Discovering Virtual Interest Groups across Chat Rooms

Hua Li, Jeff Lau and Rafael Alonso
SAIC, Inc. 1710 SAIC Drive, McLean, VA 22102, U.S.A.

Keywords: User Modeling, Machine Learning, Virtual Interest Group, Chat, XMPP, IRC, Reinforcement Learning.

Abstract: Chat has becoming an increasingly popular communication tool in our everyday life. When the number of related concurrent chat rooms gets large, tracking them 24x7 becomes very difficult. To address this research problem, we have developed VIGIR (Virtual Interest Group & Information Recommender), a tool for automatic chat room monitoring. The tool builds adaptive interest models for chat users, which are used to provide a number of personalized services including finding virtual interest groups (VIGs) for chat users. Dynamic identification of the VIG addresses the distributed user collaboration challenge, which is acute problem especially in military operations. VIGIR extends our prior work in user interest modeling into the domain of real-time text-based communications. We have evaluated the effectiveness of VIGIR in two studies. The first is a user-centred evaluation where we have achieved a precision at 60% and recall at 80% for VIG identification. In the second study using military chat data, we have demonstrated an average precision of 45% to 50%. In addition, we have shown that the precision for predicting VIG increases over time as more data become available.

1 INTRODUCTION

In recent years, chat has becoming an increasingly important communication tool in civilian life as well as in military operations. In operational settings, multiple concurrent chat rooms are typically setup to support different functions (e.g. “intel”, “fires”, and “sustainment” in JFCOM data described below) and units (e.g. “IBCT”, “MEB”, “CDR”, and “CAV” in JFCOM data). Chat users are often unaware of the situations happening in rooms they are not participating. To increase their situation awareness, which is critical for the effectiveness of soldiers in the battlefield, an operator is usually necessary to manually watch all the chat rooms and direct snippets of the chat stream to relevant user. They can also connect a user with certain information needs to another with that information. Unfortunately, when the number of related concurrent chat rooms increase, it becomes next to impossible for an operator to manually monitor all the rooms 24x7. For example, in the JFCOM data, there are about 100 chat rooms and at least 10 active rooms at any given time.

To address this problem, we have developed VIGIR (Virtual Interest Group & Information Recommender), a tool for automatic chat room monitoring (Figure 1). The tool builds adaptive interest models for chat users, which are used to provide a number of personalized services such as recommending relevant chat snippets or virtual interest groups (VIGs) for the users. In this paper, we focus our research on VIG, which addresses the distributed warfighter collaboration challenge, because automatic and dynamic identification of VIGs can assist the warfighter in finding collaborators or experts faster and across organizational, geographical or temporal boundaries.

Our approach is interesting in several aspects when compared with existing research on social matching, expertise finding, online communities, and awareness systems (Schleyer et al., 2012) and (Terveen and McDonald, 2005). The first interesting characteristic is that user interest models underlying the VIG are dynamically built on the content of the chat messages. In this way, total strangers from different chat rooms may be found in a VIG because they have similar interests or information needs, not because they directly or indirectly communicate with each other, e.g. through personal social networks. The second is the explicit representation of the user interest model, which may consist of weighted terms, named entities, and topics as time-based facets. The transparent nature of the user

Li H., Lau J. and Alonso R..
Discovering Virtual Interest Groups across Chat Rooms.
DOI: 10.5220/0004131501520157
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2012), pages 152-157
ISBN: 978-989-8565-31-0

model makes it easy to interpret the model and thus gain the user's trust of the system. Also interesting is that VIG identified is serves to augment situation awareness in terms of informing the user who to turn to seek further information on topics of common interests. Finally, the VIGIR system discovers VIG across concurrent chat rooms, and more importantly, with member that the user does not know.

