Using Twitter to Recommend Real-Time Topical News *

Owen Phelan, Kevin McCarthy & Barry Smyth
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin Ireland
[firstname.lastname]@ucd.ie

ABSTRACT

Recommending news stories to users, based on their preferences, has long been a favourite domain for recommender systems research. In this paper, we describe a novel approach to news recommendation that harnesses real-time micro-blogging activity, from a service such as Twitter, as the basis for promoting news stories from a user's favourite RSS feeds. A preliminary evaluation is carried out on an implementation of this technique that shows promising results.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Theory

Keywords

Content-based recommendation, News recommendation, Real-time recommendation, Social recommendation, Twitter

1. INTRODUCTION

There is a long history of using recommender systems to help users to navigate through the sea of stories that are written and published everyday [1, 5]. Recommender systems promise to promote the most relevant stories to a user based on their learned or stated preferences or their previous news consumption histories, helping the user in question to keep up-to-date and to save valuable time sifting through less relevant stories. Content-based and collaborative filtering techniques have been used to good effect and the recent growth of services such as Digg (*www.digg.com*) demonstrate the value of recommendation techniques when it comes to deliver a more relevant and compelling news service.

For all the success of recommender systems there are some aspects of news recommendation that are not well catered for. For example, in this short paper we will consider the problem of identifying niche *topical news stories*. Current recommender systems are limited in their ability to identify such stories because, typically, they rely on a critical mass of user consumption before such stories can be recognised. In this paper, we consider a novel alternative to conventional recommendation approaches by harnessing a popular micro-blogging service such as *Twitter* (*www.twitter.com*) [2, 4] as a source of current and topical news. To this end we describe our initial attempts to mine Twitter information with a view to identifying emerging topics of interest, which can be matched against recent news coverage in an RSS feed, as the basis for story recommendation.

In the next section we describe our recommender system, *Buzzer*, focusing on the system architecture, and highlighting how real-time information is mined from Twitter and RSS feeds to provide a basis for matching and recommendation. Each user specifies a set of RSS feeds that they are interested in, as well as a recommendation strategy they wish to employ (this will be explained further below). In Section 3 we describe the results of a small, preliminary user evaluation to compare how different users respond to different types of RSS recommendations based on a number of different recommendation strategies.

2. RECOMMENDING TOPICAL NEWS

RSS (Really Simple Syndication) and Twitter are two important Web 2.0 technologies. The former is a data format that is designed to provide access to frequently updated content. Most commonly, RSS is used as a way to syndicate or distribute news information in the form of short-updates that can be linked back to complete stories. RSS Readers then allow users to aggregate the updates from many different feeds to provide a one-stop-shop to breaking news, although as users subscribe to tens of RSS feeds this introduces a niche information overload problem [8].

Twitter in contrast is a so-called *micro-blogging* service. It allows users to submit their own short (max 140 characters) status update messages, called *tweets*, while *following* the status updates of others. Recently there has been much interest in Twitter, partly because of its growth [2], and partly because of its ability to provide access to thoughts, intentions and activities of millions of users in real-time. The starting point for this paper is the idea that mining tweets

*This material is based on works supported by Science Foundation Ireland under Grant No. 07/CE/11147 CLARITY CSET

RecSys'09, October 23–25, 2009, New York, New York, USA.

can provide access to emerging topics and breaking events and that this information can be used as the basis for a novel approach to ranking RSS news feeds so that topical articles can be effectively promoted.

