Butterfly: A Conversation-Finding Agent for Internet Relay Chat

Neil W. Van Dyke, Henry Lieberman, Pattie Maes
Media Laboratory
Massachusetts Institute of Technology
20 Ames Street #E15-305
Cambridge, MA 02139, USA
{nwv, lieber, pattie}@media.mit.edu

ABSTRACT

The Internet enables groups of people throughout the world to interact to discuss issues, get assistance, learn, and socialize. However, when there are thousands of loosely defined groups in which a user could potentially participate, the problem becomes finding the groups of most interest. In this paper we focus on the domain of Internet Relay Chat real-time text messaging, and describe a "social butterfly" agent called Butterfly that samples available conversational groups and recommends ones of interest. We discuss Butterfly's motivation, usage, real-world design constraints, implementation, and results. Finally, we introduce work in progress on a multi-agent approach that has grown out of our experience with Butterfly.

Keywords

Agents, Internet, conversation, information filtering

INTRODUCTION

The Internet allows groups of people with similar interests to interact with each other, with little regard to geographic location. Popular group interaction media that are layered atop the Internet substrate include electronic mailing lists, Usenet newsgroups, and real-time text chat systems. We focus in this paper on Internet Relay Chat (IRC) [7], a major real-time textual group messaging system, although the general approach we describe is applicable to other Internet-based group media.

Conversational groups on IRC are defined by *channels* ("chat rooms"), most of which have both regular and participants and drop-in visitors. Users explicitly *join* channels in which they wish to participate, and any message sent to a channel is seen by all users joined to it. Channels are created on demand by any IRC user, and each channel exists until the last participant has left.

There are typically over ten thousand IRC channels at any given moment,[1] each identified by a short string name that often does not give much indication as to its content. Each channel can optionally have a one-sentence *topic* string that states the intent of the channel, although in practice the topic string is often not used for this purpose. There is no hierarchy or organizing mechanism to aid a user in finding channels. Thus, a user interested in a certain topic is reduced to trying likely channel names and manually searching through a list of thousands of channels' names and topic strings, guessing at the content of each.

We propose augmenting the user's interface to IRC with a software agent that deals with the information overload of ten thousand potential conversations. Our first such prototype agent, called *Butterfly*, samples the content of all of the channels and makes recommendations using a keyword-based model of interest. Since it is deployed in an existing real-world environment, Butterfly's design has been influenced by privacy concerns and architectural properties of the IRC infrastructure.

USAGE SCENARIO

Butterfly's usage is perhaps best illustrated by an example. First, the user sends Butterfly an IRC message containing keywords for some of the user's interests.

> ***User:*** "I am interested in agents, communityware, ecommerce, sociology, and HCI."

Butterfly immediately returns a message recommending a channel that it remembers contained some of the keywords:

> ***Butterfly:*** "OK. You may be interested in channel #VirSoc (communityware, sociology)."

[1] In a representative sample of the IRC network used for evaluation, EFnet, there were 38,367 connected users in 15,951 channels, distributed among 56 servers. The actual base of users is much larger, as most do not remain connected continuously.

Butterfly also remembers the keywords for the user's interests, and later when it discovers one of them being used in a channel, it notifies the user:

> ***Butterfly:*** "You may be interested in channel #hahvahd (ecommerce)."

The pseudo-natural language interface is described in more detail later in this paper.

INTEREST AND CONTENT MODELS

The user interest model presently used by Butterfly is based on a simple term vector with positive and negative weights. For an example of differing weights in the user interest profile:

> ***User:*** "I am interested in sprinting."
>
> ***User:*** "I'm very interested in Aikido, but not in Seagal."
>
> ***User:*** "Who am I?"
>
> ***Butterfly:*** "The interest profile I have for you is: aikido(200), seagal(-50), sprinting(100)"

The current version of Butterfly uses fixed constants for normal interest, great interest, and negative interest weights, although more variable weights could be supported if the interest profile were learned, such as by watching the user's Web browsing behavior [5].