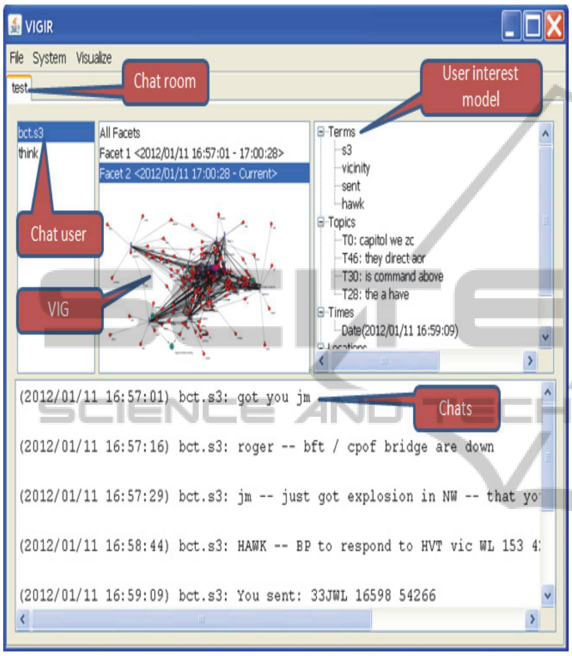

Figure 1: A screenshot of the chat GUI in the VIGIR prototype implementation. Real-time chats from multiple chat rooms are monitored. Adaptive user interest models and dynamic VIGs are generated for chat users.

2 VIGIR PROTOTYPE SYSTEM

The VIGIR prototype (Figure 2) has three logic modules: Input, Core, and Interfaces module. We describe them separately below.

2.1 The Input Module

Internet Relay Chat (IRC) or Extensible Messaging and Presence Protocol (XMPP) chat traffic is the main external input to the system. The VIGIR system can also take archived chat or other type of communications such as email or voice transcripts. In addition, the system is capable of accessing the Web or databases to get input for the knowledge recommender service discussed below.

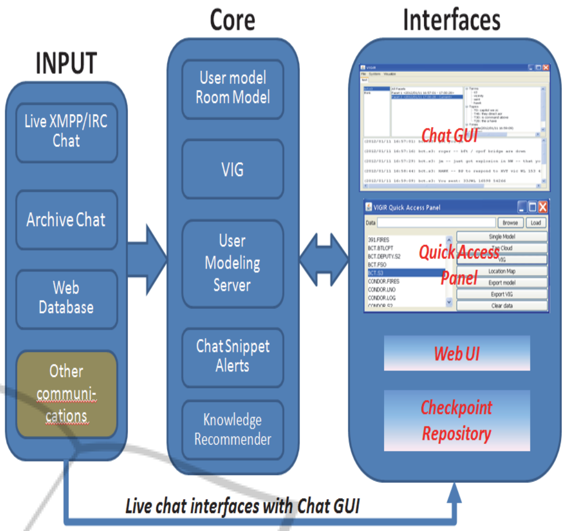

Figure 2: The current VIGIR prototype system architecture.

2.2 The Core Module

The core module contains the User Modeling Server (UMS) and several user model enabled services. The UMS processes the chat traffic from multiple chat rooms in real time to build and adapt user models for chat participants. These models capture chat users' interests, information needs, and expertise. UMS uses a modified version of the Reinforcement and Aging Modeling Algorithm (RAMA) to model the user's interests. RAMA is based on our previous research (Alonso and Li, 2010; 2005); (Alonso et al., 2003). The algorithm is driven by user events, also known as a relevance feedback. In the case of a chat system, each individual chat message is a positive user event and implicitly contains elements that indicate user's interests. The details of the RAMA algorithm can be found elsewhere (Alonso and Li, 2010). For completeness we present the pseudo-code for the RAMA algorithm is shown below.

1) Extract the raw text content from the user event.

2) Pre-process the raw text if necessary.

3) Using NLP tools to extract concepts from the event. The concepts are interest elements which may take the form of terms, named entities (e.g. persons, places, time mentions, or organizations), and topics.

4) Age the current model by applying a forgetting function to all concepts in current model. The **attenuation weight** is a system parameter that controls the rate at which the weight of concepts will decay.

