Research Repository UCD

Title	Terms of a feather : content-based news discovery and recommendation using Twitter
Authors(s)	Phelan, Owen; McCarthy, Kevin; Smyth, Barry; Bennett, Mike
Publication date	2011-04-19
Publication information	Clough, P.; Foley, C.; Gurrin, C.; Jones, G.J.F.; Kraaij, W.; Lee, H.; Murdoch, V. (eds.). Advances in Information Retrieval : 33rd European Conference on IR Resarch, ECIR 2011, Dublin, Ireland, April 18-21, 2011, Proceedings
Conference details	Paper presented at the 33rd European Conference on Information Retrieval (ECIR-11), 18-21 April, 2010, DCU, Dublin, Ireland
Publisher	Springer
Link to online version	http://dx.doi.org/10.1007/978-3-642-20161-5_44
Item record/more information	http://hdl.handle.net/10197/2950
Publisher's statement	The final publication is available at springerlink.com
Publisher's version (DOI)	10.1007/978-3-642-20161-5_44

Terms of a Feather: Content-based News Recommendation and Discovery using Twitter*

Owen Phelan, Kevin McCarthy, Mike Bennett and Barry Smyth

CLARITY: Centre for Sensor Web Technologies
School Of Computer Science & Informatics
University College Dublin
owen.phelan@ucd.ie

Abstract. User-generated content has dominated the web's recent growth and today the so-called *real-time web* provides us with unprecedented access to the real-time opinions, views, and ratings of millions of users. For example, Twitter's 200m+ users are generating in the region of 1000+ tweets per second. In this work, we propose that this data can be harnessed as a useful source of recommendation knowledge. We describe a social news service called *Buzzer* that is capable of adapting to the conversations that are taking place on Twitter to ranking personal RSS subscriptions. This is achieved by a content-based approach of mining trending terms from both the public Twitter timeline and from the timeline of tweets published by a user's own Twitter friend subscriptions. We also present results of a live-user evaluation which demonstrates how these ranking strategies can add better item filtering and discovery value to conventional recency-based RSS ranking techniques.

1 Introduction

The real-time web (RTW) is emerging as new technologies enable a growing number of users to share information in multi-dimensional contexts. Sites such as *Twitter* (`www.twitter.com`), *Foursquare* (`www.foursquare.com`) are platforms for real-time blogging, messaging and live video broadcasting to friends and a wider global audience. Companies can get instantaneous feedback on products and services from RTW sites such as *Blippr* (`www.blippr.com`). Our research focusses on the real-time web, in all of its various forms, as a potentially powerful source of recommendation data. For example, we consider the possibility of mining user profiles based on their Twitter postings. If so, we can use this profile information as a way to rank items, user recommendation, products and services for these users, even in the absence of more traditional forms of preference data or transaction histories [6]. We may also provide a practical solution to the cold-start problem [13] of sparse profiles of users' interests, an issue that has plagued many item discovery and recommender systems to date.

Online news is a well-trodden research field, with many good reasons why IR and AI techniques have the potential to improve the way we consume news online. For a start there is the sheer volume of news stories that users must deal with, plus we have varied tastes and preferences with respect to what we are interesting in reading about. At the same time, news is a biased form of media that is increasingly driven by the stories that are capable of selling advertising. Niche stories that may be of interest to a small portion of readers often get buried. All of this has contributed to a long history of using *recommender systems* to help users navigate through the sea of stories that are published everyday based on learned profiles of users. For example, *Google News* (`http://news.google.com`) is a topically segregated mashup of a number of feeds, with automatic ranking strategies based on user interactions (click-histories & click-thrus) [5]. It is an example of a hybrid technique for news recommendation, as it utilises a user's search keywords from Google itself as a support for explicit ratings. Another popular example is *Digg* (`www.digg.com`), whose webpage rating service generally leads to a high overlap of selected topical news items [12].

