Buzzer - Online Real-Time Topical News Article and Source Recommender

Owen Phelan, Kevin McCarthy and Barry Smyth

School of Computer Science and Informatics,
UCD Dublin, Ireland
firstname.lastname@ucd.ie

Abstract. The significant growth of media and user-generated content online has allowed for the widespread adoption of recommender systems due to their proven ability to reduce the workload of a user and personalise content. In this paper, we describe our prototype system called Buzzer, which harnesses real-time micro-blogging activity, such as Twitter, as the basis for promoting personalised content, such as news articles, from RSS feeds. We also introduce several new features, that include a technique for recommending community articles from the pooled resources of all system users and also a mechanism for recommending source RSS feeds to which the user does not subscribe.

1 Introduction

The Web is comprised of billions of items (e.g. video, blog posts, websites, tweets, images, users, documents), and while there is infinite potential for the *creation* of these items (by the explosion of internet connected device availability), more work is needed to develop novel ways and means of *using* and *promoting* these items. Recommender systems promote new items to users for item discovery, as well as making the link between user and item, with the potential of serendipitous exploration by the user to related items.

One example of popular online items are *news and current events* sources. These websites typically contain news articles ranked by recency and the considerations of an editorial team, however little is done to present the user with topical and novel news material ranked on real-life events and conversations from the public domain. For example, a glance at any major news organization's website during and immediately after the Inauguration of U.S. President Barack Obama would have efficiently described the event, mostly because of its perceived novelty. However, little was done by those organizations to promote it as a popular news item on the basis of explicit popularity among consumers and viewers. One disadvantage this promotion by the major news organizations of this single event is, of course, that the many hundreds of other news items of the day may be muted from public attention.

There is a long history of using recommender systems techniques to help users to navigate through the myriad of news stories that are written and published everyday [1, 2, 6]. These systems can 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 [7] and collaborative filtering techniques have been used to good effect and the recent growth of services such as *Digg*[1], a social bookmarking system, demonstrate the value of collaborative filtering recommendation techniques when it comes to delivering 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 suited. Many current recommender systems are limited in their ability to identify topical stories because they typically rely on a critical mass of user consumption before such stories can be recognised (cold-start problem). Such an example is *Google News*[2] [2], which, although a successful system, still relies on click-histories of users for personalisation without much consideration for the actual content itself (this will be discussed further in the next section).

In our earlier work [8], we developed a prototype system, called Buzzer, which takes advantage of a novel content-based approach for finding news stories among a users' set list of feeds. This content-based approach harnesses a popular micro-blogging service, such as *Twitter* (*www.twitter.com*) [3, 5], as a source of current and topical news. Co-occurring terms between the current Twitter trends on either the public feed or among users' friends are used as a basis for recommending content from a users personalised list of sources. In this paper, we extend the system with several new recommendation features. Firstly, we recommend content from other users through the use of a community pool of articles, and secondly, we recommend new RSS sources the user may not be aware of, but that are nonetheless relevant to them.

2 Recommending News & Sources

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 for breaking news. However, as users subscribe to tens of RSS feeds this introduces a niche information overload problem [10].

Our primary focus is to find a means of recommending content based on current topicality on super-active and dynamic social communications sites such as Twitter. This site is a so-called *micro-blogging* service that allows users to submit their own short (maximum of 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 popular growth [3], and also

[1] Digg - http://www.digg.com
[2] Google News - http://news.google.com

because of its ability to provide access to thoughts, intentions and activities of millions of users in real-time. Buzzer mines these tweets with the intention of discovering emerging topics and breaking events, and then 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.

In this paper, we explore extending this technique to discovering content among the pool of articles across the user-space. We also introduce a technique for recommending RSS feeds themselves.

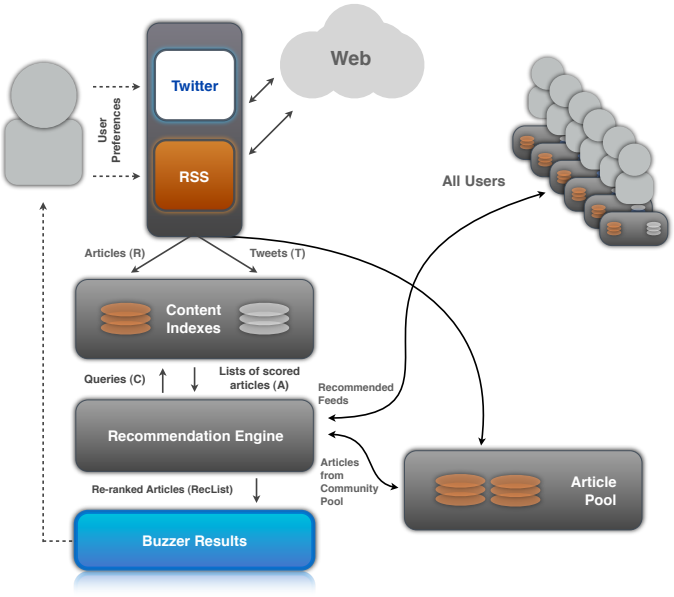

Fig. 1. Buzzer architecture: The diagram shows the user-level architecture, where the user has specified sources of information, and is presented after a recommendation step with a set of recommended results. This recommendation step also analyses content from other users' on the system to find and discover feeds and also recommend content in a pool of articles from the entire community.

