CrowdTracker: Enabling Community-Based Real-Time Web Monitoring

James Caverlee, Zhiyuan Cheng, Brian Eoff,
Chiao-Fang Hsu, Krishna Kamath, and Jeff McGee
Texas A&M University
College Station, TX 77843
{caverlee,zcheng,bde,drakihsu,kykamath,jeffamcgee}@cse.tamu.edu

ABSTRACT

CrowdTracker is a community-based web monitoring system optimized for real-time web streams like Twitter, Facebook, and Google Buzz. In this demo summary, we provide an overview of the system and architecture, and outline the demonstration plan.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.4 [Information Storage and Retrieval]: Systems and Software

General Terms: Algorithms, Design, Experimentation

Keywords: community, crowd, real-time web, monitoring

1. INTRODUCTION

The real-time web is fast becoming an indispensable resource, opening up fresh opportunities for enabling a new generation of applications for monitoring, analyzing, and distilling information from the *prospective web* of real-time content that reflects the current activity of the web's participants. As a step toward realizing this vision, we are investigating techniques and developing effective methods for automatically discovering and tracking the evolution of transient crowds from the massive scale of the real-time web. A transient crowd is an ad-hoc collection of users that reflects the real-time interests and affiliations of users [2]. Unlike the more static and perhaps staler group-based membership offered on many social networks (e.g., members of the Texas A&M community, residents of Ohio), transient crowd discovery promises organic and highly-temporal group interest identification and provides a powerful mechanism for analyzing the topical, demographic, and social characteristics of crowds with respect to multiple spatial granularities. (e.g., neighborhood-based interests, city-based trends, statewide movements, etc.). Successful crowd monitoring can positively impact online event detection, content personalization, social information discovery, as well as lead to new crowd-based applications in areas like search, advertising, and emergency management.

2. SYSTEM OVERVIEW

The CrowdTracker system supports crowd-oriented web monitoring optimized for real-time web streams like Twitter, Facebook, and Google Buzz. Concretely, the system supports three overlapping crowd perspectives: (i) *communication-based*, reflecting groups of users who are actively messaging each other, e.g., users coordinating a meeting; (ii) *location-based*, reflecting groups of users who are geographically bounded, e.g., users posting messages from Houston, Texas [1]; and (iii) *interest-based*, reflecting groups of users who share a common interest, e.g., users posting messages about a presidential debate. While each of these crowd perspectives can be monitored separately, in many cases it will be important to monitor cross-cutting crowds, e.g., users in Houston (location-based), messaging each other (messaging-based) about a local fire (interest-based). The crowd tracking prototype collects social media streams (primarily from Twitter for the purposes of this demo), and then passes these streams to the crowd tracking and trending component, including the three-layered crowd detection component (communication-based, geo-based, and topic-based).

Frontend: The frontend is designed to provide the user with a simple and flexible interface to maximize their options for exploring crowds. The primary display is a crowd display window, including both a global view panel and a personal workspace panel. The personal workspace panel is designed for exploration of a set of crowds marked by the user when exploring in the global view panel. Since each crowd may be associated with a starting and ending time, crowd participants, topics, messages, location data, and so on, we consider a timeline, list, and map format.

Backend: The backend relies on several modules, including a *crawler* for collecting information from real-time web streams; a *filter* for removing unwanted spam messages and low-quality content encountered by the crawler; a *grouper* for identifying crowds from the data collecting and inserting them into a database; and *HTTP JSON RESTful APIs* for exposing an HTTP-based API that clients can query to return crowds, users, and their messages. Concretely, the backend relies on several tools including *Beanstalk*, a simple, fast work queue used to connect the crawlers, filters, and groupers together; *Tornado*, a fast and scalable non-blocking web server; and *MongoDB*, a scalable, high-performance, open source, document-oriented database for storing discovered crowds.

3. DEMO PLAN

As part of the demonstration, attendees will be able to view significant crowds along a number of dimensions (e.g., by size, by growth rate), drill down for details on particular crowds (to view the timeline of the crowd, members of the crowd, and crowd-based themes), query for crowds that match simple keywords, and filter crowds by geography (e.g., to view crowds based primarily in Texas).

4. REFERENCES

[1] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *CIKM*, 2010.

[2] K. Kamath and J. Caverlee. Transient crowd discovery on the real-time web. In *WSDM*, 2011.

SIGIR'11, July 24–28, 2011, Beijing, China.
ACM 978-1-4503-0757-4/11/07.