Personalized News Recommendation Using Twitter

Nirmal Jonnalagedda and Susan Gauch
Department of Computer Science and Computer Engineering
University of Arkansas
E-mail: jonnalag@uark.edu, sgauch@uark.edu

Abstract

Online news reading has become a popular way to read news articles from a huge collection of news sources around the globe. News recommender systems help users manage this flood by suggesting articles based on user interests rather than presenting articles in order of their occurrence. We present our research on developing personalized news recommendation system with the help of a popular micro-blogging service, "Twitter." News articles are ranked based on the popularity of the article identified with the help of the tweets from Twitter's public timeline. In addition, users construct profiles based on their interests and news articles are also ranked based on their match to the user profile. By combining these two approaches, we present a hybrid news recommendation model that recommends interesting news stories to the user based on their popularity as well as their relevance to the user profile.

I. INTRODUCTION

Owing largely to the ever-increasing volume and sophistication of information on the web, we are able to access an enormous amount of information from around the globe. The key challenge today is for the users is to find relevant information based from an almost infinite source. This problem has led to the evolution of the recommender systems that proactively present users with information related to their interests.

Many organizations use recommender systems to recommend various types of products to the user. For example, Netflix recommends movies to its users, Amazon recommends products, and Pandora Radio recommends music based on users' past history and preferences. Additionally, there are many online news-reading services, such as Google News and Yahoo News. However, with so much news available, the driving problem is to identify and recommend the most interesting articles to each user so that they are not presented with a flood of information to wade through. These articles should be related to each user's interests but also include those news stories that are generating a lot of interest around the world.

In this paper, we develop a hybrid personalized news recommender system that recommends interesting news articles to the user using a micro-blogging service "Twitter". Our news recommender system ranks the news articles in different ways: (1) We consider the user's profile to recommend articles to the user; and (2) we also consider the article's popularity with the help of tweets from the Twitter's public timeline. We present a novel approach to help users find interesting articles to read by merging the above two methods of ranking articles.

II. RELATED WORK

A. Recommender Systems

Recommender systems collect data from users explicitly or implicitly and, based on the collected information, create user profiles. The user profiles are then used to generate recommendations. With explicit information collection, the user typically rates items in addition to his regular tasks. In contrast, with implicit information collection, the recommender system monitors the user's behavior with items during their normal activities and infers the user's preferences from their actions. Recommender systems are usually classified into three categories, based on how the recommendations are made [10][11].

- Content-based recommender systems: These recommender systems recommend an item to the user similar to the ones the user preferred in the past.
- Collaborative recommender systems: These systems recommend an item to the user based on the people with similar tastes and preferences have liked in the past. They have the advantage that they can recommend items for which little or no semantic information is available (music, movies, products).
- Hybrid recommender systems: These systems combine both the collaborative and content-based recommendation techniques in order to improve the accuracy of the recommendation.

978-1-4799-2902-3/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.144

B. *Popularity-Based News Recommender Systems*

News recommender systems are widely used to help readers filter through an ever-growing flood of information. Many researchers focus on using real-time social networking sites such as Facebook, Google Plus, and Twitter to identify the most popular and most current news stories. Because they are instant, and widely available, they provide a massive source of information on current events. However, because they are unmoderated, the quality of the information is variable. Alan et al. discuss a method to determine which Twitter users are posting reliable information and which posts are interesting [2].

Micro-blog posts can also be used as a way of identifying the popularity of certain events. Smyth et al. represent users and items based on micro-blogging review of movies and used this technique with various movie recommendation strategies on live-user data [1]. Phelan et al. focus on using micro-blogging activity to recommend news stories [7]. Their recommender system, *Buzzer,* is applied to RSS feeds to which the users have subscribed. *Buzzer* mines terms from RSS and Twitter feeds and uses them to rank articles. In [8, 9], they extended their work by considering the public-rank and the friend's-rank strategy rather than just considering the articles from the users' index.