5) If a concept from the event already exists in the model, it will be positively or negatively reinforced

depending on the nature of the user event. Specifically, the concept's weight in the model will either be increased for a positive event or decreased for a negative event. The **reinforcement weight** is a system parameter that controls the rate of change in the weight during reinforcement.

6) If the feedback is positive, insert top N new concepts from the feedback into the current model with a default weight modulated by their relevancy (e.g. term frequency).

7) The user model is divided into facets based on pre-specified time interval or number of user events. When the time interval expires or the specified number of events has been processed, the current facet is retired and a new facet will be created. The **facet size** refers to the number of events in a facet. It's also a system parameter that may impact the effectiveness of the system.

The UMS builds a user interest model for each known chat user using the RAMA algorithm. It can also build an information model for each chat room using the same algorithm. Thus the room model can be regarded as a team model for all participants of the room. Due to space limitation, we will not go into details about room model.

The chat user models form the basis for several user-tailored information recommendation services. We discuss these services next.

Based on user interest models, VIG service is to identify and recommend to a chat user a VIG, other system users with similar interest, information needs, and expertise. VIGs can facilitate information sharing and collaboration among warfighters because they explicitly suggest to a user other like-minded people they may talk to.

The VIG identification algorithm works by comparing different facets of the user model (Alonso & Li, 2010). The more facets are similar between two users, the more similar they are. If none of the facets are similar, the two users are not alike at all. Cosine similarity, commonly used in the vector space model, may be employed to compare the similarity of two model facets (Salton et al. 1975). The **VIG size** is a system parameter and refers to the number of member users to include in a VIG during computation.

The proactive Knowledge Recommender (aka KnR) is another service enabled by the user models. The system can automatically generate search queries using the model and retrieve relevant documents from the Web or databases on the user's behalf. Also powered by the user models is the Chat Snippet Alerts service, which monitors concurrent chat rooms and alerts the users with chat messages that contain relevant events.

2.3 The Interfaces Module

The Web user interface (UI) is a simple Java® (Oracle America, Inc.) servlet-based client graphical user interface (GUI) that displays the user models, VIGs, and routed chat snippets, all of which may have also optionally persisted locally in the Checkpoint Repository. The Quick Access Panel allows examination of user models and VIGs through extensive visualizations.

The Chat GUI interface is used to manage live XMPP or IRC chats (Figure 1). Chat servers and room connections are configured here. Real-time chats from multiple chat rooms are monitored. Adaptive user interest models and dynamic VIGs are generated for chat users. The user models and VIGs can be visualized with different graph layouts.

The Web UI serves as a demonstration tool showing much of the functionality of the system. It works like a simple web page but provides access to the list of user models that are being stored in the database. It also allows one to look at a specific user model, construct a VIG, or inspect chat snippets that have been routed to users, and learn more about the system itself.

The Checkpoint Repository periodically saves the session information for each user for offline assessment including the user model and the VIG.

The Quick Access Panel is a graphical interface for examining the products (i.e., user models and VIGs) generated and stored by the system core. It provides a variety of visualizations for these products.

3 EVALUATION STUDIES

To assess the performance of the VIG functionality, we designed a user-centred study and ran an experiment a using real operational chat data set. In this section, we describe them in detail.

3.1 User-centered Evaluation

For this study, we had an intern (political science major) to generate a chat stream. She played a total of six roles (Adams, Adrianne, Alycia, Charlotte, Princess Adrianne, and Rachelle). The chats were focused on three topics listed below.

1) T1 (BP oil spill): The impact of the BP oil spill

2) T2 (Afghanistan): The implications of the leaked Afghan reports released by WikiLeaks

3) T3 (Obama): The Obama administration handling of the BP oil spill and the Afghanistan war document leak

Note that T3 is intended to be related to both T1 and T2. Each topic is discussed by two role players (BP oil spill by Princess Adrianne and Rachelle; Afghanistan by Adams and Adrianne; and Obama by Alycia and Charlotte).

