An Online News Recommender System for Social Networks

Manish Agrawal
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
magraw@microsoft.com

Maryam Karimzadehgan
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
mkarimz2@illinois.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.illinois.edu

ABSTRACT

In this paper, an online news recommender system for the popular social network, Facebook, is described. This system provides daily newsletters for communities on Facebook. The system fetches the news articles and filters them based on the community description to prepare the daily news digest. Explicit survey feedback from the users show that most users found the application useful and easy to use. They also indicated that they could get some community specific articles that they would not have got otherwise.

1. INTRODUCTION

Web 2.0 has provided a new way for people to communicate with each other. Nowadays, there are hundreds of sources delivering content in various forms and more content is generated from users through blogs, reviews and comments than through professional writers. Users are often faced with an *information overload*. Even in the days before Internet, people found it difficult to decide what book to read, or what movie to watch. They were often guided with the opinions and recommendations of their friends.

There are also some situations, when users *do not know exactly what they want* or they may not even be actively looking for information. Generally, users have some particular interests and they will like to have WebPages, news articles, blogs or events related to their interests delivered to them. However, they are currently forced to visit multiple sources and scan through irrelevant content before finding useful information. For such a long-term information need, one of the best ways to help users is to *recommend information* to them.

With a number of social networking Web sites such as Facebook, MySpace, Orkut and LinkedIn available, it is most desirable to have a system application that could integrate information from multiple sources to provide customized information for a community, i.e. a group of users sharing some common interests. These social communities allow us to get feedback from users and develop algorithms that most users would benefit from. However, little research has been done on recommending information to social networks.

In this paper, we present an online news recommender system for Facebook and study users' feedback. Our preliminary results indicate that most users find such an application useful and easy to use.

SIGIR-SSM, July 2009, Boston, USA.

2. A FACEBOOK NEWS RECOMMENDATION SYSTEM

2.1 Facebook

Facebook is one of the fastest growing social networks. It consists of many networks, each based around a school, workplace, or a region. It has users ranging from college students to working professionals. More than 100 million users log on to Facebook at least once each day [4].

Each user has a personal profile and a user may belong to multiple networks. Also, a user may join various interest-based groups on Facebook or may even start a new group.

In general, two users may be connected in three different ways: 1) User a is a friend of User b. 2) User a and User b belong to the same *network*. 3) User a and User b belong to the same *group*.

Connection 1 and 2 are strong connections, in the sense, that a user can see the profiles of all his friends and also the profiles of all the people in his network(s). If the user finds someone on Facebook, who is neither his friend nor in his network, he will only be able to see a restricted profile.

We refer to both network and group as community, and the main task of our recommender system is to recommend information to the users of a community. The recommendation was initially based on manually created keyword description of a community. After obtaining user feedback, the system would improve recommendation decisions based on feedback information.

2.2 Content-based recommendation

Each community has a set of sources associated with it. These sources are RSS feeds from community related websites or news agencies. Data of the specified sources for the communities are crawled. This kind of focused crawling has low coverage of the Web because only a small subset of the available news sources is crawled. As a result, we also crawl news articles from Yahoo! News [1]. For each community, we search Yahoo! News by using keywords from the community.

Once the articles are crawled, we construct an *inverted keyword index* over the entire text for all the articles. For each community, we construct a separate index for each day using the Indri indexing [9]. For each community, a set of queries with weighted terms are constructed. Indri has a rich query language that provides flexibility in building such queries. As an example, the following shows the set of Indri queries used in our system for the "University of Illinois" community:

```
#uw(#syn(#2(University Illinois) UIUC #1(U of I)) news)
#uw(#syn(#2(University Illinois) UIUC #1(U of I))...
#syn(student academic events))
#uw(#syn(#2(University Illinois) UIUC #1(U of I))...
#syn(#uw(fighting illini)athletics football basketball))
```

As shown, the terms in the #syn block are used as synonyms. #2(University Illinois) matches "University * Illinois" (where * is any word or null). #1(U of I) exactly matches "U of I". #uw stands for unlimited unordered window and indicates that all constituents must occur within the current context in any order [9]. So, it is possible to construct complex queries using the Indri query language.

When, the user registers a new community with the keyword description, it may not be always possible to manually construct such high quality queries. So, in this case, a query is constructed with all keywords having equal weights. Our experiments show that even such simple queries are effective content based filters as the users were satisfied with the recommended articles. The user survey is detailed in a later section.