C. *Profile-Based News Recommender Systems*

Profile-based, or personalized, news recommender systems recommend articles to the user based solely on his/her interests. A user profile is built based on the preferences or interests of the user. In one of the earliest news recommendation systems, Pazzani et al. created *News Dude*, a personal news-recommending agent that uses TF-IDF in combination with a Nearest Neighbor algorithm in order to recommend news stories to users [4]. Similarly, Michael et al [3] describe a content-based recommendation system that recommends a story to a user based upon a description of the item and a profile of user's interests.

Wouter et al. also describe ontology-based methods to recommend news articles to the users depending on their interests [5]. The user profile is created based on the news articles browsed recently. mapped to an ontology provided by *Athena*, a framework built for news personalization service, an extension of their earlier work [12].

III. APPROACH

In this section, we present an overview of our hybrid news recommendation system. Our basic approach is to recommend interesting news articles to the user based on a combination of his past interests and stories that are currently of broad interest. The user's interests are captured in his user profile and the community as a whole's broad interest is captured from tweets collected from Twitter's public timeline. Our intuition is that user's most want to see news stories related to topics in their profile that are also creating a buzz on the blogosphere. In summary, users are shown hot stories related to their favorite topics.

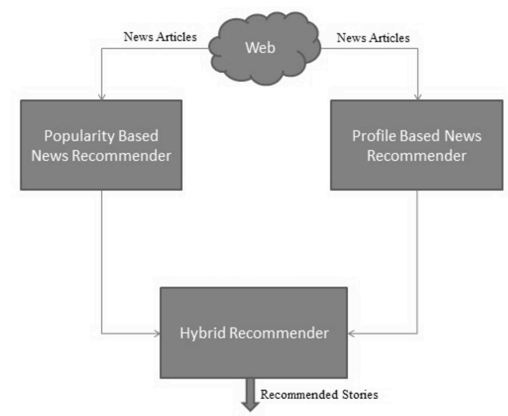

Figure 1. Architecture of the Hybrid News Recommender System

A. *High Level Design*

Figure 1 shows an architectural diagram of our Hybrid News Recommender system.

The system consists of three modules:

1. Popularity-Based News Recommender
2. Profile-Based News Recommender
3. Hybrid News Recommender

The first module recommends news articles based on the popularity of the article. The second module ranks the news articles based their similarity to a user's profile. The third module fuses the results from the above two modules to recommend news articles to the user.

B. *Popularity-Based News Recommendations*

Figure 2 shows an architectural diagram of the Popularity-based News Recommender system. First, the RSS articles are collected from a news source such as CNN or the BBC that organize their stories by category, e.g., Sports, Business, Politics, and Entertainment, etc. The article, and their associated categories are stored locally. The RSS articles are pre-processed to remove unnecessary content (html tags, numbers, etc.) while preserving the textual content. The pre-processed articles are then indexed by SOLR, an open source enterprise search platform from the Apache Lucene project [13].

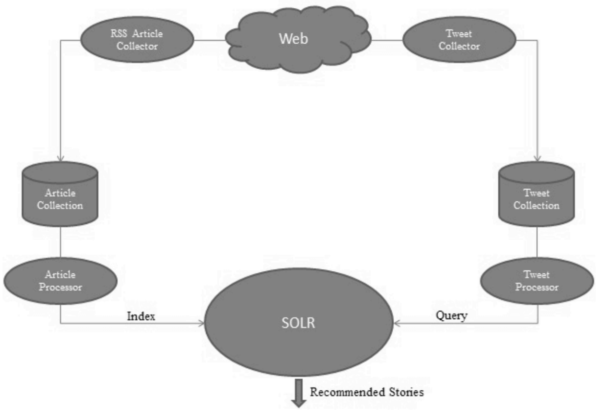

Figure 2. Architecture of the Popularity-based Recommender System

In order to identify which news stories are most popular, we also collect data from the Twitter micro-blogging site. These tweets are collected from Twitter's public timeline by the Tweet Collector that forms a request to Twitter's streaming API. The collected tweets are stored in the Tweet Collection in JSON format. The Tweet Processor is responsible for parsing the Tweet Collection by eliminating unwanted noise and preserving the tweet content. Each processed tweet is queried against the server to retrieve the articles that match the tweet contents. SOLR returns articles, and weights, based on how well each article matches the tweet according to the cosine similarity value. The weights for each article are accumulated across all tweets to produce a popularity weight for the article. Thus, the Popularity_Wt for article *i* is:

$$Popularity\_Wt_i = \sum_{t=1 \text{ to } T} CosineSimilarity(Article_i, Tweet_t)$$

where *T* is the number of tweets in the collection.

