Anthus News
Demo Report

James Doolan
10349967

August 19, 2015

1 User Scenario

Our target user:

- Gets their news from *online sources*
- Is a fairly active Twitter user (follows several accounts)
- Cares about the *relevance* of the news they read

Our target user is very broad as we aim to attract a large number of users based on catering to a wide range of tastes. Our users can be anyone who wishes to follow the news but is uninterested by a large portion of it on a generic news feed. We aim to appeal to the broadest possible base in an attempt to generate a large amount of traffic to our site.

We distributed an initial survey to determine what kind of people would be interested in using our site and what they would like to see. Some of the responses we received included:

- *"As a light social media user I would like Facebook to be included so I can create a profile too."*
- *"As a light social media user I would to create an account quickly so I can easily get relevant news hassle free."*
- *"As an active media user I would like to influence my profile so I can optimise the news I see. "*
- *"As a curious person I would like to see an analysis of my profile statistics so I can see what the feed will generate"*
- *"As a person of set tastes I would like to see articles only relating to my interests so I can avoid the irrelevant articles."*

- *"As a discoverer I would like to see random articles so I can discover new things."*
- *"As an avid news follower I would like only to see real news so I can avoid the fluff pieces."*

From these comments, and other feedback, we constructed several personas to aid our design.

2 Technical Problem

2.1 Motivations

Readers, now more than ever, care about the content they see; they want only the news that is relevant to them. This is clear from the response to our survey; several respondents highlighted the importance of relevance of the news they read, while others prioritised breaking news. Many people now obtain their news through social media, via sources such as Facebook and Reddit, where they can be assured that the stories they read come recommended by their friends or similarly-minded people.

However, users are notoriously loath to express their preferences ; many users dislike categorising their interests, even if it means that their content could be tailored to them. In fact, it has been shown, that even when they do declare an interest in a set of topics, these interests do not correspond to their behaviour.

Our project seeks to generate these preferences implicitly from a reader's supplied Twitter account, which will allow us to recommend only those articles which have relevance to the individual reader.

2.2 Similar Products

An application with similar goals and functionality to our project is News360, which attempts to recommend news stories to the user. It uses a mixture of explicit and implicit feedback, the latter from Twitter, Facebook, Google+ and Evernote. It is not clear how it uses the information from these sources, as it is a proprietary system (their blog rather vaguely describes the system as a "semantic engine").

During our research into this area, we tested the relevance of the topics suggested by News360 with a dummy Twitter account; although the account follows several politicians and pop stars, News360 failed to recommend a politics- or music-related category. Instead it suggested the Premier League, despite the test account having no interest in football.

2.3 Core Problem

The main problem can be distilled down to a few fundamental tasks:

- Inferring user interest in a topic
- Determining articles' relevance to topics
- Recommending topics/articles to users

3 Technical Solution

The system can recommend news articles via topics to users, as well as provide statistics on the user's interests. The recommended topics are chosen using:

- The user's interests as inferred from their Twitter profile
- An optional explicit rating provided by the user
- The number of times the user clicked an article related to a topic

In the following section, we discuss our methods for obtaining these three sets of data and describe the technologies we used to do so.

A key element of our project is Twitter's Lists feature. Lists can be thought of as a way of tagging accounts with identifiers; for example Katy Perry turns up in many "Music" and "Pop Star" Lists, while Barack Obama appears in Lists such as "Influential Figures" and "Politicians".

Table 1: Examples of List members

Member	List Names
Katy Perry	some some some some
Barack Obama	

Lists can be set up by any Twitter user; public Lists can be followed by any user and accessed through the Twitter API. A List has a title, description and multiple members (average of about 300 members per list in our sample). In general, the list name and description relate to the members of the list; we can exploit this fact to identify ”topic experts” - Twitter accounts which have a strong relationship with specific topics (See Table 1 for examples of some Lists and their members). We drew inspiration from a paper by Ghosh, S, Sharma, which details the process of acquiring List data and inferring topics.

Sharma et al. used three servers that were whitelisted by Twitter, and therefore had almost limitless access to the Twitter API. In addition, they had access to a snapshot of the entire Twitter social graph, which they used to identify hubs and authorities using the HITS algorithm. They could then prioritise loading Lists for the highest-ranked hubs and authorities. Even with these optimisations, Sharma managed to only process a fraction of Twitter Lists.

3.1 Inferring User Interest Using Lists

Due to time and resource constraints, we could not hope to achieve the same results as Sharma et al.; we decided to prioritise processing the top 1000 Twitter accounts by number of followers. In this way we hope to provide maximum coverage for users of our service. To date we have loaded 120,000 Lists containing 10 million unique members, representing $\sim 5\%$ of the Twitter population.

We load Lists as described above, storing the list id, members, name and description in our database. We then parsed the names and descriptions, removing stop words and domain-specific terms (”Twitter”, ”list”, ”follows”, etc.) and recorded the most frequent terms across all lists. From these top terms, we selected 16 as our topics. With this done, we identify as experts all users who are on 10 or more Lists mentioning our topics.

Inferring User Interest

To identify topics of interest to the user, we first query Twitter for the user’s friends’ IDs. These are then matched against our expert list, and a Borda count is computed for each expert; the most frequent topic is given a score of 5, the next most frequent a score of 4, etc. These scores are then summed over the entire friend list, giving us a topic distribution for the user. Using this method prevents a bias towards users who appear on many lists.

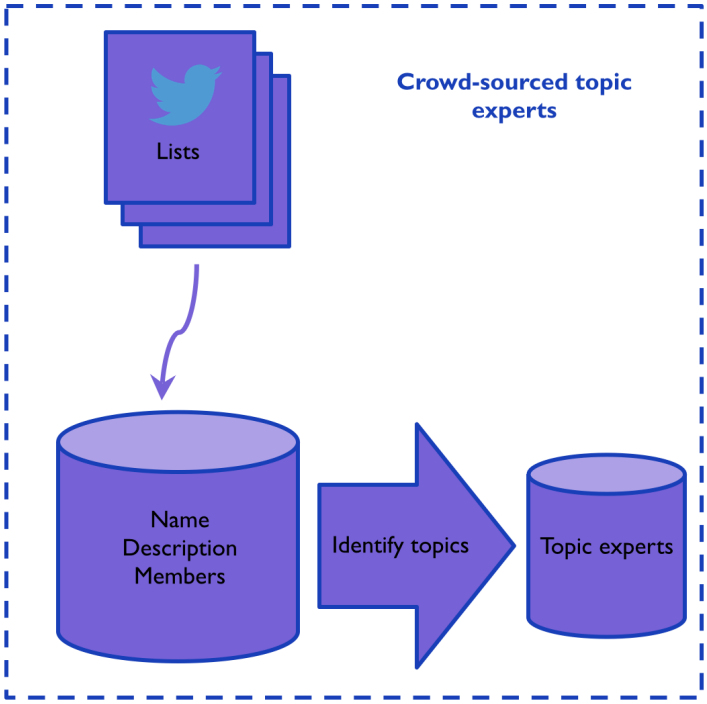

Figure 1: We load Lists as described above, storing the list id, members, name and description in our database. We then parsed the names and descriptions, removing stop words and domain-specific terms (”Twitter”, ”list”, ”follows”, etc.) and recorded the most frequent terms across all lists. From these top terms, we selected 16 as our topics. With this done, we identify as experts all users who are on 10 or more Lists mentioning our topics.

3.2 Characterising Articles

3.3 Recommending Articles