School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

Managing Your Social Networking Profile

Enabling User-Tailored Views of Your Feed

by

Geoffrey Zhu and Richard Zhang

Thesis submitted as a requirement for the degree of

Bachelor of Engineering in Computer Engineering

Submitted:	26th May 2015
Supervisor:	Dr Hye-young Paik
Assessor:	Prof Fethi Rabhi
Student ID:	3415440 & 3416477
Topic ID:	3031

Abstract

The universally used platform of Social Networking Services (SNS) faces many challenges as they gain huge user bases. One such SNS known as Facebook provides a feed that is a list of all items from friends, organisations and other entities. This feed contains an enormous amount of data that must be ordered in such a way that the user is satisfied. In this paper we will be exploring the possibilities of user modelling being used to rank these feed items as well as verifying multiple proposed algorithms in the literature. The final outcome is a more personalised algorithm for ranking items from a users Facebook feed.

Acknowledgements

We would like to thank our supervisor Dr. Helen Paik for her constant guidance and support, and our assessor Prof. Fethi Rabhi for his advice and feedback.

Contents

Chapter 1

Introduction

Social Networking Services (SNS) are platforms in which a diverse range of users are able share their interests, organise social activities and keep in touch with people. In almost all SNS, users are presented with a *feed*; this feed is a list of items generated via the user's connections throughout the SNS. The feed acts as a summary of activities that the user has subscribed to, a dashboard presented to them when they first log in. This feed contains a large amount of items that we would like to order, or rank in some way such that the items that the user finds more interesting have a higher precedence in the feed. As the use of SNS grows rapidly, so does the demand for such ranking algorithms.

While there exists many SNS currently being used, the scope of this thesis will be reduced to focus on only one of them; Facebook. This is mainly due to the time constraints involved, however the results and methodology used in both our research and implementation will be generalisable to all SNS. There are two main reasons why Facebook has been chosen as our SNS of focus.

Firstly, Facebook is currently the most used SNS; Statistica [Sta15] gives us the number of Facebook users for the second quarter of 2015 as almost 1.5 billion. This allows us to more easily gather users for our research purposes as well as have more confidence that our results can be generalised to most SNS.

Secondly, Facebook attracts many different types of users due to it's very flexible, generic nature (i.e. not a niche SNS). Each type of user will have different requirements and by having a large set of user types, we are able to more easily identify and distinguish them from one another.

With all this considered, it is clearly not possible to have one single ranking algorithm to accommodate for all users, and yet as of now, Facebook only offers one ranking, or view of a user's feed (besides chronological order). Our aim can be summarised as follows: First we will set out to identify these different user types and their needs, then we aim to create a number of different ranking algorithms based on the discovered user types. Thus offering a more personalised ranking of a user's feed. It is important to note that we do not aim to create a *better* ranking algorithm than Facebook as an enormous amount of research and time has already been put into creating said algorithm, instead we aim to offer different, more personalised rankings.

Chapter 2

Background

2.1 User Modelling

As mentioned earlier, the first half of our approach is to identify different types of Facebook users and determine what preferences they have in terms of feed items. We took a look to the literature to see if anyone had attempted user modelling for this purpose.

The first question we set out to answer was quite intuitive, that being; what are the reasons why people use Facebook? Or even SNS in general. Bon [BRR10] conducted a survey on university students to confirm the results of a research paper published in 2008 which found that the main reasons people used social networks were:

- Keeping in touch with friends
- Meeting new people
- Posting and looking at pictures

Factor analysis was performed on the survey answers and the results were tabulated into a list of loadings, representing how much certain features acted as the reason for the use of social networks. These results however, proved to be too general for our purposes as we are looking to define different, more concrete user types. On top of this, the scope of this Survey was restricted only to university students and contained a relatively small sample size of 200.

Table 1

Component Loadings for Factor Analysis

Components	Loadings
Component 1: Information Dimension	
To post social functions	.781
To learn about events	.694
To share information about yourself	.649
For academic purposes	.602
To post/look at pictures	.542
Component 2: Friendship Dimension	
To keep in touch with old friends	.945
To keep in touch with current friends	.865
To locate old friends	.767
Component 3: Connection Dimension	
For dating purposes	.839
To make new friends	.688
To feel connected	.584

Figure 2.1: Survey Results

Some interesting correlations were found with further statistical analysis, for example; men significantly used social networks more for the purpose of sharing personal information and dating compared to women. While being somewhat more specific, these correlations are still slightly too generalised. However, this being said, knowing some of the more common reasons that people use Facebook will give us direction in terms of what types of feed items we should be looking for.

Nad [NH12] published a literature review which looked at user modelling Facebook users from a more psychological frame of view. They found that there were correlations between certain Facebook behavioural patterns and a user's psychology. Some of the correlations that were found were:

- Users with higher levels of extraversion tended to use Facebook as more of a social tool rather than to replace social activities. They also generally had accumulated more social networking friends and had more addictive tendencies when it came to Facebook use

- Users with higher levels of Neuroticism Levels tended to prefer reading information in their feed whereas users with lower levels of neuroticism preferred seeing photos
- Users with significantly high or low levels of neuroticism tend to be more open to share personal information on their profiles

These findings are closer to what we are looking for, however, it is not easy to determine the psychology of a user for the application of ranking their Facebook feed. However, once again, these results allow us to get a better idea of the different types of feed objects that are to be considered, for example text vs images.