This paper extends some of the previous work presented in [15], which described an early prototype of the Buzzer system, in two ways. First, we describe a more comprehensive and robust recommendation framework that has been extended both in terms of the different sources of recommendation knowledge and the recommendation strategies that it users. Secondly, we describe the result of a live-user evaluation with 35 users over a 1 month period, and based on more than 30,000 news stories and in excess of 50 million Twitter messages, the results of which describe interesting usage patterns compared to recency benchmarks.

2 Background

Many research opportunities remain when considering how to adapt recommendation techniques to tackle the so-called *information explosion* on the web. Digg, for example, mines implicit click-thrus of articles as well as ratings and user-tagging folksonomies as a basis of content retrieval for users [12]. One of the byproducts of Digg's operation is that users' browsing and sharing activities generally involve socially or temporally topical items, so as such it has been branded as a sort of news service [12]. Difficulties arise where it is necessary for many users to implicitly (click, share, tag) and explicitly (star or digg) rate items many times for those items to emerge as topical things. Also, there would be considerable item churn, that is, the corpus of data is constantly updating and item relevances are constantly fluctuating. The space of documents themselves could be defined by Brusilovsky and Henze as an *Open Corpus Adaptive Hypermedia System* in that there is an open corpus of documents (though topic specific), that can constantly change and expand [3].

Google News is a popular service that uses (mostly unpublished) recommendation techniques to filter 4500 partner news providers to present an aggregated view for registered users of popular and topical content [5]. Items are usually between several seconds to 30 days old, and appear on the "front page" based

on click-thrus and key-word term Google—search queries. The ranking itself is mostly based on click-thru rates of items, higher ranked items have more clicks. Issues arise with new and topical items struggling to get to the "front page", as it is necessary for a critical-mass of clicks from many users. Das et al. [5] mostly describe scalability of the system as an issue with the service, and propose several techniques know to be capable of dealing with such issues. These included MinHash, Probabilistic Latent Semantic Indexing (PLSI) and Latent Semantic Hashing (LSH) as component algorithms in an overall hybrid system.

Content-based approaches are widely discussed in many branches of recommender systems [2, 9, 13, 14]. Examples such as the News@Hand semantic system by Cantador et al. [4] show encouraging moves towards considering the content of the news items themselves. The authors use semantic annotation and ontologies to structure items into groups, while matching this to similarly structured user profiles of preferred items — unfortunately the success of these are based on the quality and existence of established domain ontologies. Our approach is to look at the most atomic components of the content, the individual terms themselves.

There is currently considerable research attention being paid to Twitter and the real-time web in general. RTW services provide access to new types of information and the real-time nature of these data streams provide as many opportunities as they do challenges. In addition, companies like Twitter and Yahoo have adopted a very open approach to making their data available and Twitter's developer API provides researchers with access to a huge volume of information for example. It is no surprise then that the recent literature includes analyses of Twitter's real-time data, largely with a view to developing an early understanding of why and how people are using services like Twitter [7, 8, 11]. For instance, the work of Kwak et al. [11] describes a very comprehensive analysis of Twitter users and Twitter usage, covering almost 42m users, nearly 1.5bn social connections, and over 100m tweets. In this work, the authors have examined reciprocity and homophily among Twitter users, they have compared a number of different ways to evaluate user influence, as well as investigating how information diffuses through the Twitter ecosystem as a result of social relationships and retweeting behaviour. Similarly, Krishnamurthy et al. identify classes of Twitter users based on behaviours and geographical dispersion [10]. They highlight the process of producing and consuming content based on retweet actions, where users source and disseminate information through the network.

We are interested in the potential to use near-ubiquitous user-generated content as a source of preference and profiling information in order to drive recommendation, as such in this research context Buzzer is termed a content-based recommender. User-generated content is inherently noisy but it is plentiful, and recently researchers have started to consider its utility in recommendation. There has been some recent work [17] on the role of tags in recommender systems, and researchers have also started to leverage user-generated reviews as a way to recommend and filter products and services. For example, Acair et al. look at the use of user-generated movie reviews from IMDb as part of a movie recommender system [1] and similar ideas are discussed in [18].