C. Profile-based News Recommendations

Figure 3 shows the architectural diagram of the Profile-Based News Recommender system. The profile-based recommender system uses the same article collection as the popularity-based recommender system. Although articles are placed in only one category by the website editor, they may actually partially belong to more than one category. To allow for this, each article is classified into all 7 potential categories using a k-nearest neighbor classifier [14], the classification module of the KeyConcept project. We store the top 3 most similar categories (and their similarity scores) for use in profile matching. In order to do fast lookup by category, we again use SOLR to build a second Lucene index that maps from category ids to document ids and weights.

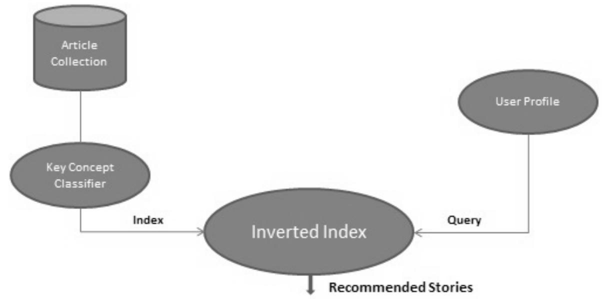

Figure 3. Architecture of the Profile-based Recommender System

Next, each user creates his or her own profile by manually scoring the categories presented on a Web form. This user profile is used to identify documents that best match their profile. The profiles and the articles can be viewed as feature vectors where each category is a feature. The similarity of each article to the user's profile is calculated using the dot product between the user profile's category vector and the article's category vector. Thus, for a given user *j*, the Personal_Wt for article *i* is:

$$Personal\_Wt_{ij} = CosineSimilarity(Article\ Profile_i, User\ Profile_j)$$

We reuse SOLR, querying with the Lucene index that stores the category vectors with the user's profile.

D. Hybrid News Recommendations

The hybrid recommender module combines the weights provided by each of the previous two modules to produce a recommendation based on both the articles popularity to users everywhere and the article's likely interest to the particular user. We first experimented with multiplying the two factors together:

$$Hybrid\_Wt1_{ij} = Popularity\_Wt_i * Personal\_Wt_{ij}$$

This module calculates a Hybrid_Wt by combining the two scores. The Hybrid_Wt for article *i* and user *j* is given by:

$$Hybrid\_Wt2_{ij} = \alpha * Popularity\_Wt_i + (1-\alpha) Personal\_Wt_{ij}$$

Hybrid_Wt2 incorporates a tunable parameter, α, that controls how much each of the two components contributes. When α is 0.0, only the Personalized_Wt is contributes to the overall weight whereas when α is 1.0, only the Popularity _Wt counts.

IV. EVALUATION

All experiments were conducted on the same collection of news articles 280 news articles collected from CNN and BBC and 202,224 tweets collected from Twitter on the same day. We collected 40 news articles for each of 7 topics (Sports, Crime, Business, Politics, Tech, Health and Entertainment). The results reported here were produced using 27 volunteer test subjects. To evaluate the relative importance of global popularity versus individual interest in a story, we varied α from 0.0 to 1.0 in increments of 0.1, a total of 11 values.