2.1 Architecture & Recommendation Approaches

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 for recommending news articles have proven successful in the past. Perhaps the earliest example of a news recommendation service, *Krakatoa Chronicle* [4], 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. Although Billsus and Pazzani [7] 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 looks at common co-occuring terms among Twitter and RSS.

The architecture of the Buzzer system (Figure 1) comprises three basic components:

1. The *Web front-end* manages the basic user registration and login 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). The interface presents multiple feeds of personalised and community gathered articles as well as new feed sources.
2. The *Content Gatherer & Indexer* components are responsible for mining and indexing the appropriate Twitter and RSS information, given the user's configuration settings. This component also manages the community pool of articles.
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. It has also been extended to compute similarities among users' co-occuring terms, gather recommended feed data, and search a pooled index of the community's articles to discover new items that the case user may not subscribe to or receive.

The process by which Buzzer generates a set of ranked RSS stories is presented in detail by the algorithm in Figure 2(a). 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[3] 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 along with their associated TF-IDF score [9, 11] . Thus each co-occuring t_i is associated with a set of articles $A_1, ...A_n$, which contain t, and the TF-IDF score for t in each of $A_1, ...A_n$ to produce a matrix as shown in Figure 3.

[3] Apache Lucene - http://apache.lucene.org

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

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

This technique is used to recommend articles from the users personal RSS articles but is also applied in recommending items from the community pool of articles. For the community recommendation technique, the RSS articles, R, represents the set of pooled community articles rather than just the user's personal articles. The rest of the recommendation process remains the same. In the current version of Buzzer, both sets of recommendations are provided to the user through the web front-end as shown in the next section.

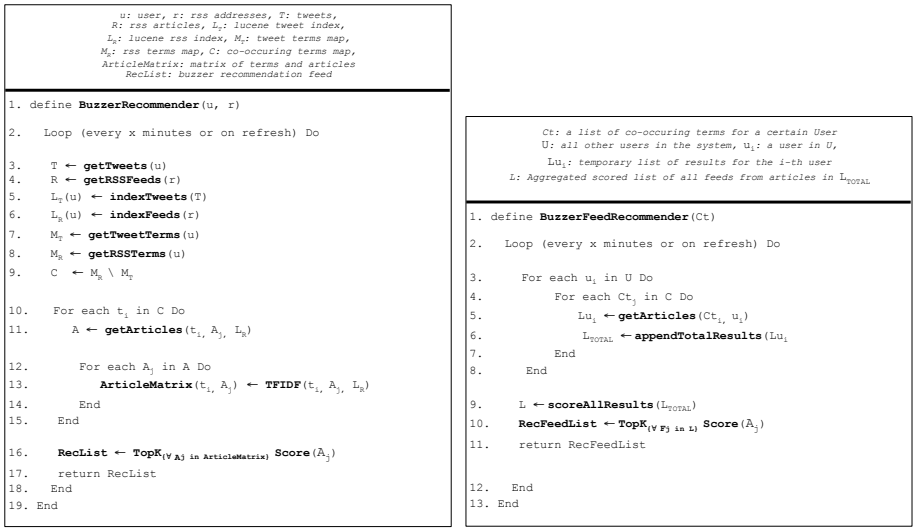

Fig. 2. Two main algorithms employed in the recommendation engine. On the left, Figure 2(a), this is the high-level description of how the user interacts and is returned feeds of article content. The algorithm on the right, Figure 2(b), shows how we recommend the feed addresses.