It eventually became clear to us that even if we were to use these results for our user modelling, it is uncertain whether they are still relevant, as we saw, the survey conducted by Bon [BRR10] set out to confirm results from a study just 2 years previous to his research paper and already found that user's behaviour changed. So we began to look at how people perform user modelling and the practices involved with it. Cle [Cle04] conducted an interview with 4 different HCI professionals based in Sweden to gain insight into their methodologies and philosophies. The following methods of performing user modelling that came from the interviews especially interested us and will most likely be used throughout our implementation:

- Usability Tests - Observing how a user interacts with a system and asking good questions to extract meaningful feedback from the user.
- Surveying - Simply asking existing users questions that will aid us in our user modelling - as seen in the survey conducted by Bon [BRR10]
- Conceptualise how users are represented in the system (as data) - By thinking about how a user's preferences will be represented in the system itself, we are

able to design our algorithms accordingly

- Creating personas and use cases/scenario - Designing a fictional character who would benefit from our system and creating a use case around it; this allows us to consider all the different types of users that can be modelled

2.2 Ranking Algorithms

A ranking algorithm will give each item a score and order them with the item with the highest score at the top and the item with the lowest score at the bottom. The score of an item will depend on a set of criteria that the ranking algorithm uses. Social networking services will use these ranking algorithms in order to provide the user with items that will interest them. Since we are focusing on Facebook, the users will be provided with a feed that contains a variety of different feed items. In Li [LTL+10] paper, they discovered that there are three major factors that could affect how interesting a user may find a particular item. They are:

- Topical Preference
- Topological Locality
- Social Influence

Topical reference is the idea that most users are interested in a limited range of topics. Topological locality refers to the fact that users are interested in the topics that their friends like and Social influence basically says that users are interested in famous people such as singers or actors. These factors do provide some insight on what a user likes but could be further generalised to topics that a user likes. The Topological locality does raise an interesting notion, that is, users are more likely to like a feed item that a friend likes. We can summarize these two into a more general categorization of Topic classification and connections. Topic classification will be classifying the topics that a user may like while connections will be a measure of how 'close' the user is with their friend based on their interactivity.

Topic classification is quite difficult we have to generalise a topic that they may like based on the feed items that they receive. In a paper by Bur [BMAC13], they analysed twitter tweets and tried to generalise a topic based on the tweets each user received. They have analysed two types of methods. They are:

- BestOverlap
- UserInfoBigram

The BestOverlap method attempts to gather a huge amount of tweets and look at the common words in those tweets. A topic can be generated by the word is overlapped the most across all the tweets. In regards to our algorithm where we have to look at Facebook feed items, the likelihood of word overlaps across a large amount of feed items is quite low. This method would not be appropriate for our purposes.

The UserInfoBigram analyses the optional text that is provided in every tweet and generalises a topic from those words. In Facebook, almost no one uses the optional text so this method will also not work.

In order to do topic classification we looked to a paper by Szo [SAC^{+}08] who proposed a system for defining topics of interest with the aid of Wikipedia. He collated a list of tags using a user's "tagging activity" which, for our purposes could mean commonly used words in past feed items. These tags are then verified using Wikipedia to check that there indeed does exist something along the lines of the tag. Wikipedia is chosen over more formal dictionaries due the the nature of social media tags not being real words, thus many of the tags will not be able to be verified by formal dictionaries such as WordNet. Wikipedia is then traversed (from the tag's wiki page) to find a super category that the tag belongs to. For example if a user generates the tag "C++", in Wikipedia, the C++ page is a subpage of the "Programming Language Families" page.

Since a single verified tag can have many categories and the final list ends up being dominated by the broader categories, only categories that meet either of these criteria are taken:

- The tag has only 1 category
- The category matches the tag name exactly
- The category is a plural of the tag

This approach nets us a list of the user's general interests, which can then be used to determine how much a certain feed item is related to the user's interests by observing the contents of the feed item and determining if they belong to any of these topics, again using Wikipedia.

Aga [ACG+14] discusses activity ranking for LinkedIn which is also a Social Networking Service. They discover two more factors that have a huge impact on whether the activity is deemed interesting or not. They used the measurement of CTR or click-through-rate which is the probability of a user clicking on the link to measure the appeal of an activity. An activity that was old had a low CTR compared to an activity that was new. It seems that the freshness of an activity or the time that the activity was made had to be taken into account in the ranking algorithm. In the graph below, we see a drop in the CTR as time progresses.

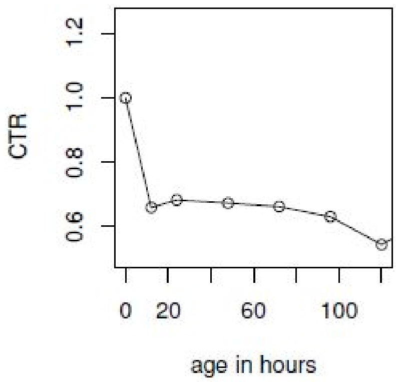


Figure 2.2: CTR vs Time Graph

Another factor was diversity. A huge drop in CTR was found when they gave users a repeated type of activity in their feed. We can see this effect in the graph below. Actor repetitions refer to activities that are posted by the same user appearing in our feed.

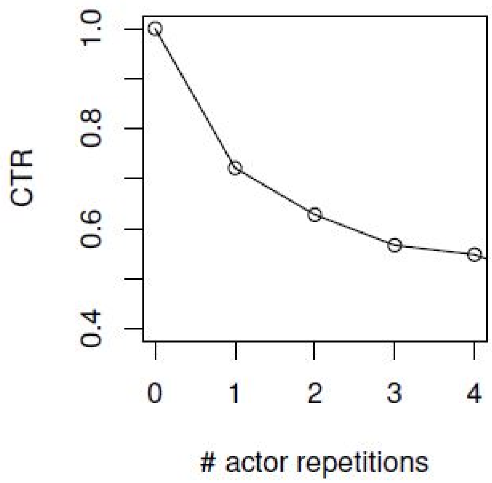


Figure 2.3: CTR vs Actor Repetitions

We can surmise that freshness and diversity are key factors that must be considered in our ranking algorithm. The method that they have used to deal with these two issues involved re-ranking the feed with a decay factor to account for time and adding a negative score to activities of the same type.

