Community Implicit and Explicit Feedback Recommender

Michael Cutter
UCSC IRKM laboratory
mcutter@ucsc.edu

ABSTRACT
The Know-My-News project is an expert system to study how pushed news story relevance is affected by a user's implicit browsing behavior. This is done with explicit, keywords and feedback data.

General Terms
The general areas to be discussed are: Design, Algorithms, Management, Economics, Experimentation, Security, Human Factors, and Legal Aspects.

Keywords
A user selected RSS feed will be denoted as RSS_{Ui} where U=user

INTRODUCTION
Know-My-News provides the user a community that keeps up to date on information that is in the shared interest of the group and the user. The user provides the Know-My-News system with implicit and explicit data. The research goal is to isolate implicit features to provide recommendations with minimal user interaction other then searching.

MOTIVATION
In order to keep up to date on news articles a user could constantly search the web with the same query, using Google alerts to automate the process. Know-My-News researches if recommendations can be made based purely on explicit and implicit use of a search engine restricted to the domain of user selected RSS feeds.

Know-My-News platform consists of seven components: A web application, database server, RSS parser, text indexer, recommender system, user forum, and a Firefox add-on to capture implicit feedback.

Web application
The web application is an interface developed in php to provide user level access to the Know-My-News platform. In order for a person to use Know-My-News they must log in and create an account and agree to the terms and conditions. Once a user has an account they will be able to select a set of RSS feeds with dynamic content. Examples include popular news websites, electronic sale sites, online classified, online auctions, and sports feeds. For each feed selected the feed index "i" will be set in the database as RSS_{Ui} . Other explicit information that is recorded is user queries and ratings.

Database server
Know-my-news is written in multiple programming languages in order to take advantage of the best open source software available. The way all these application stay interconnected in a mySQL database server. There are two databases. One is strictly for the lemur "Querylogtoolbar"; the other has been developed specifically to interconnect Know-My-News. There are the following tables: ratings, RSS_info_new, user_info, user_push_info, and user_keywords.

RSS parser
The parser component is written in Python and makes use of the open source RSS parser 'FeedParser.org'. For all RSS_{Ui} parsing occurs periodically on a schedule. In the RSS_info_new table the last title and the last publish date is stored. Every time the feed is parsed the date and title are checked to insure only new content is indexed. The output from the RSS parser is a text file in TREC format. After the entire update process has occurred the text file is added the user's index using the text indexer described in 3.4.

RSS parsing is accomplished by two python scripts, newUpdateFeed and personalizeUpdate, both of which import 'FeedParser.org'. The script personalizeUpdate is run for all users prior to calling newUpdateFeed. This order is necessary due to the way newUpdateFeed interacts with the database.

A critical aspect of a search engine is that it does not contain redundant information. For that reason it is important to only parse an RSS item once. The following will discuss my approach in dealing with this issue.

In the table RSS_info information is stored about each RSS feed. The three critical attributes of RSS_info for newUpdateFeed are the RSS URL, the last item title, and the last updateDate.

The update algorithm originally was supposed to use RSS embedded information such as syn_update or last_pubdate (xml elements). However, this approach was prone to error. The reason for this is that these elements are not standardized in even the same type of RSS feed. At first I wrote a massive try-catch scenario that attempted to handle all possible situations. After finding that some RSS did not contain any consistent date information I switched to a new approach.

NewUpdateFeed looks at the entry in the database of the last item title. It will then parse every RSS item from top to bottom until it either exhausts all items in the feed or encounters the same title. Once complete it updates the table for that specific RSS with the first items title, and stamps the current time.

The potential error in this algorithm is that some redundant titles are actually new content. In order to check against this with minimal data stored in the database my plan is to add another field to RSS_info. This adds the size of the text in the description of the last item. This information could then be compared, so if they had the same title but different content there would be no loss of information.

Text indexer

The open source text indexer lemur provides the searchable index feature to Know-My-News platform. When a user issues a query the web application calls searchANDshow.exe, a program in C++ that makes use of the lemur API to provide customizable search.

Recommender system

For all feeds being updated if any are equal to RSS_{Ui} the user keywords is queried in the new item. If there is a match Know-My-News algorithm decides if it should push it to the user. The user feedback will be used as training data to make the system adapt specifically to the user.

The recommender system learns from feedback.

$$Forall\ i='item'\ \square RSS_{ui}$$

If I have been rejected by the user then that item is deleted. The RSS feed it came from and the keyword are penalized in their respective tables.

When the system considers recommending another article from an RSS feed the feedback will determine if it is recommended and the rank it will be displayed as.

User forum

Modern day communities rely on ratings to provide user-user recommendations. One future goal of this application is to provide the capability for users to share trained classifiers on the forum. That way a new user can start with something and not have to train from scratch and simply will refine a pre-existing expert system.

Firefox add-on

All implicit feedback is captured using Lemurs 'Querylogtoolbar'. It acts like a key logger for the internet recording how long a user spends on a page, where they click and many other interesting user implicit features. All the data is sent through an apache tomcat servlet which is stored in a predefined database.

MANAGEMENT

At this point Know-My-News has not been released to the general public. When it is it will update the index periodically for each user. During the proof of concept phase the entire system checks for updates on RSS feeds every hour.

A goal of this system is to promote a community that will create specific profiles for different technical or niche fields. The motivation for this is if using the system has a low overhead there will be greater use. The more use the more expert data can be used to train the classifier.

LEGAL ASPECTS

User privacy is a major concern for any system that has personal information stored on the system. The userid is the super key for the entire database therefore user will be encouraged to pick an ID that does not imply their name. Security is another major consideration. There will be multiple safe guards in place to ensure that data is secure. The authors of the Querylogtoolbar made privacy and transparency a user manageable system. The system provides a function to users to input terms that all forms (through a regular expression) to be scrubbed out of any output sent to the server. Before the user study candidates will be trained on how to configure the toolbar to not share any of their sensitive information.

GOALS OF FUTURE USER STUDIES

The studies will attempt to answer the following questions.

Are demographical features independent of implicit browsing behavior?

What implicit features imply continued interest in a specific website?

ACKNOWLEDGMENTS

Funded by SURF-IT REU funded by the NSF. Special thanks to Richard Hughey for managing the program.

IRKM lab

Professor Yi Zhang

Graduate student advisor Anita Krishnakumar

REFERENCES

Lemur http://www.lemurproject.org/

FeedParser http://www.feedparser.org/