Both of these instances of related work look to mine review content as an additional source of recommendation knowledge (in a similar way to the content-boosted collaborative filtering technique in Melville et al. [13]), but they rely on the availability of detailed item reviews, which may run to hundreds of words but which may not always be available. In this paper, we consider trending and emerging topics on user-generated content sites like twitter as a way to automatically derive recommendation data for topical news and web-item discovery.
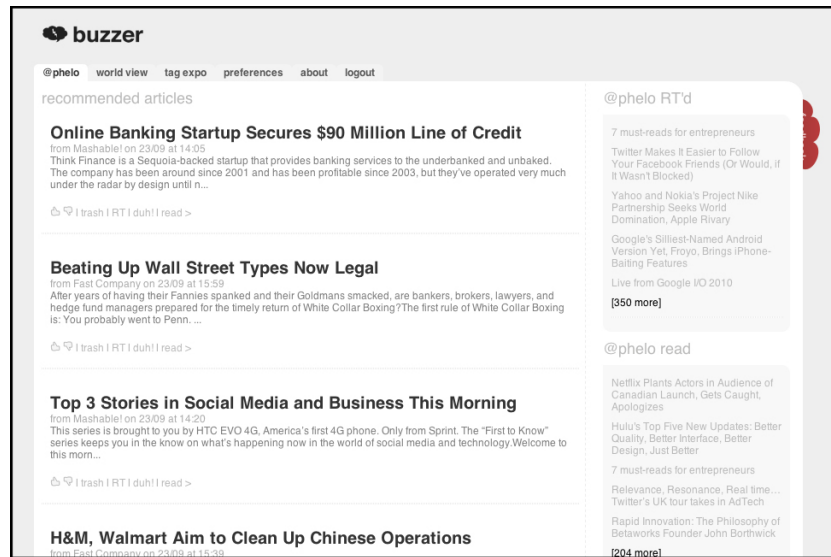

Fig. 1: *A screenshot of Buzzer, with personalized news results for a given user.*

3   The Buzzer System

People talk about news and events on Twitter all of the time. They share web pages about news stories. They express their views on recent stories. They even report on emerging news stories as they happen. Surely then it is logical to think of Twitter as a source of news information and news preferences? The challenge of course is that Twitter is borderline chaotic: tweets are little more than impressions of information through fleeting moments of time. Can we really hope to make sense of this signal and noise and harness the chaos as a way to search, filter and rank news stories? This is the objective of the research presented in this paper. Specifically, we aim to mine Twitter information, from both public data streams, and the streams of related users, as a way to identify discriminating terms that are capable of being used to highlight breaking and interesting news stories.

As such the *Buzzer* system adopts a content-based technique to recommending news articles, but instead of using structured user profiles we use unstructured real-time feeds from Twitter. In effect, the user messages (*tweets*) themselves act as an implicit ratings system for promoting and filtering content for retrieval in a large space of items of varied topicality or relevance to users.
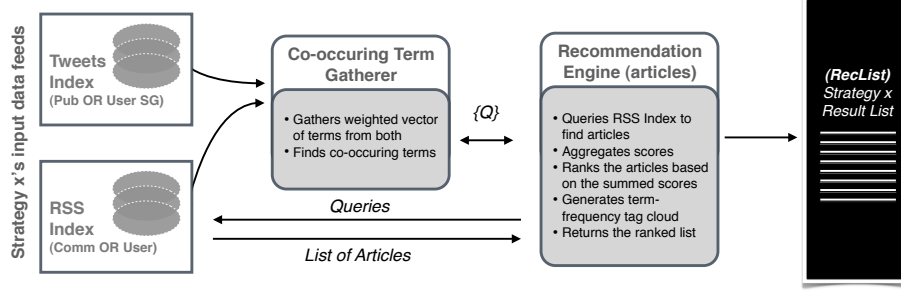

