduced set of data but lots of words still appear in the corpus only once. We remove these junk words as they are non-informative after their appearance. There are 2891 such words. The text corpus is still rather large and inspection on the data reveals that many of the short messages are quite uninformative. Following this thought we try many alternative ways to construct document structures from the utterances in the data that don't prove to be succesful. We constructed documents based on the closeness of timestamps in chat utterances (to find discussions), based on the date or time period at which the utterance took place and even tried per user models. None of these yield the results we were hoping for – however we reveal the importance of the pre-processing in a crude way. We are no longer mining the actual data but pre-

[4] Usernames have been replaced by an anonymous User-(group) tag to protect the identities of the students – all usernames were two-part names so the same user can appear twice in the lists.



(a) Most frequent words in original corpus

(b) Top words after stop word removal

(c) Top words in sentences longer or equal than nine words

Figure 8: Frequency of the top words in different phases of filtering. In 8(a) the whole corpus is processed and all chat mass is available. When stop words (very frequent in most text corpora) are removed the frequency of the top words is significantly lower as seen in 8(b). Filtering utterances of nine words or more produces quite even distribution in word frequencies even among the top words (8(c)).

processing it in to a form that would produce the results that we are expecting. In no way is this an uncommon approach in science but here the actual pre-processing becomes the essence – the pre-processing is not just a trivial phase that we would be casually mentioning here. We could try to bootstrap important utterances or play with weights on important words, utterances or whole discussions – approaches that might yield results we expect but at the same time would be very specific to our problem domain. This would be the wrong approach. Instead here should be an inclusive study on how to pre-process chat data and how it affects the various algorithms in the domain of text or chat mining. As to demonstrate the twisted approach we filter the data by only looking at the utterances that have seven or more words. Each utterance begins with two terms that are the username of the participant so to be exact we are observing chat lines of nine terms or more. See Figure 8(c) for how top word histogram changes. At this point the data has been reduced from the original 13529 chat utterances to a mere 1173 – from a statistical point of view this could be perfectly normal because most of the discussion is likely to be uninformative and the small sample might very well capture the very essence of what we are looking for. Yet we underline the fact that we are looking at about 9 % of the original chat data so we cannot ignore the fact that much information is probably lost in the seemingly unimportant utterances. The topics that we find in the nine percentage of the data are shown in Table 5.

We've already categorized utterances of nine or more words as meaningful in the process of filtering. Thus users who appear in the representative words in the six topics LDA was set to find are likely to utter meaningful sentences. Although the user group recognition did not succeed we confirm that some meaningful topics could be found in the chat data. In Table 6 we have labeled the topics based on unique words that appear in the top-30 representative words in each topic. There is no coherent distribution in the user names among the topics – members of all groups appear in each topic.

Although the approach wasn't succesful it reveals information about how to approach chat data mining and topic models. Reason why the method wasn't succesful can be attributed to the similar vocabularity between the groups. The topics that are found are generic over all groups instead of group specific. Even if some vocabularity in the groups is distinctively their own it is still a minor part in the surrounding discussion. In case the vocabularities were completely separate – as is between different languages – we could expect to identify topics of each language. User names would then appear together often in the topics of their own language

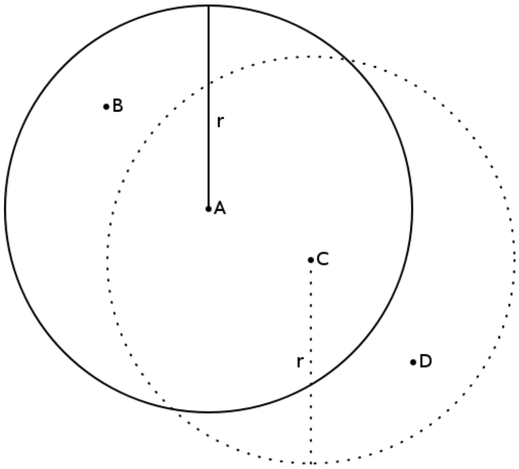


Figure 9: Link threshold distance limits the range over which chat messages can be "heard". Here node A can shout over to nodes B and C which are withing the threshold distance r. Node D is not in the range of A and thus communication between these nodes is impossible. Node C would be able to shout over to A and D and thus strenghten the community bondage with them.

which could be used to classify same language users in the same group.