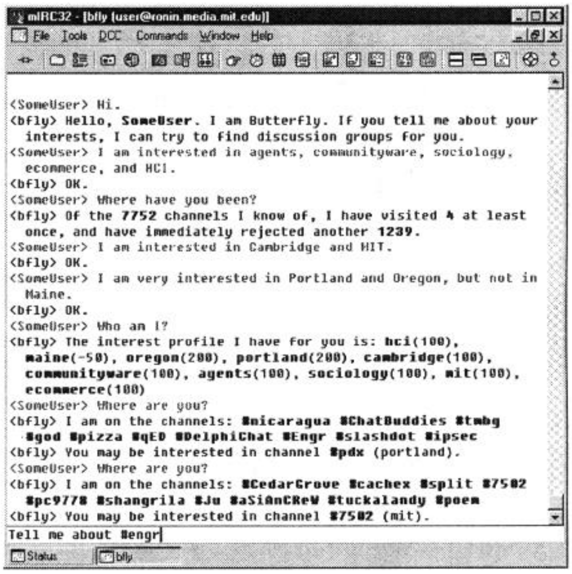
Channel content is also represented as a term vector, with the weights corresponding to frequency of occurrence. Relevance of a channel to an interest profile is determined by summing the weights of the dot product of the two vectors. If the sum is over some threshold, then the channel is recommended. The incremental building and decay of the channel content term vectors is tied to the channel-sampling behavior.

CHANNEL SAMPLING

Our decision to have Butterfly operate on the vast real-world data of an active IRC network, rather than on a relatively small static data set, significantly affected the design of the prototype.

The first design constraint concerned privacy. An IRC user knows who else is on a channel that he or she is on, and has an understanding that only those people will be able to hear what is said on the channel. Surreptitiously monitoring all traffic through an IRC server might violate reasonable expectations of privacy. Therefore, Butterfly connects to an IRC server and appears as a normal user client joined to any channel it is sampling.

This solution to the first design constraint in turn imposed a second design constraint for the prototype. Normal IRC user clients typically cannot join more than about ten channels at once. Therefore, instead of monitoring thousands of channels continuously, Butterfly employs a scheduled visiting behavior. It samples up to ten channels at once, and does not stay on any channel for more than 30 seconds at a time. Channels that have never been visited take priority over those that have.


mIRC32 - [bfly [user@ronin.media.mit.edu]]
File Tools DCC Commands Window Help
Status
bfly

```
<SomeUser> Hi.
<bfly> Hello, SomeUser. I am Butterfly. If you tell me about your
  interests, I can try to find discussion groups for you.
<SomeUser> I am interested in agents, communityware, sociology,
  ecommerce, and HCI.
<bfly> OK.
<SomeUser> Where have you been?
<bfly> Of the 7752 channels I know of, I have visited 4 at least
  once, and have immediately rejected another 1239.
<SomeUser> I am interested in Cambridge and MIT.
<bfly> OK.
<SomeUser> I am very interested in Portland and Oregon, but not in
  Maine.
<bfly> OK.
<SomeUser> Who an I?
<bfly> The interest profile I have for you is: hci(100),
  maine(-50), oregon(200), portland(200), cambridge(100),
  communityware(100), agents(100), sociology(100), mit(100),
  ecommerce(100)
<SomeUser> Where are you?
<bfly> I am on the channels: #nicaragua #ChatBuddies #tmbg
  #god #pizza #qED #DelphiChat #Engr #slashdot #ipsec
<bfly> You may be interested in channel #pdx (portland).
<SomeUser> Where are you?
<bfly> I am on the channels: #CedarGrove #cachex #split #7502
  #pc9778 #shangrila #Ju #aSiAnCReW #tuckalandy #poem
<bfly> You may be interested in channel #7502 (mit).
Tell me about #engr
```

Fig. 2: Screenshot of interactions

Butterfly has an optional keyword list that it uses to reject channels based on their names or topic strings alone. During testing of Butterfly, this list was principally composed of vulgarities, to spare the authors' delicate sensibilities.

While it is visiting a channel, Butterfly builds a term vector of occurrence counts of keywords it extracts from channel conversation. Not counted are closed-class words such as prepositions. When Butterfly leaves a channel, it decays the previous term vector of the channel by 0.5 and adds to it the vector for the visit. Then Butterfly determines if it should recommend this channel based on the new information about its content.

A third design constraint was that we wanted Butterfly to be scalable to many people using it at once. Therefore, rather than having a thousand people's individual Butterfly agents overloading the IRC network, a single agent works on behalf of multiple users at once. It identifies, and maintains separate interest profiles for, each user.

USER INTERACTION

As shown in the previous examples and Fig. 2, Butterfly uses normal IRC text messages to communicate with its users. This is a convenient interface that avoids the need for multiple users to install and run additional software to use Butterfly.