Different queries may return the same document as a relevant result. Also, different sources may report the same news article. These factors produce redundant articles in the result set that should be put together. This is achieved by clustering the documents based on their content using Indri's clustering routine. We used the default centroid-based agglomerative clustering algorithm that uses cosine similarity as the distance measure. The clusters are sorted on the score of the most relevant document in each cluster and the top 5 clusters are presented to the user. We kept the similarity threshold high (0.70) to allow only highly similar documents to merge together and avoid generating large and general clusters.

Presently, the *Daily Newsletters* is prepared based on the retrieval results using such queries (i.e., a *content-based filter* on the news feeds). However, the *Popular Articles* list is prepared by aggregating the user feedback (i.e., *collaborative filtering*). Thus our system integrates content-based filtering with collaborative filtering. Below we describe how feedback is done.

3. COLLABORAITVE USER FEEDBACK

The system is designed to capture both implicit and explicit user feedback. This feedback later, can be leveraged to provide useful features in the application and improve the filtering accuracy. $F = \{(u_i, a_j, f_{ij})\}$ is a feedback where u_i is the user, a_j is the article and f_{ij} is a parameter that aggregates all kinds of feedback for article a_j from user u_i. The user feedback is captured in three different ways:

[1]http://news.search.yahoo.com/

1. Clickthroughs of articles ($f1$).
2. Ratings of articles ($f2$).
3. Recommendation of an article by a user to another user ($f3$).

for each article a_j, we have potentially three different kinds of feedback from each user u_i. We then need an algorithm to aggregate these feedbacks together. Let b be the base rating of the articles recommended by our system. Only if a user rating is more than the parameter b, it will be considered positive feedback from the user. We use a very simple algorithm for feedback.

$f1_{ij}$ denotes the feedback through clickthrough for article a_j from user u_i. $f2_{ij}$ denotes the feedback through rating of article a_j from user u_i. $f3_{ij}$ denotes the feedback through recommendation of article a_j from user u_i.

The parameters are defined using the ternary operator. If the predicate in between the brackets is true, the variable takes the value after "?", otherwise the variable takes the value after ":". m denotes the scale on which the feedback is measured. So, m is the maximum feedback value that an article can get and r is the rating given by user u_i for article a_j.

$f1_{ij} = (u_i \text{ clicked article } a_j) \ ? \ m : 0$
$f2_{ij} = (u_i \text{ rated article } a_j) \ ? \ (r-b)*(m/(m-b)) : 0$
$f3_{ij} = (u_i \text{ recommended article } a_j) \ ? \ m : 0$

And F_{ij} aggregates all these feedbacks as follows:

$$F_{ij} = \sum_{k=1}^{3} \lambda_k * fk_{ij} \quad (1)$$

where λ_k denotes the weight we want to put on each kind of feedback.

$F(a_j)$ is the feedback for article a_j aggregated for all users which is defined as follows:

$$F(a_j) = \sum_{i=user} W_i * F_{ij} \quad (2)$$

where W_i is the weight we want to put on each user.

Let *support* be the number of users used for calculating the feedback. We set $\lambda_k = 1/3 \quad \forall \quad k$; this means that we are equally weighing the three kinds of feedback and $W_i = 1/support$; which implies the feedback from all users are given equal weight. m is set to the best rating that a user can give i.e. 5 and b is set to 2 since a rating of 3 or more is considered positive feedback (as per user guidelines).

Let us see some scenarios to better understand the feedback model. We have dropped the subscript i and j to make the scenarios more readable. Note that F means F_{ij}.

Case 1 (No feedback available) :
$f1 = 0; f2 = 0; f3 = 0$
$\Rightarrow F = 0$ (No Feedback)

Case 2 (Bad article) :
User does not even click the main article but based on the synopsis, rates it as 1 (minimum rating).
$f1 = 0; f2 = (1-2)*(5/(5-2)) = -5/3; f3 = 0$
$F = (f1 + f2 + f3)/3 = -5/9$ (Negative Feedback)

Case 3 (Best article) :
User reads the article, likes it, recommends it and rates

it as 5.
$f1 = 5; f2 = (5 - 2) * (5/(5 - 2)) = 5; f3 = 5;$
$F = (f1 + f2 + f3)/3 = 5;$ (Maximum Positive Feedback)

These scenarios show how the parameter $F(a_j)$ can be interpreted to understand the collaborative user feedback. In our system, we use this feedback for Popular Articles feature, where Articles are ordered by $F(a_j)$ as long as $support(a_j) > support_{min}$.