Fig. 2: *Generating results for a given strategy. System mines a specified RSS and Twitter source and uses the co-occuring technique described to generate a set of results, which will be interleaved with other sets to produce the final list shown to users.*

3.1 System Architecture

The high-level Buzzer system architecture is presented in Figure 2. In summary, Buzzer generates two content indexes, one from Twitter (including public tweets and Buzzer-user tweets as discussed below) and one from the RSS feeds of Buzzer users. Buzzer looks for correlations between the terms that are present in tweets and RSS articles and ranks articles accordingly. In this way, articles with content that appear to match the content of recent Twitter chatter (whether public or user related) will receive high scores during recommendation. Figure 1 shows a sample list of recommendations for a particular user. Buzzer itself is developed as a web application and can take the place of a user's normal RSS reader: the user continues to have access to their favourite RSS feeds but in addition, by syncing Buzzer with their Twitter account, they have the potential to benefit from a more informative ranking of news stories based on their inferred interests.

3.2 Strategies

Each Buzzer user brings two types of information to the system — (1) their RSS feeds; (2) their Twitter social graph — and this suggests a number of different ways of combining tweets and RSS during recommendation. In this paper, we explore 4 different news retrieval strategies ($S1 - S4$) as outlined in Figure 3. For example, stories/articles can be mined from a user's personal RSS feeds or from the RSS feeds of the wider Buzzer community. Moreover, stories can be
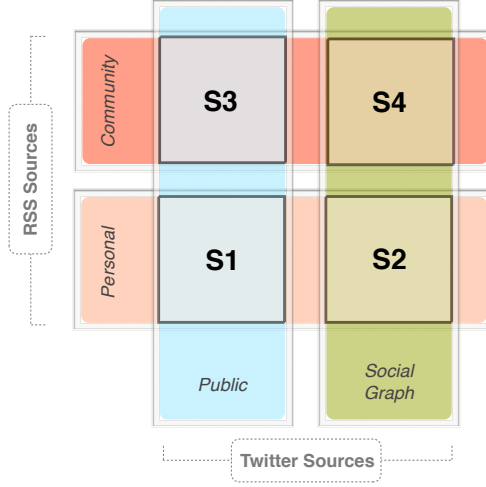
Fig. 3: *Buzzer Strategy Matrix*

ranked based on the tweets of the user's own Twitter social graph, that is the tweets of their friends and followers, or from the tweets of the public Twitter timeline. This gives us 4 different retrieval strategies as follows (as visualized in Figure 3):

1. *S1 — Public Twitter Feed / Personal RSS Articles*: mine tweets from the public timeline, searches the user's index of RSS items.
2. *S2 — Friends Twitter Feed / Personal RSS Articles*: mine tweets from people the user follows, searches the user's index of RSS items.
3. *S3 — Public Twitter Feed / Community Pool of RSS Articles*: mine tweets from the public timeline, searches the entire space of RSS items gathered from all users's subscriptions.
4. *S4 — Friends Twitter Feed / Community Pool of RSS Articles*: mine tweets from the public timeline, searches the entire space of RSS items gathered from all users's subscriptions.

In the evaluation section of this paper we will add a 5th strategy as a standard benchmark (ranking stories by recency).

As explained in the pseudo-code in Figure 3(a), the system generates four distinct sets of results based on varied inputs. Given a user, u, and a set of RSS articles, R, and a set of Tweets, T, the system separately indexes both to produce two *Lucene* indexes [1]. 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.