Aga [ACG+14] also reinforces the idea that people are interested in what their friends like when they analysed the activities of co-workers and colleagues. Like Li [LTL+10], they found that there was an increase in the click-through-rate of activities they were made by co-workers in the same organisation and colleagues. This emphasizes the importance of the factor of connections. We can see a huge increase in CTR in the graph below if the viewer has some relationship with the actor.

Viewer-Actor Relationship	CTR Lift
Same Company	66%
Same Job Function	25%
Same Industry	33%
Same Geo Region	24%

Figure 2.4: Table of Relationships with CTR

Chapter 3

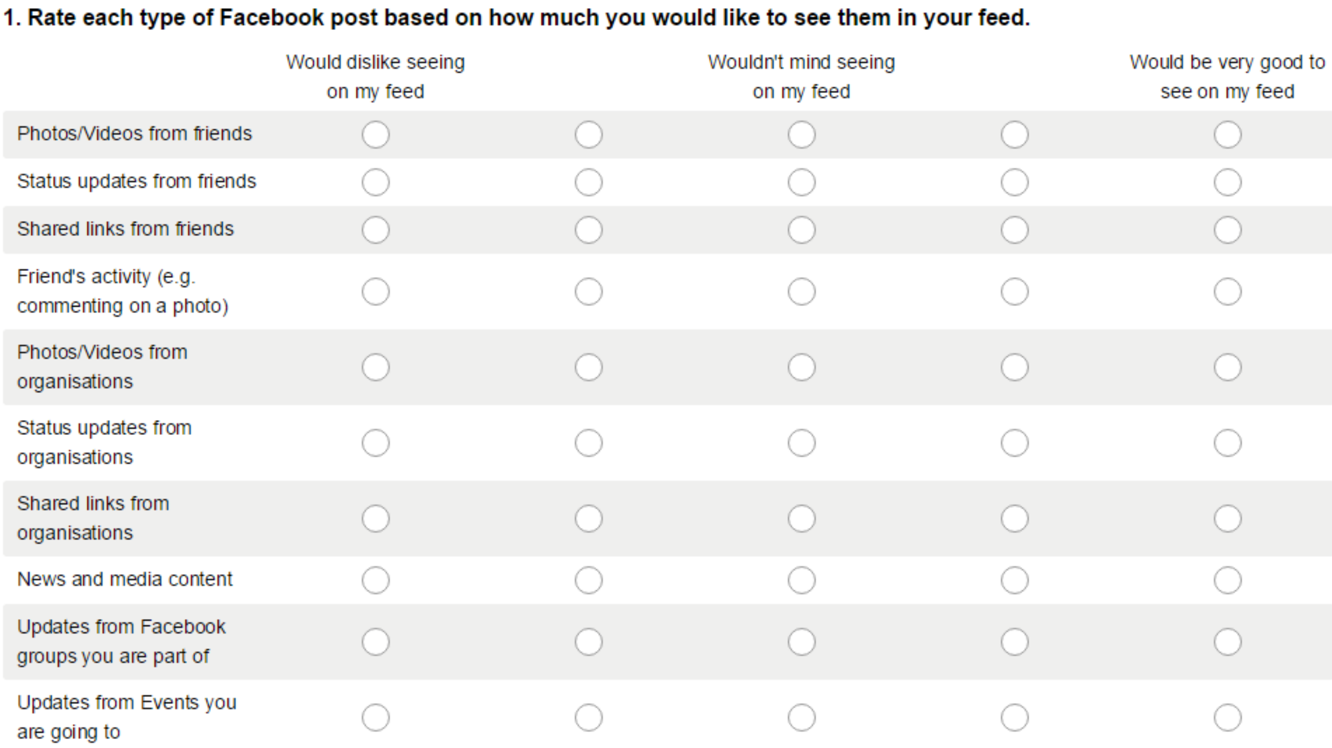
Design

3.1 User Modelling

After reviewing the previous works in user modelling, we chose to conduct a survey similar to the ones performed in the literature to help determine whether there are certain patterns in feed preferences for users and to group those patterns into user types. For example, finding that there is a significant group of users that prefer to see friend's social updates, photos and event information, we can classify those set of preferences as a user type.

This approach is further verified by Cle [Cle04] who had found that HCI professionals commonly used surveying as a part of user modelling.

Our survey was formulated with ideas from the survey conducted by Bon [BRR10] and will serve as the basis of our user modelling. The bulk of our survey can be seen in Figure 3.1.



1. Rate each type of Facebook post based on how much you would like to see them in your feed.

	Would dislike seeing on my feed		Wouldn't mind seeing on my feed		Would be very good to see on my feed
Photos/Videos from friends	○	○	○	○	○
Status updates from friends	○	○	○	○	○
Shared links from friends	○	○	○	○	○
Friend's activity (e.g. commenting on a photo)	○	○	○	○	○
Photos/Videos from organisations	○	○	○	○	○
Status updates from organisations	○	○	○	○	○
Shared links from organisations	○	○	○	○	○
News and media content	○	○	○	○	○
Updates from Facebook groups you are part of	○	○	○	○	○
Updates from Events you are going to	○	○	○	○	○

Figure 3.1: Survey Questions

There was a possibility that the results of this survey did not yield any significant patterns, in which case we had planned to simply display the survey questions themselves in our application to give the user full customisation of their feed ranking. This approach was not desired as it may have discouraged users of the system due to the amount of time it may take to answer the questions. fortunately the survey did yield some significant patterns in feed preferences and the former approach could be taken, as will be explained in the implementation section.