In future research a better set of pre-processing steps should be evaluated. It would be also interesting to combine topic models and social networks based on spatial information for chat data mining.

6.2 Experiences with Walktrap

On this approach to find groups we employ information about where and when group members utter something (see Tables 3, 4). Notable difference to our previous attempt is that we don't consider chat contents, only individual chat utterance events. We approach the problem by using an intuitive idea. People who communicate with each other need to be a) in the same place, b) at the same time. We construct a query to return links from speaker to listeners. Each such utterance adds link weight between each speaker-listener pair. To meet our intuitive idea the query needs to limit the distance over which utterances can be heard and require that the listeners are within that threshold distance at the time of the utterance (see Figure 9).

Due to the recording conditions the utterances and spatial coordinates weren't recorded in the same database with the same timestamps. The spatial coordinates

Table 5: The 30 most representative words for each topic as discovered by LDA.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**	**Topic 6**
u	project	week	User12-(2)	d	u
User12-(2)	t	just	User12-(2)	u	User37-(4)
User37-(4)	don	User49-(3)	model	User18-(3)	User37-(4)
schedule	User25-(3)	use	start	ll	th
User52-(4)	User39-(6)	time	d	User18-(3)	User15-(2)
User9-(3)	User39-(6)	User49-(3)	User28-(7)	User21-(2)	User15-(2)
User25-(3)	User25-(3)	ok	User28-(7)	User21-(2)	live
User25-(3)	User13-(7)	User18-(3)	User40-(6)	User26-(2)	User19-(2)
User9-(3)	User13-(7)	User18-(3)	User40-(6)	model	ur
did	think	User17-(7)	just	v	sync
User32-(4)	know	User17-(7)	drawings	User26-(2)	folder
think	team	User32-(4)	cost	User13-(7)	User22-(5)
User32-(4)	really	team	work	send	User19-(2)
time	mean	u	estimate	User13-(7)	email
User15-(2)	need	portal	need	drawing	User22-(5)
User15-(2)	make	guys	like	User55-(3)	n
know	guys	need	room	ok	team
guys	work	work	finish	details	r
model	time	User41-(6)	think	ya	new
ok	User17-(7)	let	schedule	need	send
User39-(6)	User17-(7)	meeting	User35-(2)	User55-(3)	id
User39-(6)	working	meet	floor	prepare	d
User40-(6)	simvision	good	drawing	graddus	mail
columbia	manager	report	yeah	time	wat
User22-(5)	file	write	User35-(2)	couple	User34-(4)
activities	sure	User32-(4)	possible	User51-(5)	guess
need	able	think	mail	building	com
User28-(7)	bridge	know	ok	working	report
work	right	hour	doodle	make	hav
team	lot	User36-(5)	students	floors	fan

Table 6: Suggested labels for each of the six topics based on hand picked unique words among the top-30 words in each topic.

Topic	**Suggested label**	**Hand picked top words**
Topic 1	Schedule	schedule, model, columbia, activities
Topic 2	Project	project, manager, file, simvision, bridge
Topic 3	Meeting / reports	week, portal, meeting, meet, report, write
Topic 4	Planning	model, start, drawings, cost, estimate, room, finish, schedule, mail, doodle, students
Topic 5	Building	model, drawing, details, prepare, building, floors
Topic 6	Sharing	live, sync, folder, email, send, mail, report