---

[1] Lucene (`http://lucene.apache.org`) is an open-source search API, it proved useful when dealing with efficiently storing these documents and has native TF-IDF support

```
1. define RecommendArticles(R, T, k)

2.     L_T ← indexTweets(T)
3.     L_R ← indexFeeds(R)

4.     M_T ← mineTweetTerms(L_T)
5.     M_R ← mineRSSTerms(L_R)

6.     Q  ← findCoOccuringTerms(M_R, M_T)


7.     For each q_i in Q Do
8.         a ← retrieveArticles(q_i, a_j, L_R)

9.         For each a_j in A Do
10.            S_j ← S_j + TFIDF(q_i, a_j, L_R)
11.        End
12.     End

13.     RecList ← Rank All a_j by Score S_j

14.     return top-k(RecList, k)
15.    End
16. End
```

	a_0	a_1	a_2	a_3
q_0	0.12			
q_1		0.23	0.37	
q_2				0.3
q_3	0.24	0.22		
q_4	0.11		0.2	
q_5		0.15		
Σ	0.47	0.6	0.57	0.3

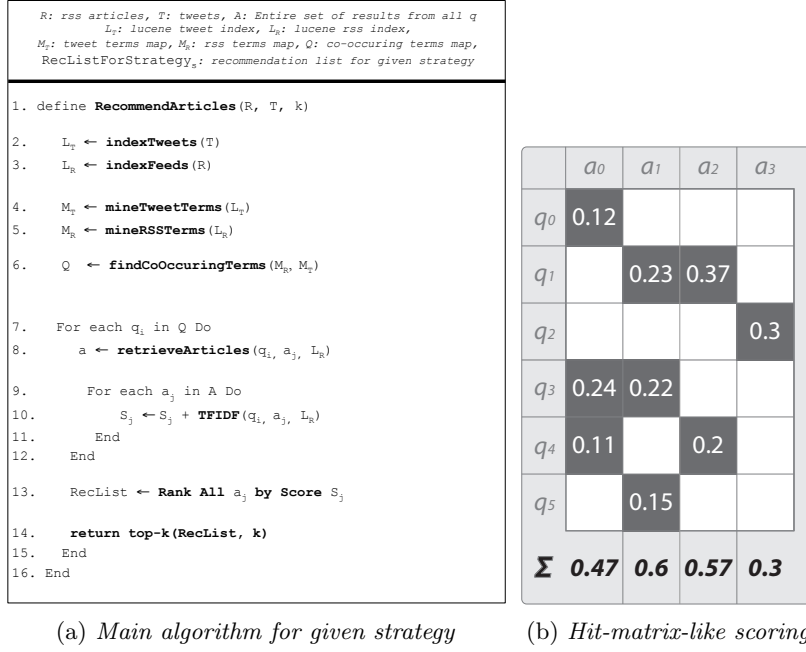(a) *Main algorithm for given strategy*   (b) *Hit-matrix-like scoring*

Fig. 4

We then identify the set of terms, Q, 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 technique. Each term, q_i, is used as a query against the RSS index to retrieve the set of articles A that contain q_i along with their associated TF-IDF (term frequency inverse document frequency) score [16]. Thus each co-occurring term, q_i is associated with a set of articles $a_1, ...a_n$, which contain t, and the TF-IDF score for q_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 higher TF-IDF scores are preferred to articles that contain fewer tweet terms with lower TF-IDF scores. Finally, retrieving the recommendation list is a simple matter of selecting the top k articles with the highest scores. Each time Buzzer mines an individual feed from a source, the articles are copied into both the user's individual article pool, and a community pool. Each article has a differing relevance score in either pool, as their TF-IDF score changes based on the other content in the local directory with it.

$$Score(a_i) = \sum_{\forall q_i} element(a_i, q_i) \tag{1}$$

For a user, four results-lists are generated, and the fifth recency-based list is gathered by collecting the latest to 2-day old articles (as the update windows on each feed can vary). Items from each of these strategies are interleaved into a single results list for presentation to the user (this technique is dependent on our experimental setup, explained further in the next section). Finally, the user is presented with results and is encouraged to click on each item to navigate to the source website to read the rest of its contents. We capture this click-thru as well as other data such as username, the position in the list the result is, the score and other data, and consider the act of clicking it as a metric for a successful recommendation.