3.2 Ranking Algorithm

Following a review of the previous works, we have chosen to combine modules proposed from multiple papers to construct our ranking algorithm. We concluded that the main factors affecting our ranking algorithm were the following:

- Topic Classification
- Connections
- Freshness
- Diversity
- User Modelling

User modelling was mentioned previously and will play a role in our algorithm. Topic Classification, Connections, Freshness and Diversity implementation will come from ideas from various algorithms suggested by works that we have researched.

3.3 System Architecture

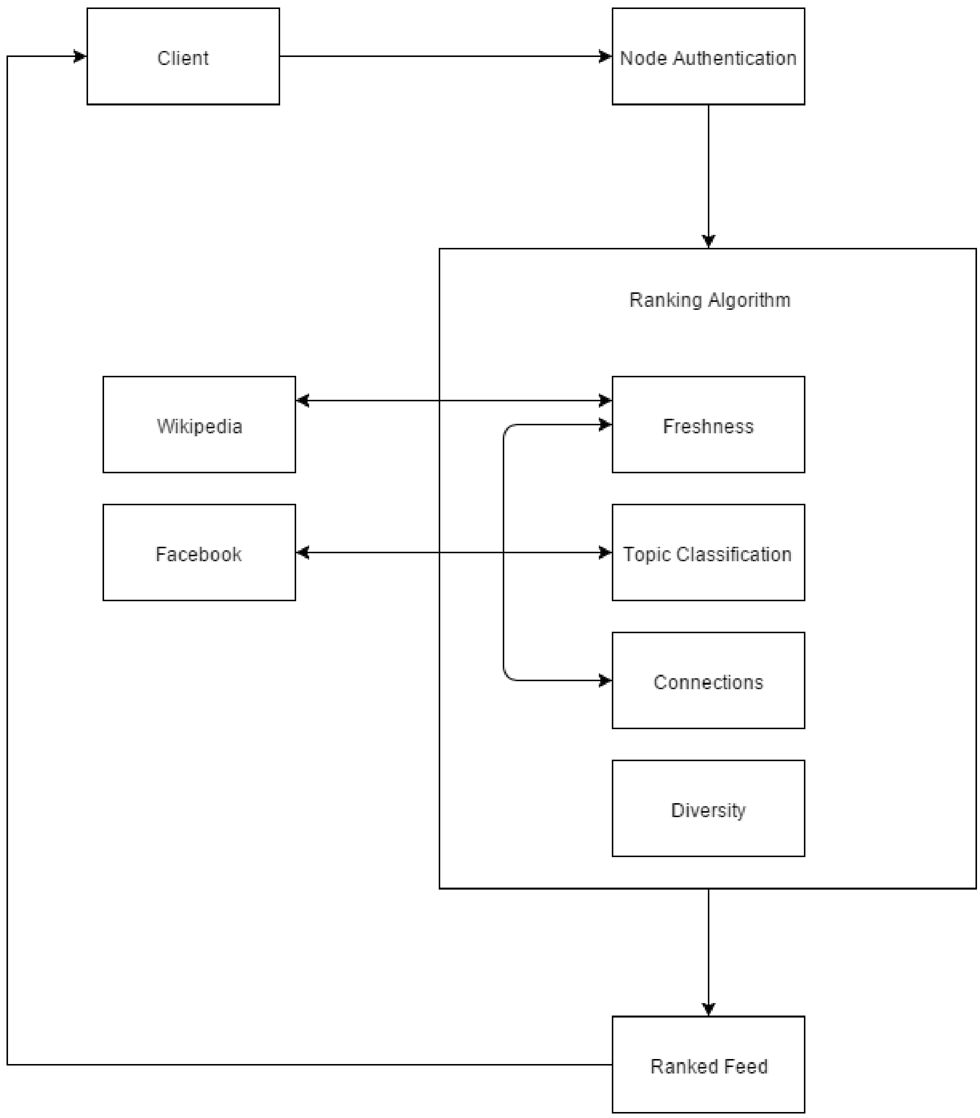


Figure 3.2: Block Diagram of System

In our design in Figure 3.2, we will have a front end module which will be the website (Node Authentication) that is seen by the user. The website will have a login screen for user authentication which will allow us to pull the data from their Facebook feed. Included with the login form will be a set of checkboxes indicating which user types they think they are with a brief description of each type.

When the user logs in we note which checkboxes they had selected and the associated weights e.g. feed items from friends, feed items from news / media, etc. from the user types selected and proceed to pull the users feed using their credentials. Once we have the users feed, it is pushed through our ranking algorithm which applies the modules specified earlier in the order listed below:

- The feed will be first be passed over to the freshness module. We will use Aga's [ACG+14] method and assign a decay factor depending on their time of creation.
- Then the topic classification module will decide which feed items contain content that the user will most likely be interested in, using the method proposed by Szo [SAC+08].
- After this, connection to the poster will be considered in the connections module which will simply look at how often the user has interacted with the person who posted that item and give a score based on that. This is an implementation of the solution proposed by Li [LTL+10]
- This feed will then be re-ranked based on the new scores and passed over to the diversity module. In this module, we will re-rank the feed and add a negative score to feed items of the same type. This method was also proposed by Aga's [ACG+14].
- Lastly, the newly ranked feed will be displayed in the frontend website in front of the user.

It should also be noted that our application does not contain a database as we deemed it unnecessary for our purposes. All of the users feed items can and must be pulled via the Facebook API and there is no need to create accounts as we already need the users Facebook accounts to authenticate.