3.1 Newsletters System

The *Newsletters* application uses Facebook API to find out the networks of the user. The newsletters for the networks are available as *tabs* on the top of the page. The newsletter is presented as a list of articles, each with its title, news synopsis, links to original article and the locally cached page. Users can rate an article on a scale of 1 to 5. In the newsletter, the clusters are sorted on the score of the most relevant document in each cluster. Only one result per cluster is presented to the user, but the user can look at the other results by navigating through the *"Similar pages"* link. Only the top 5 results are presented to the user. The users can also *recommend* particular news articles to their friends through a recommendation button provided beside each news result. When a user makes a recommendation, his friend is sent a notification. The snapshot of the system is shown is figure 1.

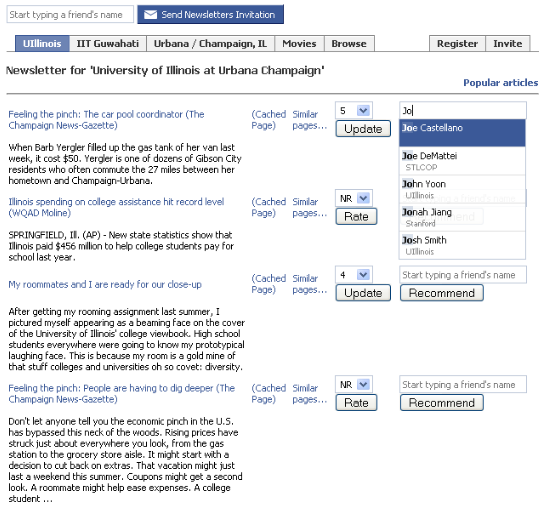

Figure 1: University of Illinois NewsLetter

4. PILOT STUDY

We conducted a pilot study to understand how users respond to such a recommendation system. The *Newsletters* application was launched on Facebook for three months[2]. The application was advertised amongst university students by "word of mouth" publicity. During this period, sixty people added the application on Facebook from a number of universities. There were initially 3 seed communities on the application with available newsletters but during three months, the number of communities increased to 25. All the new communities are user initiated.

[2]After we got the first user feedback and the usefulness of such an application, we have been working on developing new algorithms for feedback since then. So the user statistics is gathered for about three months.

Statistics for all the communities:	
No. of people registered:	60
No. of clicks:	350
No. of inter-person recommendations:	15
Average rating (out of 586 ratings):	3.41
Average rating for clicked article:	3.71
Average rating for recommended articles:	4.07
Statistics for U of I community:	
No. of clicks:	145
No. of inter-person recommendations:	5
Average rating (out of 220 ratings) :	3.76
Average rating for clicked article:	4.14
Average rating for recommended article:	4.11

Table 1: Usage Statistics

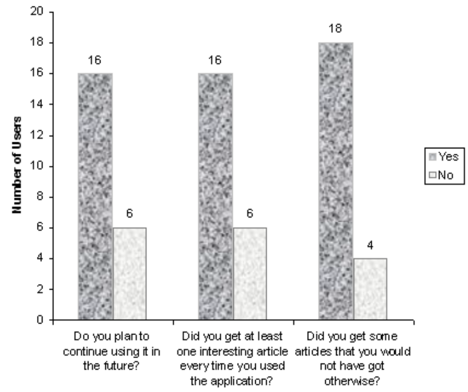

Figure 2: User Survey I

A set of University of Illinois (U of I) students, were asked to use the application on a *regular basis*. They used the application for about one month and then a user survey was conducted to garner explicit feedback from them. The basic aim was to evaluate the utility, accuracy and scope of the application. 22 users participated in the survey. Before presenting the user survey results, let us first look at the user data automatically captured by our system. Table 1 shows the basic statistics of user clickthroughs and ratings. We can see that the average rating of a news article is 3.41 (out of 5), and it is higher for clicked and recommended articles. U of I was one of the seed communities of the application and also most of our users were U of I students. So, let us look at the statistics from U of I newsletter. Average rating is higher in this case (3.76). The rating of the articles that were either clicked or recommended was more than 4.1. From these results, it seems that the articles that the users clicked or recommended are the best candidates for including in the user feedback for improving the language model of the community. Figure 2 shows a part of the user response gathered by the user survey. Most users plan to continue using the application in future and said that they at least got one *interesting* article every time they used the application. 18 users (81.8%) said that they got some article that they *would not have got otherwise*, through their newspaper