A. *Evaluating the Hybrid Recommender System*

Each volunteer test subject was presented with a web page on which they entered weights from 0 - 10 indicating their personal interest each of the seven categories such that they added to 10. These category/weight pairs form their user profile. For each of the 11 values of α evaluated, we identified the top 10 articles recommended by the hybrid system and presented them to the user to be rated as very interesting, interesting, or not interesting. In order to avoid bias, we randomized the order of presentation of the news articles to the user. Although subjects could potentially be asked to judge 110 articles, the average number of unique articles judged by the subjects was 45.

i. *Analyzing HybridWt1*

Since users only really look at the top 10 documents, we used metrics based on the ratings of articles in the top 10 recommendations. We analyzed our results using two different metrics: Average rating and Cumulative rating. Table 1 shows the average rating of news articles in the top 10 for all three approaches. The average rating fluctuates and there is no clear trend, other than the hybrid system presents, on average, a relevant document at every rank, the personal system does best at showing a relevant document as the top result, and the popularity-based system is not as effective as the other two.

Rank Order	Popularity	Personal	Hybrid
1	0.88	1.208	0.958
2	0.95	1.166	1.25
3	0.91	1.347	1.04
4	0.75	0.875	1.24
5	1.24	1.208	1.4
6	1.48	0.875	1.125
7	1.54	1.086	1.12
8	0.66	1.083	1.25
9	0.66	1.166	1
10	1.13	1.25	1.083

Table 1. Average Rating of Top 10 Articles

To take the rank order of relevant documents within the set of 10 recommended documents into account, we employ a cumulative rating metric that calculates, at each rank position in the top 10, the total relevance of all documents at or above that rank. We took the average of those ratings across all users.

Rank Order	Popularity	Personal	Hybrid
1	0.81	1.07	0.85
2	1.59	2.07	1.93
3	2.37	3.19	2.85
4	3.00	3.93	3.96
5	4.11	4.96	5.22
6	5.44	5.70	6.19
7	6.78	6.59	7.19
8	7.33	7.52	8.26
9	7.89	8.52	9.07
10	8.81	9.59	10.00
Total	48.15	53.15	55.52

Table 2. Cumulative Rating of the Top 10 Articles

From Table 2, we see that the personal and hybrid strategies consistently outperform the popularity-based strategy. Users are only interested in, well, things they are interested in. Very popular articles outside their normal areas of interest are just not very relevant to them. We can also observe that the relevance of the highest ranked articles is essentially tied for the personal and hybrid algorithms. However, the articles ranked 5 - 10 by the hybrid system are more relevant than those ranked by the profile-based system alone. Overall, users see more relevant new stories in the top 10 when these stories are recommended based on a combination of the story's popularity and their interest in the topic.

ii. *Analyzing HybridWt2*

Since it was the most informative, taking into account the ranking of the relevant articles, we evaluated our second method, HybridWt2, using only the cumulative rating of the top 10 articles. For this experiment, we used a subset of 15 of the 27 test subjects who were available. We chose to evaluate this second metric because, by adjusting α, we can tune the relative contributions of the popularity and personal weighting algorithms.

The mean of the cumulative rating over all users is depicted graphically in Figure 8. The worst performance, 47.4, arises when only the popularity measure is considered (α = 1.0). When the only user's profile is considered (α = 0.0), we achieve a cumulative rating of 54.5. These results are remarkably consistent with, and thus reconfirm, the performance of those approaches when we evaluated Hybrid_Wt1.