Figure 2(b) describes the method by which Buzzer recommends new RSS feeds to users based on querying each other users indexes to find new articles. The system queries all of the other users indexes using the same criteria as when it scans a given users index for articles. It aggregates the results in a

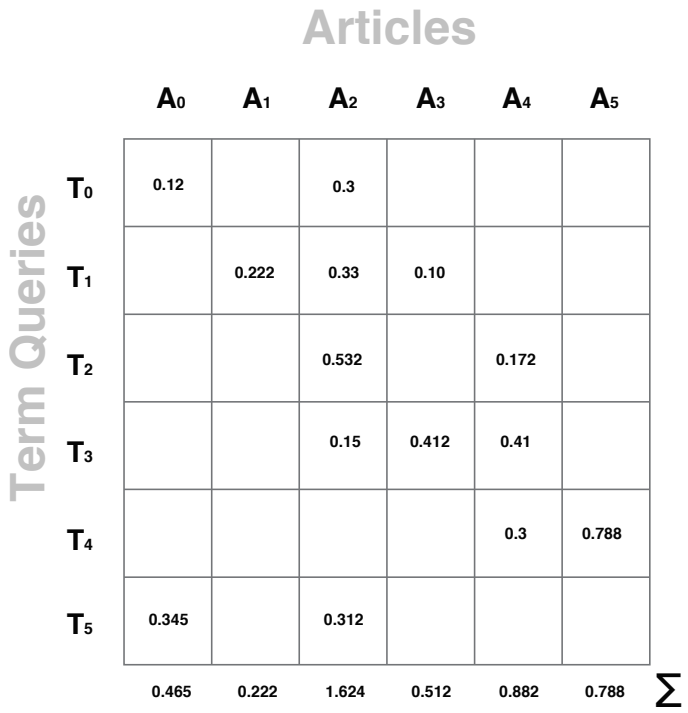

Fig. 3. Buzzer's co-occurence matrix: each cell contains the Lucene TF-IDF score (from the RSS index) of the given term in the given article.

similar fashion and returns parent RSS Feed addresses (example: CNN Headlines www.cnn.com/headlines.rss, etc.). These addresses are returned to the user in a list in the User Preferences page of the system (See Figure 5). Each of these feeds are new, as in the current user has not selected to follow them before. We discard feeds that are already part of the users list of feeds.

2.2 Example Session & User Interface

In this section, we will describe some usage scenarios of the system. The user logs into the system using their Twitter login details[4] (used by the Twitter API). The user then configures the system by providing the RSS feeds and a selected strategy. Users can choose a strategy which examines the public Twitter feed or their personal friends' feeds, but can also select not to use Twitter at all. The system then collects the latest RSS and Twitter data and makes a set of recommended Buzzer feeds for that user. The system gathers the top 100 frequent co-occuring terms between the articles and the tweets contained in the

[4] As mentioned earlier, the user does not have to provide their Twitter login as access to the twitter public timeline does not require it.

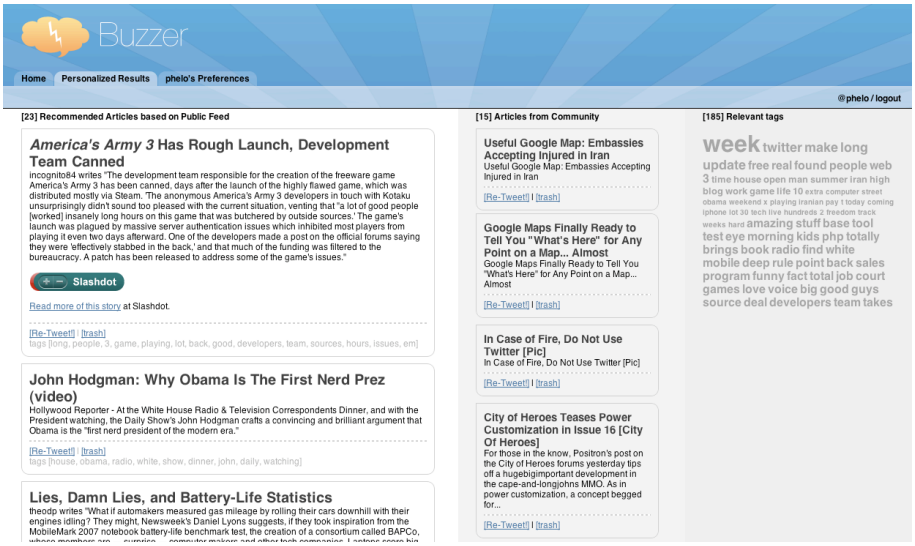

Fig. 4. A screenshot of recommended articles for a given user.

user index. This is a basis of inferring relevant and novel descriptive terms of a user, and we can use this to both search article indexes and also compute user-user similarities.

The screenshot in Figure 4 shows the recommendation page of Buzzer. The first column of the interface contains the personalised articles that have been recommended using the user-specified content. The personalised article content also shows associated tags with each article, which aids the user's understanding as to why the system chose to rank a certain article in a certain way. The second (middle) column shows the recommended articles that are from the community pool of articles, that have been gathered based on the co-occuring terms searched across the pool's index. Each of the articles in this column do not appear in the users feeds, they are new articles that the user would not see in the primary column. The articles in these columns have been ranked based on their compound relevance score, as seen in the co-occurance matrix in Figure 3. The third column 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 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.