4   User Evaluation

We undertook a live-user evaluation of the Buzzer system, designed to examine the recommendation effectiveness of its constituent social, RTW, retrieval and filtering techniques ($S1 - S4$) alongside a more typical recency-based story recommendation strategy ($S5$). Overall our interest is not so much concerned with whether one strategy is superior to others — because in reality we believe that different strategies are likely to have a role to play in news story recommendation — but rather to explore the reaction of users to the combination of strategies.

4.1   Evaluation Setup

As part of this evaluation, we re-developed the original Buzzer system [15] with a more comprehensive interface providing users with access to a full range of news consumption features. Individual users were able to easily add their favourite RSS feeds (or pick from a list of feeds provided by other users) and sync up their Twitter accounts, to provide Buzzer with access to their social graph. In addition, at news reading time users could choose to trash, promote, demote, and even re-tweet specific stories. Moreover, users could opt to consume their news stories from the Buzzer web site and/or sign up to a daily email digest of stories. In this evaluation we focus on the reaction of users to the daily digest of email stories since it provides us with a consistent and reasonably reliable (once-per-day) view of news consumption.

This version of Buzzer was configured to generate news-lists based on a combination of the 5 different recommendation strategies: $S1 - S4$, and $S5$, as described in Section 3. Each daily email digest contained 25 stories in 5 blocks of 5 stories each. Each block of 5 stories was made up of a random order of one story from each of $S1 - S5$; this the first block of 5 stories contained the top-place recommendations from $S1 - S5$, in a random order, the second block contained the second-place stories from $S1 - S5$, in a random order, and so on. We did this to prevent any positional bias, whereby stories from one strategy might always appear ahead of, or below, some other strategy. Thus every email digest contained a mixture of news stories as summarized in Section 3.

The evaluation itself consisted of 35 active users; these were users who had registered with Buzzer, signed up to the email digest, and interacted with the system on at least two occasions. The results presented relate to usage data gathered during the 31 days of March 2010. During this timeframe we gathered a total of 56 million public tweets (for use in strategies $S1$ and $S3$) and 537,307 tweets from the social graphs of the 35 registered users (for use in $S2$ and $S4$).

In addition, the 35 users registered a total of 281 unique RSS feeds as story sources and during the evaluation period these feeds generated a total of 31,137 unique stories/articles. During the evaluation, Buzzer issued 1,085 emails. We considered participants were fairly active Twitter users, with 145 friends, 196 followers and 1241 tweets sent, on average.

4.2 Results

As mentioned above, our primary interest in this evaluation is understanding the response profile of participants across the different recommendation strategies. To begin with, Figure 5(a) presents the total per-strategy click-thrus received for stories across the 31 days of email digests, across the participants. It is interesting to note that, as predicted all of the strategies do receive click-thrus for their recommendations, as expected.

Overall, we can see that strategies $S1$ and $S2$ tend to outperform the other strategies; for example, $S1$ and $S2$ received about 110 click-thrus each, just over 35% more than strategies $S3$ and $S4$, and about about 20% more than the default recency strategy, $S5$. Strategies $S1$, $S2$, and $S5$ retrieve stories from the user's own registered RSS feeds, and so there is a clear preference among the users for stories from these sources. However, stories from these feeds that are retrieved based on real-time web activity ($S1$ and $S2$) attract more click-thrus than when these stories are retrieved based on recency ($S5$). Clearly users are benefiting from the presentation of more relevant stories due to $S1$ and $S2$.

Moreover it is interesting to note that there is little difference between the relevance of stories (as measured by click-thru) ranked by the users own social graph ($S2$) compared to those ranked by the Twitter public at large ($S1$). Of course both of these strategies mine the user's own RSS feeds to begin with and so there is an assumed relevance in relation to these stories, but clearly there is some value, for the end user, in receiving stories ranked by their friends' activities and by the activities of the wider public. Participants responded less frequently to stories ranked highly by strategies $S3$ and $S4$, although it must be said that these strategies still manage to attract about 30% of total click-thrus.

