System Demo Report

Katharine Cooney
katharine.cooney@ucdconnect.ie

Liam Creagh
liam.creagh@ucdconnect.ie

James Doolan
james.doolan@ucdconnect.ie

Shuyu Huang
shuyu.huang@ucdconnect.ie

Kang Li
kang.li@ucdconnect.ie

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 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 comments that we felt would be interesting to implement were:

- *"As a light social media user I would to create an account quickly so I can easily get relevant news hassle free.*"
- *"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 an active media user I would like to influence my profile so I can optimise the news I see.* "

The other responses we received included:

- *"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 [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 interests do not correspond to their behaviour[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.

2.2 Similar Products

An application with similar goals and functionality to our project is News360[3], 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".

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[4], 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.

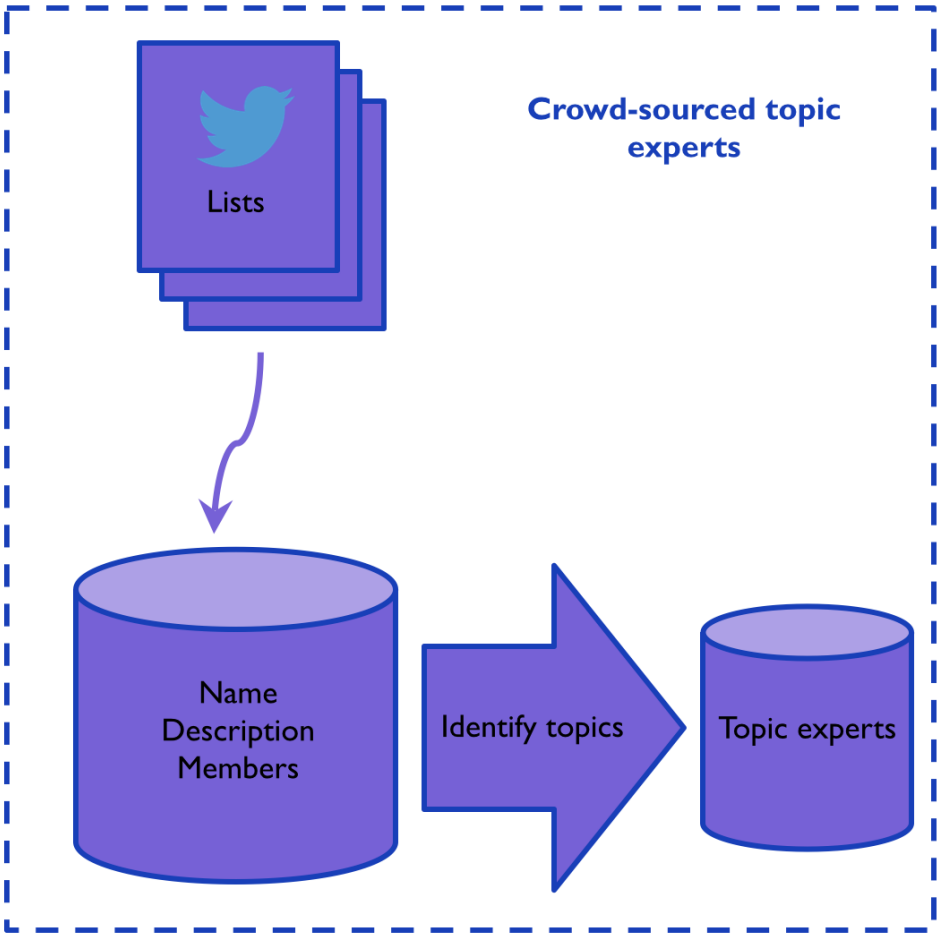

Figure 1: We check if the user are following any accounts that we have identified as topical experts. We then infer user interest in these topics.

3.1 Inferring User Interest Using Lists

In 2014 a group from the same Indo-German collaboration published a second paper which outlines a methodology for inferring Twitter users' interests from List data [5]. Due to time and resource constraints, we could not hope to achieve the same results; 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 600,000 Lists containing ~ 10 million unique members, representing around 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 accounts which appear on many lists. (See Figure 2).