Infact, our application is completely stateless, in that no data obtained is persistent and we ask the user to authenticate and select their user types for every time they wish to view their feed using our ranking. This is primarily because the incurred delay from pulling the users feed every time is negligible and we are only aiming to prove the effectiveness of the multiple proposed algorithms implemented in our design rather than focusing on efficiency and user experience.

Chapter 4

Implementation

4.1 Approach

Our aim was to create a more personalized view of the Facebook feed. The top down approach was used to break the problem into smaller modules and to create a solution for each module. The problem was broken down into the following parts:

- Pulling data from the Facebook feed
- Ranking Algorithm
- User Interface

To implement this solution we have used Nodejs for a number of reasons:

- It is a very flexible and diverse language and can be used for a huge variety of purposes, allowing us to create all aspects of our system using it.
- Many user-made modules that allow us to reuse solutions, e.g. interfacing with the Facebook api
- It is essentially just javascript which is perfect for a light application such as ours

4.2 Facebook API and App

Facebook requires the creation of an app in order to allow a website to pull any data from a users account. In order to create the application we registered as Facebook developers and were given a unique App ID and secret for our application. This is used to let Facebook know that our application is being used when we make calls to Facebook's API.

To grab the feed data from Facebook we utilized the Nodejs module written by Thuzi. This module communicates with the Graph API provided by Facebook that is used to gather information from the feed. Another module that was used was the passport-facebook module written by Jared Hanson. This module was mainly used to allow ease of authentication for Facebook. This means that we do not directly control the authentication and we are not capable of storing usernames or passwords.

Before we can begin pulling data from the feed we first need to get permissions from the user about which data that we need. The permissions that we desired were:

- read stream (feed)
- user likes
- read mailbox

The reason for getting each of these permissions will be explained in the ranking algorithm.

We receive the users feed as a json object, containing a huge amount of data on each feed item. We proceed to extract the required information and create our own feed item object, containing the fields shown in the figure 4.1, below.

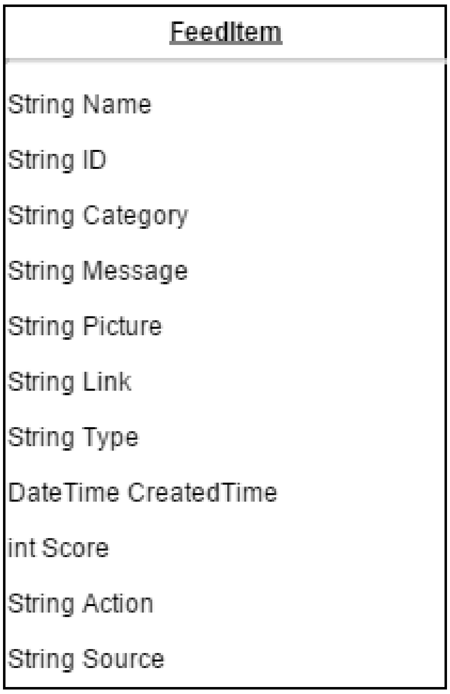


Figure 4.1: FeedItem UML

The name is the name of the person or organisation that posted the feed item.

The ID is the ID of the person or organisation that posted the feed item.

The category is Facebooks generated category of the feed item.

The message is the message of the feed item that appears in text.

The picture is a url of a feed item that points to a picture. This is relevant if the type is photo.

The link is a url that points to the actual feed item.

The type is the type of the feed item. It could be a photo or video.

The action is the action that was taken on the feed item.

The source is a url of a feed item. This is only relevant if the type is a video.

The createdTime is the time the feed item was created.

4.3 Ranking Algorithm

The algorithm for ranking the feed from Facebook comes from the combination of many other algorithms that have been specified in our design. The algorithm contains the following key components:

- Topic Classification
- Connections
- Freshness
- Diversity
- User Modelling

Each set component will provide either a positive or negative score to each feed item that we have received from the feed.

4.3.1 Topic Classification Module

The Topic classification module followed an idea from Szo [SAC+08] where a search of Wikipedia can be used on tags in order to generalise the topic. We decided to go through the user's likes and to do a wikipedia search in order to determine the topic of interest. To get this data we require the read likes permission from Facebook. The Wikipedia search used a nodejs module called wikipedia-js written by kenshiro. When we get a feed item from the feed, we would check parts of the feed for a match for the topics we generalised from wikipedia. A match would mean that the topic of interest of the user has appeared in the particular feed item and hence, we would add a positive score to the feed item.

4.3.2 Connections Module

The Connections module was based on the idea from Li [LTL+10] where a user may prefer a feed item that their friend likes or feed items. Our implementation of this module involved looking at the friends that the user has recently messaged. This required the read mailbox permission from Facebook. We do not directly look into the message but do consider the friend that the user has recently talked to. We consider the items from the feed where the person who posted the feed item is a friend of the user. We would add a positive score to the feed item if the poster and the user are friends that have recently interacted.

4.3.3 Freshness and Diversity Module

The Freshness module implementation followed an idea from Aga [ACG+14] where we assigned an initial score to each feed item and decreased the score depending on the feed items creation time. For the diversity module, we used an array to store all the types of feed items that we have not seen before. As the feed items come, we compare the feed items with the array of feed items in the array. The fields that we consider are type, category, name and link. If there is a match then we have seen a similar feed item before and should subtract score in order to make our feed more diverse. If there isnt a match then we add it to the array.

4.3.4 User Modelling

User modelling came from Bon [BRR10] and Nad [NH12] who concluded users have different reasons for using Facebook. This module hopes to find a generalized user and adapt the feed to their particular needs. Some types of users we generalised were:

- A **Socialite** is a user who mainly uses Facebook to see feed items from friends or families.

- A **News Reader** is a user who mainly uses Facebook to keep up-to-date with the news.
- A **Follower** is a user who uses Facebook to see feed items from organisations.

