# 

# Anthus News

# anthusnews.com

# System Demo Report

# Pink Team:

James Doolan

Kang Li

Katharine Cooney

Liam Creagh

Shuyu Huang

## 1. User Scenario: The Characters (300 words approx.)

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 sent out a survey to try and help determine what people kind of people would be interested in using our site and what they would like to see. Some of the personas that we created from 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."

## 2. Technical Problem: The Setting (500 words approx.)

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, through 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 [1]; 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 do not necessarily match their actual interests [2].

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.

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

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 account; although the account used follows several politicians and pop stars, it failed to recommend a politics- or music-related category, instead suggesting the Premier League, despite the test account having no interest in football.

## 3. Technical Solution: The Plot (1,800 words approx.)

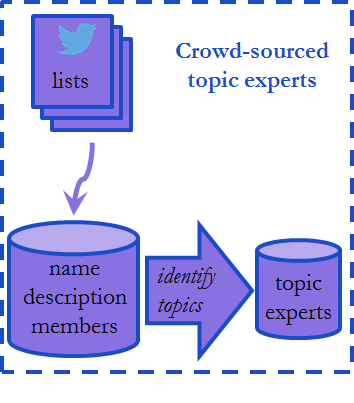
The system can recommend news articles to users, as well as provide statistics on the user's interests.

Identification of topic experts

A key element of our project is Twitters 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".

Lists can be set up by any Twitter user and public Lists can be followed by any Twitter 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. We drew inspiration from a paper by Ghosh, S, Sharma, which details the process of acquiring List data and inferring topics.

Due to time and resource constraints, we are unable to load all Lists to our database; we have therefore prioritised loading the Lists containing the top 1000 Twitter users. In this way we hope to provide maximum coverage for users of our app. To date we have loaded over 110,000 Lists containing over 9.5 million unique members, representing around 5% of the Twitter population.





Inferring User Interests

To identify which topics a user is interested in, we check if they are following any accounts that we have identified as an expert. We then infer that the user has an interest in that expert's topics.



Characterising News Articles

We process the article text using the same method of parsing as for Lists (stopword removal, case-folding, identifying bigrams and entities etc.). Using a method of document tagging described by Schonhofen, we use the Wikipedia category hierarchy to assign topics and factors to the articles.

Complete System:

***User Interface***



The user interface is a simple website with 3 different possible menus on display. The first will display when the user is not logged-in. This is comprised of a Register page, Login page and contact page. The user can sign in if he/she already has an account and register if they do not. Upon login, the user will be presented with the following options:

My News displays the user's personalised news feed if they have authorised their Twitter account, and an invitation to do so if they have not. The user enters their details on Twitter and authorises our app to use their account; Twitter will then redirect back to our site.

My Profile allows users to edit profile information such as password and email as well as other details after the Twitter profile is created.

After the user has authorised our application with Twitter, the following options will be made available:

Analytics gives users an insight into how their profile was generated. Various statistics are displayed here, such as which topics our system identified as being relevant to the user, and which of the user's followed accounts on Twitter contribute to these topics.

My Profile will now allow users to directly influence their news recommendations if they wish to do so, by adding or removing topics from their profile.

**Technology Stack**

We used Python for the majority of this project, using Django for handling requests and general web processes and various other packages fulfilling processing roles on the back end. Some of the more important packages we used include:

Django Registration Redux

This package provides a convenient framework for maintaining user profiles, and made setting up registration, login, signups very straightforward. The registration consists of a simple register step, that allow users to create an account with our site. The Django Registration Redux package comes with many features and provides a number of different pages for a all the possible different account related steps in the process such as changing password, account activation, logging in and out and registering. All of these account related events have a special section for handing these steps and are preceded by "/account/page.html" to segment the account related events. Registered users can be seen and edited in the administration section in django.

Feedparser & Newspaper

These were used in the back end for acquiring content from RSS feeds. The process is follows: the feeds in our database are checked for new articles using Feedparser, the URLs of which are passed on for processing by Newspaper. Newspaper uses a trained model of what the content of a news article looks like in a HTML document, which means that it can be used to scrape text from almost any page without the need to use CSS or HTML selectors.

Natural Language Toolkit (NLTK)

Used extensively throughout the project; we wrote a parser for processing List titles and articles using a variety of NLTK's resources (tokenizers, part-of-speech tagging, entity recognition).

Tweepy

This package is a Python wrapper around the Twitter API we used it for authenticating users with Twitter, obtaining List data and querying user's followed accounts.

**Back-end**

We used PostgreSQL as our database management system, using a combination of Django models and raw SQL to interact with the database. A short summary of the content of the database is as follows:

* Lists
  + Members
  + Names
  + Descriptions
* Wikipedia
  + Article titles
  + Categories
* Articles
  + URLs
  + Text
  + Publish date
* User-specific Data
  + Topics
  + Twitter credentials
  + Login credentials
* Example input/output and interaction with different components

## 4. Integration: The Sub-plots (400 words approx.)

The website is integrated with the user's Twitter account, allowing us to query their profile and retrieve their followers.

Similarity to Existing Technologies

Most of the ideas presented in this project have been applied in isolation in an academic setting. For example, the paper examining the use of Lists for inferring user interest did not discuss its utility in recommending products or news. Similarly, the research done on tagging articles using Wikipedia did not go any further, merely contrasting their results with other methods such as LDA and clustering via NMF.

* How does your technical solution interact with, support or enhance related technologies?

## 5. Impact: The Resolution (400 words approx.)

Our site will offer users a new and exciting way to consume relevant news. It will present any potential user the opportunity to create a more interesting and personalised news feed virtually instantly.

Anthus News Mission Statement:

Our main goal is to create a diverse and centralised news feed, with quick profile creation that offer the general user a personalised and unique user experience.

We aim to create a far more accurate and semantic experience for users through deep and thorough linguistic analysis and advanced premeditated methodologies. Our system will have an impact based upon its strengths which derive from our research into and choice of the most effective processes implemented and studied in previous academic and practical settings.

We hope it will make an immediate impact when it is initially release in production. The intended ease of use and simplicity make it immediately useful and accessible with its modularity making it very scalable allowing for future growth and development. The design allows for many features to be added while still being useful in its current minimalistic state.

## 6. Reflections: The Reviews (800 words approx.)

* How successful is your solution?
* How appropriate were the choices you made about technologies to use?
* What were the biggest challenges you faced as a team in working on this project?
* How effective were the project management and software development methodologies that you used?
* What lessons have you learned by developing this system? (what problems are hard/easy, has your perspective changed since the start?)

## 7. References and Key Resources