This is perhaps to be expected; for a start, both of these strategies sourced their result lists from RSS feeds that were not part of the user's regular RSS-list; a typical user registered 15 or so RSS feeds as part of their Buzzer sign-up and the stories ranked by $S3$ and $S4$, for a given user, came from the 250+ other feeds contributed by the community. By definition then these feeds are likely to be of lesser relevance to a given user (otherwise, presumably, they would have formed part of their RSS submission). Nevertheless, users did regularly respond favourably to recommendations derived from these RSS feeds.
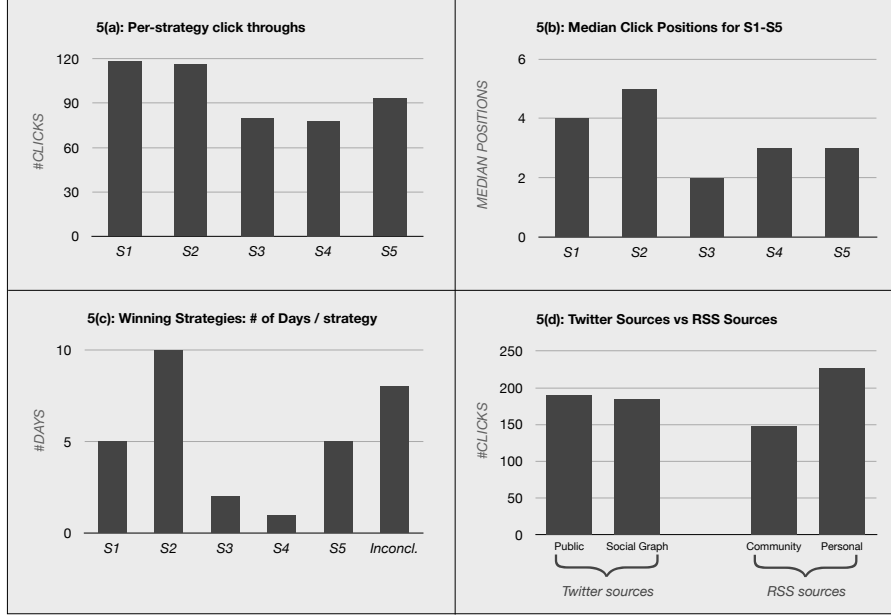
Fig. 5: *Main Results*

We see little difference between the ranking strategies with only fractionally more click-thrus associated with stories ranked by the public tweets than for stories ranked by the tweets of the user's own social graph.

It is also useful to consider the median position of click-thrus in the result-lists across the different strategies. Figure 5(b) shows this data for each strategy, calculated across emails when there is at least one click-thru for the strategy in question. We see, for example, that the median click-thru position for $S1$ is 4 and $S2$ is 5, compared to 2 and 3 for $S3$ and $S4$, respectively, and compared to 3 for $S5$. On the face of it strategies $S3$ and $S4$ seem to attract click-thrus for items positioned higher in the result lists. However, this could also be explained by the fact that the high click-thru rates for $S1, S2, S5$ mean that more items are selected per recommendation list, on average, and these additional items will have higher positions by definition.

Figure 5(c) shows depicts the *winning* strategy S_i on a given day d_j if S_i receives more click-thrus than any other strategy during d_j, across the 31 days of the evaluation. We can see that strategy $S2$ (user's personal RSS feeds ranked by the tweets of their social graph) wins out overall, dominating the click-thrus of 10 out of the 31 days. Recency ($S5$) comes a close second (winning on 8 of the days). Overall strategies $S3$ and $S4$ do less well here, collectively winning on only 3 of 31 days.

5 Discussion & Conclusion

These results support the idea that each of the 5 recommendation strategies has a useful role to play in helping users to consume relevant and interesting news stories. Clearly there is an important opportunity to add value to the default recency-based strategy that is epitomized by $S5$. The core contribution of this work is to explore whether Twitter can be used as a useful recommendation signal and strategies $S1 - S4$ suggest that this is indeed the case. It was not our expectation that any single strategy would win outright, mostly because each strategy focuses on the recommendation of different types of news stories, for different reasons, and for a typical user, we broadly expected that they would benefit from the combination of these strategies.