These user types were created based on intuition, which is why we carried out the survey specified in the design section. We created our survey using an online survey tool called SurveyMonkey. This allowed us to input our questions (as seen in figure 3.1) into an online platform to share with participants.

To distribute our surveys in a way that the participants would be diverse and unbiased, we made use of SurveyMonkeys feature of purchasing responses. This process works as follows; when a participant completes a survey with SurveyMonkey, they are (at random) asked whether they would like to join their panel, which would involve being invited to complete surveys where the profits from the purchased responses go towards charity. This creates an organic, unbiased panel of participants ready to give responses to surveys.

We chose to make our survey available to participants in the US of all ages, ethnicities and gender, with the only restriction being that they must be Facebook users, of course. We received over 100 responses which we analysed for any notable patterns. The responses were of surprisingly good quality and quite diverse and in depth. Some interesting points we observed were:

- Some users suggested we consider more specific post types such as weather alerts, free giveaways and local stories, while this would be great, given our time constraint it would be impractical for us to implement.
- A significant amount (10 percent) of users provided quite a surprising response in regards to connections, that was that they would prefer to see content from friends who they did not interact with often. These types of users could be accommodated for in our system as a subtype of socialite, however we did not increase the number of user types as we wanted to keep it general and lessen how

much the user had to think about their user type. This decision was made after seeing how difficult it was for users to determine their own user types even with our 3 basic, generalised types. This is further discussed in the evaluation section.

The raw data can be seen in Figure 4.2

	Would dislike seeing on my feed	(no label)	Wouldn't mind seeing on my feed	(no label)	Would be very good to see on my feed	Total	Weighted Average
Photos/Videos from friends	5.00% 5	5.00% 5	29.00% 29	24.00% 24	37.00% 37	100	3.83
Status updates from friends	6.06% 6	10.10% 10	26.26% 26	10.10% 10	47.47% 47	99	3.83
Shared links from friends	8.00% 8	18.00% 18	31.00% 31	23.00% 23	20.00% 20	100	3.29
Friend's activity (e.g. commenting on a photo)	16.00% 16	20.00% 20	31.00% 31	15.00% 15	18.00% 18	100	2.99
Photos/Videos from organisations	25.00% 25	31.00% 31	31.00% 31	9.00% 9	4.00% 4	100	2.36
Status updates from organisations	27.00% 27	31.00% 31	30.00% 30	7.00% 7	5.00% 5	100	2.32
Shared links from organisations	33.00% 33	30.00% 30	25.00% 25	8.00% 8	4.00% 4	100	2.20
News and media content	16.16% 16	29.29% 29	30.30% 30	8.08% 8	16.16% 16	99	2.79
Updates from Facebook groups you are part of	11.00% 11	10.00% 10	37.00% 37	28.00% 28	14.00% 14	100	3.24
Updates from Events you are going to	8.00% 8	6.00% 6	37.00% 37	24.00% 24	25.00% 25	100	3.52

Figure 4.2: Table of users

These results only indicate the opinions of the sample in general, to observe for patterns in user types, we had to analyse each response individually, categorise them as either Socialite, Follower, News Reader, or none. This process yielded the following graph:

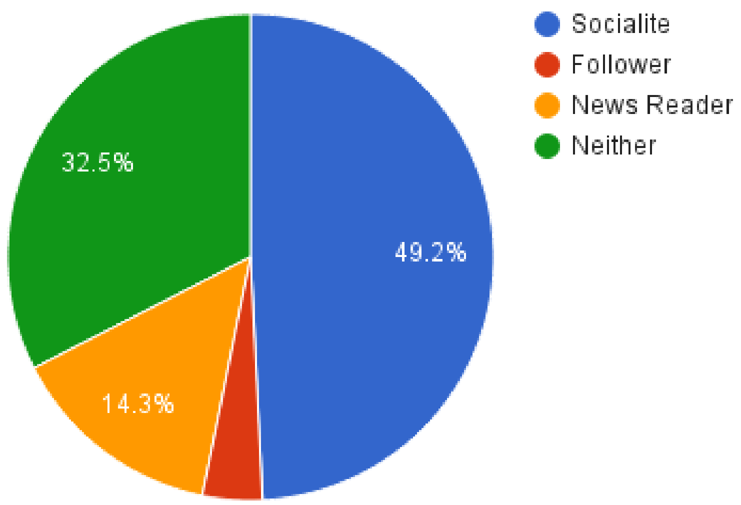


Figure 4.3: Pie chart of user type distribution

One could argue that the follower type may not be a significant portion of the results, however we struggled to find any other significant pattern of users from our survey responses and the follower type definitely did stand out to us. From this analysis, we were quite confident in using the aforementioned user types for the user modelling component of our algorithm.

The weightings for each of these user types are explained as follows:

- Users who identified as socialites had additional score added to their feed items if the feed item was generated by the action of a friend (e.g. liked the feed item) and even more score was added if the feed item was actually posted by the friend.
- Users who identified as News readers had additional score added to their feed items if the category field contained news in the string. This created some problems with the diversity algorithm as the score would then be subtracted due to

feed items having the same category, so we bypassed that section of the diversity module if the user identified as a news reader.

- Users who identified as Followers acted oppositely to a socialite, that is, we added additional score to the feed item if it was generated by the action of an entity who is not a friend and additional points if it was posted by that entity themselves.

4.4 User Interface

A typical user that uses our app will first be welcomed with a welcome screen as shown in the figure below.