The experiment protocol for the chat stream generation is as follows.

1) Start mIRC, an IRC client available at http://www.mirc.com/, and connect to the specified IRC server

2) Set chat message logging to true

3) Join the specified chat room as one of the six users at a time

4) Write required number of chat messages: 30 messages on the assigned topic and each message is a typical short chat, 5-20 words

5) Leave the chat room

6) Repeat for all six players. If we have two subjects chat with each other, then repeat for each topic.

7) Generate VIG for each user

8) Evaluate system output Compute performance metrics: precision and recall for VIG and Snippets Routing

The VIG ground truth was known by design. For a given player's VIG, his or her VIG members should include players discussing the same topic or related topics, e.g., Adams' VIG should have Adrianne, Alycia and Charlotte.

We added noise data in this study in order to increase the difficulty of identifying the VIGs. We preloaded two chat archives into the system:

1) DWE Chat from APG

2) NIST Chat Data

The DWE chat data were received from the THINK program office on June 9, 2009. The training was held Feb. 4-8 and the actual exercise was held Feb. 11-15. The chat data are in one 362KB text file with a total of 69 users and 1,918 chat messages.

The NIST chat data set was acquired on July 23, 2009, from the National Institute of Standards and Technology (NIST). It was generated from a NIST study (O'Connell et al., 2009). The goal of the study was to investigate how people interact with each other in a virtual gaming world to solve the same set of puzzles. The data consist of four text files, one for each experimental condition. File size ranges from 12KB to 17 KB. This archive contains a total of 16 users and 927 messages.

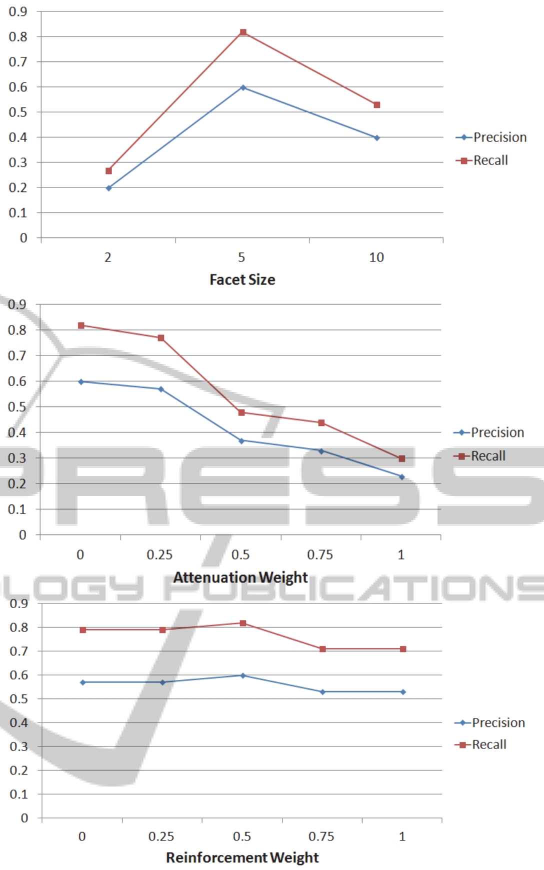

Figure 3: The effect of facet size, attenuation weight, and reinforcement weight on VIG performance.

How difficult is the VIG identification task with the noise data? Let's compute the probability of getting a VIG right by chance. With noise, we now have a grand total of (6 + 69 + 16) or 91 users. The chance of getting 1, 2, 3, 4, or 5 VIG members correct is 0.011, 0.00025, 8.5E-6, 3.9E-7, or 2.2E-8, respectively!