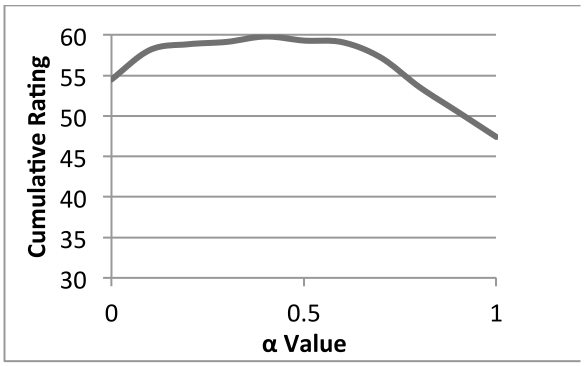

Figure 8. Average Cumulative Rating versus α

Our results show that as measured by the cumulative rating, α = 0.4 outperforms all other α values with a cumulative rating of 59.8 followed by 59.3 for the α = 0.5. These results indicate that the most relevant articles are those that are selected using a relatively equal combination of user interests and general public interests. Overall, Hybrid_Wt2 with α = 0.4 outperforms Hybrid_Wt1 by 7.8%, Personal_Wt by 9.7%, and Popularity_Wt by 26.2%. We performed a student's t-test analysis (2 tailed, matched) and confirmed that, although the Hybrid_Wt2 improvement versus the Personal_Wt was not quite statistically significant, its improvement versus the Popularity_Wt was (p=0.045) was significant.

V. CONCLUSION

In this paper, we presented the design and implementation of a news recommender system that incorporates a novel approach to recommend interesting news articles to the user.

We implemented four different strategies to recommend news articles to the user that are interesting to read.

We have evaluated each of the strategies and found that:

1. Both hybrid approaches outperform the popularity and personal recommendations.
2. The personal recommender provides better recommendations than the popularity-based recommender.
3. The tunable hybrid algorithm with α = 0.4 provided the best overall performance.

We can extend this work in several ways. In particular, the accuracy of our news recommender system can be improved by considering other features such as location or temporal activity. The recommender system could be improved by implicitly inferring the users' interests based on their reading habits.

REFERENCES

[1] S. G. Esparza, M. P. O'Mahony, and B. Smyth, "On the Real-time Web as a Source of Recommendation Knowledge," in *RecSys 2010*, Barcelona, Spain, September 26-30 2010.

[2] A. Jackoway, H. Samet and J. Sankaranarayanan, "Identification of Live News Events using Twitter," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, New York, New York, USA, 2011.

[3] M. Pazzani and D. Billsus, "Content-based Recommendation Systems," in *The Adaptive Web, pages 325–341*, 2007.

[4] D. Billsus and M. J. Pazzani, "A Personal News Agent that Talks, Learns and Explains," in *The Third Annual Conference on Autonomous Agents, pages 268–275*, May 1999, ACM.

[5] W. IJntema, F. Goossen, F. Frasincar, Hogenboom, "Ontology-Based News Recommendation," in *International Workshop on Business intelligence and the Web*, New York, USA, 2010.

[6] I. Cantador, A. Bellog´ın, and P. Castells, "News@hand: A Semantic Web Approach to Recommending News," in *AH 2008, pages 279–283*, Berlin, Heidelberg, 2008.

[7] O. Phelan, K. McCarthy and B. Smyth, "Using Twitter to Recommend real-time Topical News," in *Proceedings of the Third ACM Conference on Recommender Systems*, October 23-25, New York, New York, USA, 2009.

[8] O. Phelan, K. McCarthy, M. Bennett and B. Smyth, "Terms of a Feather: Content-based news recommendation and discovery using Twitter," in *Proceedings of the Thirty-Third European Conference on Advances in Information Retrieval*, Berlin, 2011.

[9] O. Phelan, K. McCarthy, M. Bennett and B. Smyth, "On using the Real-time Web for News Recommendation & Discovery," in *Proceedings of the 20th International Conference Companion on World Wide Web*, March 28-April 1, Hyderabad, India, 2011.

[10] A. Tuzhilin and G. Adomavicius, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," in *IEEE Transactions on Knowledge and Data Engineering, pages 734-749*, New Jersey, USA, 2005.

[11] M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," in *Communications of the ACM, pages 66-72*, New York, New York, USA, March 1997.

[12] F. Frasincar, J. Borsje, and L. Levering, "A Semantic Web-Based Approach for Building Personalized News," in *International Journal of E-Business Research, pages 35–53*, 2009.

[13] Apache SOLR website, http://lucene.apache.org/solr/

[14] L. Baoli, L. Qin and Y. Shiwen, "An adaptive k-nearest neighbor text categorization strategy," in *Journal of ACM Transactions on Asian Language Information Processing (TALIP), pages 215–226*, 2004.