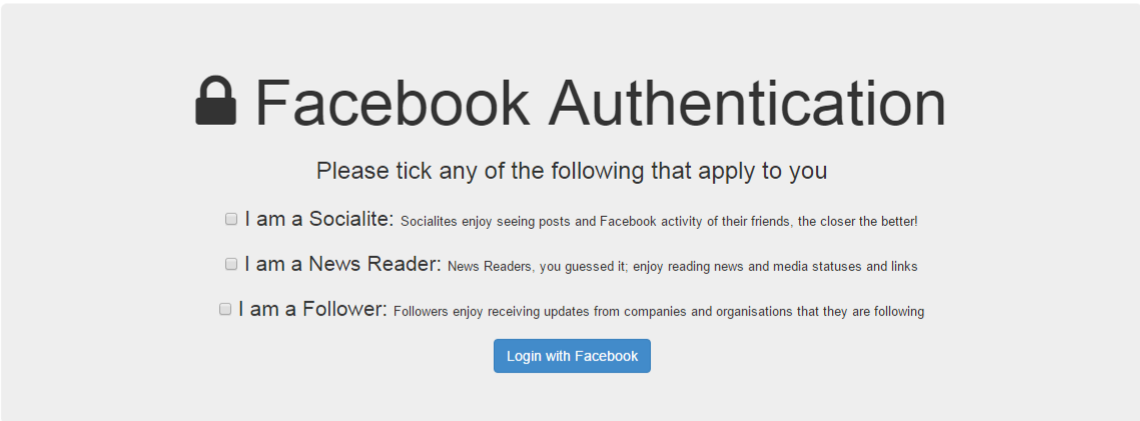


Figure 4.4: Welcome screen

The user can choose one or more of the options before they click log in. When the login is clicked, the user is taken to a typical login screen from Facebook.

After logging in, the user will be able to see a Facebook feed.

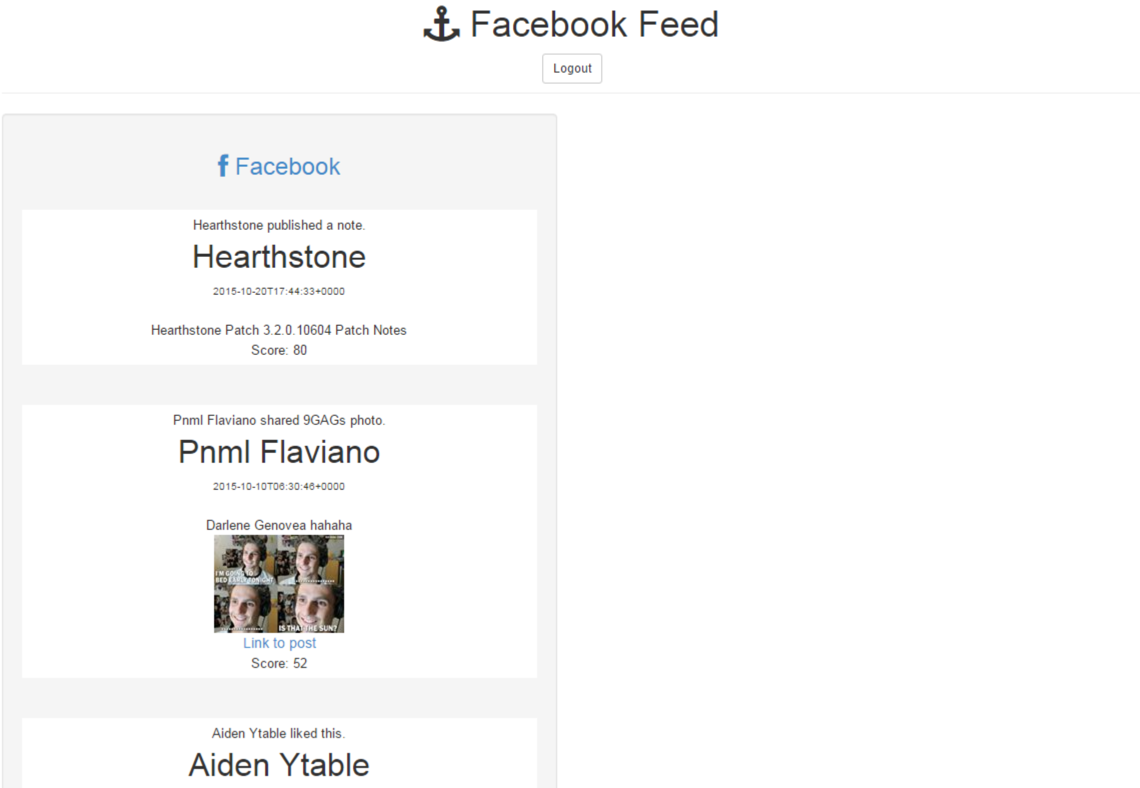


Figure 4.5: Profile Page with ranked feed

4.5 Problems Encountered

A typical user receives an enormous amount of items to their feed daily. This means that we would be required to pull a large amount of feed items in order to decide which items that the user would prefer to see. Unfortunately Facebook has a limitation on how many API calls can be made in a period of time. Only 600 API calls can be made every 10 minutes. This limitation has prevented us from getting more data about the user such as searching through the feed to see if a friend has liked a particular feed item.

Additionally it may seem that some of our modules such as the connections module seem far too simple. This was also due to restrictions placed on us by the Facebook API; one such limitation of the Facebook API involved the friends list. The Facebook API only allows us to get friends who have also installed the app. For our particular time frame, this is very impractical, thus we were unable to even see a list of the user's friends, let alone data about how close they are.

Another limitation that we encountered was not having access to the users own past posts, this meant that we had to resort to only looking at the users liked pages to define their topics of interest, instead of looking at their past posts as suggested by Szo [SAC+08].

Chapter 5

Evaluation

5.1 Considered Evaluation Methods

We considered two different ways of evaluating our system.