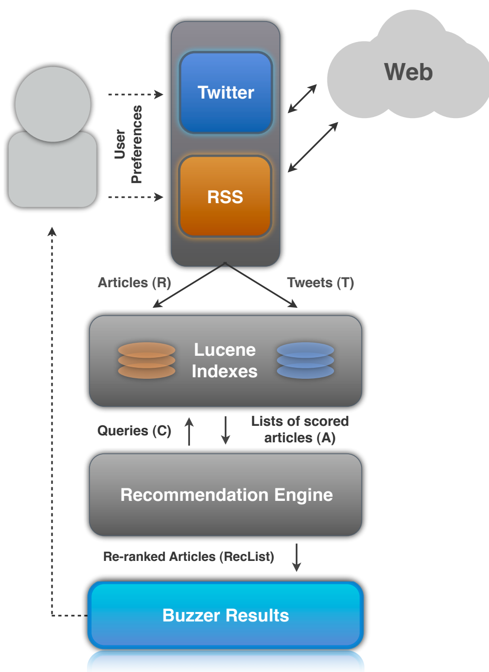

Figure 1: Buzzer back-end architecture

2.1 Recommendation Approach & Architecture

Buzzer adopts a content-based recommendation technique, by mining content terms from RSS and Twitter feeds as the basis for article ranking. Content-based approaches to recommending news articles have proven successful in the past. For example, perhaps the earliest example of a news recommendation service, *Krakatoa Chronicle* [3] represented user profiles as a weighted vector of terms drawn from the articles that a given user liked, and matched this weighted vector against a new set of articles to produce a ranked list for presentation to the user. Similarly, Billsus and Pazzani's *News-Dude* [1] harnessed content based representations and multi-strategy learning techniques to generate short-term and long-term user profiles, as the basis for news recommendation. Billsus and Pazzani [6] argue that content-based approaches to finding trends and topics in news articles are difficult because of the sheer random bag-of-words unstructured nature of articles, and the complexity of natural-language processing. We bypass this consideration because our technique finds common co-occuring terms among Twitter and RSS. The basic system architecture of Buzzer is shown in Figure 1.

The Buzzer system comprises three basic components:

1. The web-based *Configuration Interface* manages the basic user registration process and allows users to provide their Twitter account information and a list of RSS feeds that they wish to follow; in fact providing Twitter account information is optional since, as discussed later, Buzzer can use Twitter's *public timeline* as an alternative source of tweets, as opposed to tweets only from friends on Twitter.

2. The *Lucene Indexer* is responsible for mining and indexing the appropriate Twitter and RSS information, given the user's configuration settings.

3. The *Recommendation Engine* generates a ranked list of RSS stories based on the co-occurence of popular terms within the user's RSS and Twitter indexes.

The process by which Buzzer generates a set of ranked RSS stories is presented in detail by the algorithm in Figure 2. Given a user, u, and a set of RSS feeds, r, the system first extracts the latest RSS articles, R, and Twitter tweets, T and separately indexes each article and tweet to produce two Lucene indexes. The resulting index terms are then extracted from these RSS and Twitter indexes as the basis to produce RSS and Twitter term vectors, M_R and M_T, respectively.

Next, we identify the set of terms, t, that co-occur in M_T and M_R; these are the words that are present in the latest tweets and the most recent RSS stories and they provide the basis for our recommendation technique. Each term, t_i, is used as a query against the RSS index to retrieve the set of articles A that contain t_i along with their associated TF-IDF score [7] . Thus each co-occuring t_i is associated with a set of articles $A_1, ...A_n$, which contain t_i, and the TF-IDF score for t_i in each of $A_1, ...A_n$ to produce a matrix as shown in Figure 3.

To calculate an overall score for each article we simply compute the sum of the TF-IDF scores across all of the terms associated with that article as per Equation 1. In this way, articles which contain many tweet terms with high TF-IDF scores are preferred to articles that contain fewer tweet terms with lower TF-IDF scores. Obviously this is a simple scoring mechanism but it does serve to provide a straightforward and justifiable starting point.

$$Score(A_i) = \sum_{\forall t_i} element(A_i, t_i) \qquad (1)$$

Finally, producing the recommendation is a simple matter of selecting the top K articles with the highest scores.

2.2 Example Session & User Interface

In this section, we will describe a typical usage scenario. The user logs into Buzzer using their Twitter login details (this is used for the Twitter API). The user then sets up their preferences for RSS feeds and recommendation strategy. The system then collects the latest RSS and Twitter data and makes a recommended Buzzer feed. The user is presented with a set of articles ranked by their relevance score. The user can select from a range of three *recommendation strategies*, as follows:

1. *Public-Rank* - this strategy used the basic technique describe above but mined tweets from the public timeline (that is, the most recent public tweets across the entire Twitter user-base).