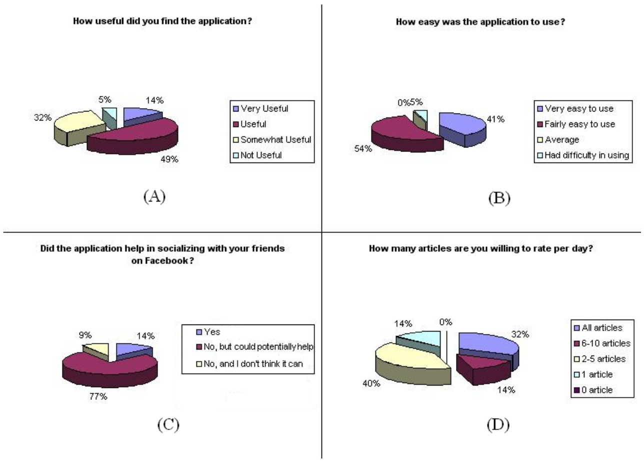

Figure 3: User Survey II

or regular web browsing. This is an important result as it shows that the application was successful in retrieving localized information for the communities that are generally ignored in the nationwide newspapers. Figure 3 shows 4 pie charts showing the user response. 95% of the users found the application from somewhat useful to very useful. 95% found it fairly easy to use or better. Only 14% think that the application helped them to socialize with their friends but another 77% feel that the application can potentially do that. In the present form, the application has limited support to interact with other people. The only way one can socialize is by recommending news to their friends. But it is encouraging to know that most of the users think that this application has the potential to help them socialize. In the future, more features can be added to the application that may extend the users' social networking experience.

We asked the users that if they had a perfect news recommender system, what delivery methods are most preferred by them. 14 users (63.64%) will like to receive newsletters on Facebook. So, the users see Facebook as the *best platform* for receiving such content. 10 users also expressed interest in receiving newsletters through email. This facility can be integrated in our system where the users can register specific newsletters for delivery via email.

5. RELATED WORKS

Recommender systems have been studied for many years. Adomavicius et al. [1] have categorized recommendations systems in three main categories: Content-based filtering, collaborative filtering, and Hybrid approach which combines the previous two methods. Collaborative filtering systems, recommend items based on similarity of users and their preferences. Examples of such a method are [8, 2, 6]. In Content-based recommendation, items similar to the ones that a user has seen in the past are recommended. Examples of such systems are [7] for news recommendation and [10] for book recommendation.

Although there has been some previous work on recommendations in social networks [11, 5, 3], and many applications have been developed for Facebook, there has been no application for recommending news information on Facebook, which is our goal.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel system for recommending news articles to communities of a social network. Our main contribution is building a novel news recommender system and integrating it with Facebook and gathering user feedback. Our recommendation approach is a combination of content-based filtering and collaborative filtering. To the extent of our knowledge, this is the only application of its kind on Facebook. The system is automatic, sustainable and scalable to a large number of communities by user participation. User studies indicate that most users of this application find it useful and efficient, demonstrating the feasibility of recommending information through social networks.

For future, we plan to do the followings:
Presently the newsletters is community specific. It can be personalized for individual Facebook user by exploiting the user profile and their friends' profiles. Adding some features to help users socialize is another interesting direction. Tagging entities in the news articles would also be interesting. In addition, we plan to develop and evaluate more advanced feedback algorithms based on machine learning and incorporate them with the system.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.

[2] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *ICML*, pages 46–54, 1998.

[3] P. Bonhard, C.Harries, J. McCarthy, and M. A. Sasse. Accounting for taste: Using profile similarity to improve recommender systems. In *ACM CHI*, pages 1057–1066, 2006.

[4] Facebook. Facebook statistics. *http://www.facebook.com/press/info.php*, 2009.

[5] G. Groh and C. Ehmig. Recommendations in taste related domains: Collaborative filtering vs. social filtering. In *ACM Group*, pages 127–136, 2007.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *ACM SIGIR*, pages 230–237, 1999.

[7] H.-J. Lai, T.-P. Liang, and Y. C. Ku. Customized internet news services based on customer profiles. In *ICEC*, pages 225–229, 2003.

[8] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[9] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40(5):735–750, 2004.

[10] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *ACM DL*, pages 195–204, 2000.

[11] A. Seth and J. Zhang. A social network based approach to personalized recommendation of participatory media content. In *AAAI*, pages 109–117, 2008.