In order to express the performance metrics, we define the following terms:

- True VIG: one defined by the ground truth
- Found VIG: one generated by the UMS
- A VIG Owner: the user whose VIG is created
- Hit: a found VIG member belongs to the true VIG of the same owner
- Total Answer: the size of the found VIG
- Total Correct Answer: true VIG size

Precision and recall are calculated as follows:

$$Precision = \#Hits/(TotalAnswer)$$
$$Recall = \#Hits/(TotalCorrectAnswer)$$

We assessed the effects of three underlying user modeling parameters, namely facet size, attenuation weight, and the reinforcement weight on the VIG identification precision and recall (**Figure 3**). Facet size refers to the number of messages we use to build each time-based facet in the user model. We found that for the current study, a facet size of 5 gave the highest precision at 60% and the best recall at 80% (Figure 3, top panel).

We found that a very slow attenuation rate yielded better performance with best precision (60%) and recall (80%) when there was no attenuation (Figure 3, middle panel). This result may be attributed to the short duration nature of the chat stream. There were three 10- to 20-minute sessions, one on each topic.

The effect of reinforcement weight is only modest, with better performance for medium weight range (Figure 3, middle panel). The best precision (60%) and recall (80%) were seen at a reinforcement weight of 0.5.

3.2 Evaluation with JFCOM Chat Data

In this evaluation experiment, we used the JFCOM chat data provided by U.S. Joint Forces Command Joint Futures Laboratory on September 23, 2010. It consists of 14 Excel® (Microsoft Corporation) chat data files and one metadata file, which were generated in an experiment conducted in 2006. The data contains chats from a total of 10 functional user groups and 541 users. The position or role for each user was also defined in the metadata file.

We tested two hypotheses this experiment:

1) The THINK tool can generate VIG with precision that is significantly better than a random predictor.

2) The precision for predicting VIG increases over time when more data are becoming available.

The experiment protocol consisted of these steps:

1) Build ground truth for VIG based on metadata

2) Use VIGIR tool to build VIG for each user at various time points of the exercise

3) Compute precision of VIG at different time points

4) Compare the results with those of a random predictor

The VIGIR tool derived VIG results from the JFCOM chats. A VIG for a user (i.e., VIG owner) refers to other users that share similar interests and information needs as the owner. The members of the VIG may come from different groups and different positions. If we assume that, in general, user's activities are largely dependent on the roles they play, then two users would have similar interests and information needs if they play same or similar roles. Thus one way to evaluate the accuracy of the VIGs is to see if the VIG members have the same role as the VIG owner. To perform this type of evaluation, we needed to establish the ground truth on the roles each user plays. In the metadata file, there are 526 types of positions (see the attached Excel file, sheet "Positions"). Some positions are very similar and probably should be grouped into the same role.

With the help of an in-house military subject matter expert, we had the 526 positions grouped into six distinctive roles (functions): cmd, ops, intel, plans, logistics, and admin.

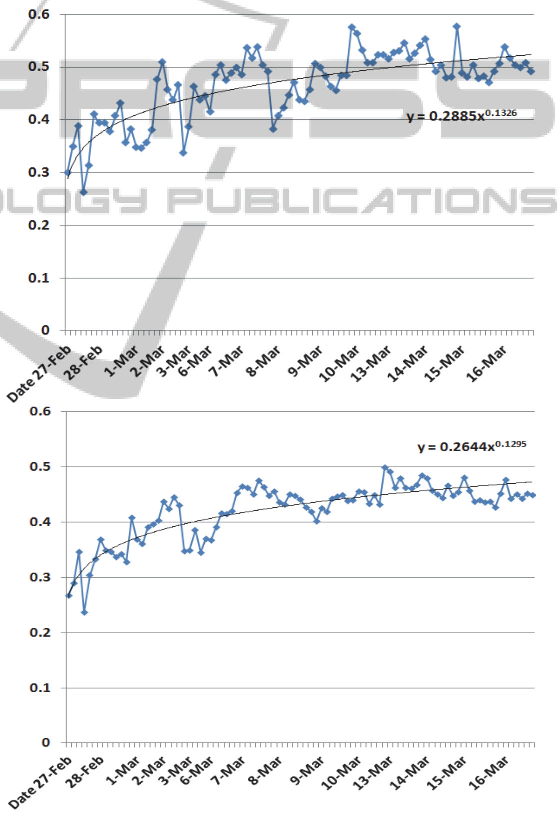

Figure 4: Average VIG precision over time at VIG size of 1 (top) and 3 (bottom).