2. *Friends-Rank* - this strategy mined tweets from the user's Twitter friends.

```
u: user, r: rss addresses, T: tweets,
R: rss articles, L_T: lucene tweet index,
L_R: lucene rss index, M_T: tweet terms map,
M_R: rss terms map, C: co-occuring terms map,
ArticleMatrix: matrix of terms and articles
RecList: buzzer recommendation feed

define BuzzerRecommender(u, r)
  Loop (every x minutes or on refresh) Do
    T ← getTweets(u)
    R ← getRSSFeeds(r)
    L_T(u) ← indexTweets(T)
    L_R(u) ← indexFeeds(R)
    M_T ← getTweetTerms(L_T (u))
    M_R ← getRSSTerms(L_R (u))
    C  ← M_R ∩ M_T

    For each t_i in C Do
      A ← getArticles(t_i, A_j, L_R)

      For each A_j in A Do
        ArticleMatrix(t_i, A_j) ← TFIDF(t_i, A_j, L_R)
      End
    End

    RecList ← TopK_{∀ Aj in ArticleMatrix} Score(A_j)
    return RecList
  End
End
```

Figure 2: High-level algorithm

Term Queries / Articles	A_0	A_1	A_2	A_3	A_4	A_5
T_0	0.12		0.3			
T_1		0.222	0.33	0.10		
T_2			0.532		0.172	
T_3			0.15	0.412	0.41	
T_4					0.3	0.788
T_5	0.345		0.312			
$\sum$	0.465	0.222	1.624	0.512	0.882	0.788

Figure 3: Buzzer's co-occurence matrix: each cell contains the Lucene TF-IDF score (from the RSS index) of the given term in the given article.

3. *Content-Rank* - this benchmark strategy did not use Twitter but instead ranked articles based on term frequency alone, by scoring articles according to the frequency of occurence of the top-100 RSS terms.

Figure 4 shows an example set of recommended articles based on the users selected recommendation strategy (in this case, the user has selected the *Friends-Rank* approach, which uses their own friends' Twitter content).

The resulting articles include a hyperlinked title, the main content of the RSS article (usually a condensed / sample of the full article), and a set of explanatory metadata from the Buzzer system, namely the recommendation score, and the associated terms that were used to gather that particular article (See Figure 4). This explanation aids the users understanding as to why the system chose to rank a certain article in a certain way.

The results page also shows a standard term/frequency tag cloud that includes terms ordered and sized based on the frequency of each term. This is also useful in explaining to the user the term space that the results were derived from. For example, if the user has selected a twitter-based strategy, such as using the public feed, these terms are the co-occuring terms between the specified RSS feeds of that user, and the Twitter database. The frequency is determined based on these co-occuring terms' frequencies in the Twitter database.

3. PRELIMINARY EVALUATION

The basic Buzzer system provides users with an alternative way to access RSS stories. They can use the Buzzer interface as an RSS reader or, alternatively, the Buzzer recommendation lists can be published as RSS feeds themselves and thus incorporated, as a summary feed, into the user's normal RSS reader.

Ultimately we are interested in how well the recommendations produced by this novel news recommendation technique are received by end-users. To test this we have carried out a preliminary evaluation using a small group of 10 participants, over a period of 5 days. Each participant configured the system by providing up to 10 of their favourite RSS feeds along with their Twitter account information, and periodically selected a different recommendation strategy to re-rank the content.

During the study users were asked to use Buzzer as their RSS reader. To begin with, they were asked explore the different types of recommendation strategies at their leisure. As a basic evaluation measure we focused on the click-through frequency for articles across the 3 different recommendation strategies. The users were told the names of each of the recommendation strategies, and were given a simple explanation of each upon initial registration for the evaluation.