The first method was to simulate real users by creating Facebook accounts and attempting to mimic behaviour of each user type. We would then create a ground truth regarding how that type of user would like their feed ranked and compare the output of our ranking algorithm to the ground truth. We found quite a few flaws in this method, the major one being how difficult it would be to simulate a real user. Creating social interactions and simulating connections between users would prove very difficult. On top of this, the ground truths that we would be creating could be affected by confirmation bias. This left us with a very questionable evaluation method, so we arrived at our second one.

The second evaluation method takes the form of gathering real users and performing a form of usability test. In this test, we will ask participants to order their feeds how they would like it to be seen, this forms an unbiased ground truth. We then run our ranking algorithm on their feeds and compare the ground truth they gave us earlier to the output. In addition to this ground truth comparison, we will ask the user to compare our ranking algorithm with the one provided by Facebook, without telling them which is which. This will give us some subjective results as to whether our ranking algorithm has succeeded in personalising the user's feed.

5.2 Evaluation Approach

From the two methods that were considered, we chose to use our second method. However, we decided to use a modified version of it since the purpose of our algorithm is to provide different views of the users feed, not necessarily a better algorithm than Facebooks which would prove extremely difficult considering the amount of data, time and resources put into their algorithm.

We gathered users and conducted a usability test on each of them. We pulled 50 items from their feed and had each user choose 10 items that they would like to see at the top of the feed. We then used our ranking algorithm and measured how many of the items they chose ended up in the top 10, 25 and finally 30 of our ranked feed. If we have a majority of the items that interest them at the top of the feed then we can say that our algorithm was a success.

We performed our evaluation on a very small scale as we felt that the background research connected to our design decisions are sufficient to back up our algorithm and our intention was for these sessions to act as a simple sanity check for our application rather than a large scale verification of our algorithm.

S - Socialite, F - Follower, N - News Reader

User No.	User Type	Top 10	Top 25	Top 30
1	S+N	3	4	5
2	S+N+F	3	5	6

Figure 5.1: Table of sample users

In figure 5.1 the top 10 refers to the the amount of items that the user has chosen that appeared in the top 10. The top 25 refers to the amount of items that the user has chosen that appeared in the top 25. Similarly top 30 refers to the amount of items that the user has chosen that appeared in the top 30 items of the feed in our ranking algorithm. From the table we can see that our algorithm is not performing as well as intended. Approximately 30 percent of the items in the feed that the user enjoys is at the top 3 with 50 percent being in the top 25.

5.3 Discussion

From our results, it may appear as though our algorithm is not performing well, however, on further analysis, when we look at the posts the user had interest in and the selected type, we find that users dont seem to be selecting the user type that best describes themselves. Some users chose the user type to be socialite, yet they have selected many posts that belong to the user type of Follower. Since we also pull a small amount of feed items compared to the large pool that Facebook has, there will likely be many items that the user does not want to see. This makes it difficult for them to select the best 10 items as they may not like any of the 50 that we have provided. This idea could skew our results.

Chapter 6

Conclusion

A plethora of research has already been conducted into general ranking algorithms for SNS feeds. We successfully aggregated and implemented common algorithms proposed in the literature by Aga [ACG+14], Li [LTL+10] and Szo [SAC+08] as a basis to our ranking algorithm and on top of that incorporated the results of our user modelling to enable a more personal factor to the algorithm.

As discussed in the evaluation section, it proved to be quite difficult for users to determine what type of posts they prefer, let alone generalise their preferences to certain user types. This makes it very difficult for us to confirm how effective our algorithm is exactly. Some suggestions in regard to this are proposed in the future work section.

6.1 Future Work

Our algorithm only contains a few user types, these user types could be refined into more specific categories thus accommodating a larger variety of users, but doing so would require more justification via surveying. There could also be more work dedicated to making the user understand each user type, for example providing examples of typical posts that may be scored highly for each user type. This will allow the user to more easily and correctly select the user type that best defines themselves.

And most importantly, the solution that this paper proposes would best be used as a framework to implement similar ranking algorithms across other SNS and further, more extensive and large-scale evaluation should be performed if such an algorithm was to be seriously considered for use within these SNS.

Bibliography

[ACG+14] Deepak Agarwal, Bee-Chung Chen, Rupesh Gupta, Joshua Hartman, Qi He, Anand Iyer, Sumanth Kolar, Yiming Ma, Pannagadatta Shivaswamy, Ajit Singh, et al. Activity ranking in linkedin feed. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1603–1612. ACM, 2014.

[BMAC13] Matthew Burgess, Alessandra Mazzia, Eytan Adar, and Michael J Cafarella. Leveraging noisy lists for social feed ranking. In *ICWSM*, 2013.

[BRR10] Jennifer Bonds-Raacke and John Raacke. Myspace and facebook: Identifying dimensions of uses and gratifications for friend networking sites. *Individual Differences Research*, 8(1):27–33, 2010.

[Cle04] Torkil Clemmensen. Four approaches to user modelling-a qualitative research interview study of hci professionals' practice. *Interacting with computers*, 16(4):799–829, 2004.

[LTL+10] Huajing Li, Yuan Tian, Wang-Chien Lee, C Lee Giles, and Meng-Chang Chen. Personalized feed recommendation service for social networks. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 96–103. IEEE, 2010.

[NH12] Ashwini Nadkarni and Stefan G Hofmann. Why do people use facebook? *Personality and individual differences*, 52(3):243–249, 2012.

[SAC+08] Martin Szomszor, Harith Alani, Ivan Cantador, Kieron OHara, and Nigel Shadbolt. *Semantic modelling of user interests based on cross-folksonomy analysis.* Springer, 2008.

[Sta15] Statista. Global social networks by users 2015 — statistic, 2015.