The JFCOM data spans two weeks of time. We compute the average VIG precision at various time points. To do that, we first compute the VIG precision for each individual user at that time point. The VIG precision of all users known at that time point was averaged to get the average VIG precision at that time point.

The results of the average VIG precision are shown in Figure 4. With VIG size of 1, the average

VIG precision across about 108 users (users join and leave at various times, this is an average number of users per time point) gradually increased from around 30% on Feb. 27 to about 50% on March 26 (top panel). The same trend held with a VIG size of 3 (bottom panel). The average precision gradually increased from about 25% on Feb. 27 to 45% on March 16. These results seem to support the hypothesis that the precision for predicting VIG increases over time as more data become available.

To get some perspective on the performance, let's calculate the chance of correctly getting VIG. With an average of 108 users, the probability of getting 1, 2, or 3 correct VIG members is 0.0093, 0.00018 and 5.0E-6. Thus, regarding hypothesis 1, the VIG performance is significantly better than a random predictor.

4 CONCLUSIONS

This paper addresses the problem of identifying the VIGs across multiple concurrent chat rooms. To address this problem, we have developed VIGIR tool for automatic chat room monitoring. The tool builds adaptive interest models for chat users, which are used to provide a number of personalized services including finding VIGs for chat users. We have evaluated the effectiveness of VIGIR in two studies. The first is a user-centred evaluation where we have achieved a precision at 60% and recall at 80% for VIG identification. In the second study using military chat data, we have demonstrated an average precision of 45% to 50%.

There are a couple areas for future research, including incorporating other user communications and enhancing user models with themes or topics. Besides chat traffic, warfighters may use many other forms of text discourse, such as emails, to achieve their objectives. Each form of communication may reveal a different aspect of a user's interests, information needs or expertise. The more variety of inputs VIGIR gets the more complete and accurate the models will potentially be. Incorporating topics or discourse themes into user models allow capture user's interests at a more abstract level. The resultant user models will have a mixture of weighted terms and topics.

ACKNOWLEDGEMENTS

This work was supported by the Research Development and Engineering Command (RDECOM) Communications-Electronics Research, Development & Engineering Center (CERDEC) under the auspices of the U.S. Army Research Office Scientific Services Program administered by Battelle. We would also like to thank Teri O'Connell of NIST for providing chat data used in this research.

REFERENCES

Alonso, R. and Li, H.: Incremental user modeling with heterogeneous user behaviors, *International Conference on Knowledge Management and Information Sharing 2010 (KMIS2010)*.

Alonso, R. and Li, H.: Model-guided information discovery for intelligence analysis. *Proceedings of the 14th ACM international conference on Information and knowledge management, ACM,* Bremen, Germany, 2005.

Alonso, R., Bloom, J. A., Li, H. and Basu, C.: An adaptive nearest neighbor search for a parts acquisition ePortal. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM,* Washington, D.C., 2003.

O'Connell, T. A., Grantham, J., Workman, K. and Wong W.: Editor-in-Chief's Corner: Leveraging Game-Playing Skills, Expectations and Behaviors of Digital Natives to Improve Visual Analytic Tools, *Journal of Virtual Worlds Research*, 2(1), ISSN: 1941-8477, April 2009.

Terveen, L. and McDonald, D. W., 2005. Social Matching: A Framework and Research Agenda. *ACM Trans. Comput. Hum. Interact. 12, 3 (Sep. 2005), 401-434.*

Schleyer, T., Brian S. Butler, Mei Song, and Heiko Spallek, 2012. Conceptualizing and advancing research networking systems. *ACM Trans. Comput.-Hum. Interact.* 19, 1(May 2012).