3.2 Characterising Articles

We trained a classifier using hand-labelled articles from several RSS feeds; the content of each article was parsed, and a TF-IDF transformation applied. Feature selection was achieved using a linear support vector classifier (SVC), and the transformed data was passed to the classifier. The classifier we used was a stochastic gradient descent (SGD) classifier with a modified Huber loss function. The classifier was applied in a one-vs-all strategy, fitting each class against all

Table 1: Examples of List members

Member	List Name	List Description
Katy Perry	Music	Musicians, producers, studio’s, record labels
	Music-News-Artists	Music tweets by artists, labels, fans, organizations, in all - people in the wide world of music.
Barack Obama	News & Politics	Politicians, current events and news-things to keep me in touch with the world.
	Politics	News, Pols, Pundits

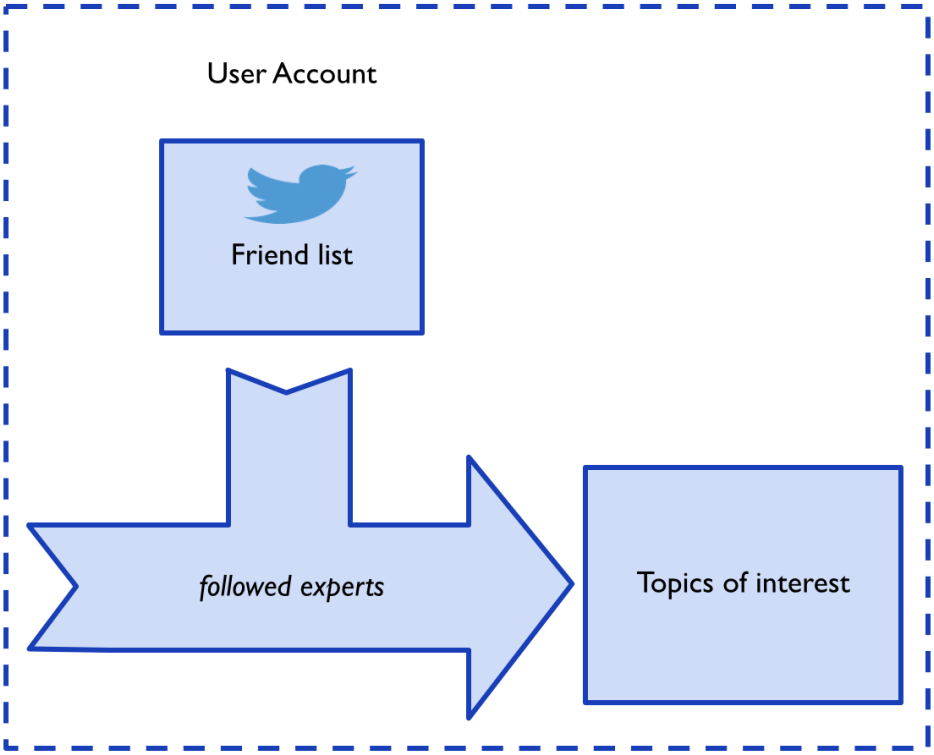

Figure 2: We check if the user are following any accounts that we have identified as topical experts. We then infer user interest in these topics.

the others. This classifier gave the best results out of several tested, as seen in Table 2:

3.3 Recommending Articles

Three metrics are taken into account when retrieving articles for a user, as follows:

Twitter Ratings: The topical interest vector as obtained from the user's Twitter account.

Explicit Ratings: The user is provided with sliders to adjust the topic rating generated from their Twitter profile or to add and adjust new topics not automatically recommended for them.

Table 2: Comparison of Various Classifiers

Classifier	**Accuracy**
Naive Bayes	0.52
SGD (hinge loss)	0.73
SVC	0.45
SGD (huber loss)	0.87

Click-Through Rating: The number of times the user clicked articles relating to a topic, divided by the number of articles from that topic that they have been shown.

The three vectors are weighted as shown in Figure 3, and the resultant vectors are added and the result normalised to sum to 1. When it comes to recommending articles, we take this topic vector and multiply it by the number of articles we want to retrieve; this gives us a number of articles in each topic proportional to interest in that topic. We then group the article list into equal chunks, such that each chunk has the same distribution of articles as the overall list, and high-ranking topics occur higher up in each chunk. This means that the user will see an immediate change in their feed when they change their profile rating.

3.4 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:

