Multiple Event Information Extraction from Semi-Structured Pure Text (PROPOSAL)

Peter Groves
March 11, 2003

Abstract

This project is part of a planned larger effort to build a system that is able to extract upcoming event information from streams of weekly-calendar-style emails and compile them into a database. The database could then be queried by a dynamic web page, news feed, personal planner application, or any other applications that present event information. This project will make use of the General Architecture for Text Engineering (GATE) package developed at the University of Sheffield for most of its information extraction (IE) functionality. The focus of this course project is to resolve the issues associated with porting the GATE routines to a domain other than the one it was developed for. This includes identifying addresses given only partial information and creating rules for labeling tokens as “event titles.” Furthermore, and likely most difficult, complete event descriptions will have to be extracted from a single text containing varying numbers of dates, times, locations, titles, descriptions, etc.

1 Introduction

There are currently two main ways that local event information is disseminated over the internet. One is community calendar websites, where people must manually submit events, and interested users can check back periodically to find out about events in their community. This has a serious drawback of being quite laborious for submitters, and demands that users check back frequently to prevent missing anything, while the low volume of submissions may cause the content to change infrequently. Furthermore, multiple calendars may exist that cover the same area, which requires multiple submissions if one wants to ensure the maximum number of people see the announcement.

The second method of information delivery is mass emailing from organizations that a user can opt-in to. These are normally the most reliable sources of event information, as it is a simple way to inform people of an event who have explicitly stated they are interested in what the organization does. The drawback here is that users must know about the organization, and have signed up for their mailing list, in order to learn of events.

This project would like to begin work on a third way. We propose a system that receives as many distinct mass mails as possible, extracts the relevant event information, and compiles that information into a structured database. That database can then be queried by any of a number of applications to provide users with up-to-date, geographically relevant data on upcoming events. End-user applications might include a web page where a user enters their zip code and the furthest they are willing to drive, a news-feed that either feeds recently added events or events that are happening that day, or directly adding events to a user’s personal calendar.

Because of this last possible application (downloading events to one’s calendar), it is evident that another use of the proposed system could be easily implemented if the system is realized.
That is, the proposed system could be used as a plug-in to personal management software that has integrated email an calendar functionality. The plug-in could detect when emails contain event/appointment information and ask the user if they wished to add it to their calendar (or possibly add it automatically if it was from a certain trusted source, but these are issues that won’t need to be resolved until later).

2 Class Project Goals

As developing such a system is a major undertaking, the goals for the cs497 class project are much more focused. We plan to create and test the part of the system that, given an email message, extracts the pertinent attributes of any events described in that email. This includes date, time, title/brief description, category (meeting, lecture, etc.), cost, location name, and address. We plan to use the General Architecture for Text Engineering (GATE) package developed at the University of Sheffield for most of the Information Extraction (IE) tasks including the Named Entity functions.

While GATE will provide a basic framework, there will be much work in porting the functionality to the current domain. First of all, GATE was designed primarily to work with news articles, and therefore is optimized for english prose. The documents we’ll be dealing with will be plain text, but will be formatted in various styles, usually using some kind of pseudo-bullet list structure. Consider, for example, the excerpt at the bottom of the page from a Central Illinois concert newsletter. Notice that in many ways the information is structured much better for an extraction system that might be built from scratch than if the information was in typical sentence form. However, most of the current research that deals with plain text (read “not xml”) assumes typical sentence form and, as such, significant work is likely to be necessary to adapt it to this domain. Also notice that multiple events are listed under each location. It is equally likely that another mailing list will format their mailings by having multiple events and locations under a single date, or a general event title such as 'meeting' followed by a list of dates and locations. This will require significant work in associating various event attributes with each other to come up with a list of fully-described events. In fact, this process of attribute association is the main focus of this project.

By the end of this project, we will have a system that takes in an email and is able to identify all of the information of an event, for all of the events mentioned in the email. There will be two

Breadstretchers / 322 E. Monroe Springfield, IL / 217-753-3366

* 14 Fri Resident Genius / Sea Power // 8:00 / $3

Canopy Club / 708 S. Goodwin Champaign, IL / www.canopyclub.com / 18+
(TM) / All show times 10 pm unless noted

February

11 Tue Greenscene & Full Range // $2
13 Thu Mr. Blotto w/ Public Display of Funk
14 Fri Umphrey’s McGee // $8 (advance)
15 Sat Chirs Robinson & New Earth Mud // $10 (advance)
17 Mon Sound Tribe Sector 9 // $10
18 Tue Tryptich // $2
data sets available to the system, one test and one train. At the very least the test set will have to be manually tagged with event information (the real event info to be predicted), and possibly the train data as well if the algorithm developed is supervised. Whether or not the algorithm uses supervised learning has not been determined, as the algorithm is under development.

3 Timeline

Here is the schedule for this project:

April 10 Collection of examples is compiled with “correct” information manually specified for at least a portion to be used as a test set.

April 18 All necessary software is installed and working. Skeleton code is implemented that provides all the functionality of the tests that will be run. At this stage, this means that the event information code can simply take in an email and make jibberish-event fields, which will be given a score of 0. All the code in this process, however, runs correctly.

May 2 All code completed, working satisfactorily.

May 6 All test runs completed

May 13 Report, presentation completed.