The second screenshot (Figure 5) depicts the user preferences page on Buzzer. This page includes preferences such as their chosen personal RSS feeds, as well as options for their Twitter influences (either the public, or friends feeds, or no Twitter influence at all). More importantly, the page provides the user with a list

of recommended RSS feeds based on the algorithm discussed previously. These feeds are most relevant to the user, but are also new in the sense that the user does not already subscribe to them.

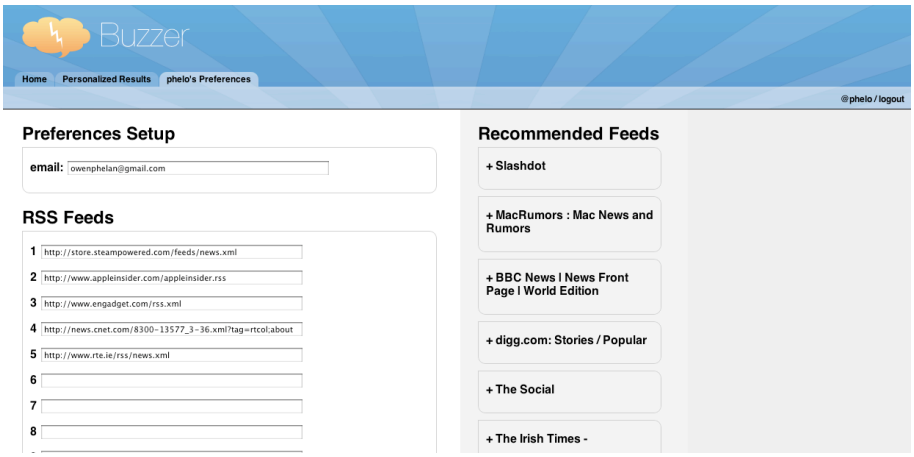

Fig. 5. A screenshot of the user preferences page, with feed recommendations in the second column that can be easily added to the user's list on the left.

3 Discussion of Evaluation

We ran a small-scale evaluation with 10 participants using a prototype system which included the basic content-based system for individual users (the news feed and pooled article recommendations were unavailable here) [8]. Users could 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. Each participant configured the system by providing up to 10 of their favourite RSS feeds along with their Twitter account information. The system was further configured to provide users with access to 3 different recommendation strategies, including two variations on our Twitter-based ranking technique, as follows:

1. *Public-Rank* - this strategy used the basic technique described 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 its tweets from the user's Twitter friends.
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.

To begin with the users were asked to 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 resulting usage patterns were interesting. For instance we found 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 found a preference among the users for the *Friends-Rank* recommendations compared to the recommendations derived from Twitter's Public Timeline (*Public-Rank*). This suggests that users were 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-trial questionnaire, which indicated a strong preference for the *Public-Rank* articles; 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.

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 be more inclined to benefit 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.

We are currently preparing a second live user study to evaluate the extensions to Buzzer described in this paper. We believe that the enhancements will improve user satisfaction by recommending a more diverse set of topical news articles as well as introducing users to new news feeds, to which they were previously not subscribed.

4 Conclusions

This paper has outlined a novel news recommendation technique that harnesses real-time Twitter data as the basis for ranking and recommending articles and sources from a collection of RSS feeds. The prototype system has been developed to show a proof of concept, along with an extensible architecture to adapt future work. In this paper, we have introduced community-based article feeds and news feed recommendations to the Buzzer system. Users are not only recommended articles from their own feeds but also interesting articles from the community of users. Now users are also recommended RSS news feeds to which they were

not previously subscribed. To this end, we can see the Buzzer system providing considerable opportunity for further innovation and experimentation as a test-bed for real-time recommendation. There are many ways in which the content-based recommendation technique may be improved, rather than using single terms, we hope to adopt n-gram analysis, which may provide a way of capturing more meaningful phrases from Twitter data and RSS articles to further improve the recommendation ranking. We also wish to move into recommending friends and potential contacts with services such as Twitter, and indeed explore further content analysis of individual users' indexes as a different support. 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 relevant people to follow on Twitter.

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. Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 271–280, New York, NY, USA, 2007. ACM.
3. 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.
4. Tomonari Kamba, Krishna Bharat, and Michael C. Albers. The krakatoa chronicle - an interactive, personalized, newspaper on the web. In *In Proceedings of the Fourth International World Wide Web Conference*, pages 159–170, 1995.
5. 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.
6. 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.
7. Michael Pazzani and Daniel Billsus. Content-based recommendation systems. *The Adaptive Web*, pages 325–341, 2007.
8. Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *ACM RecSys 2009 (Submitted)*. ACM, October 2009.
9. Fabrizio Sebastiani and Consiglio Nazionale Delle Ricerche. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.
10. B. Smyth and P. Cotter. A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3):53 – 59, 2000.
11. Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.