The command messages have a simple syntax that results in commands that look like English, and Butterfly sends English-like messages to the user. The intent is not to anthropomorphize Butterfly for the sake of affecting how users respond to it, but to have a syntax that is easy both to remember and for the casual observer to understand [2]. Unlike many of the "chatterbots" that inhabit some online chats [6], Butterfly does not attempt to fool other IRC users into thinking that it is a human. It is intended as an interface extension for users, not an interest group participant itself.

EXPERIENCE

The Butterfly prototype as described is operational, and we have conducted limited tests of it. One difficulty we anticipated is that the brevity of the sample periods means that Butterfly must run for a long time before it acquires a term vector representative of the content a channel.

Another problem did not become apparent until after Butterfly was running and we could see all the traffic it sampled. IRC allows users to make a channel *secret* so that it does not show up in the channel list. Approximately half of the channels on the IRC network we used (EFnet) are secret. An informal poll of experienced IRC users revealed that most of the channels that they consider valuable are secret. Butterfly cannot practically find all of the secret channels and so may never encounter many of the best channels.

FUTURE DIRECTIONS

One of the primary reasons that users make a channel secret is that they have established a group of interest and do not want to dilute its value with inappropriate traffic. A common problem is that once a channel is secret it starts to become stale for lack of fresh participants. This suggests a need for agents that operate on behalf of established groups trying to attract desirable new participants.

As a solution, we are investigating a multi-agent architecture in which each group and individual has its own agent. Group agents know about properties of the group and the properties that are wanted of new participants. Individuals' agents know properties of the individual and what properties he or she is looking for in a group.

Properties of a group include its purpose, topics it considers, format (e.g., open discussion, question-and-answer), tone of interaction (e.g., politeness), social rules, and group dynamics. Some information about properties is explicitly communicated to the group agent, and other information is extracted by the agent by constantly observing the group's interactions. For example, a group may tell its agent that the group is concerned with discussion of network administration in general, but the agent may notice that the group seems to talk about certain kinds of network routers a lot. That extracted information would be useful in matchmaking with an individual interested specifically in one of those routers.

RELATED WORK

Yenta [3] uses a privacy-safe referral mechanism to discover clusters of interest among people on the Internet, and presently builds user profiles by examining the users' E-mail and Usenet messages. Butterfly fills a somewhat different role, helping people to find groupings that have been established explicitly by people, and characterizes a group by sampling its interactions.

Some related systems [1, 4] refer individuals to other individuals on demand for purposes of getting assistance.

Others have tackled the Internet information overload problem through filtering of interactions within large groups (e.g., [8]). By contrast, Butterfly reflects a desire to build smaller, more focused groups, which perhaps will foster stronger interpersonal connections with attendant familiarity, trust, and willingness to provide assistance.

ACKNOWLEDGEMENTS

Support for this research was provided in part by grants through the Media Laboratory's Digital Life Consortium, News in the Future Consortium, British Telecom, IBM, and other sponsors of the MIT Media Lab.

REFERENCES

1. Ackerman, M., McDonald, D. Answer Garden 2: Merging Organization with Collaborative Help, in Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work (CSCW'96), 1996.

2. Don, A., Brennan, S., Laurel, B., Shneiderman, B. Anthropomorphism: from Eliza to Terminator 2, panel position statements in *Conference Proceedings on Human Factors in Computing Systems* (CHI'92), May 1992.

3. Foner, L. Yenta: A Multi-Agent, Referral-Based Matchmaking System, in *Proceedings of the First International Conference on Autonomous Agents* (Agents'97), Marina del Rey, CA, USA, February 1997.

4. Kautz, H., Milewski, A., Selman, B. Agent Amplified Communication, in *AAAI'95 Spring Symposium Workshop Notes on Information Gathering in Distributed, Heterogeneous Environments*, 1995.

5. Lieberman, H. Letizia: An Agent That Assists Web Browsing, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, August 1995.

6. Mauldin, M. Chatterbots, TinyMUDs, and the Turing Test: Entering the Loebner Prize Competition, in *Proceedings of the Twelfth National Conference on Artificial Intelligence* (AAAI'94), Seattle, WA, USA, August 1994.

7. Oikarinen, J., Reed, D. Internet Relay Chat Protocol. Internet Network Working Group RFC 1459, May 1993. http://info.internet.isi.edu/in-notes/rfc/files/rfc1459.txt

8. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews, in *Proceedings of Conference on Computer Supported Cooperative Work* (CSCW'94), October 1994.