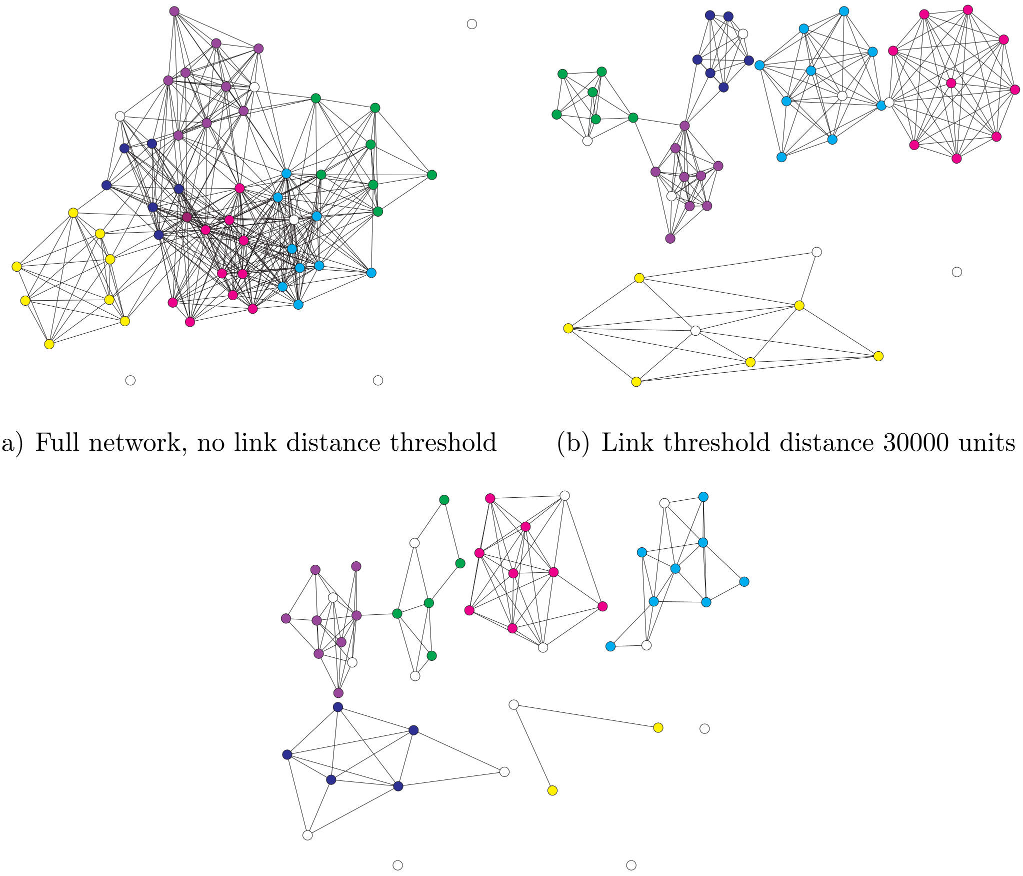
were recorded in 10 second intervals whereas chat utterances were recorded on event basis – every utterance was recorded just as it happened with the exact timestamp. The databases could be merged by assigning the spatial coordinate of the speaking user to the closest timestamp of each utterance. This merging produces a database in which each utterance is assigned to a certain coordinate in the VE. Merging causes an error margin on the exact location of the speaker. If the spatial locations were recorded on each utterance event we would know the spatial locations exactly with no error margin at all.

It is then possible to efficiently execute queries in which speaker is assigned as the origin (nodes A and C in Figure 9) at the time of the utterance and close spatial-temporal neighbors (nodes B and D respectively) are found within the given threshold distance in the spatial database. Maximum inaccuracy in this query is the recording interval of the spatial data – in this case the distance which a user can move in five seconds. It is worth mentioning that often when people are engulfed in discussion they hardly move at all. In fact, this behaviour can make it challenging to mine subtle clusters with kernel estimation because few long and intense discussions make the rest of the data look flat and uninteresting.

In our research data there exist multiple ways to build the network – it is important to build the network so that community finding algorithms can be fully exploited. We found that counting link strenghts by speaker-listener relationship captures community relationships reasonably well. Our query returns a list of speakers, utterances and listeners – the network is build by summing these speaker-listener relationships. The resulting network is a weighted undirected graph. Other viable ways to build the network could involve closeness of the discussing pair or amount of people near the speaker. More sophisticated approaches could grade the importance of the utterance – and as such the weight of the link – with topic modelling methods.

Figure 10 displays how the network structure changes depending on the link threshold distance. Nodes are colored according to their group. White nodes do belong to some group but because they did not utter anything that anyone could hear they aren't assigned any color. The colors indicate how we would like the graph to be partitioned – each node would be assigned to the same group with the other members of the same group. However that is the human perception of the groups because we already know how the members were assigned in the groups. Walktrap approaches the problem from mathematically founded angle by optimizing modularity of the partition. Although this may produce a different partition than that we expect it is not viable to say that it is wrong. Because all partitions of a network can be measured with the modularity measure we can measure the modularity of the original group assignments and those that Walktrap finds. The goodness comparison can be done by comparing the modularity values of the partitions. The higher the modularity the better the partition.