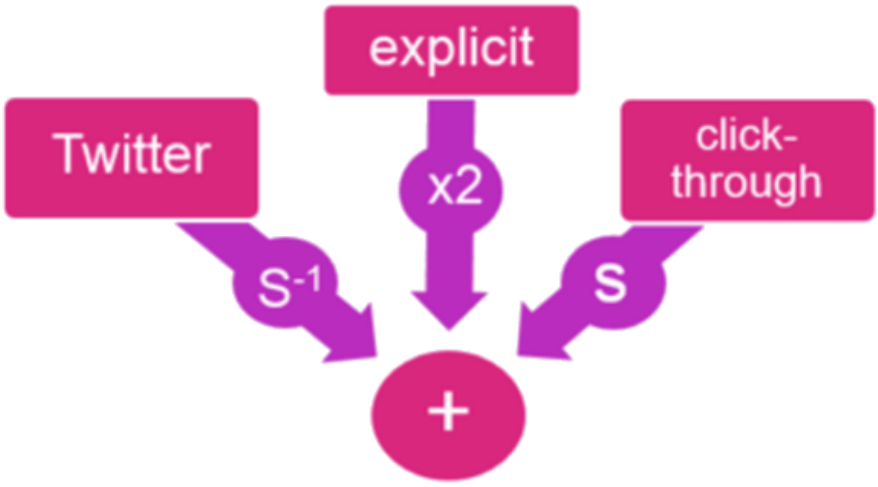

Figure 3: The user's explicit feedback is given twice the weight of the other 2 items so that the user sees a response to any feedback given. The purpose of the sigmoid function (S) is to prioritise the Twitter profile information initially, but as the click-through counts are collected, the system will learn more about the user, and the Twitter data will become less important.

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.

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.

3.5 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 registra-

tion 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 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 an 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)[6]

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.

PostgreSQL

We used PostgreSQL as our database management system, using a combination of Django models and SQL commands to interact with the database.

4 Integration

4.1 Integration of System Components

We feel that we made excellent choices in regards to our technology stack. The PostgreSQL database was well equipped to handle all the various and large data we were storing for this project. Our largest data sets were our Experts and List with over 500k entries each, which PostgreSQL handled with ease. For our front-end we used Bootstrap with a free theme, Andia[7]. This offered simple and quick scalable design with a flexible framework for the layout of our site.

For parsing and receiving articles we used feedparser in conjunction with newspaper which complemented each other as we could pass feedparser entire feeds that would in turn pass individual articles to newspaper which would get all the article related information. Wrapping this component in a Celery task running every half-hour ensures that our article database stays up to date.

4.2 Similarity to Existing Technologies

Most of the ideas presented in this project have been applied in isolation in an academic setting. For example, the Sharma paper examining the use of Lists for inferring user interest did not discuss its utility in recommending products or news. We discuss here how the topics returned by our system compare to those returned by WhoLikesWhat.

4.2.1 Comparison to WhoLikesWhat

For each topic in our database, we selected an account suggested by the WhoLikesWhat system [8] and made sure it was covered in the Anthus system (Anthus has less coverage as it only has a small percentage of Twitter Lists). We then compared the top 3 results in the two systems; the results of this evaluation are seen in Table 3. We noted the following during this evaluation:

1. We registered what we called a “tech bias”, that is our system identifying a higher level of “tech” component than WhoLikesWhat.

2. A better match would have been the word “movie” rather than film. Direct text matching will not cope with semantic differences like this.

3. We did not have good matching on sports topics. Again note the semantic difference in the meaning of “football”.

5 Impact

Anthus News Mission Statement:

Provide a diverse and centralised news feed, with quick profile creation that offers the general user a personalised and unique user experience.

Our site will offer users a new and exciting way to consume relevant news. It will present any potential user with the opportunity to create a more interesting and personalised news feed virtually instantly. We aim to create a far more accurate and personalised experience for users through linguistic analysis and advanced premeditated methodologies. Our system will have an impact based upon our research into the most effective processes implemented and studied in previous academic and practical settings.

We expect it will make an immediate impact when it is released 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.

We would expect peer recommendations to quickly expand demand for the app. Anthus News is also available on mobile so we would see that as an additional channel for recruiting users. Our promotion of Anthus News via Twitter and Facebook should also bring in new users.

6 Reflections

6.1 Successes

We succeeded in many aspects of our project such as determining user interest via Twitter, giving the user profile flexibility and including varied content. We compared the topics returned by our system to those suggested by WhoLikesWhat[8], the results of which can be seen in Table 3. We see decent agreement with the topics returned, which suggests that, despite the limitations we faced regarding resources and time, we have implemented a successful inference system.

Topic	Twitter account	WhoLikesWhat (Top 3)	AnthusNews (Top 3)	Notes
art	@tate	art, culture, design	art, design, tech	
business	@FT	politics, finance, media	business, tech	1
celebrity	@JimCarrey	celebrity, entertainment, actors	entertainment, celebrity, art	
design	@mashable	media, tech, design	tech, design	
education	@edutopia	education, news, media	education	
entertainment	@theellenshow	celebrity, entertainment, TV	entertainment, art, music	
fashion	@StellaMcCartney	fashion, design, news	fashion, design	
film	@kevinspacey	entertainment, movies, celebrities	entertainment, art, film	2
food	@Bourdain	celebrity, entertainment, foodies	food, entertainment, travel	
health	@MichaelPollan	food, health, news	food, health, tech	
music	@justinbieber	music, movies, celebrity	music, art	
politics	@whitehouse	politics, gov, news	politics, tech	1
science	@newscientist	science, journalists, tech	tech, science	1
sport	@nfl	sports, players, football	sport	3
technology	@ijustine	socialmedia, youtubers, tech	tech, design, music	
travel	@lonelyplanet	travel, blogger, media	travel	

Table 3: Comparison with WhoLikesWhat

We feel that another success for our project was the fact that we satisfied our design objectives, most notably the analytics and profile features. We received positive feedback on these from the UX survey, and we therefore feel that our target users and personas would be satisfied with the system.