The results shown in Figure 5 (A) represent the average per-user click-throughs for each of the recommendation strategies and there is a clear difference in the behaviour of users when comparing the Twitter-based strategies to the default content-based technique. For example, we see that, on average, the Twitter-based strategies resulted in between 8.3 and 10.4 click-throughs per user compared with only 5.8 article click-throughs for the content-based strategy; a relative click-through increase of between 30% and 45% for the Twitter-based strategies.

We also see that these usage results suggest a preference for the *Friends-Rank* recommendations compared to the recommendations derived from Twitter's Public Timeline (*Public-Rank*). This suggests that users are more likely to tune in to the themes and topics of interest to their friends than those that might be of interest to the Twitter public at large. Interestingly, however, this is at odds with the feedback provided by participants as part of a post-

Figure 4: Screenshot of results (articles reranked based on users friends' Tweets)

trial questionnaire, which indicated a strong preference for the *Public-Rank* articles as shown in Figure 5 (B); 67% of users indicated a preference for *Public-Rank* recommendations compared with 22% of users indicating a preference for *Friends-Rank* recommendations. Incidentally, none of the participants favoured the *Content-rank* strategy and 11% didn't know which strategy they preferred.

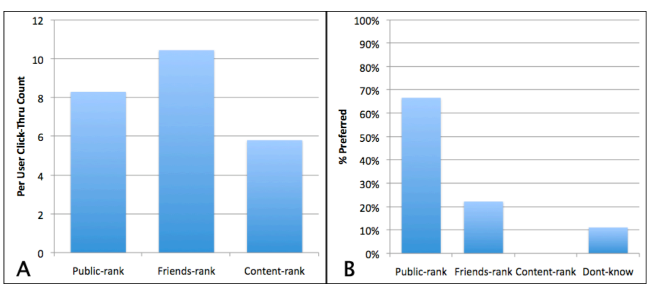

Figure 5: A) Average per user click-through for different recommendation strategies. B) Preferred recommendation strategies

Interestingly when we compared the ratio of *Public-Rank* to *Friends-Rank* click-throughs to the number of friends the user follows on Twitter we found a correlation coefficient of -0.89, suggesting that users with more friends tend to benefit more so from the *Friends-Rank* recommendations, compared to the recommendations derived from the public timeline.

Although our initial user study was preliminary, the Buzzer recommender system was well received and we found that participants preferred the Twitter-based recommendation strategies. The Buzzer feed provided the participants with interesting and topical articles which were viewed in greater detail by clicking-through to the full article text.

4. CONCLUSIONS

In this short paper, we have outlined a novel news recommendation technique that harnesses real-time Twitter data as the basis for ranking and recommending articles from a collection of RSS feeds. A prototype system has been developed and deployed and early evaluation results suggest that users do benefit from the recommendations that are derived from the Twitter data.

Looking to the future, the Buzzer system provides considerable opportunity for further innovation and experimentation as a test-bed for real-time recommendation. We are currently extending the feedback options that are presented to users to facilitate negative as well as positive feedback. There are also many ways in which the content-based recommendation technique may be improved. For example, moving from single terms to more complex phrases, which may improve the recommendation ranking. We also hope to focus on perfecting the scoring technique. Moreover, the Buzzer system has the potential to act as a collaborative news service with a number of opportunities to provide additional recommendation services such as recommending new RSS feeds to users or recommending relevant people to follow on Twitter.

5. REFERENCES

[1] Daniel Billsus and Michael J. Pazzani. A personal news agent that talks, learns and explains. In *In Proceedings of the Third International Conference on Autonomous Agents*, pages 268–275. ACM Press, 1999.

[2] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, NY, USA, 2007. ACM.

[3] Tomonari Kamba, Krishna Bharat, and Michael C. Albers. The krakatoa chronicle - an interactive, personalized, newspaper on the web. In *Proceedings of the Fourth International World Wide Web Conference*, pages 159–170, 1995.

[4] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24, NY, USA, 2008. ACM.

[5] Ken Lang. Newsweeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[6] Michael Pazzani and Daniel Billsus. Content-based recommendation systems. *The Adaptive Web*, pages 325–341, 2007.

[7] G. Salton and M. J. Mcgill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[8] B. Smyth and P. Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3):53 – 59, 2000.