Graph in Figure 10(a) represents a network in which we haven't used a threshold value to limit the distance over which links from speaker to listener are connected. Only requirement for link to be created is that the two persons are online at the same time and another one speaks. From this graph it is visually difficult to recognize communities. By defining a threshold distance of 30000 units (approximately the width of a room) we get a much more refined network, as seen in Figure 10(b). Here it is possible recognize the 6 study groups although three of the groups are still connected to each other. However if we set the threshold value too low we might lose links even between group members. Too low estimate for link threshold distance causes loss of information and badly connected network (10(c)). Here we have approximated good link threshold distance by manually trying different values. Other estimation methods could be used instead. One way to estimate the threshold distance would be to measure average distance to a closest listener. This distance would probably be a quite good approximation about how close listeners usually are to a speaker. Then multiply that distance with a fixed multiplier value. The larger the multiplier is the more complex the network becomes as more users fall within the radius. The more users are within the radius the more information is captured in the network. The tradeoff is between the amount of information and the efficiency of the community finding algorithm.



(a) Full network, no link distance threshold

(b) Link threshold distance 30000 units

(c) Link threshold distance 15000 units

Figure 10: Effect of link threshold distance in network structure. In figure 10(a) all the communication links in the network are mapped as if everyone heard what anyone said as long as they were online at the same time. In figure 10(b) we've set the maximum distance over which a discussion can be heard to 30000 units (approximately a room width) – six different communities begin to show in the visualized network. Setting link threshold distance too low will result in a loss of important network structure as can be seen in figure 10(c). Nodes are colored according to their group – white nodes did not utter anything that anyone could hear.

We then run the Walktrap algorithm on the full network (Figure 10(a)) and the one with threshold limit of 30000 units (Figure 10(b)). Walktrap finds the correct groups with path lengths $t = 3$ and $t = 4$. This result is not the optimal as measured by the modularity though – the partition only yields modularity $Q = 0.615239$. Other path lenghts ($t \in \{2, 5, 6, 7, 8, 20\}$) that were tested result in a partition with a slightly higher modularity of $Q = 0.618103$ – this is a known problem of parametrized algorithms that may provide different (better) results with different parameters. Increase in modularity is possible by merging one of the administrators (only one who was active, uttering four messages in the beginning of spatial data recording period) from group 1 to group 4. Later he was just idling far outside the conference hall just to record the data without intefering in the interactions of the groups. It would be reasonable to remove edges from the network that are below a certain threshold value. There is no "hidden information" in utterances that someone happens to accidentally intercept and thus create a link between the pair. These links however can unnecessarily increase the complexity of the network. In our network a good link strenght threshold could be around 20 – although this was not further tested. In Figure 10(c) the distance threshold of 15000 units is clearly too strict. Although some group members are still internally connected there are many internal links missing. The reduction in network degree distribution loses important information about group structures.

Running time of the algorithm was 0.281 seconds on the full network and 0.25 seconds on the 30000 unit threshold network. This comparison is not completely accurate because reading the network and printing out the results takes considerable share of the total time. On larger networks these variables could be neglected. Regardless the overall time to find the optimal partition is very small.

7 Conclusions

We've first introduced the history of distance education (DE) and virtual learning environments. The history of DE extends from the mid 19th century continuing the search for more efficient ways to teach and learn. Collaborative virtual environments (CVEs) can be categorized as one form of distance education although there is not necessarily need for a teacher as users themselves are set to solve a problem before entering the VE altogether. Users in the VE themselves find out the best way to approach the given problem. Way to the virtual worlds was paved by text based environments that were limited by both amount of users and inconvenience of sharing content. As technology has evolved more comprehensive solutions have become available – yet the first virtual environments have come into existence over 150 years ago between Morse code operators. That is the trend that is still ongoing.