In Figure 5(d), we summarize the click-thru data according to the framework presented in Figure 3 by summing click-thru data across the diagram's rows and columns in order to describe aggregate click-thrus for different classes of recommendation strategies. For example, we can look at the impact of different Twitter sources (public vs. the user's social graph) for ranking stories. Filtering by the Twitter's public timeline $(S1 + S3)$ delivers a similar number of click-thrus (about 185) as when we filter by the user's social graph $(S2 + S4)$, and so we can conclude that both approaches to retrieval and ranking have considerable value. Separately, we can see that drawing stories from the larger community of RSS feeds $(S3 + S4)$ attracts fewer click-thrus (approximately 150) than stories that are drawn from the user's personal RSS feeds (strategies $S1 + S2$), which attract about 225 click-thrus, which is acceptable and expected. These community strategies do highlight the opportunity for the user to engage and discover interesting and relevant content they potentially wouldn't have been exposed to from their own subscriptions.

We have explored a variety of different strategies by harnessing Twitter's public tweets, as well as the tweets from a user's social graph, for the filtering of stories from both personal and community RSS indexes. The results of the live-user evaluation are positive. They demonstrate how different recommendation strategies benefit users in different ways and overall we see that most users respond to recommendations that are derived from a variety of different strategies. Overall users are more responsive to stories that come from their favourite RSS feeds, whether ranked by public tweets or the tweets of their social graph, than stories that are derived from a wider community repository of RSS stories.

There are many opportunities for further work within the scope of this research. Some suggestions include considering preference rankings and click-thrus as part of the recommendation algorithm. Also, it will be interesting to consider whether the *reputation* of users on Twitter has a bearing on how useful their tweets are during ranking. Moreover, there are many opportunities to consider more sophisticated filtering and ranking techniques above and beyond the TF-IDF based approach adopted here. Finally, there are many other application domains that may also benefit from this approach to recommendation: product reviews and special offers, travel deals, URL recommendation, etc.

References

1. Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. Recommender system based on consumer product reviews. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 719–723, Washington, DC, USA, 2006. IEEE Computer Society.
2. Marko Balabanovic and Yoav Shoham. Combining content-based and collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.
3. Peter Brusilovsky and Nicola Henze. N.: Open corpus adaptive educational hypermedia. In *The Adaptive Web: Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science. Springer, 2007.
4. Iván Cantador, Alejandro Bellogín, and Pablo Castells. News@hand: A semantic web approach to recommending news. In *AH 2008*, pages 279–283, Berlin, Heidelberg, 2008. Springer-Verlag.
5. Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW '07*, pages 271–280, New York, NY, USA, 2007. ACM.
6. Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth. On the real-time web as a source of recommendation knowledge. In *RecSys 2010*, Barcelona, Spain, September 26-30 2010. ACM.
7. Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *SSRN eLibrary*, 2008.
8. Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop*, pages 56–65, 2007.
9. 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.
10. 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.
11. Haewoom Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW '10*, pages 591–600, 2010.
12. K. Lerman. Social Networks and Social Information Filtering on Digg. *Arxiv preprint cs.HC/0612046*, 2006.
13. Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *in Eighteenth National Conference on Artificial Intelligence*, pages 187–192, 2002.
14. Michael Pazzani and Daniel Billsus. Content-based recommendation systems. *The Adaptive Web*, pages 325–341, 2007.
15. Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 385–388, New York, NY, USA, 2009. ACM.
16. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002.
17. Shilad Sen, Jesse Vig, and John Riedl. Tagommenders: connecting users to items through tags. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 671–680, New York, NY, USA, 2009. ACM.
18. René T. A. Wietsma and Francesco Ricci. Product reviews in mobile decision aid systems. In *PERMID*, pages 15–18, Munich, Germany, 2005.