On the other hand, the classification of articles was not entirely successful. While training the classifier, we performed k-fold cross-validation on our training data. This produced an accuracy of around 87%; however, when we inspected the performance of the classifier on real data, we found that the overall accuracy was closer to 70%. The results from this informal evaluation can be seen in Table 4 The classification of articles is probably the least successful component of the system; this is due to several factors, namely:

- The topics we selected from the Twitter Lists are too broad, with large overlaps between them.
- Our training set was too small (20-30 articles per topic)

While these classification issues affect the performance of our system, they could be rectified by simply spending more time training the classifier. In a full-scale version of the system, we would also generate more topics from Lists, and potentially introduce a topic hierarchy to deal with the overlaps between topics. A more sophisticated approach to identifying topical experts, perhaps by clustering Lists based on their membership, might also improve the classification. This could also mitigate the effects of synonymous List topics such as “film” and “movies” as well as polysemous terms such as (American) football and football (soccer).

Table 4: Classifier Performance per Topic

Topic	Rating	False Positives	Topic	Rating	False Positives
Art	100%		Business	40%	sport
Entertainment	70%	sport	Celebrity	50%	sport, tech
Music	55%	sport	Tech	70%	design
Education	75%		Film	70%	sport, celebrity
Science	65%		Sport	100%	
Design	40%	sport	Food	90%	
Politics	75%		Travel	50%	range of topics
Fashion	100%		Health	55%	business

6.2 Challenges

Getting started and organised was a big challenge at the outset. We spent a lot of time talking and discussing all the different directions the project could go in. The initial stages of the project were mainly spent researching different methods for inferring user interests; we feel that this time was spent well, as once we began writing code, we had a clear idea of what needed to be done, and gave us focus for our first minimal viable products.

Overcoming the Twitter rate limits was another challenge initially; Twitter only allows 15 API calls to get a user's List memberships in a 15-minute window. This is clearly insufficient to aggregate the large number of Lists we required. We circumvented this by applying for 10 developer keys, and switching between them when we exceeded the rate limits. This ensured that we had a constant stream of incoming List data.

We were quite successful when managing the project and task involved in creating the software. We started off with an agile Scrum approach using Trello as a task manager for the first half of the project. We divided up the tasks, self-delegated and completed them in our own time which worked well as many of us were busy during the beginning of the project. As the project progressed, we became more available which allowed us to have regular face-to-face meetings; we decided on various tasks in the morning, communicating and collaborating throughout the day. We had a set plan of what needed to be achieved within increments of 2 week periods much like a sprint.

This style of project management worked well for us, as it allowed us to throw out ideas and discuss them as a team, and all members had a clear vision of our progress at all times.

6.3 Lessons Learned

We learned many new technologies such as Python and all the available associated libraries, PostgreSQL, Bootstrap, HTML, CSS, Javascript and jQuery among others. We all learned some skills from each other coming from a diverse background with nobody coming directly from a computer science undergraduate course. Katharine and Kang came from an Engineering background while Liam and James came from Media and Physics respectively. In addition, we had studied a wide variety of subjects between us during the previous semesters. This brought greatly-varying skills and approaches that contributed to the planning, research, implementation and presentation aspects of the project.

References

[1] Doychin Doychev, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. An analysis of recommender algorithms for online news. CLEF, 2014.

[2] Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. User attitudes towards news content personalization. *International journal of human-computer studies*, 68(8):483–495, 2010.

[3] `https://news360.com/`.

[4] Saptarshi Ghosh, Naveen Sharma, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Cognos: crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 575–590. ACM, 2012.

[5] Parantapa Bhattacharya, Muhammad Bilal Zafar, Niloy Ganguly, Saptarshi Ghosh, and Krishna P Gummadi. Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 357–360. ACM, 2014.

[6] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[7] `http://goo.gl/EaytN0`.

[8] Saptarshi Ghosh, Naveen Sharma, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. http://twitter-app.mpi-sws.org. 2014.