Then the main concepts of interaction in VEs were introduced. These user model concepts are the basis in how VEs fundamentally work both technically and as a social environment. Each VE has a slightly different implementation of the user model which usually directs people how they act, interact and behave in each VE. There are CVEs that are solely built for learning, those that are for social leisure, for short detachment and enjoyment from daily routines, for long time building of an online character among a vast mix of VEs that involve these characteristics. The invention and evolution of CVEs cannot be attributed to any single authority or domain. We can however certainly discern that the VEs came to be long before they were widely recognized as learning environments in any branch of scientific research.

Community finding problem was introduced in Chapter 3. We divide the problem to two separate concepts: finding dense subgroups in a graph and finding discriminative topics in document corpus. Dense subgroup finding problem is closely related to a graph partitioning problem. The aim is to find a good division of a given graph to smaller groups that form the community. Modularity measure was introduced that can be used to measure communityness of a network partition. The higher the modularity is the better the community structure is. Topic modelling is used to find distinct topics in large document collection quickly. The topic model can be then used to classify other documents in the topics that were discovered in the learning set. However we were just interested in the topic model that can be found in our chat corpus. We chose LDA topic modelling algorithm that has shown good results compared to earlier algorithms.

We then analyzed a collection of data from a study group in Second Life. We evaluated two distinctly different methods to find groups in the data. First we employed a topic modelling algorithm on the premise that each group would develop a vocabularity distinctly of their own – and thus groups and group members would be identifiable through the topics they discuss. This approach did not prove to be succesful but interesting findings were discovered in the progress. Importance of the pre-processing became prominent as the results of LDA topic modelling algorithm varied greatly depending on the pre-processing steps. Another discovery was that identifiable topics can be found in chat data. It might very well be plausible to identify topic or topics that a single user or a user group is discussing in real time. Through topic identification it'd be possible to build filtering applications to easily find interesting discussions.

By using spatial information combined with chat data we were able to find groups in the social network with an algorithm called Walktrap. These groups can be found fast. The algorithm finds dense clusters on an undirected graph that is built upon chat relationships between interacting users. We were able to reduce the network size by defining a threshold distance over which chat utterances could be heard, as if each user in the VE had a voice that can only reach a certain distance. Interesting improvement ideas revolve around how to evaluate the link strenght between each speaker-listener pair. Trivial approaches could weight the link more when the listener is close to the speaker or there is only one or few listeners present. More sophisticated approach would be to evaluate *what* the speaker is saying and weight the link based on that estimate. This approach would combine the topic modelling and the spatial modelling so that each link would have a semantic meaning. In turn this would yield a semantic network in which two users might have connections at multiple different semantic levels. For instance work colleagues are likely to discuss work related issues while friends discuss about leisure time activities. The relationship would not need to be limited to a single classification – for example a work colleague might be one's friend as well. As documents are classified with a certain degree of belief to a topic we might classify relationships in the same fashion.

Future research might involve recognition of topics of discussions (in communities) and guiding the users in the world based on their own interests. A far-reaching idea is that people could join a virtual world and be guided to a place where experts of their domain of interest are gathered. An algorithm could find the communities and what the users in them discuss about and thus identify meaningful places. In social evolution some places in the VE would gradually become centralized domains of certain interests. These domains would become inherently attractive to people who are interested in the subject and thus solidify the existence of such a place. Whether this development actually happens will remain to be seen and depends not only on the technology but the popularity of the existing and coming virtual worlds.

References

ACEC07 Arango, F., Chang, C., Esche, S. K. and Chassapis, C., A scenario for collaborative learning in virtual engineering laboratories. *Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference*, 2007, pages F3G–7 – F3G–12.

BDPd^{+}90 Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C. and Mercer, R. L., Class-based n-gram models of natural language. *Computational Linguistics*, 18, pages 18–4.

BF93 Benford, S. and Fahlén, L., A spatial model of interaction in large virtual environments. *Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work*, 1993, pages 109 – 124.

BG98 Becker, B. and G., M., Social conventions in collaborative virtual environments. *Proceedings of CVE'98*, 1998, pages 47–55.

BGLL08 Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E., Fast unfolding of communities in large networks.

BH Bower, B. and Hardy, K., From correspondence to cyberspace: Changes and challenges in distance education.

BL07 Blei, D. M. and Lafferty, J. D., A correlated topic model of science, Aug 2007.

BNJ03 Blei, D. M., Ng, A. Y. and Jordan, M. I., Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, pages 993–1022.

BRSP00 Bernard, R., Rubacava, B. and St-Pierre, D., *Collaborative online distance learning: Issues for future practice and research*, volume 21. 2000.

Cai05 Cai, J., A social interaction analysis methodology for improving e-collaboration over the internet. *Electronic Commerce Research and Applications*, 4,2(2005), pages 85–99.

DBA89 Dunn, R., Beaudy, J. and A., K., *Survey of research on learning styles.* Number 46. 1989.

DDF$^+$90 Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R., Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, pages 391–407.

Dun98 Dunbar, R., *Grooming, Gossip, and the Evolution of Language*. Harvard University Press, October 1998.

FLM04 Fortunato, S., Latora, V. and Marchiori, M., Method to find community structures based on information centrality. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70,5 Pt 2(2004).

FM07 Fortunato, S. and M., B., Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104,1(2007), pages 36–41.

HBBS08 Hicks, I. V., Balasundaram, B., Butenko, S. and Sachdeva, S., Clique relaxations in social network analysis: The maximum k-plex problem, 2008.

Hof99 Hofmann, T., Probabilistic latent semantic indexing. *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 1999, ACM, pages 50–57.

Hol Holmberg, B., The evolution of the character and practice of distance education.

Kol81 Kolb, D., *Learning Styles and disciplinary differences*. A. W. Chickering (Ed.), San Francisco: Jossey-Bass, 1981.

Lev09 Levine, N., *CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations*. 2009.

MK89 Mason, R. and Kaye, A., Mindweave: communication, computers and distance education. In *Mindweave: communication, computers and distance education*, Mason, R. and Kaye, A., editors, Pergamon Press, Oxford, 1989, pages 85–87.

MSLC01 McPherson, M., Smith-Lovin, L. and Cook, J. M., Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27,1(2001), pages 415–444.

New04 Newman, M. E. J., Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, page 066133. URL `doi:10.1103/PhysRevE.69.066133`.

NG03 Newman, M. E. J. and Girvan, M., Finding and evaluating community structure in networks, August 2003.

NP03 Newman, M. E. J. and Park, J., Why social networks are different from other types of networks. *Phys. Rev. E*, 68,3(2003), page 036122.

NRT04 Novak, J., Raghavan, P. and Tomkins, A., Anti-aliasing on the web. *WWW '04: Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA, 2004, ACM, pages 30–39.

PL05 Pons, P. and Latapy, M., Computing communities in large networks using random walks. 2005, pages 284–293.

Pre01 Preece, J., Sociability and usability in online communities: Determining and measuring success.

SB88 Salton, G. and Buckley, C., Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 1988, pages 513–523.

Sco88 Scott, J., Social network analysis. *Sociology*, 22,1(1988), pages 109–127.

SHT06 Schroeder, R., Heldal, I. and Tromp, J., The usability of collaborative virtual environments and methods for the analysis of interaction. *Presence: Teleoper. Virtual Environ.*, 15,6(2006), pages 655–667.

Smi99 Smith, M. A., *Communities in Cyberspace.* Routledge, first edition, February 1999.

SN02 S., R. and N., N., Collaborative virtual environments to support communication and community in internet-based distance education. *Journal of Information Technology Education*, 1,3(2002), pages 201–211.

Sta00 Standage, T., *The Victorian Internet.* 2000.

Ste92 Steuer, J., Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42,4(1992), pages 73–93.

TSW03 Tromp, J., Steed, A. and Wilson, J. R., Systematic usability evaluation and design issues for collaborative virtual environments. *Presence: Teleoper. Virtual Environ.*, 12,3(2003), pages 241–267.

TT04 Tuulos, V. and Tirri, H., Combining topic models and social networks for chat data mining. *Web Intelligence, IEEE / WIC / ACM International Conference on*, 0, pages 206–213.

WdVdJ07 Westerveld, T., de Vries, A. and de Jong, F., Generative probabilistic models. *Multimedia Retrieval*, Data-Centric Systems and Applications. Springer Berlin Heidelberg, 2007, pages 177–198.

Wik09 Wikipedia, Graph partition — wikipedia, the free encyclopedia, 2009.